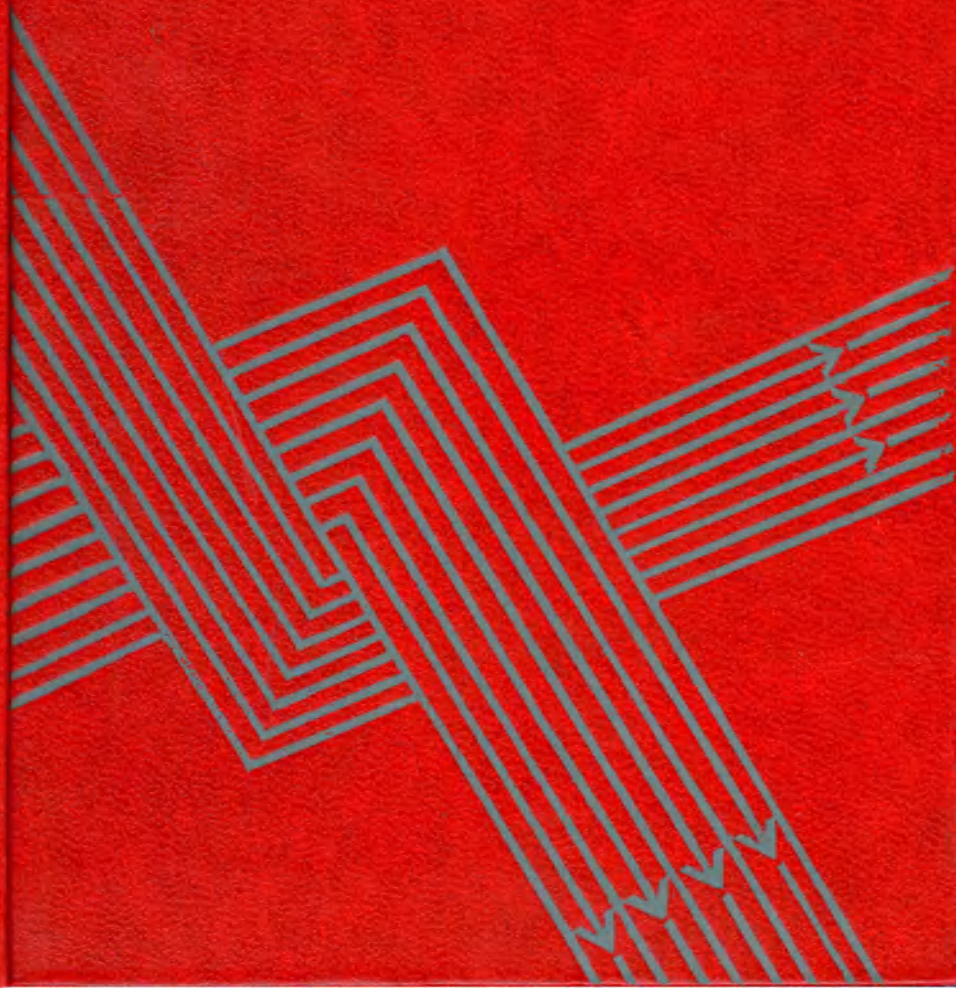


МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ
И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ
И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ



**MULTIPROCESSORS
AND PARALLEL
PROCESSING**

COMTRE CORPORATION
Philip H. Enslow, Jr., Editor

A WILEY-INTERSCIENCE PUBLICATION
JOHN WILEY & SONS, New York · London · Sydney · Toronto

МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

Под редакцией
Ф. Г. ЭНСЛОУ

Перевод с английского
Ю. С. ГОЛУБЕВА-НОВОЖИЛОВА и А. Л. ЩЕРСА

ИЗДАТЕЛЬСТВО «МИР»
МОСКВА 1976

Книга посвящена одной из наиболее важных областей современной вычислительной техники — мультипроцессорным и многомашинным системам, позволяющим создавать высокопроизводительные и надежные вычислительные комплексы. Рассмотрены как аппаратный, так и программный аспекты таких систем. Значительное внимание уделено обстоятельному описанию ряда реально существующих мультипроцессорных систем, используемых в различных областях человеческой деятельности.

Предназначена для широкого круга специалистов, связанных с проектированием и эксплуатацией вычислительных систем в народном хозяйстве.

Редакция литературы по новой технике

ПРЕДИСЛОВИЕ К РУССКОМУ ИЗДАНИЮ

Развитие работ по проектированию и внедрению автоматизированных систем управления (АСУ), широкое использование вычислительной техники для научных и инженерно-технических расчетов, создание Государственной сети вычислительных центров (ГСВЦ) в нашей стране выдвинули в качестве первоочередной задачи построение мощных вычислительных центров, обеспечивающих автоматизированный сбор, накопление и обработку информации в интересах различных предприятий, научно-исследовательских организаций, министерств и ведомств. Основные требования к таким вычислительным центрам заключаются в обеспечении возможности их работы в режиме коллективного пользования, высокой надежности, приспособляемости к изменениям функций управления объектами, возможности наращивания вычислительной мощности и т. д. Все эти требования могут быть эффективно реализованы прежде всего путем создания многомашинных и мультипроцессорных вычислительных систем.

Исследования показывают, что современные электронные схемы, работающие в наносекундном диапазоне, близки к возможному пределу быстродействия. Теоретически этот предел определяется скоростью света, равной 30 см/нс. Таким образом, дальнейшее повышение быстродействия вычислительных комплексов возможно прежде всего за счет создания особых структур, позволяющих совмещать во времени выполнение нескольких вычислительных операций, т. е. вести параллельную обработку информации. Это достигается в мультипроцессорных системах посредством дублирования блоков ЭВМ (процессоров, оперативной памяти, процессоров ввода-вывода и др.) и разработки специальных программных управляющих средств для разделения вычислительного процесса решения задачи на отдельные независимые операции и их распределения по различным устройствам ЭВМ для одновременного выполнения. Дублирование элементов ЭВМ обеспечивает, кроме того, повышение надежности вычислительной системы, а наличие специального программного обеспечения позволяет наращивать мощность системы подключением добавочных

модулей обработки, запоминания и ввода-вывода информации. Благодаря таким преимуществам мультимикропроцессорные и многомашинные комплексы в настоящее время становятся основой вычислительных центров коллективного пользования и больших информационно-вычислительных сетей.

Следует отметить, что интерес к подобным системам возник в нашей стране давно. Монографии, посвященные проблеме создания многомашинных и мультимикропроцессорных систем, появились в нашей стране почти десять лет назад и были, по-видимому, одними из первых в мире самостоятельных книг на эту тему¹. Однако в них рассматривались в основном теоретические вопросы создания многомашинных комплексов и их функционирования, а в качестве примеров были приведены системы, далекие от совершенства.

Предлагаемая вниманию читателя книга, написанная под редакцией одного из известных американских специалистов по вычислительной технике Ф. Энслоу, позволяет в значительной степени расширить наши знания в области создания современных больших вычислительных систем. Она является наиболее полной монографией, специально посвященной рассмотрению проблемы мультимикропроцессорования, и рекомендуется специалистами США в качестве универсального учебного и справочного пособия по этой проблеме.

В книге представлен и обобщен большой опыт разработки конкретных мультимикропроцессорных систем, накопленный ведущими американскими фирмами начиная с 1958 г. по настоящее время. Это дает возможность читателю ознакомиться с последними достижениями в этой области вычислительной техники.

Изложены вопросы как технической реализации мультимикропроцессорных и многомашинных систем, так и их математического обеспечения, что позволяет рассмотреть проблемы мультимикропроцессорной обработки комплексно.

Естественно, что подробно обсудить все эти проблемы в рамках одной книги невозможно, однако в постановочном плане в монографии освещены все наиболее важные вопросы создания мультимикропроцессорных систем.

Для переводчиков наиболее сложной оказалась задача выбора терминологии. Она усложнялась тем, что фактически книга была написана большим числом авторов, каждый из которых отличался своим стилем изложения, а в некоторых случаях и терминологией. Поэтому в процессе перевода приходилось не только давать русские эквиваленты совершенно новым английским терминам и словосочетаниям, но и согласовывать понятия и определения различных разделов монографии.

¹ Э. В. Евреиннов и Ю. Г. Косарев, «Однородные универсальные вычислительные системы высокой производительности», изд-во «Наука», Новосибирск, 1966 г.; Ю. С. Голубев-Новожилов, «Многомашинные комплексы вычислительных средств», изд-во «Советское радио», М., 1967 г.

Рассматриваемая область создания и использования вычислительной техники является сравнительно новой и бурно развивающейся; поэтому авторы перевода должны были идти на известное новаторство в выборе русской терминологии по мультипроцессорным системам, понимая, что некоторые термины могут вызвать возражения и не выдержать испытания временем. Для решения некоторых терминологических вопросов и для удобства читателя в книге приведен словарь английских терминов и их русских эквивалентов. Мы надеемся, что терминологические трудности не помешают читателю разобраться в сути вопросов создания и функционирования мультипроцессорных систем.

Книга рассчитана на широкий круг читателей, занимающихся вопросами создания и применения вычислительной техники. Она послужит хорошим учебным пособием для студентов старших курсов и аспирантов, специализирующихся в области разработки мощных информационно-вычислительных систем.

Ю. С. Голубев-Новожилов

А. Л. Щёрс

ПРЕДИСЛОВИЕ

По мере того как значение и доступность мультипроцессорных цифровых вычислительных систем возрастают, многие специалисты по вычислительной технике, не имевшие ранее опыта работы с такими системами, начинают ощущать недостаток ясного понимания как общей сущности этих систем, так и смысла некоторых элементов систем, с которыми они знакомятся по обычной технической литературе. Сейчас уже не возникает сомнений относительно важной роли, которую должны сыграть мультипроцессоры в будущем развитии цифровых вычислительных систем различного назначения и производительности. Цель этой книги — лишь подготовить читателя к вступлению в «эру мультипроцессоров». Это, однако, не означает, что мы утверждаем или предсказываем, что все потребности в новых системах будут полностью удовлетворяться за счет мультипроцессоров или что они по крайней мере станут определяющим фактором на рынке ЭВМ. Тем не менее в противоположность прошлому, когда мультипроцессоры были чем-то уникальным и необычным, в будущем они станут массовым явлением благодаря не только своей производительности и экономичности, но и другим качествам, таким, как надежность и готовность.

В течение долгого времени разработчиками и пользователями мультипроцессорных систем были военно-воздушные силы США (ВВС США). Большая часть первых работ в этой области финансировалась ВВС США с целью создания технических средств для систем управления, функционирующих в особо сложных условиях. Среди этих первых проектов были: система Ramo-Wooldridge RW-400 («Полиморфная ЭВМ»), наземная полуавтоматическая система управления средствами ПВО SAGE, система управления Стратегического командования ВВС (SACCS) и система Burroughs D825. Это были крупные стационарные системы; однако ВВС США, оценив количественные и качественные тенденции в развитии цифровой вычислительной техники, стремились использовать мультипроцессорные системы и на более низких уровнях управления, которые по своему характеру должны быть мобильными и вместе с тем требую

такой надежности и готовности, которые обычно обеспечиваются мультипроцессорными системами. Несколько проектов было разработано для автоматизации Центра управления тактической авиацией (ТАСС). Один из них под названием «Разработка критериев выбора и оценки для мультипроцессорной системы в качестве центральной вычислительной системы ТАСС в перспективе на период после 1985 г.» был выполнен фирмой COMTRE. Заключительные отчеты «Technical Report — Derivation of Selection Criteria and Comparative Analysis» (TR-5700-03) и «Technical Report — Multiprocessor Test Requirements» (TR-5700-04) были опубликованы 30 апреля 1971 г. Хотя основное внимание в этих работах уделялось особенностям применения мультипроцессорных систем в ТАСС, сравнительный анализ, представленный в приложениях к отчетам фирмы COMTRE, был выполнен в достаточно широком масштабе и охватил в качестве примеров военные и коммерческие системы, характеризующие современное состояние данной области исследований. Этот анализ явился побудительным стимулом к написанию данной монографии, а также был положен в основу приложений к книге.

Содержание книги характеризуется следующими существенными особенностями:

Это первая работа, в которой рассматриваются исключительно мультипроцессорные системы.

В работе впервые дано полное рассмотрение организации мультипроцессорных систем, охватывающее как аппаратные средства, так и программное обеспечение.

В книге показано, каким образом усилия разработчиков по повышению производительности ЭВМ при помощи совместного и параллельного использования программ и аппаратуры привели к логическому результату — созданию мультипроцессоров.

В книге дано сжатое описание большинства наиболее важных мультипроцессорных систем, находящихся сейчас в производстве.

Ф. Г. Энслоу мл.

*Лондон, Англия,
январь 1974 г.*

ACKNOWLEDGMENT

The coverage of a subject area as broad as multiprocessors would be impossible without a large amount of assistance. All of the manufacturers involved have participated in the review of the descriptions of their own systems. However, many other individuals have provided substantive input to the main portion of the text. I would like to thank specifically Elmer Branyan, Harvey Cragon, Ken G. Day, A. U. Elser, Jr., John W. Esch, Robert A. Mosier, E. Rudofsky, John E. Schier, R. Stoke, Joe Watson, and Jay Wolf for their greatly appreciated assistance.

Philip H. Enslow, Jr.
Editor

I acknowledge the considerable efforts of those staff members of The Comtre Corporation who participated in the various post 1985 Multiprocessor system studies leading to this book. Special thanks are due to Harry C. Collins for his great contribution, to Steve Tatusko for his cooperation, to Thomas G. Shack, Jr. for his counsel, and Jane Otto and Lana Hutton for their unending secretarial support, without whom this book could have not been possible. In addition, I thank the Editor, Dr. Philip H. Enslow, Jr., for his dedicated and effective work on this project. Lastly, an expression of gratitude to TeleDynamics Inc. for their valuable assistance. This book is dedicated to individuals in search of tomorrow's visions who synthesize today's applications.

M. James Errico
President, The Comtre Corporation

The material presented in this book does not represent the official position of any agency of the U. S. Department of Defense. The mention or discussion of specific systems is not an endorsement of the manufacturer or the product itself.

ПРИЧИНЫ СОЗДАНИЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ И СИСТЕМ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИНФОРМАЦИИ

УЛУЧШЕНИЕ ХАРАКТЕРИСТИК СИСТЕМ

В ответ на постоянно растущие требования к цифровым вычислительным системам, способным быстрее и надежнее решать все более сложные задачи, разработка таких систем успешно осуществляется в нескольких различных, но взаимосвязанных направлениях.

Аппаратные элементы и схемы. Новые или усовершенствованные элементы, которые могут быть использованы в более быстродействующих и более надежных схемах, являющихся основными строительными блоками для конструктора логических основ системы.

Архитектура системы. Алгоритмы выполнения основных функций, таких, как арифметические и логические операции.

Организация системы. Топология взаимных связей основных узлов вычислительной системы, таких, как устройство управления, модули памяти, контроллеры ввода-вывода, арифметико-логическое устройство, а также спецификация правил, управляющих потоком данных и сигналов управления между этими узлами, и средства, обеспечивающие возможность одновременного существования нескольких путей для потока данных.

Программное обеспечение. Управляющие и сервисные программы, необходимые для максимального использования технических возможностей системы.

Книга в основном посвящена исследованию одного метода совершенствования организации систем, который заключается в объединении множества функциональных блоков в мультипроцессорные системы или системы параллельной обработки информации. Хотя в дальнейшем нужно будет расширить предлагаемое определение, для начала мультипроцессор, или система параллельной обработки, может быть определен как единая система, состоящая из двух или большего числа устройств обработки информации, которые способны работать одновременно. Основное внимание в этой книге уделено физической организации таких систем, однако эту тему невозможно рассматривать в отрыве от других. Иногда возникает необходимость также обсудить архитектуру и программное обеспечение этих

систем. Однако полное исследование последних двух тем потребовало бы гораздо больше места, чем имеется в этой книге, и поэтому читатель должен четко представлять, что здесь приведено главным образом описание общей организации мультипроцессорных систем.

ТЕНДЕНЦИИ РАЗВИТИЯ ХАРАКТЕРИСТИК СИСТЕМ

В течение последних 20 лет происходит регулярное улучшение характеристик электронных схем для вычислительных систем. В среднем его можно оценить при помощи коэффициента 10 за де-

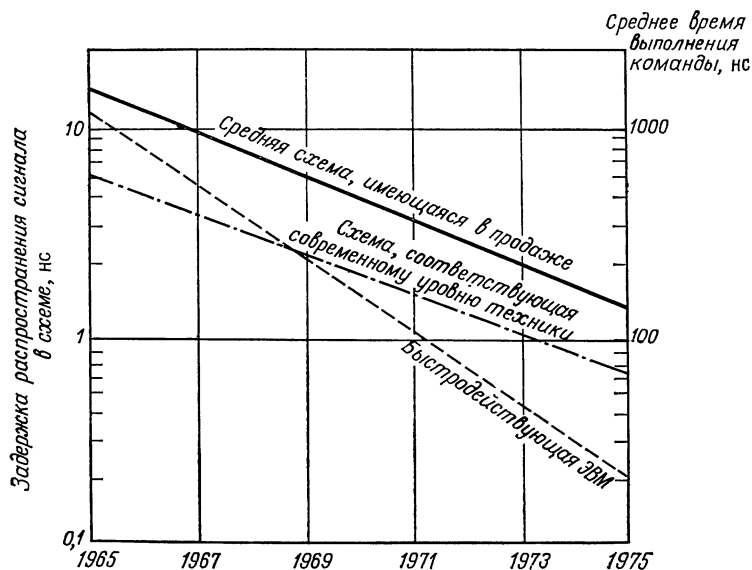


Рис. 1.1. Улучшение характеристик систем (по данным фирмы Quantum Science Corporation).

сятилетний период. Однако при использовании достижений в области основной архитектуры систем (улучшение алгоритмов, увеличение быстродействия) удастся добиться повышения темпов роста скорости выполнения основных функций (средней скорости выполнения одной команды) в больших быстродействующих ЭВМ в три раза по сравнению с тем, что было указано выше. Рис. 1.1 иллюстрирует значение двух указанных факторов, а также улучшение характеристик новейших схем, которые еще не повлияли на производство вычислительных устройств за последние 4—5 лет. Заметим, что успехи в разработке современных устройств вычислительной техники

меньше, чем достигнутое улучшение характеристик «средних схем», имеющих в продаже. На основании данного анализа становится очевидным, что в будущем наибольшее внимание следует уделять алгоритмам, которые будут использовать режим совпадения операций для обеспечения общего высокого темпа роста средней скорости выполнения команд.

В 1944 г. ЭВМ на релейных элементах Mark I затрачивала на выполнение одной операции сложения 333 мс. Через 7 лет характеристики ЭВМ были улучшены более чем в 1000 раз; ЭВМ UNIVAC I в 1951 г. выполняла операцию сложения за 282 мкс. Следующее улучшение характеристик в 1000 раз было достигнуто в 1964 г. — в ЭВМ CDC 6600 время операции сложения снизилось до 300 нс. Можно ли продолжить такое тысячекратное улучшение характеристик и создать машину, которая будет производить одно сложение за 300 пс? Это весьма маловероятно, так как этому может помешать ограниченность максимальной скорости прохождения электрических сигналов в схемах. Электрический сигнал распространяется со скоростью несколько меньше 30,5 см/нс. Быстродействие порядка 300 пс означает, что за время выполнения одной инструкции сигнал должен пройти меньше 10,2 см. Это быстродействие может быть достигнуто в пределах одного сумматора; но необходимо также учесть время, которое потребуется сигналу для прохождения из памяти в сумматор, поскольку упомянутые 10,2 см не соответствуют размерам запоминающих устройств очень большой емкости. Поэтому необходимо рассматривать и другие варианты решения этой проблемы, а не только сокращение времени работы отдельной схемы. Наиболее вероятным способом дальнейшего увеличения эффективной скорости выполнения операций в ЭВМ может стать использование одновременных или совпадающих операций.

СОВПАДЕНИЕ¹ ОПЕРАЦИЙ

В связи с тем, что в любое время быстродействие и надежность электронных приборов и схем для вычислительных устройств однозначно определяется технологией их изготовления, ближайшей задачей развития архитектуры, организации и обеспечения вычислительных систем должно быть создание такой системы, которая обеспечивает параллельное, совпадающее или одновременное выполнение нескольких программ и эффективно использует эти возможности. Такая цель ставится не только при создании мультипроцессоров, она является общей для всех больших вычислительных систем.

¹ Следует различать по смыслу два используемых в книге термина — «совпадение» и «одновременность». Совпадающие явления происходят на протяжении одного и того же интервала времени, одновременные — в один и тот же момент времени.

Совпадение желательно и в большей или меньшей степени возможно для всех типов операций, которые должны выполняться системой:

Совпадение при выполнении нескольких программ различных пользователей (мультипрограммирование).

Выполнение операций ввода-вывода одновременно с программами пользователя (перекрытие операций ввода-вывода, переплетение).

Выполнение многих операций ввода-вывода одновременно с передачей данных.

Совпадение при выполнении в центральном процессоре различных операций.

Последнее может быть достигнуто при использовании следующих двух подходов:

Обеспечение параллелизма, когда одинаковые функциональные устройства выполняют в режиме совпадения различные задачи (основной тип мультипроцессорной обработки).

Применение магистральных систем, в которых каждая команда разбивается на составные части, и операции над потоком данных выполняются по частям, по мере того как каждый операнд проходит определенный участок магистрали.

Совпадение является чрезвычайно эффективным способом повышения производительности систем. Для современных систем, в том числе и для мини-ЭВМ, стала обычной возможностью одновременного ввода-вывода или ввода-вывода с перекрытием. Другие методы обеспечения совпадения, первоначально разработанные и используемые в больших вычислительных системах, будут также распространяться и на более мелкие системы в связи с повышением требований, предъявляемых пользователями к производительности малых ЭВМ.

ПОВЫШЕНИЕ НАДЕЖНОСТИ И ГОТОВНОСТИ

Требования высокой надежности и в особенности наибольшей готовности при минимальной конфигурации системы, имеющей значительную рабочую нагрузку, побудили к проведению большей части работ, связанных с созданием мультипроцессорных систем, включая их проектирование и разработку оборудования, системной организации, средств управления прохождением прикладных программ и программного обеспечения. Системы, спроектированные с учетом в первую очередь указанных выше требований, получили название систем, устойчивых к ошибкам.

Требования высокой надежности и готовности для бортовых ЭВМ в пилотируемых космических аппаратах стали основной причиной появления в NASA программ по исследованию мультипроцессорных систем. Основная деятельность NASA была сосредоточена в бывшем

Научно-исследовательском центре NASA по электронике в Кембридже (шт. Массачусетс) и в Дрейперовской лаборатории (лаборатории приборостроения) Массачусетского технологического института [21, 27, 34]. В этих работах не были использованы возможности увеличения производительности мультипроцессорных систем, поскольку отдельные устройства применялись в основном для взаимного контроля характеристик. Некоторые особенности общей организации системы, разработанной по этой программе, будут описаны ниже при рассмотрении организации аппаратных средств различных систем.

Имеется два основных подхода для обеспечения высокой надежности и соответствующей высокой готовности всей системы либо главной ее части. Эти подходы лучше всего представлены в системах с реконфигурацией и дуплексных системах. Почти все типы организации мультипроцессорной системы, рассмотренные во 2-й главе, предусматривают использование только первого метода. Его техническая реализация предполагает применение нескольких идентичных устройств или компонентов каждого типа, организованных и взаимосвязанных таким образом, что каждый из них может выполнять различные программы. В случае выхода из строя одного функционального блока исходная система несколько снижает свою эффективность и производительность вследствие реконфигурации и перераспределения нагрузки (постепенная деградация). Конечно, для идентификации сбоя в каждом аппаратном устройстве или блоке программного обеспечения должен быть предусмотрен полный набор средств обнаружения ошибки. Другой подход, известный как дуплексный, предполагает параллельную обработку данных для одной и той же задачи двумя или большим числом функциональных блоков или несколькими целыми системами. Любые расхождения в ответах, полученных отдельными устройствами, являются первым признаком сбоя. Затем может быть проведена полная проверка, и неисправное устройство выводится из работы, с тем чтобы восстановить его, если это возможно. Производительность не уменьшается, так как дублирующее устройство в состоянии взять на себя всю нагрузку (хотя в данном случае значительно уменьшаются возможности обнаружения ошибки). Дублированные или параллельные функциональные блоки, используемые в дуплексном методе, могут быть взаимосвязаны таким же образом, как и в других мультипроцессорах; однако для защиты средств связи от сбоев были разработаны специальные организационные методы, которые будут описаны ниже, после рассмотрения вопросов общей организации систем.

Требования высокой надежности и готовности предъявляются не только к системам, используемым для пилотируемых космических полетов, но и к Системе управления воздушным движением (АТС) Федерального авиационного управления США (FAA) [17]. Однако в АТС требования устойчивости к ошибкам сочетаются с требованиями очень высокой производительности, которые в настоящее вре-

мя еще не ставятся перед бортовыми ЭВМ. Поскольку в наземном оборудовании ограничения по физическим размерам не являются критическими, и более важными факторами являются время ответа и экономическая эффективность, FAA использует в своих системах уже имеющееся или незначительно модифицированное стандартное оборудование. Это отразилось на структуре системы АТС, которая обеспечивает лишь незначительные резервные или запасные возможности, которые могут быть использованы для сохранения работоспособности системы в процессе исправления отказавших элементов. Стопроцентное резервирование оборудования, которое характерно для бортовых ЭВМ космических аппаратов, здесь, как правило, не используется.

К военным командным системам и системам управления предъявляются такие же требования, как и к системе АТС. Первые военные системы, которые были использованы на практике в сложных ситуациях, такие, как SAGE и BMEWS, были чисто дуплексными системами; однако сейчас существует тенденция к применению мультипроцессоров, так как последние дают более эффективные решения с точки зрения экономичности.

Одиночный процессор в принципе не может быть надежным из-за его простой организации, основанной на последовательных действиях. Для достижения высокой надежности и готовности необходимо по крайней мере дублирование операций. Но такой подход является зачастую неэффективным, поскольку дополнительное оборудование не увеличивает производительности всей системы. При мультипроцессорной организации эти факторы снижения эффективности перестают действовать, так как все функциональные блоки выполняют различные программы до тех пор, пока не произойдет сбой. При этом система благодаря реконфигурации может выполнять наиболее важные задачи, несмотря на то что ее суммарная производительность уменьшилась. В тех случаях, когда важно сохранить эту производительность, целесообразно иметь в системе один или два запасных функциональных блока каждого типа, которые можно использовать при выходе из строя основных блоков.

ОДНОМАШИННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Для того чтобы лучше понять возможности и функциональные характеристики мультипроцессорных систем, читатель должен иметь хорошее представление об архитектуре одномашинных вычислительных систем. Удивительно, что многие специалисты-практики в области программирования и даже в области аппаратного обеспечения ЭВМ не имеют такого представления. Поэтому мы рекомендуем читателю, прежде чем начать изучение многомашинных систем, еще раз рассмотреть те основные идеи и средства их реализации, которые

были развиты в процессе создания одноплатных систем с ЭВМ третьего поколения.

В первых вычислительных системах все операции ввода-вывода выполнялись главным образом при помощи устройств считывания с перфокарт, печатающих устройств выходных данных и картонных перфораторов. При этом наибольшее влияние на производительность всей системы оказывали две характеристики. Первая из них — это низкая скорость обмена данными, обеспечиваемая перфокарточным оборудованием и устройствами печати. Даже современное быстродействующее оборудование имеет следующие характеристики скорости обмена:

Устройства считывания с перфокарт — от 800 до 1400 80-столбцовых перфокарт в минуту или 1400 символов в секунду (1050 карт/мин).

Картонный перфоратор — от 300 до 600 перфокарт в минуту, или 700 символов в секунду (525 карт/мин).

Построчное печатающее устройство — от 1000 до 2000 120- или 132-символьных строк в минуту или 3600 символов в секунду (1500 строк/мин)¹.

Однако реальные временные ограничения, связанные с выводом данных на перфораторы или печатающие устройства, обычно приводят к еще более низким скоростям обмена, чем указано выше. В любом случае эта скорость очень низка, вследствие чего процессор часто простаивает, ожидая результатов ввода данных или их вывода.

Уже первые устройства записи на магнитную ленту, непосредственно подключенные к большим ЭВМ, имели скорость обмена около 15 000 символов в секунду и выше. Поэтому в дальнейшем был использован логический путь повышения производительности. Были созданы отдельные специализированные автономные устройства (конвертеры) для чтения 80-столбцовых перфокарт и непосредственной записи этих данных на магнитную ленту блоками по 80 символов. Эта магнитная лента затем может быть использована как входная для большой ЭВМ вместо перфокарт. Подобным образом были построены специализированные автономные устройства для переноса данных с выходной магнитной ленты на перфокарты или печатные сводки. Проблема скорости обмена данными была частично решена, но оставалась еще одна старая проблема и появилась одна новая.

Нерешенной оставалась одна из первоначальных проблем. Форматы данных на входных и выходных магнитных лентах, посылаю-

¹ В этом списке не указаны устройства вывода информации из ЭВМ на микрофильм. Такие устройства обеспечивают в некоторых специальных областях применения значительно большие скорости вывода, однако в обычных условиях они пока широкого распространения не получили.

щих данные на конвертеры и получающих данные от них, соответствовали данным, поступающим с выхода или на вход периферийных устройств, работающих с единичными записями (устройства чтения перфокарт, печатающие устройства). Эти форматы включают сравнительно небольшие блоки данных, в результате чего много времени уходит на перемещение ленты между блоками. Коэффициент использования магнитных лент с низкими характеристиками достигает лишь около 20%, что приводит к уменьшению скорости обмена до 3000 символов в секунду. (Для современных устройств магнитной записи с высокими характеристиками скорость обмена могла бы упасть до 6%.) Рассмотренные специальные конвертеры не обладают способностью объединять по нескольку записей в один блок и распаковывать эти блоки при их выводе.

Новой проблемой явилась стоимость трех специальных конвертеров, которые редко работают с большой нагрузкой и играют существенную роль лишь при рассматриваемом режиме обработки данных.

Хотя это и будет сильным упрощением, есть все основания сказать, что одним из главных движущих факторов развития всех многомашинных систем было стремление разгрузить центральный процессор (ЦП) от выполнения операций ввода-вывода единичных записей. Поэтому перед тем, как приступить к обсуждению многомашинных систем, целесообразно рассмотреть достижения в организации простых, или однопроцессорных систем. До сих пор обсуждение было сосредоточено на периферийных устройствах и их влиянии на производительность вычислительных систем. Теперь мы распространим его на центральный процессор.

ОСНОВНАЯ ПЯТИБЛОЧНАЯ ЭВМ — МАШИНА ФОН НЕЙМАНА

Самую простую организацию имеет основная пятиблочная автономная ЭВМ, или однопроцессорная система, показанная на рис. 1.2. Такая структура была у всех первых ЭВМ с запоминаемыми программами, и на практике использовалось большое число коммерческих систем такого типа. Их основной недостаток связан с тем, что все операции ввода-вывода осуществляются через посредство арифметико-логического устройства. Это делается с целью совместного использования дорогих буферных регистров и других аппаратных средств в этих операциях. Проблема заключается в том, что все виды обработки прекращаются на время выполнения ввода или вывода.

Указанная организация систем преобладала до 1958 г. Однако уже в середине шестидесятых годов в ЭВМ CDC 3600 все сообщения, поступающие с консольного телетайпа оператора или, наоборот, на телетайп, проходили через накапливающий сумматор.

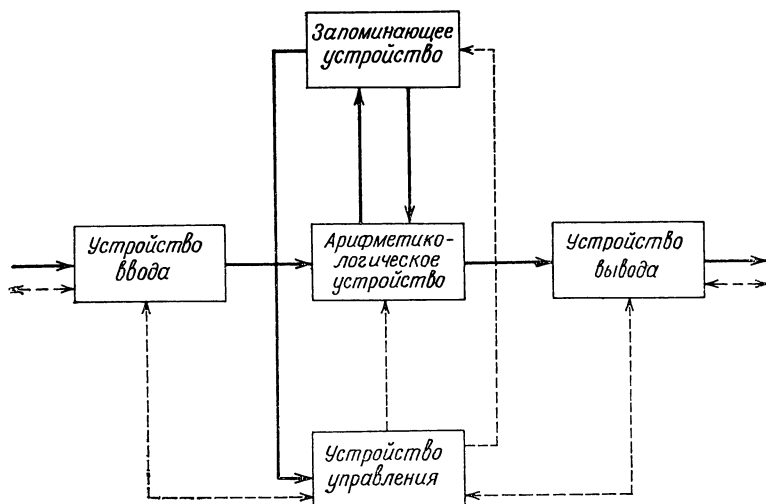


Рис. 1.2. Схема обычной ЭВМ.

— данные и команды; - - - управляющие сигналы.

ПРЯМОЙ ДОСТУП К ПАМЯТИ

В середине пятидесятых годов были изменены пути движения входного и выходного потоков данных в ЭВМ с целью обеспечения прямого доступа к памяти (рис. 1.3). Это позволило разгрузить ап-

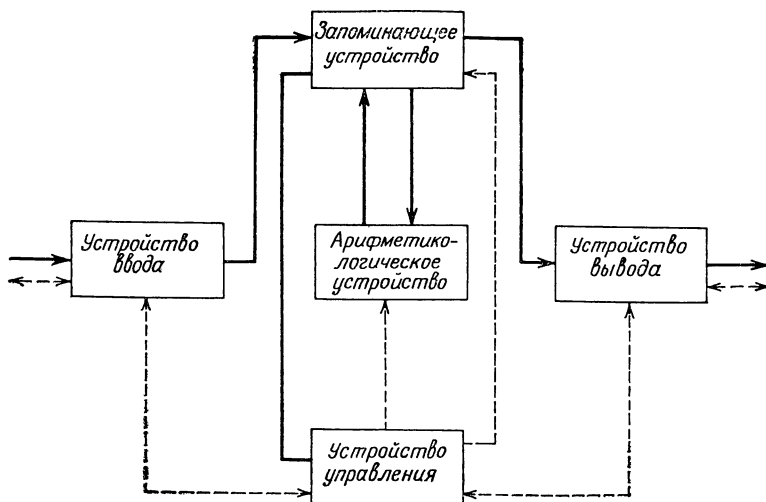


Рис. 1.3. Прямой доступ к памяти.

— данные и команды; - - - управляющие сигналы.

паратные средства арифметико-логического устройства, однако устройство управления все еще осуществляло подробный контроль и управление операциями ввода-вывода, а производительность процессора во время процедур ввода-вывода снижалась.

КАНАЛ ВВОДА-ВЫВОДА

В середине 50-х годов в состав ЭВМ был включен независимый канал ввода-вывода. Хотя и сегодня еще применяются некоторые старые методы буферизации или обеспечения частичной независимо-

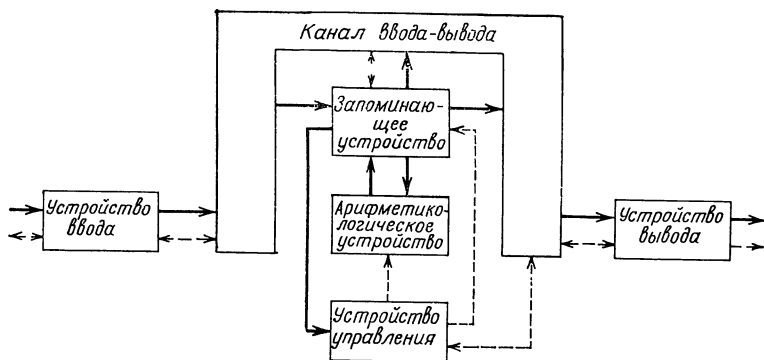


Рис. 1.4. Включение в ЭВМ канала ввода-вывода.

— данные и команды; — — — управляющие сигналы.

сти операций ввода-вывода, наиболее типичным для современных вычислительных систем является наличие канала ввода-вывода. Этот канал, как показано на рис. 1.4, обеспечивает не только независимый доступ к памяти, но и автономное управление операциями ввода-вывода. Фактически канал представляет собой небольшую ЭВМ. Он имеет очень ограниченные возможности, однако способен выполнять свои функции параллельно (одновременно) с центральным процессором. Программы канала генерируются операционной системой центрального процессора и размещаются в центральной памяти, доступной для канала. В этих программах должны быть определены области памяти, предназначенные для канала, объем передаваемых данных, применяемые внешние устройства и виды операций, выполняемые устройствами ввода-вывода. Центральное устройство управления информирует канал о месте размещения этой программы и передает ему команды о выполнении программы. Центральный процессор при этом остается свободным и может продолжать свои действия до тех пор, пока не произойдет прерывание от канала, указывающее, что канал выполнил заданную ему программу или

что возможна непредвиденная ситуация, например появились ошибки ввода-вывода. Следует отметить, что эффективное быстроедействие центрального процессора все же будет ниже потенциально возможного из-за обращений к памяти, необходимых для обслуживания операций ввода-вывода. Для более полного ознакомления с каналами, особенностями их работы и программирования целесообразно обратиться к книге Флореса «Организация ЭВМ» [23].

ПРОЦЕССОРНЫЙ БЛОК

В предыдущем параграфе термин «процессор» был использован для сокращенного обозначения совокупности устройства управления и арифметико-логического устройства (рис. 1.5). Это название пи-

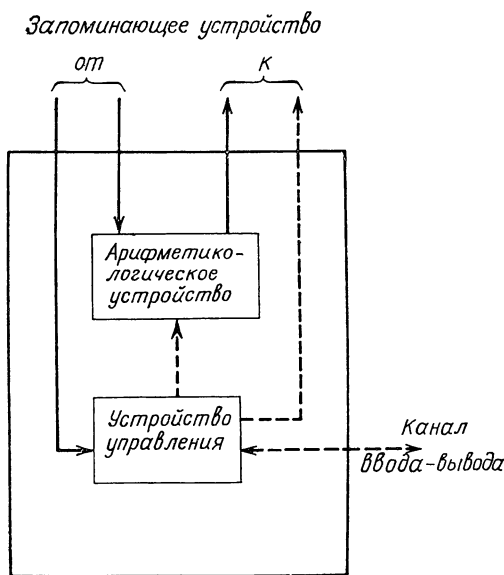


Рис. 1.5. Процессорный блок.

роко используется при обсуждении организации вычислительных систем. На схемах показаны только связи между устройствами, предназначенные для потоков данных, если в движении управляющих сигналов нет ничего принципиально нового или необычного.

МНОГОМАШИННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Многомашинные системы известны почти так же давно, как и современные цифровые ЭВМ. Самые первые из них представляли собой две одинаковые или почти одинаковые машины, которые вы-

полняли независимо одну и ту же программу с одними и теми же данными и сравнивали как промежуточные, так и окончательные результаты с целью проверки правильности работы оборудования. Такие системы состояли из вычислительных компонентов на электронных лампах, причем для повышения надежности на всех уровнях, начиная с отдельных узлов и кончая ЭВМ в целом, использовались дублирующие блоки. Наиболее известной системой с полным дублированием была наземная полуавтоматическая система управления средствами ПВО SAGE. Система на электронных лампах AN/FSQ-7 относится к так называемым дуплексным вычислительным машинам. Одна ЭВМ действует как «активная», другая — как «резервная». Их можно менять ролями с помощью автоматических или полуавтоматических методов. В этой системе процесс обработки входных данных не дублируется. Активная ЭВМ периодически передает часть базы данных в резервную ЭВМ через магнитный барабан, соединенный с обеими машинами. В случае переключения с одной ЭВМ на другую новая активная машина может начать пусковые процедуры, используя полученную часть базы данных. Программа резервной ЭВМ просто получает информацию об основных данных и хранит ее в дублирующих таблицах. Система AN/FSQ-7 использовалась в центрах наведения. Аналогичная дуплексная система Q-8 применялась в центрах управления боем. Если не считать слабо согласованные устройства ввода-вывода, эти системы не могут быть отнесены к мультипроцессорным; однако такие дуплексные системы внесли определенный вклад в разработку мультипроцессоров, демонстрируя один из путей достижения лучшей надежности или по крайней мере уверенности в том, что получены правильные результаты. В других отношениях эти системы менее интересны, поскольку в них взаимодействие между ЭВМ ограничивалось лишь сравнением результатов.

Многомашинные системы, рассмотренные ниже, обеспечивают взаимодействие между ЭВМ в течение всего процесса выполнения программы или комплекса программ, но только на уровне обмена данными. Другой признак, общий для всех описываемых систем и характеризующий их как многомашинные в отличие от мультипроцессорных, заключается в том, что каждая ЭВМ использует другую как канал или устройство ввода-вывода, каким бы ни был их общий аппаратный интерфейс.

Основная характеристика, позволяющая дифференцировать обсуждаемые ниже четыре типа организации многомашинных систем, — это применяемый в них способ соединения отдельных ЭВМ. Каждый тип организации в свою очередь характеризуется единством используемых физических и логических средств. Названия для этих систем выбраны по принципу наибольшей наглядности. К сожалению, относительно названий систем, как и по многим другим вопросам вычислительной техники, у специалистов нет единого мнения. Одна-

ко приведенные названия (за исключением названия «спутниковые ЭВМ») широко применяются по отношению к определенным системам.

СПУТНИКОВЫЕ ЭВМ

Когда появились первые автономные периферийные системы, они были названы спутниками. В дальнейшем после введения комплексирования, т. е. физического или электрического соединения двух систем, название «спутник» сохранили для обозначения следующего поколения автономных систем, что, очевидно, явилось неудачным использованием одного и того же термина. Возможно, наилучшим выходом из этого положения будет сослаться на стандартное словарное определение спутника как «помощника важного лица» и указать, что все системы, рассмотренные ниже, включают «спутниковые ЭВМ».

НЕСВЯЗАННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С АВТОНОМНОЙ ПЕРИФЕРИЙНОЙ ЭВМ

В результате создания IBM 1401 — малой ЭВМ, разработанной специально для посимвольной обработки информации типа банковских счетов и платежных ведомостей, было найдено решение сразу двух проблем, связанных с низкой скоростью устройств ввода-вывода и стоимостью специального оборудования. ЭВМ IBM 1401 вместе с устройством считывания с перфокарт, перфоратором, печатающим устройством и устройствами записи на магнитную ленту стоила меньше, чем три специальных преобразователя (конвертера). Кроме того, общие характеристики устройств ввода-вывода ЭВМ IBM 1401 были лучше (хотя конвертеры, безусловно, были бы усовершенствованы, если бы не появилась эта ЭВМ). Конкретные достоинства этой системы обусловлены возможностью обнаруживать ошибки, редактировать данные, а также при необходимости объединять информацию в блоки или распаковывать блоки информации. При этом не только уменьшается нагрузка на центральный процессор, но значительно возрастает эффективная скорость обмена данными с входными и выходными лентами.

Вскоре выяснилось, что при использовании таких систем все же приходится выполнять каждое задание по отдельности, что связано с необходимостью частой смены магнитных лент и другой вспомогательной работой. Поэтому в дополнение к новому аппаратному обеспечению, которое допускало применение программируемого конвертера ввода-вывода, было разработано специальное программное обеспечение для организации последовательной обработки «пакета» заданий определенного типа без остановок или вмешательства человека для повторного запуска программ, за исключением случаев,

когда выявляются ошибки. Достаточно полное обсуждение этого вопроса включено в разделы об операционных системах, поскольку оно не относится к проблемам организации аппаратного обеспечения систем.

Несвязанные системы разрабатывались для обеспечения ввода-вывода информации в режиме off-line с целью разгрузки центрально-

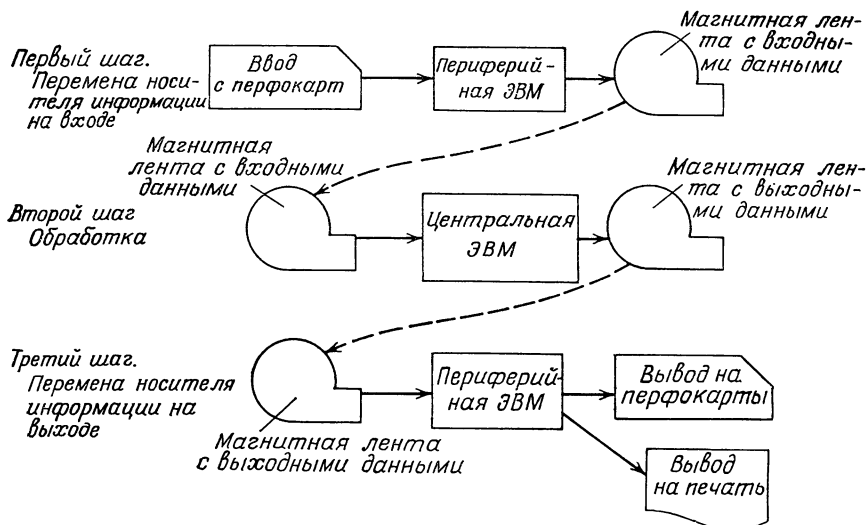


Рис. 1.6. Несвязанная вычислительная система с автономными периферийными ЭВМ.

го процессора. Между периферийным и центральным процессорами не имеется прямого физического соединения. Здесь отсутствуют также какие-либо совместно используемые аппаратные средства, которые есть в других системах, рассмотренных ниже. Целесообразность создания этой системы определялась обеспечением совпадения операций и экономичности, так как небольшая и недорогая система выполняет медленные операции ввода-вывода (чтение карт, печать), а центральная система выполняет высокоскоростные операции, такие, как обмен с магнитными лентами и дисками (рис. 1.6).

Одним из первых примеров организации несвязанной системы было использование ЭВМ IBM 1401 совместно с ЭВМ IBM 709 или IBM 7090¹. Это объединение ЭВМ было осуществлено с учетом

¹ Такое использование ЭВМ IBM 1401 как периферийного процессора не следует путать с другими специальными применениями ее в качестве канала ввода-вывода режима on-line для большой ЭВМ, такой, как AN/FSQ-1,3-32 или другие ЭВМ ВВС США.

взаимно дополняющих друг друга характеристик каждого компонента системы:

ХАРАКТЕРИСТИКИ ОБОРУДОВАНИЯ

IBM 1401

ЭВМ с посимвольной обработкой данных.

Высокая экономическая эффективность процедур ввода-вывода, редактирования, объединения данных в блоки и т. д.

Небольшая производительность (по сравнению с IBM 7090).

IBM 709 или IBM 7090

Двоичная ЭВМ с пословной обработкой данных.

Более высокая производительность по сравнению с IBM 1401.

ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАЦИЙ

Перенос информации с перфокарт на магнитную ленту в периферийной системе.

Подключение магнитной ленты к центральной ЭВМ после того, как комплектация пакета заданий закончена. (Сначала это делалось вручную путем переноса ленты с периферийных лентопротяжных устройств на устройства, подключенные к центральному процессору. Позже к лентопротяжным устройствам был добавлен переключатель для электрического переключения всего лентопротяжного устройства от одного процессора к другому.)

Выполнение вычислительных операций в центральной системе.

Подключение магнитной ленты с выходными данными к периферийной системе.

Печать выходных данных в периферийной системе.

РЕЗУЛЬТАТ

Эффективность всей системы в значительной степени определяется указанным режимом ее работы.

Система имеет более высокую производительность. Выполняется большее число заданий, центральная система используется эффективнее, поскольку центральный процессорный блок не простаивает в ожидании, пока канал выполняет операции ввода-вывода одиночных записей.

С другой стороны, увеличивается полное время получения результата для всех пользователей в связи с необходимостью составления пакетов заданий и временной задержкой, обусловленной тем, что приходится ждать окончания процедур по преобразованию данных на входе и выходе системы для всего пакета заданий.

СВЯЗАННЫЕ СИСТЕМЫ

Введение. Связанные системы имеют электрические соединения или, в более общем случае, совместно используют общие аппаратные средства в режиме совпадения. Обычно это два процессора, совершенно различные по производительности и рабочим характеристикам. Способы объединения аппаратных средств, которые дают хорошие результаты для систем с периферийными процессорами, как правило, подходят и для связанных, если возможно электрическое сопряжение между процессорами.

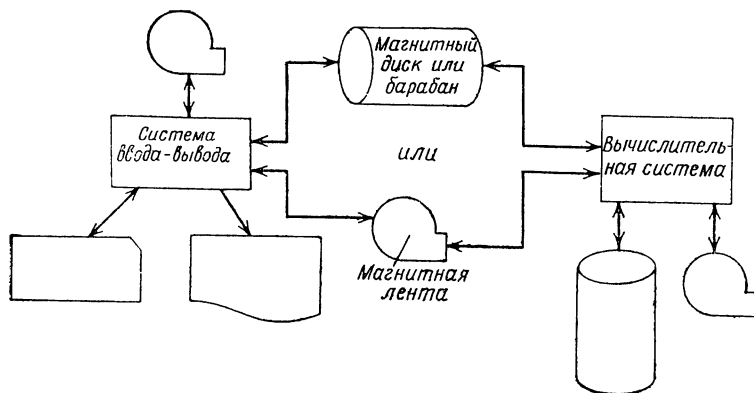


Рис. 1.7. Косвенно, или слабо связанная система.

В связанной системе два процессора могут выполнять две программы совершенно автономно или во взаимодействии друг с другом. В общем случае одна система функционирует по отношению к другой как блок ввода-вывода или коммуникационный процессор. Взаимодействие между двумя процессорами остается на уровне обмена данными, и основной процессор использует другой процессор только как устройство ввода-вывода.

Косвенная, или слабая связь. Слабо связанные системы имеют в совместном пользовании часть оборудования ввода-вывода типа дисков или магнитных лент (рис. 1.7).

В подобной системе невозможно какое-либо взаимодействие между программами, выполняемыми в каждой отдельной системе, за исключением информационного обмена. Одна из систем вводит совокупность данных в общее запоминающее устройство, используя принцип «почтового ящика». Другая система не знает о наличии этих данных, пока не «заглянет» в почтовый ящик, или же первая система сигнализирует второй о том, что в почтовом ящике уже что-то есть.

Прямая связь. Прямо связанные системы имеют более жесткое электрическое соединение либо за счет общей адресуемой быстродействующей памяти, либо в результате прямого соединения двух высокоскоростных каналов (рис. 1.8). В качестве примеров можно привести следующие системы с прямой связью:

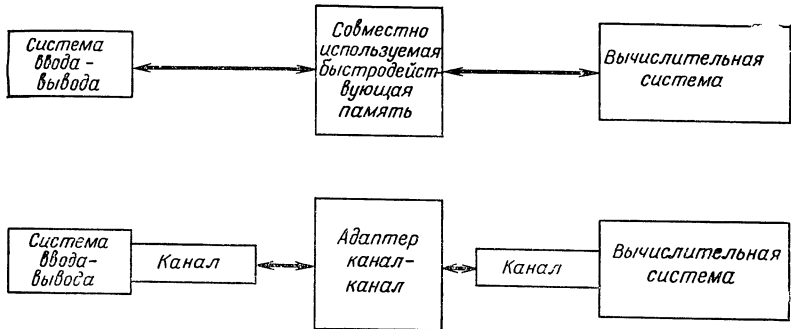


Рис. 1.8. Прямо связанная система.

Honeywell 8200

H-4200 (процессор с посимвольной обработкой информации для ввода-вывода)

H-1800 (центральная ЭВМ) — реконструированная система

H-1800 с повышенной производительностью.

Система IBM, созданная в начале 60-х годов

704X (для ввода-вывода)

709X (для расчетов и обработки).

Большие системы фирмы Control Data Corporation

CDC 6400 (с 10 периферийными процессорами)

CDC 6600 (с 10 периферийными процессорами)

CDC 7600 (с 20 периферийными процессорами).

Различие между прямо связанной системой и мультипроцессорной системой заключается в характере или степени взаимодействия между двумя ЭВМ на уровне математического, или программного обеспечения. Даже при наличии прямой связи взаимодействие между системами происходит только на информационном уровне.

Система с вспомогательным процессором. Термин «система с вспомогательным процессором» (ASP) относится к слабо связанным системам, содержащим до четырех ЭВМ фирмы IBM. Одна из ЭВМ управляет распределением работы всего оборудования. Все системы могут вести обработку заданий. Основная цель такой организации

заклучается в передаче части работ небольшой, менее мощной системе, выполняющей роль устройства ввода-вывода и другие вспомогательные функции¹. [Предусмотрен также вариант с одной ЭВМ, получивший название локальной ASP (LASP).]

Идея ASP первоначально возникла в результате развития системы с прямой связью 7040/7090. Эта идея предполагает нежесткое

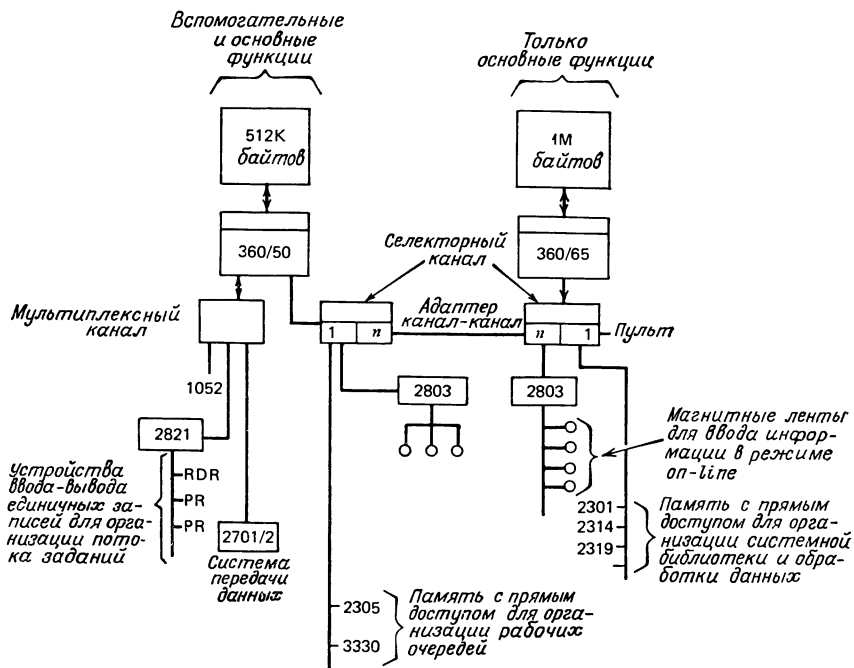


Рис. 1.9. Типичная система IBM ASP, в которой используется адаптер канал-канал.

соединение процессоров при помощи адаптера канал-канал. Примером может служить система ASP S/360, показанная на рис. 1.9. Здесь нет совместного использования центральной памяти, хотя периферийные устройства могут быть использованы совместно при помощи двухканальных переключателей.

В отличие от большинства других форм передачи по трактам ввода-вывода связь одного процессора с другим через адаптер канал-

¹ Эти функции включают: считывание и ввод заданий в систему, подготовку данных для каждого задания, обслуживание пользователей в режиме разделения времени и выдачу результатов выполнения всех локальных пакетов заданий.

канал требует их согласованной совместной работы и определенной последовательности действий. Запрашивающий процессор использует адаптер для сообщения другой системе о своем намерении передать или получить данные из памяти другого процессора. Обе ЭВМ должны затем привести в действие соответствующие программы чтения и записи, прежде чем адаптер канал-канал разрешит передачу.

«Адаптер служит устройством управления для каждого канала, к которому он подключен. Он запрашивается каналом и посылает ответ точно так же,

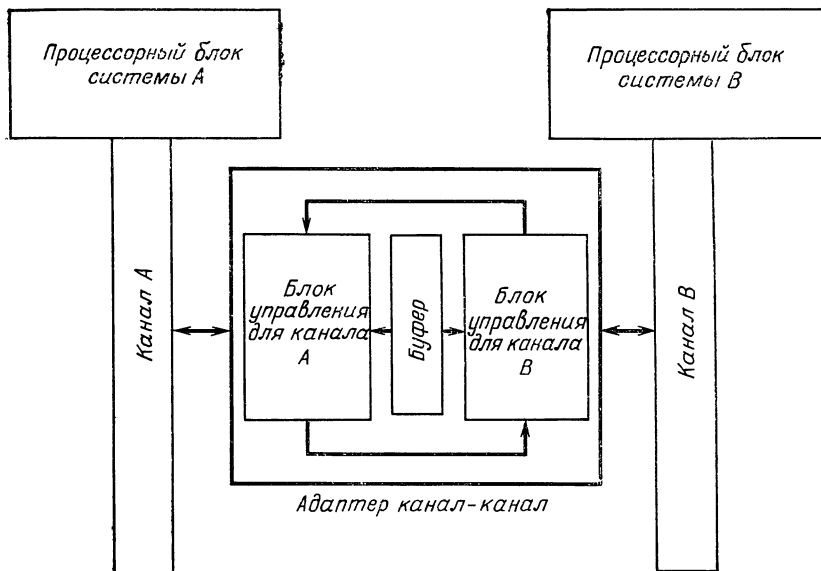


Рис. 1.10. Организация потока данных через адаптер канал-канал в мультипроцессорной системе IBM.

как любое устройство управления, и аналогично устройству управления принимает и расшифровывает команды из канала. Однако адаптер отличается от устройства управления тем, что не использует эти команды для приведения в действие или управления устройствами ввода-вывода; при помощи этих команд он открывает путь между двумя соединяемыми каналами и затем обеспечивает синхронизацию операций, выполняемых обоими каналами.

В функциональном отношении адаптер состоит из двух управляющих блоков, соединенных и взаимодействующих друг с другом при помощи общего однобайтового буферного регистра и нескольких сигнальных линий. Один из этих двух управляющих блоков обслуживает один канал, другой блок обслуживает другой канал. (IBM System/360 Special Feature, Channel-to-Channel Adapter, IBM Form GA 22-6892-1.) (См. рис. 1.10.)

Фирма IBM указывает следующие достоинства ASP:

1. Пользователи имеют дело с образом «одной системы». Несколько ЭВМ представляются программисту и обслуживающему пер-

соналу как одна машина. ASP обеспечивает равномерную загрузку каждой машины комплекса. Пользователю представляется возможность заранее определить, какая именно машина будет выполнять его задание, так что в комплекс могут быть включены процессоры с разными характеристиками. Такой выбор применяется, например, когда один процессор имеет эмулятор или другие специальные средства, которые необходимы для обработки задания. Наряду с соединением систем посредством внутренних каналов ASP позволяет соединять системы через линии связи.

2. Исключается необходимость использования машинного времени центральных процессоров для выполнения вспомогательных функций (таких, как вывод данных на печать). Поскольку служебные функции возлагаются на вспомогательный процессор, центральные процессоры не разделяют свое время между выполнением вспомогательных функций и прикладных программ. Поэтому можно с максимальной эффективностью использовать все ресурсы центрального процессора для прикладных программ.

3. Устройства ввода-вывода для обмена наборами данных имеют высокоэффективный алгоритм управления. Этот алгоритм был специально разработан, чтобы согласовать требования к обмену данными, характеристики наборов данных и имеющиеся аппаратные средства. В программах ввода-вывода всегда известно положение механизмов доступа, что обеспечивает минимальное время поиска при передаче данных на устройства. Эта функция распределения заданий улучшает совмещение во времени работы устройств ввода-вывода и центрального процессора, в результате чего повышается эффективность использования всего оборудования.

Белл и Ньюэлл возражают против некоторых из этих положений, утверждая: «В идеале одиночный процессор с мультипрограммированием или мультипроцессорная структура могли бы легко обеспечить все вышеназванные достоинства без излишеств в виде большой оперативной памяти в двух ЭВМ (имеющих почти одинаковые операционные системы)» [10].

ОСНОВЫ ПОСТРОЕНИЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

После того как было рассмотрено несколько многомашинных систем, не относящихся к мультипроцессорам, будет легко оценить особенности и дать определение мультипроцессорных систем. Определение мультипроцессоров, предложенное Американским национальным институтом стандартов (ANSI), не очень удачно, но все же содержит ряд ключевых положений:

«Э В М, содержащая два или несколько устройств обработки, функционирующих под единым управлением». («Словарь по обработке информации». American National Standard X3. 12-1970.)

ОПРЕДЕЛЕНИЕ МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМЫ

Более полное определение учитывает характеристики как аппаратных средств, так и программного обеспечения системы:

Система включает одно или несколько устройств обработки информации (центральных процессорных блоков).

Некоторые авторы к этому определению добавляют требование, что устройства должны иметь примерно одинаковые характеристики (т. е. симметричные процессоры). Другие допускают включение в группу и асимметричных систем, если они удовлетворяют другим требованиям данного определения.

Центральная память должна находиться в общем пользовании и должна быть доступна для всех процессоров системы.

Некоторые авторы требуют, чтобы *вся* память системы была общей, однако, как будет показано ниже на ряде примеров, часто бывает весьма целесообразно иметь в процессорах некоторую собственную память. Совместное использование всей памяти приводит к усложнению системного программного обеспечения. Поэтому большинство определений допускает наличие собственной памяти у каждого процессора.

В системе должен быть общий доступ ко всем устройствам ввода-вывода, включая каналы, устройства управления и периферийное оборудование.

В системе должна быть *единая* интегрированная операционная система, осуществляющая общее управление аппаратными и программными средствами.

В системе должна быть предусмотрена возможность тесного взаимодействия элементов аппаратного и программного обеспечения:

- на уровне системного программного обеспечения при выполнении системных задач;

- на программном уровне при выполнении частей одной и той же программы по очереди несколькими процессорами или при выполнении какой-либо независимой задачи одного задания в одном процессоре, когда основная задача решается на другом (возможность перераспределения заданий);

- на уровне обмена данными;

- на уровне аппаратных прерываний.

Взаимосвязь основных аппаратных средств мультипроцессорной системы может быть проиллюстрирована схемой на рис. 1.11. Следует отметить, однако, что взаимодействие аппаратных и программных средств зависит не только от организации взаимных связей, но и от системного программного обеспечения и процедур обработки данных.

Таким образом, мультипроцессорная система включает:

Два или больше процессорных блока.

Общую память.

Устройства ввода-вывода с общим доступом.

Единую интегрированную операционную систему.

Аппаратное и программное взаимодействие на всех уровнях.

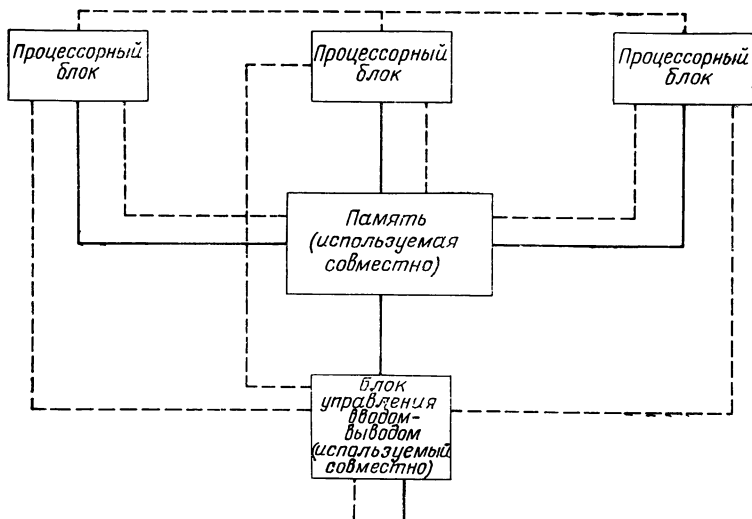


Рис. 1.11. Базовая конфигурация мультипроцессора.

— данные и команды; — — — управляющие сигналы.

РАЗВИТИЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ И ИХ НАЗНАЧЕНИЕ

Разработка и применение многомашинных систем, рассмотренных выше, были подчинены одной общей цели. Эта цель заключается в разгрузке главного процессора (обычно вычислительного устройства большой мощности) от операций ввода-вывода и передача этих операций небольшому и более подходящему для этих функций устройству. Совсем по-другому обстоит дело с мультипроцессорными системами. Некоторые результаты, полученные при разработке этих систем, похожи на аналогичные результаты в многомашинных комплексах, однако мотивы и причины их появления совершенно иные.

Первые исследования и практические приложения мультипроцессоров были связаны с созданием военных командных и управляющих систем. Сначала основной целью этих систем было достижение высокой боевой готовности, которая обеспечивалась путем объединения в группы одинаковых устройств, способных выполнять одни и те же функции и в результате реконфигурации создавать необхо-

димую работоспособную систему, даже когда сохраняется лишь часть исходной системы. Только после того, как эта цель была достигнута, разработчики универсальных вычислительных машин стали использовать возможности мультипроцессоров для повышения производительности систем. Необходимо было решить следующие задачи:

Обеспечить высокую готовность системы за счет:

повышения надежности, достигнутой путем резервирования различных устройств;

способности системы к реконфигурации.

Повысить производительность системы за счет:

обеспечения возможности параллельного выполнения независимых задач;

повышения эффективности работы и улучшения распределения нагрузки в системе;

обеспечения наиболее экономичного обслуживания экстренных заданий и заданий при пиковых нагрузках;

достижения наибольшего общего коэффициента «эффективного использования ресурсов» без создания новых типов архитектуры системы.

ИСТОРИЧЕСКАЯ ЭВОЛЮЦИЯ КОНЦЕПЦИИ МУЛЬТИПРОЦЕССОРА

При разработке многих проектов цифровых ЭВМ, в том числе и самых первых, инициатива их создания, обоснование необходимости, финансирование исследований, опытно-конструкторские работы и серийное производство были обусловлены специальными требованиями военных организаций. В частности, потребность в более надежных процессорах для командных систем и систем управления ВВС США явилась фактической причиной разработки конкретных проектов в этой области. Некоторые из них оказались неудачными или слишком дорогими, но несколько проектов, и в первую очередь различные варианты системы D825, увенчались успехом; они легли в основу научных знаний и послужили первыми моделями для коммерческого использования мультипроцессоров.

К сожалению, широкое распространение и использование системной организации, обеспечивающей совпадающую или параллельную обработку информации, произошло значительно позже, чем можно было бы ожидать. Возможны два объяснения этой задержки. Отчасти это запаздывание явилось следствием того, что большинство наиболее квалифицированных специалистов в гражданских организациях были заняты разработкой следующих поколений стандартных ЭВМ. Вне военной сферы разрабатывалось лишь несколько проектов мультипроцессорных систем (в некоторых университетах и промышленных лабораториях). Тем не менее опыт, накопленный при этих

Таблица 1.1

Краткая история мультипроцессорных систем и систем параллельной обработки информации

Время создания системы	Фирма-изготовитель системы и обозначение модели	Примечания
1958 г.	Национальное бюро стандартов, PILOT	Три независимых ЭВМ, которые могут функционировать в едином комплексе
1958 г.	IBM, AN/FSQ-31,-32	Полупроводниковая система SAGE; чисто дуплексная система
1960 г.	Ramo-Wooldridge, TRW-400	«Полиморфная система», применяемая в качестве командной и управляющей системы ВВС США. Структура системы не завершена. Имела важное значение для ранних концепций многомашинных систем
Май 1960 г.	UNIVAC LARC	Один процессор ввода-вывода и один центральный процессор, способные действовать параллельно. Один экземпляр системы был установлен в лаборатории Livermore AEC. Система не является «истинным» мультипроцессором
Май 1961 г.	IBM Stretch (7030)	Первоначально конструкция системы предполагала наличие отдельных процессоров для посимвольной обработки информации и двоичной арифметики. Однако эта идея не была реализована, и поэтому система не стала мультипроцессором. Все же система содержала перспективные идеи. Заказчикам поставлено только семь экземпляров системы
Октябрь 1962 г.	Burroughs D825 (Система имеет много различных обозначений в зависимости от основной военной системы, в состав которой она входит)	Первая модульная система с несколькими идентичными процессорами. Вся память используется всеми процессорами совместно. Система включает до 4 процессоров, 16 модулей памяти, 10 устройств управления вводом-выводом и 64 внешних устройства. Важной характеристикой системы является наличие одного из первых образцов операционной системы — Программы автоматического планирования и управления вычислительным процессом (ASOP)

Время создания системы	Фирма-изготовитель системы и обозначение модели	Примечания
Февраль 1963 г.	Burroughs B-5000	Один или два процессора. До восьми модулей памяти. Программы не связаны с физическими адресами. Супервизором является Главная управляющая программа (Master Control Program, MCP). Используются принципы виртуальной организации памяти и аппаратной реализации некоторых алгоритмических функций. Машинный код базируется на польской бесскобочной записи. Пользователи программируют только на языках АЛГОЛ и КОБОЛ. В ноябре 1964 г. начат серийный выпуск модели B-5500
1963 г.	IBM 704X/709X (7040 или 7044 и 7090 или 7094)	«Прямо связанная система»
1963 г.	Bendix G-21 (позднее CDC)	Мультипроцессорный вариант системы G-20, разработанный Технологическим институтом Карнеги. Система с перекрестным коммутатором
Сентябрь 1964 г.	CDC 6600	Содержит совокупность арифметикологических устройств, каждое из которых может выполнять только небольшую часть из всего набора команд. Десять периферийных процессоров составляли одну общую часть системы. В 1969 г. их число возросло до 20. Эти периферийные процессоры фактически образуют мультипроцессорную систему. В общем система является примером несимметричного мультипроцессора
Ноябрь 1964 г.	Burroughs B-5500	Усовершенствованная система B-5000 (см. февраль 1963 г.)
1964 г.	GE 645 (сейчас Honeywell)	Заказана по проекту MAC в Массачусетском технологическом институте
Май 1965 г.	GE 645 (сейчас HIS-645)	Разработана по проекту MAC в Массачусетском технологическом институте. Используются нестандартные аппаратные средства, однако операционная система MULTICS изготавливается серийно

Время создания системы	Фирма-изготовитель системы и обозначение модели	Примечания
Декабрь 1965 г. 1965 г.	UNIVAC 1108 SOLOMON 1	Только проект. Первый большой матричный процессор
Март 1966 г.	IBM S/360, модель 67	Специальная двухпроцессорная система, работающая в режиме разделения времени
Апрель 1966 г.	CDC 6500	Состоит из двух ЭВМ CDC 6400
Декабрь 1966 г. 1966 г.	XDS Sigma 7 SOLOMON II	Только проект
Июнь 1967 г.	CDC 6700	Состоит из двух ЭВМ CDC 6600
Август 1967 г. 1968 г.	XDS Sigma 5 CDC 7600	ЭВМ, очень похожая на CDC 6600, но с большим быстродействием и иерархической системой центральной памяти
Апрель 1969 г.	IBM S/360, модель 65 MP	Двухпроцессорный вариант стандартной модели 65. Истинная мультипроцессорная система
Июнь 1970 г. Октябрь 1970 г.	XDS Sigma 6 Burroughs B-5700	Похожа на систему B-5500, но обладает способностью к увеличению памяти. Имеется возможность совместного использования памяти на дисках одновременно четырьмя системами B-5700
Февраль 1971 г.	Honeywell 6050, 6060, 6080	
Июнь 1971 г.	Burroughs B-6700	
Сентябрь 1971 г.	Digital Equipment System 10/1055, 10/1077	
Сентябрь 1971 г.	XDS Sigma 8, 9	
Ноябрь 1971 г. 1971 г.	UNIVAC 1110 SDC, PEPE (ансамбль параллельных элементарных процессоров)	Прототип системы обработки радиолокационных данных для систем противоракетной обороны
Январь 1972 г.	Honeywell 2088	

Продолжение таблицы 1.1

Время создания системы	Фирма-изготовитель системы и обозначение модели	Примечания
Сентябрь 1972 г.	ILLIAC IV	Матричный процессор. 64 элементарных процессора. Управляется стандартным мультипроцессором, который играет роль буферной управляющей ЭВМ
Февраль 1972 г.	Burroughs B-7700	
1972 г.	CDC, Cyber 72, 73, 74, 76	
1972 г.	Goodyear STARAN	Параллельная ассоциативная система
1972 г.	Texas Instruments ASC (Научно-исследовательская ЭВМ)	Объединение мультипроцессорной и магистральной обработки информации
1973 г.	CDC STAR-100	Система магистральной обработки информации
1974 г.	IBM S/370 модели 158 MP и 168 MP	Системы с общей реальной и виртуальной памятью

разработках, был достаточен для того, чтобы показать, что основная проблема при создании мультипроцессоров связана с системным программным обеспечением, — операционной системой. Потребовались дополнительные исследования в этой области, прежде чем были созданы удовлетворительные мультипроцессорные операционные системы. Фактически многие специалисты чувствуют, что это пока остается ахиллесовой пятой реализации данной концепции. Большой список существующих мультипроцессорных систем и систем параллельной обработки информации приведен в гл. 4, а в приложении дано описание более 20 из них. Не удивительно, что почти все эти системы первоначально поставлялись вообще без соответствующего операционного программного обеспечения, а многие из них не имеют его до сих пор, и этот пробел должен заполнять сам пользователь. Во многих случаях пришлось вернуться к простейшим типам мультипроцессорных операционных систем, функционирующих в режиме «ведущий — ведомый», а получение высокой производительности всей системы становилось часто трудной задачей, решение которой отнимает много времени и у изготовителей, и у пользователей.

По-видимому, целесообразно будет завершить эту вводную главу обобщенной сводкой конкретных достижений в области создания мультипроцессорных систем. Лучше всего это сделать в виде краткой хронологической справки, представленной в табл. 1.1.

ГЛАВА 2

АППАРАТУРА

ОСНОВНЫЕ ТРЕБОВАНИЯ

Используя в качестве отправного пункта основное определение мультипроцессорной системы, сравнительно легко перечислить те возможности, которые должны быть обеспечены ее аппаратными средствами. Мультипроцессорная система состоит из двух или большего числа процессорных блоков с примерно равными возможностями, каждый из которых:

Имеет доступ к общей памяти.

Имеет общий доступ по крайней мере к части устройств ввода-вывода.

Управляется одной общей операционной системой, которая обеспечивает требуемое взаимодействие между процессорами и выполняемыми ими программами на уровнях задания, задачи, шага работы, набора данных, элемента данных, аппаратуры.

Наиболее очевидные выводы из этого определения касаются основных характеристик топологии соединительной сети между различными функциональными блоками системы. Должно существовать несколько групп многоканальных путей — либо постоянных физических, либо логических, создаваемых соединительной сетью по мере надобности. Эти пути должны обеспечивать следующие условия работы системы:

Любой процессор может управлять передачей данных к и от любой ячейки памяти (хотя может оказаться более удобным снабдить каждый процессор небольшой *собственной* памятью, что будет рассмотрено ниже).

Любой процессор может направлять команды управления к любому контроллеру канала ввода-вывода.

Любой канал ввода-вывода может передавать данные к и от любой ячейки памяти.

Любой канал ввода-вывода может управлять передачей данных между центральной памятью и любым устройством ввода-вывода.

Такие связи позволяют полностью выполнить требование всеобщего разделения ресурсов. Остальные возможности аппаратуры, которые должны быть предусмотрены в мультипроцессорах, относятся

больше к обеспечению общих функций управления, чем к разделению ресурсов.

Для того чтобы операционная система, управляющая мультипроцессорной системой в целом, функционировала эффективно и надежно, необходимо обеспечить несколько особых характеристик аппаратуры. В зависимости от условий работы операционной системы некоторые из этих характеристик имеют существенное значение, а другие только желательны. Ниже указаны эти характеристики.

Для того чтобы гарантировать сохранность таблиц или наборов данных при обращениях одного процессора, должна существовать аппаратная защита, предотвращающая вход другого процессора (специфические особенности такой защиты и обоснование ее необходимости рассмотрены в гл. III).

Должна быть предусмотрена возможность работы с переменными логическими адресами или именами процессорных каналов, запоминающих устройств и устройств ввода-вывода, а не с фиксированными физическими адресами.

Один процессор должен иметь возможность передать другому процессору сигнал или прервать его работу для того, чтобы запросить выполнение какой-либо функции или узнать, работает ли другой процессор. Это можно осуществить при помощи механизма прерывания или метода передачи сообщений через «почтовый ящик» (так называемого «мягкого» прерывания).

Если один из процессоров отказал, а другой, обнаружив это, желает перестроить свою работу так, чтобы взять на себя функции отказавшего процессора, то он должен получить доступ ко всей необходимой для этого информации, если даже некоторые данные находятся внутри самого процессора.

Наконец, может потребоваться, чтобы один из процессоров мог запускать или перезапускать другой процессор независимо от состояния последнего, если только он еще сохраняет работоспособность.

Как уже говорилось во введении, на блок-схемах систем, описываемых ниже, показаны только пути передачи информации. Этого вполне достаточно, так как основную функцию любой системы можно понять, рассматривая поток данных через нее. Не следует, однако, забывать, что последняя группа требований к аппаратуре, перечисленных выше, касается главным образом передачи управляющих сигналов, которые не показаны на схемах в настоящей главе.

ТИПЫ ОРГАНИЗАЦИИ СИСТЕМ

Так как основное внимание будет сосредоточено на потоках данных и на степени параллельности или совпадения во времени работы, которая может быть достигнута в мультипроцессорных системах,

очевидно, важнейшее значение приобретают топология и способ работы сети, которая соединяет между собой различные функциональные блоки системы. Приведенный ниже материал охватывает все общие типы конфигураций, которые были использованы до сего времени для построения «истинных» мультипроцессоров. Для полноты изложения рассмотрено также несколько других системных организаций, так как их часто относят к мультипроцессорам, хотя они и не удовлетворяют всем четырем требованиям нашего определения, сформулированного в начале главы. При обсуждении организации таких систем предполагается, что система сосредоточена в одном пункте физического пространства, так что расстояния между отдельными блоками системы позволяют передавать информацию между ними со скоростью работы ЭВМ. Это условие исключает из рассмотрения системы, состоящие из элементов, соединенных между собой линиями связи. Ни одна из таких систем, разработанных или проектируемых, не удовлетворяет всем четырем критериям нашего определения. Однако если учесть работы, которые ведутся в настоящее время в области высокоскоростных распределенных управляющих сетей, таких, как ARPANET, шлейфовые и кольцевые сети, то можно предположить, что недалек тот день, когда функциональные элементы «истинного» мультипроцессора будут отделены друг от друга сотнями и тысячами километров.

При выборе порядка изложения мы не руководствовались какой-либо особой идеей. Рассмотрение первых трех типов системной организации основывается на предыдущем материале. После первых трех конфигураций, каждая из которых может быть использована для построения «истинных» мультипроцессоров, описаны четыре типа квазимультипроцессоров. Таким образом, будут рассмотрены следующие семь типов системной организации:

Системы с общей или разделенной во времени шиной.

Системы с перекрестной коммутацией.

Многошинные — многовходовые системы.

Асимметричные или неоднородные системы (некоторые реализации этой конфигурации очень близки к «истинным» мультипроцессорам).

Матричные или векторные процессоры.

Процессоры с магистральной обработкой.

Системы, устойчивые к сбоям или отказам. Некоторые из реализаций являются мультипроцессорами, но основной целью этой конфигурации является достижение устойчивости против отказов и сбоев.

СИСТЕМЫ С ОБЩЕЙ ИЛИ РАЗДЕЛЕННОЙ ВО ВРЕМЕНИ ШИНОЙ

Эти системы обладают наиболее простой организацией, которая может быть реализована с минимальными затратами. Схема такой системы весьма проста (рис. 2.1). Между функциональными блока-

ми нет постоянных связей. Передача информации между модулями памяти и другими блоками осуществляется методами разделения времени или мультиплексирования.

Низкая стоимость достигается не только благодаря тому, что у каждого блока имеется только один интерфейс, но и за счет разделения памяти, т. е. доступности ее для всех блоков. Однако вследствие того, что для *всех* потоков информации имеется только один путь, временные задержки в такой системе больше, чем в системах с многоканальными путями, описанных ниже. Убедительным дово-

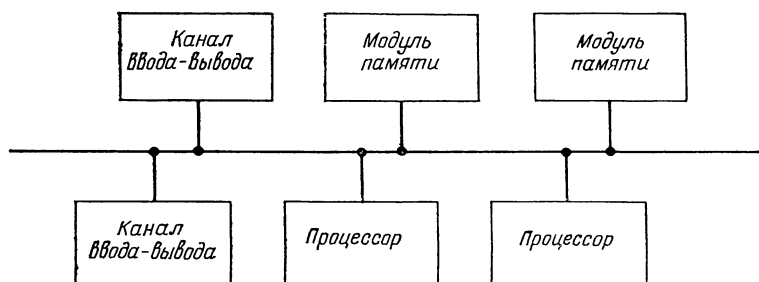


Рис. 2.1. Организация системы с общей или разделенной во времени шиной. Вариант с одной шиной.

дом в пользу рассматриваемой организации является ее гибкость и простота добавления или изъятия модулей и функциональных блоков.

Все модули подсоединены параллельно к одной шине, которая может иметь ширину одного полного слова или одного байта, или же по ней можно передавать одновременно только один бит информации. Чем меньше ширина шины, тем сложнее становятся функции управления.

Блоки процессоров и периферийных устройств могут быть подсоединены либо к одной двусторонней шине (рис. 2.1), либо к нескольким однонаправленным шинам (рис. 2.2). Во втором случае путь передачи образуется при помощи шинного модификатора (на схеме — крайний слева блок). При выборе того или иного способа здесь приходится сравнивать главным образом реализации одного двустороннего интерфейса и двух однонаправленных интерфейсов. Управляющая логика второго способа проще, однако первый способ имеет преимущество использования единственного буферного регистра в интерфейсе и более простого монтажа.

Возможно применение и более одной шины с временным разделением, как показано на рис. 2.3. Такая схема приближается по топологии к следующей рассматриваемой нами системной конфигурации — системе с перекрестной коммутацией. Отличительная особен-

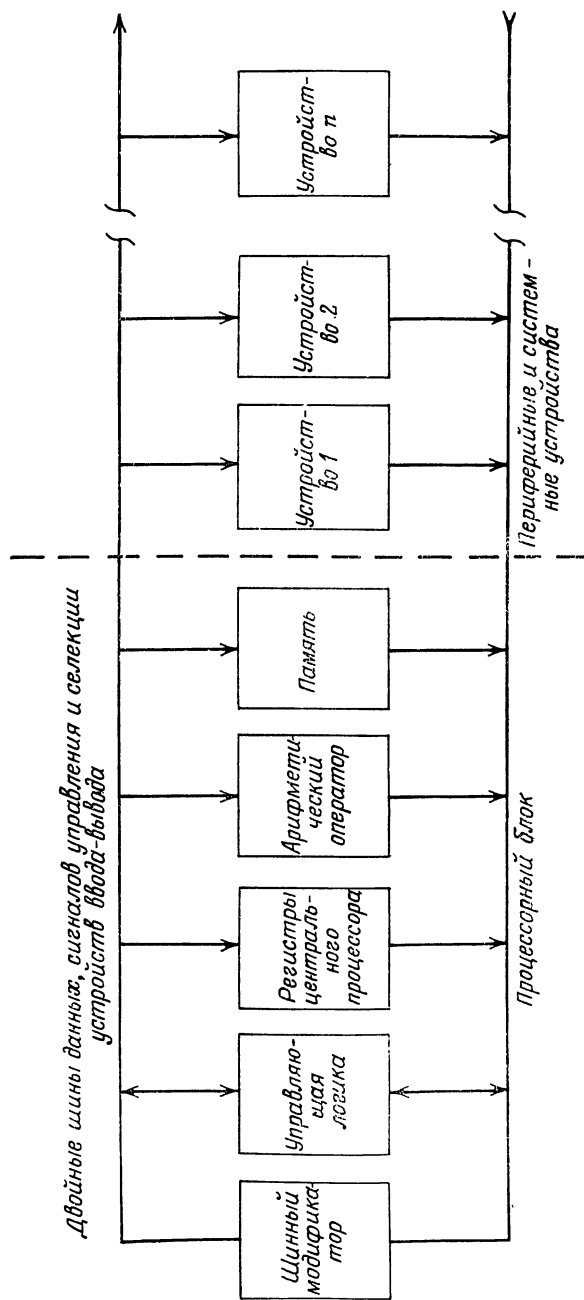


Рис. 2.2. Организация системы с общей или разделенной во времени шиной. Вариант с двумя однонаправленными шинами. (Схема соответствует организации системы GRI-99.)

ность системы с общей шиной состоит в том, что, если даже она содержит равное число процессоров и запоминающих устройств, они не могут функционировать одновременно из-за временного разделения путей передачи.

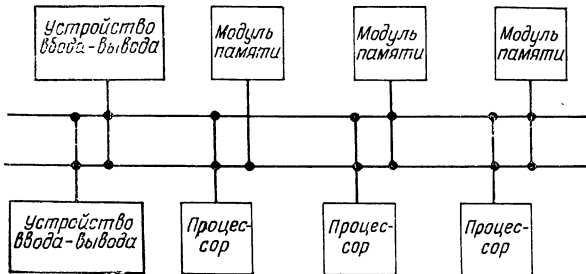


Рис. 2.3. Организация системы с многоканальной общей или разделенной во времени шиной.

Каждый пакет информации, переданный на шину, должен содержать данные, подлежащие передаче, и адрес блока, куда они должны быть направлены. Здесь не существует проблемы конфликтов между несколькими пакетами, поступающими к какому-либо блоку одновременно, так как в каждый момент времени шина содержит лишь один пакет, а другой источник информации должен ожидать, пока

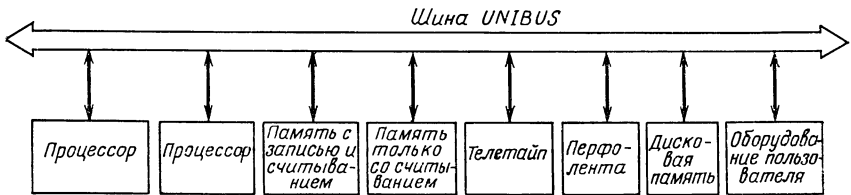


Рис. 2.4. Упрощенная блок-схема системы PDP-11 с шиной UNIBUS.

не освободится шина для приема пакета от этого источника. Хотя разрешение конфликтов производится автоматически и не представляет сколько-нибудь серьезной проблемы, конфликты все же возникают, и они существенно замедляют работу комплекса в целом. Каждый блок, подключенный к шине, должен содержать необходимую аппаратуру для распознавания адреса в пакете и для формирования соответствующей реакции.

В качестве примера одношинной системы рассмотрим систему PDP-11 фирмы Digital Equipment Corporation, в которой полностью использованы возможности шины UNIBUS (рис. 2.4). Шина содержит 56 проводов, из которых только 16 используются для передачи

данных (машинное слово ЭВМ PDP-11 состоит из 16 разрядов). Функции всех проводов шины указаны в приведенной ниже таблице.

Функция	Число проводов
Передача данных	16
Адрес (одного регистра памяти или устройства)	18
Управление (В, ИЗ, ПАУЗА, БАЙТ)	2
Главный синхросигнал (с указанием адреса и управляющей информации)	1
Подчиненный синхросигнал (реакция на главный синхросигнал)	1
Проверка на четность	2
Запрос шины (периферийным устройством)	4
Предоставление шины (процессором)	4
Другие сигналы запроса шины	3
Прерывание	1
Сигнал занятости шины	1
Различные сигналы управления	3
Всего	56

Недостаток, состоящий в том, что система способна одновременно выполнять только одну операцию передачи, уравнивается принципиальной простотой и возможностью наращивания системы. Выбор такой конфигурации, однако, может привести к недопустимо большому времени ожидания при обмене пакетами данных по мере расширения системы и возрастания нагрузки. Можно назвать еще одно преимущество системы с общей шиной; оно заключается в возможности полного блокирования (защиты) данных одним процессором, которое легко осуществляется путем кратковременной монополизации шины этим процессором. Для этого не нужны ни специальные программы, ни аппаратура.

К недостаткам рассматриваемой системной организации относится низкая надежность системы. Всегда существует опасность катастрофического отказа, если шина содержит активные элементы или если в системе имеется только один блок управления шиной. Надежность системы повышается, если применяется полностью пассивная шина без контроллера, т. е. если шина представляет собой простой кабель с отводами.

Примерами мультипроцессорных систем с организацией рассмотренного типа могут служить системы:

IBM Stretch.

UNIVAC LARC.

CDC 6600 — для передачи информации между центральной памятью и периферийными процессорами.

DIGITAL System 10.

Data General NOVA — для коммутации сообщений (хотя каждый процессор имеет собственную память) [36].

Контроллер Plessey Laboratories — для небольших телефонных коммутаторов [36].

Последний вариант рассматриваемого типа системной архитектуры предусматривает использование нескольких шин с временным разделением, причем число шин меньше числа процессоров (рис. 2.3). Этот вариант был рассмотрен Кэртином [47]. Такая топология обеспечивает некоторую избыточность и повышает общую скорость передачи данных.

СИСТЕМЫ С ПЕРЕКРЕСТНОЙ КОММУТАЦИЕЙ

Следующей системной организацией, которая может быть реализована на основе одновходовых функциональных блоков, является конфигурация с перекрестной коммутационной матрицей (рис. 2.5).

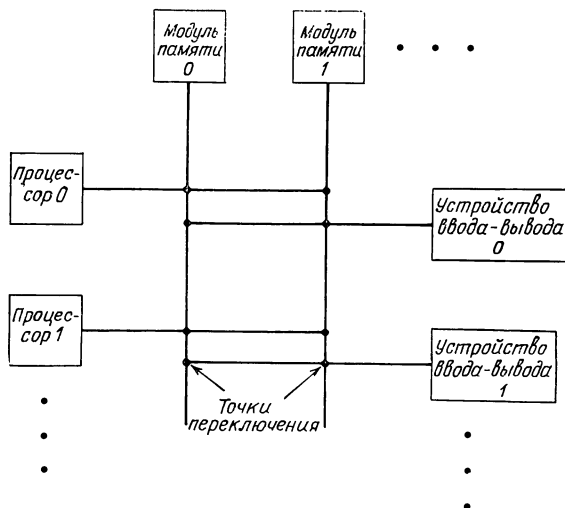


Рис. 2.5. Организация системы с перекрестной коммутацией.

Такая организация системы позволяет подсоединять любой модуль памяти к любому процессору или к любому устройству ввода-вывода. Между любыми двумя блоками устанавливается физический контакт на все время передачи информации. В отличие от коммутации с

временным разделением, реализуемой в системах с общей шиной, рассматриваемый метод переключения часто называют коммутацией с пространственным разделением. Он очень похож на метод, применяемый на большинстве центральных телефонных станций.

Хотя рассматриваемая система обладает несколько меньшей гибкостью, чем система с одной шиной, тем не менее она позволяет сравнительно просто добавлять новые модули, если коммутационная матрица обладает достаточной емкостью. Размеры системы не ограничены пропускной способностью отдельных функциональных блоков, так как все они подсоединяются через единственный вход.

Конфликты при запросах одного и того же модуля памяти разрешаются в коммутационной матрице каким-либо из нескольких возможных методов. Так как в системе устанавливается контакт на все время передачи, эффективная скорость передачи может быть выше, чем в одношинной системе с временным разделением. Кроме того, имеется возможность установить одновременно несколько путей передачи.

Коммутационная матрица полностью отделена от функциональных блоков и может быть построена также по модульному принципу, что упрощает ее расширение. Однако вследствие сложности функций, которые могут быть возложены на коммутатор, он может стать чрезмерно большим и сложным. Коммутационная матрица вместе с управляющими схемами при максимальной конфигурации системы H4400 фирмы Hughes (8 процессоров или контроллеров ввода-вывода и 16 модулей памяти) содержит в два с половиной раза больше элементов, чем один процессорный блок [31]. В другом случае вычислительная система содержит:

24 процессора.

32 модуля памяти.

Слова с 32 информационными и 4 контрольными разрядами.

Адреса с 16 информационными и 4 контрольными разрядами.

Число схем, требуемых для коммутатора такой системы, в 2—3 раза больше, чем для процессора ЭВМ IBM S/360, модель 75 [6].

В системе с перекрестной коммутацией интерфейсы отдельных блоков очень просты, так как они не должны ни разрешать конфликты, ни распознавать направляемые к ним данные. Эти функции выполняются логическими схемами коммутационной матрицы.

Устройства ввода-вывода могут подсоединяться к обоим контроллерам ввода-вывода точно так же, как в случае однопроцессорной системы, с привязкой к одному каналу или с распределением между несколькими каналами. Для обеспечения максимальной гибкости можно также ввести дополнительную коммутационную матрицу на той стороне коммутаторов, где находятся устройства ввода-вывода (рис. 2.6). Этот способ используется почти во всех больших вычислительных системах фирмы Burroughs, включая военные (D825)

и коммерческие (B5500, B6700 и B7700) системы¹. Интересный вариант организации предложен для системы Multi-Interpreter фирмы Burroughs, в которой использована группа одинаковых микропрограммных процессоров [18]. В этой системе одни и те же блоки

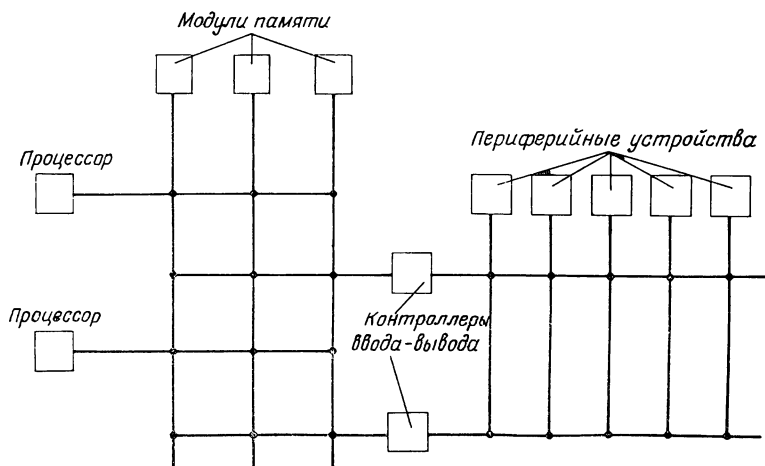


Рис. 2.6. Организация системы с перекрестной коммутацией и с отдельной коммутационной матрицей для устройств ввода-вывода.

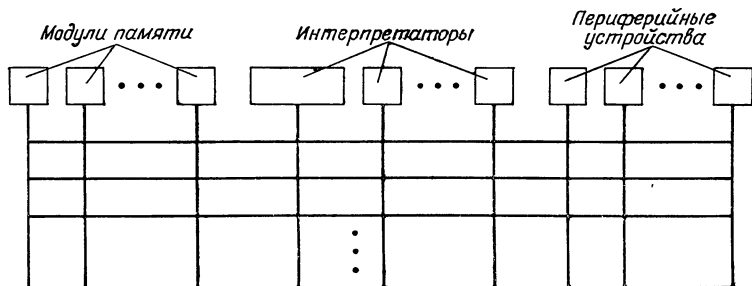


Рис. 2.7. Организация системы Multin-Interpreter фирмы Burroughs.

используются либо в качестве центральных процессоров, либо в качестве контроллеров ввода-вывода посредством перезагрузки блоков микропрограммной памяти. Это позволяет подключить все указанные периферийные устройства к той же коммутационной матрице, к которой подключены модули памяти и процессоры (рис. 2.7).

¹Следует заметить, что логика коммутации в системе D825 имеет распределенную, а в других системах фирмы Burroughs — централизованную структуру.

Еще один вариант перекрестной конфигурации реализован в одной из самых ранних мультипроцессорных систем — в системе TRW-400 фирмы Thompson-Ramo-Wooldridge. Эта система, которую называют также «полиморфной ЭВМ», имеет только одну коммутационную матрицу, точно так же, как и система Multi-Interpreter. Однако к матрице системы TRW-400 подсоединены периферийные

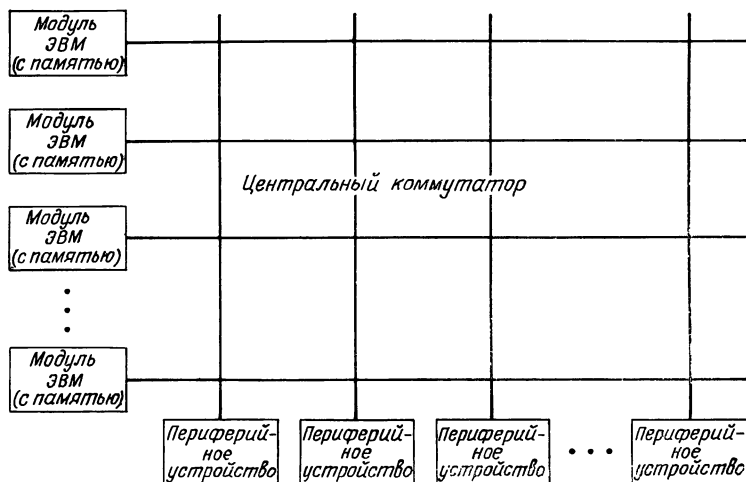


Рис. 2.8. Организация системы RW-400 фирмы Ramo-Wooldridge — «Полиморфная ЭВМ».

устройства и самостоятельные ЭВМ, каждая из которых обладает собственной памятью (рис. 2.8). Блоки процессоров и памяти не рассматриваются здесь как самостоятельные устройства. В системе была сделана попытка осуществить связь непосредственно между процессорами и перекрестный доступ к запоминающим устройствам путем замыкания соответствующего набора пересечений. Громоздкость и неэффективность этого метода очевидны.

Ниже приведены примеры систем с перекрестной коммутацией.

RW-400 фирмы Ramo-Wooldridge. Одна из самых первых систем, разработанных для ВВС США. Примечательная особенность этой системы — отсутствие отдельных блоков памяти. Каждое запоминающее устройство привязано к своему процессору и образует, таким образом, самостоятельный центральный процессор. Для управления коммутационной матрицей используется специальный управляющий процессор. Этот процессор содержит большое число элементов и снижает общую надежность системы. Хотя некоторые функциональные блоки были изготовлены, система RW-400 в целом так и не была закончена.

D825 фирмы Burroughs [AN/GYK-3(V)]. Наряду с системой RW-400 она относится к первым истинным мультипроцессорным системам. Коммутационное устройство системы D825 было распределено между блоками памяти. Нормальный вариант системы содержит три модуля ЭВМ, каждый из которых подсоединен к одной из пяти коммуникационных шин, доступных любому модулю памяти. Остальные две шины памяти обычно использовались для устройств ввода-вывода. Интерфейсы шин памяти содержат логические схемы, необходимые для приема и составления очереди запросов на доступ к памяти, поступающих одновременно. Модули ЭВМ содержат регистры относительного адреса для облегчения перемещения программ при мультипрограммной работе.

CDC 6600. Использовалась для связи между периферийными процессорами и каналами ввода-вывода.

H 4400 фирмы Hughes. Состоит из 8 процессоров или контролеров устройств ввода-вывода и 16 модулей памяти.

PRIME (Калифорнийский университет). Для управления внешним доступом используется перекрестный коммутатор [33].

Multi-Interpreter фирмы Burroughs с микропрограммным управлением [18].

МНОГОШИННЫЕ — МНОВХОДОВЫЕ СИСТЕМЫ

Эти системы являются примером еще одной конфигурации, обеспечивающей несколько путей одновременной передачи информации (рис. 2.9). Для осуществления организации такого типа очень важ-

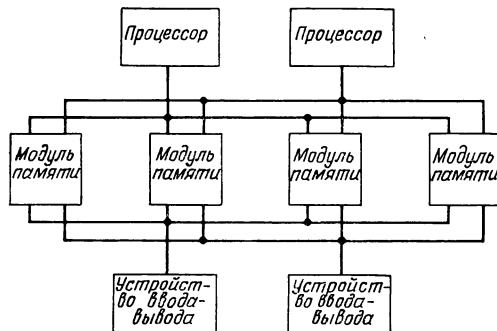


Рис. 2.9. Упрощенная схема организации многошинной — мновходовой системы.

но, чтобы модули памяти имели по несколько входов и управляющие схемы для разрешения конфликтов в тех случаях, когда два или

большее число процессоров или устройств ввода-вывода требуют доступа к одному и тому же модулю памяти в пределах одного цикла памяти.

Реализация такой топологической схемы обходится дешевле, чем перекрестный коммутатор, так как здесь меньше точек, в которых нужно разрешать конфликты. Это преимущество, однако, компенсируется тем фактом, что максимальная возможная конфигурация

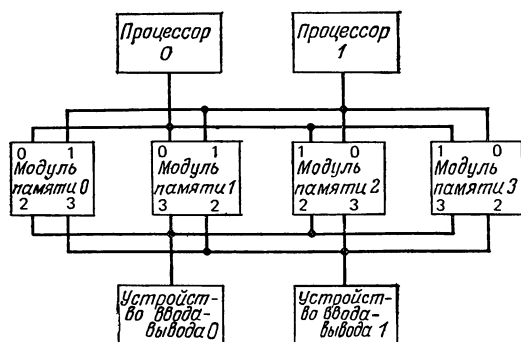


Рис. 2.10. Назначение приоритетов доступа к памяти в системе с многоходовой памятью.

ограничена числом входов модуля памяти. Объяснение этого, по-видимому, заключается в том, что основные конструктивные и производственные решения были приняты задолго до того, как были рассмотрены все возможные и желательные варианты системной организации. Чтобы увеличить число точек взаимного соединения, можно было бы добавить «расширители» входов, или мультиплексоры. Однако скорость передачи такой системы все же не может быть больше, чем допустимая скорость для одного входа. Другая особенность модулей памяти с несколькими входами состоит в том, что в них часто возникают приоритеты, зависящие от физической точки соединения. Эти приоритеты можно положить в основу разрешения конфликтов при одновременных запросах, причем первенство отдается тому процессору или блоку ввода-вывода, который обращается к «своему» модулю памяти, как показано на рис. 2.10.

Как и в системах с перекрестной коммутацией, ширина пути передачи данных выбирается из условий удобства и экономичности. Если основной единицей при хранении информации является слово, а ширина пути передачи данных меньше, чем слово, то в точки интерфейса нужно включить специальные регистры для составления и разложения слов, а также специальные схемы контроля, гарантирующие, что в путь передачи не будет введена другая информация

и что он не будет разорван, когда передача слова еще не окончена.

Нет необходимости, чтобы каждый модуль памяти подсоединялся к любому процессору. Фактически в некоторых системах существенную роль играет то обстоятельство, что каждый процессор имеет «собственную память», в которой хранятся специальные таблицы для функций управления, восстановления, распределения частных ресурсов и т. д. (рис. 2.11). Однако с наличием собственной памяти

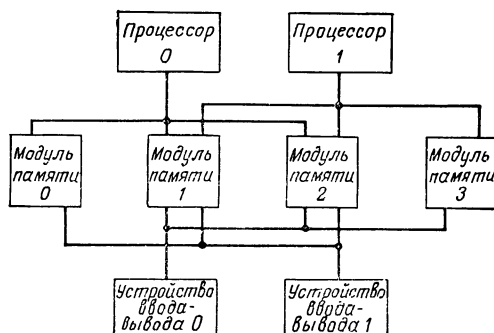


Рис. 2.11. Многоходовая система с «собственными» модулями памяти.

связаны и недостатки, такие, как уменьшение надежности, усложнение ремонта и т. д. Если какой-либо процессор вышел из строя и прерванную работу надо закончить с помощью другого процессора, то может оказаться, что новому процессору будет недоступна управляющая информация, необходимая для выполнения этой задачи.

Система сильно теряет в универсальности, если каждый процессор не имеет доступа к любому модулю памяти. При этом утрачивается гибкость в обеспечении перемещения объектных программ и в работе операционной системы. Преимущества работы с единственным экземпляром операционной системы очевидны. Серьезный недостаток такой организации, когда на доступ процессоров к модулям памяти наложены ограничения, связан с возможностью выхода из строя модуля памяти или же процессора.

Во всех системах подобной конфигурации модуль памяти должен распознавать и обрабатывать запросы на доступ к определенным ячейкам памяти. Его устройство управления также обеспечивает разрешение конфликтов при одновременном обращении и уведомляет запрашивающее устройство о том, что ему разрешен доступ к памяти.

Ниже приведены примеры многоспинных—многоходовых систем.

UNIVAC 1108. Содержит до 4 банков, или модулей памяти. Использует два блока управления многоходовой памятью, каждый

из которых может обрабатывать и разрешать конфликты между пятью путями передачи данных — двумя от контроллеров устройств ввода-вывода (с высшим приоритетом) и тремя — от процессоров.

HIS 635.

CDC 3600.

XDS-Sigma 7. Оснащена шестивходовой памятью с фиксированными приоритетами.

HIS (GE) 645 (система MULTICS).

IBM S/360, модель 67. Многовходовые запоминающие устройства применяются повсеместно в моделях систем IBM S/360 и S/370.

UNIVAC 1832, AN/UYK-7 и ARTS-III.

АСИММЕТРИЧНЫЕ И НЕОДНОРОДНЫЕ СИСТЕМЫ

Хотя до сих пор это специально не подчеркивалось, в основе всех рассмотренных систем лежало предположение об однородности процессоров в составе одной группы и аналогично об идентичности модулей памяти и устройств ввода-вывода. Однако это условие выполняется не всегда. На практике могут существовать большие различия между элементами системы, даже относящимися к одному типу. Обычно это является результатом объединения универсальных и специализированных устройств в составе одной системы или создания систем, состоящих только из универсальных устройств, часть которых предназначена для специальных целей. Следует еще раз подчеркнуть, что рассматриваемые типы организации можно квалифицировать как истинные мультипроцессорные системы, так как они удовлетворяют следующим основным критериям:

В них имеется разделение основной памяти или общий доступ к ней.

Они работают под управлением единой интегрированной операционной системы.

В них осуществлено аппаратное и программное взаимодействие на уровне программ и задач в качестве основного режима работы.

Одним из самых ранних примеров системы с организацией такого типа была система PILOT Национального бюро стандартов. Она состоит из трех процессоров, каждый из которых выполняет одну из следующих основных задач:

Вычисления.

Регистрация и учет.

Ввод и вывод информации.

Эстрин (Estrin) предложил принципиальную конструкцию неоднородной мультипроцессорной системы, состоящую из трех частей:

Неизменной части системы.

периферийные процессоры выполняют большинство функций контроллеров устройств ввода-вывода, имеющих в других системах.

Один из периферийных процессоров обычно обслуживает только пульт оператора с дисплеем, а другой выполняет роль главного системного монитора, который ставит задачи центральному процессору и остальным периферийным процессорам. Интересной особенностью системы является то, что прерывания не передаются от процессора к процессору. Все управляющие директивы и ответы на них передаются в обоих направлениях методом «почтового ящика».

Известно несколько других проектов и экспериментальных исследований неоднородных систем. Одним из этих проектов является предложенный Апенбреннером с соавторами «истинный мультипроцессор» [3]. Главной особенностью этого проекта были типы строительных блоков, а именно последовательные устройства управления (подобные обычным процессорам, но не способные выполнять арифметические операции) и совместно используемый набор функциональных арифметических блоков, таких, как суммирующие и множительные устройства.

Следует заметить, что система Multi-Interpreter фирмы Burroughs может также рассматриваться как представитель этой категории мультипроцессорных систем, так как любой интерпретатор по мере необходимости может быть определен или переопределен в реальном времени путем изменения микропрограмм, в результате чего он превращается в «стандартный» процессор, «специальный» процессор или процессор ввода-вывода.

СИСТЕМЫ С МАГИСТРАЛЬНОЙ ОБРАБОТКОЙ

Эти системы основаны на применении иного метода, обеспечивающего выполнение операций в параллельном режиме или в режиме совпадения. Они обычно фигурируют во всех материалах, посвященных мультипроцессорам. В то время как в матричных системах совпадение достигается одновременным выполнением одних и тех же операций в нескольких потоках данных, в ЭВМ с магистральной обработкой одна и та же операция повторно выполняется над некоторыми переменными в одиночном потоке данных.

Лучшим примером операций такого типа может служить операция сложения с плавающей запятой двух векторов, содержащих по n элементов¹, $\bar{A} + \bar{B} = \bar{C}$.

¹ Для более полного знакомства с аппаратурой, применяемой при магистральном методе обработки, и с преимуществами этого метода рекомендуется статья Халлена и Флинна «Использование магистрального метода обработки при выполнении арифметических функций», IEEE Trans. Comput., 880—886 (август 1972 г.).

Операция сложения с плавающей запятой может быть разбита на пять шагов:

1. Нормализация (исключение нулей в старших разрядах путем сдвига) обоих операндов с соответствующим изменением их порядков.

2. Сравнение порядков.

3. Сдвиг мантиссы с меньшим порядком для выравнивания с мантиссой с большим порядком.

4. Прямое сложение мантисс.

5. Нормализация результата.

Для примера выберем следующие значения времени выполнения каждого шага:

1. 100 нс.

2. 60 нс.

3. 100 нс.

4. 120 нс.

5. 100 нс.

Тогда каждая операция сложения с плавающей запятой потребует по крайней мере 480 нс. В машине с магистральной обработкой каждый шаг выполняется отдельным комплектом аппаратуры, после чего операнд перемещается на следующую позицию. Затем в магистраль вводятся новые операнды и т. д.:

Время							
Шаг 1	a_1, b_1	a_2, b_2	a_3, b_3	a_4, b_4	a_5, b_5	...	a_{i+4}, b_{i+4}
Шаг 2		a_1, b_1	a_2, b_2	a_3, b_3	a_4, b_4	...	a_{i+3}, b_{i+3}
Шаг 3			a_1, b_1	a_2, b_2	a_3, b_3	...	a_{i+2}, b_{i+2}
Шаг 4				c_1	c_2	...	c_{i+1}
Шаг 5					c_1	...	c_i

Теперь необходимо установить продолжительность каждого шага 120 нс, с тем чтобы иметь достаточно времени для завершения шага 4. Первый результат будет получен через 600 нс вместо 480 нс. Однако каждый последующий результат будет появляться на выходе через 120 нс. Очевидно, подобный метод может быть полноценным только при условии, что выполняемые вычисления подходят для него, а программное обеспечение и аппаратура автоматически его реализуют.

На рис. 2.13 показаны два примера того, как можно по-разному соединить набор базовых арифметических блоков, чтобы выполнять различные операции. Эта схема соответствует магистральному арифметическому устройству системы ASC фирмы Texas Instruments.

В статье Бэра [6], посвященной большим системам, поставлены следующие три вопроса относительно ценности и полезности применения магистрального метода в универсальных арифметических устройствах:

Сложение с плавающей запятой Умножение с фиксированной запятой

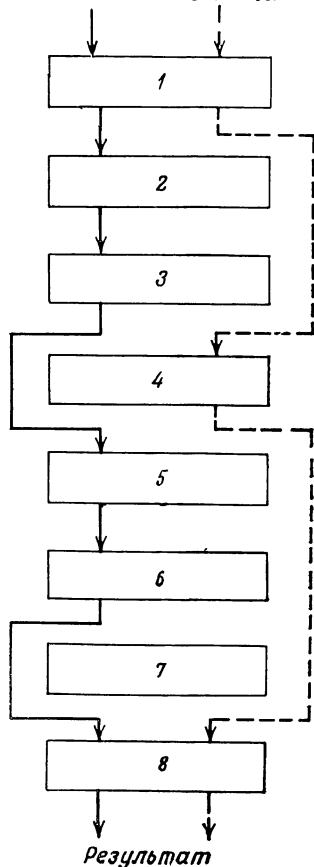


Рис. 2.13. Примеры магистральной обработки для двух различных команд (система ASC фирмы Texas Instruments).

1 — приемный регистр, 2 — вычитание порядков, 3 — выравнивание, 4 — умножение, 5 — сложение, 6 — нормализация, 7 — накопление, 8 — выход.

Можно ли применить магистральный метод к двум различным операциям, например к сложению и умножению?

Можно ли применить магистральный метод к одной и той же операции над различными форматами данных, например к коротким и длинным операциям с плавающей запятой?

Можно ли применить магистральный метод к двум операциям одного и того же типа, при одинаковом формате данных, но с операндами, которые ничем не связаны друг с другом? Иначе говоря, можно ли применить магистральный метод к двум операциям $A=B+C$ и $D=B+C$? (Заметим, что две операции $A=B+C$ и $D=A+E$ не могут быть выполнены с использованием магистрального метода.)

Методы выполнения операций с сильным расчленением и взаимным перекрытием, используемые, например, в ЭВМ IBM S/360, модель 91, также являются вариантами магистрального метода. Однако они резко отличаются от «векторного» или «поточного» магистрального метода, который был рассмотрен здесь. Главное отличие состоит в том факте, что в системах ASC фирмы Texas Instruments и STAR-100 фирмы Control Data магистральный метод использован для выполнения одной и той же арифметической операции над последовательностью операндов по мере их продвижения по магистрали.

Главная забота при использовании возможностей магистральной системы — обеспечить заполнение магистрали. Для большинства систем это требование не очень критично, так как магистраль содержит не так уж много ступеней, или шагов. По-

этому даже по основному замыслу и целям магистральные системы, примененные в модели 91 и в системе STAR фирмы CDC, сильно различаются.

Система STAR представляет собой истинную линейно-матричную систему, и качество ее работы сильно понизится, если обработка не будет состоять в выполнении длинной последовательности одинаковых операций над потоком данных. В модели 91 нет такого большого числа шагов и не требуются такие длинные серии операций. В некоторых системах обеспечение загруженности магистрали возложено на программиста. В системе S/360, модель 91, максимальное совпадение операций достигается при помощи аппаратных средств.

Очевидно, магистральный метод дает тем больший выигрыш, чем сложнее выполняемые операции. Как указывают Халлин и Флинн (Hallin, Flynn), при использовании магистрального метода эффективность повышается на 40% для операции сложения и на 230% для операции умножения.

Последнее замечание касается обработки прерываний в магистрали. Так как в магистральном процессоре должно выполняться большое число операций над различными наборами данных в режиме совпадения, проблема распознавания типа прерывания сильно усложняется.

ПРИМЕРЫ СИСТЕМ С МАГИСТРАЛЬНОЙ ОБРАБОТКОЙ

IBM S/360, модель 91. Магистральная обработка применена в обоих блоках обработки данных с плавающей запятой; магистраль состоит всего лишь из двух шагов.

CDC 7600. Во всех девяти арифметических функциональных блоках применен магистральный метод с различным числом шагов в каждом.

CDC STAR-100. В дополнение к другим формам обеспечения параллельной обработки, таким, как применение периферийных станций, которые настолько велики, что могут рассматриваться как самостоятельные системы, выполняющие операции по вводу-выводу и по управлению файлами, и параллельных арифметических блоков в центральном процессоре, здесь применены также две магистрали и «поточный» блок. Система задумана как ЭВМ научно-исследовательского назначения с весьма высокими характеристиками, оптимально выполняющая операции со строками и матрицами (отсюда ее название STAR—STring ARray). Число шагов в магистрали различно для различных операций. Например, для выполнения арифметических операций с плавающей запятой имеется около тридцати шагов длительностью 40 нс.

ASC (Advanced Scientific Computer — усовершенствованная научно-исследовательская ЭВМ фирмы Texas Instruments). Содержит

центральный процессор магистрального типа, выполняющий операции по обработке как векторных, так и скалярных величин. 8 периферийных процессоров придают системе некоторые свойства, присущие асимметричным мультипроцессорным системам. Имеются центральные процессоры с 1, 2, 3 и 4 магистралями. Продолжительность основного цикла работы магистрали составляет 60 нс.

ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ. МАТРИЧНЫЕ И ВЕКТОРНЫЕ ПРОЦЕССОРЫ

Во всех рассмотренных выше конфигурациях, за исключением магистральной, совпадение или параллельность выполнения операций достигается за счет увеличения числа процессорных блоков, каждый из которых способен автономно выполнять отдельную последовательность команд над отдельным потоком данных. Главной особенностью матричных и векторных операций является то, что одна и та же команда должна выполняться над большой совокупностью элементарных данных, которые как-то связаны между собой. Часто эту операцию можно производить над всеми этими данными одновременно. В этом случае достаточно иметь только одно устройство управления, которое могло бы дешифровать одиночный поток команд и вырабатывать управляющие сигналы для некоторой совокупности, или матрицы элементарных процессоров.

Схематическое изображение конфигурации матричного процессора приведено на рис. 2.14. Так как отдельные элементарные процессоры не являются законченными центральными процессорами и не могут работать самостоятельно, рассматриваемые здесь структуры являются скорее параллельными системами, чем мультипроцессорами. Единственная степень свободы элементарного процессора, работающего в матричной системе, состоит в том, что он может выполнять или не выполнять некоторую частную команду. Сигналы, реализующие эту возможность, вырабатываются и распределяются также управляющим процессором.

Первым большим достижением в этой области была разработка системы SOLOMON I [32]. За системой SOLOMON I последовала система SOLOMON II, а затем — система ILLIAC IV [8]. Работа над обоими вариантами системы SOLOMON представляла собой предварительное исследование, которое привело к разработке проекта системы ILLIAC IV; последняя система была построена и находится в настоящее время в эксплуатации. Грандиозный проект системы SOLOMON предусматривал 1024 процессорных, или исполнительных, элемента, объединенных в одну матрицу и управляемых одним управляющим процессором. Каждый из этих элементарных процессоров обладал невысокой производительностью и небольшой собственной памятью (емкостью всего 4096 битов) [45].

В систему SOLOMON входит матрица размерностью 32×32 , состоящая из элементарных процессоров, которыми управляет один центральный управляющий процессор. Центральное устройство управления содержит память программ, обладает средствами для поиска и интерпретации команд, хранящихся в памяти, и при помощи многомодальной логики может обеспечить выполнение этих команд в процессорной матрице. Таким образом, в каждый данный момент времени несколько элементарных процессоров системы могут

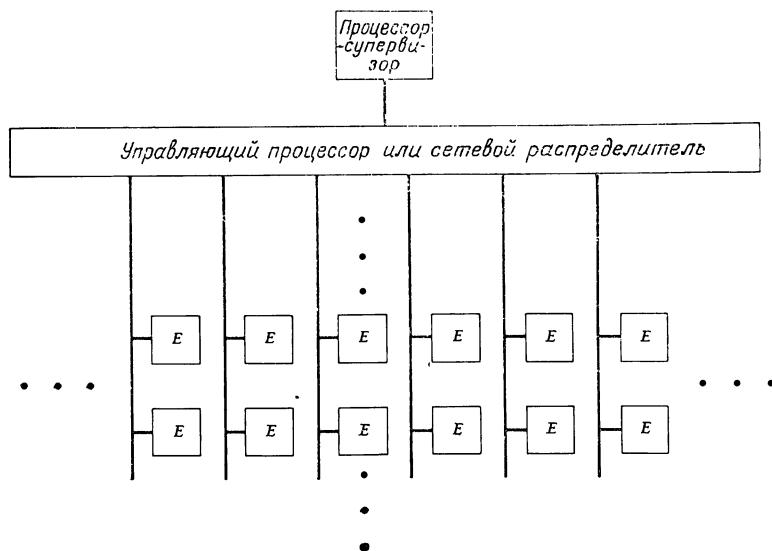


Рис. 2.14. Базовая конфигурация матричного процессора.

Е — исполнительные, или элементарные процессоры, обладающие большинством возможностей автономного процессора и имеющие собственную память.

выполнять одну и ту же команду над операндами, хранящимися в одноименных ячейках памяти различных элементарных процессоров. Благодаря тому, что каждый элементарный процессор снабжен собственной ферритовой памятью, эти операнды могут быть различными.

Каждый элементарный процессор может общаться с четырьмя своими «соседями». Свободные входы крайних элементарных процессоров, не имеющих полного набора «соседей», соединены с устройствами ввода-вывода. Система центрального управления может передавать константы для использования их во всех элементах матрицы.

В каждом элементарном процессоре матрицы имеется регистр режима работы. Команды, поступающие в элементарный процессор от центрального устройства управления, выполняются только тогда, когда сигналы режима работы, переданные центральным устройством управления, совпадают с режимом работы, записанным на регистре элементарного процессора [38].

Проект SOLOMON был подвергнут критике как за свои размеры, так и за «довольно неудобный механизм передачи информации от одного элементарного процессора к другому и относительно малую

гибкость, обусловленную фиксированной локальной структурой памяти» [45].

Проекты SOLOMON I и SOLOMON II отличались друг от друга в основном техническим исполнением и быстродействием элементарных процессоров.

Преемником системы SOLOMON явилась система ILLIAC-IV. Базовая структура системы должна была включать 256 элементарных процессоров, каждый из которых обладал значительно более высокой производительностью по сравнению с элементарными процессорами, предложенными для системы SOLOMON. Все 256 элементарных процессоров сгруппированы в четырех квадрантах по 64 в каждом (матрица 8×8), причем каждый квадрант имеет свое собственное устройство управления. Каждый элементарный процессор снабжен собственной памятью значительно большей емкости (2000 64-разрядных слов), имеет прямой доступ к системе ввода-вывода и связан с четырьмя соседними элементарными процессорами (как и в проекте SOLOMON). В качестве управляющего процессора использована ЭВМ B6700 фирмы Burroughs, которая выполняет оперативные программы (такие, как программы загрузки) и вспомогательные программы (такие, как программы компиляции). К настоящему времени изготовлен и находится в эксплуатации один квадрант, состоящий из 64 элементарных процессоров и одной ЭВМ B6700. Эта часть системы, обладающая всеми свойствами рассматриваемой системной организации, вероятно, вполне достаточна для оценки метода параллельной матричной обработки информации.

Хотя предварительная оценка показала, что система ILLIAC IV может произвести обращение матрицы размерностью 700×700 за 1 с, причем стоимость этой процедуры составляет приблизительно полтора доллара, главной проблемой и препятствием для реализации этой формы мультипроцессирования является трудность подготовки программ, которые использовали бы все возможности системы, и разработки трансляторов и других вспомогательных программ, которые могли бы автоматически анализировать рабочие программы и вырабатывать сигналы для параллельной работы.

Необычность системы ILLIAC IV состоит еще и в том, что в ней предусмотрена иерархия мультипроцессоров. В системе имеется управляющий процессор для каждого квадранта, состоящего из 64 элементарных процессоров и, кроме того, центральный управляющий процессор, который координирует работу четырех квадрантов (рис. 2.15).

Ниже приведены примеры других параллельных процессоров, которые уже созданы или находятся в процессе разработки.

STARAN фирмы Goodyear Aerospace. В этой системе используются как матричная организация, так и метод ассоциативной обработки, благодаря чему повышается степень параллельности

работы. Система предназначена специально для обработки большого объема данных радиолокационного слежения при управлении воздушным движением, хотя ее можно использовать и в других областях, таких, как хранение и поиск данных. Каждый элементарный процессор системы не обладает большим быстро-

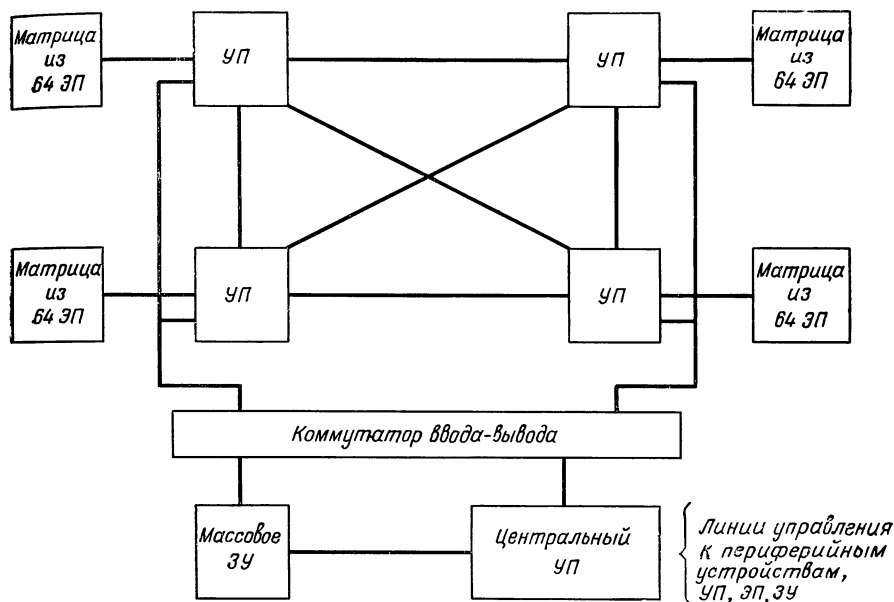


Рис. 2.15. Организация полной (четырехквadrатной) системы ILLIAC IV.
УП — управляющий процессор, ЭП — элементарный процессор.

действием, однако сочетание ассоциативных и матричных характеристик дает в результате довольно высокую общую производительность при обработке радиолокационных данных. Базовая конфигурация системы STARAN может включать до 32 ассоциативных матриц, каждая из которых имеет память емкостью 65K бит (256×256) и содержит 256 элементарных процессоров. PERE (ансамбль параллельных элементарных процессоров). Система была разработана для Управления перспективных разработок по противоракетной обороне армии США фирмами Bell Telephone Laboratories, System Development Corporation и Honeywell (последняя создала прототип упрощенного варианта системы). Общий размер окончательной конфигурации еще не определен. Система содержит большой асимметричный мультипроцессор (CDC 7600), используемый в качестве устройства управления (рис. 2.16), и дополнительные специализированные устройства:

блок управления корреляционными устройствами, блок управления арифметическими устройствами и блок управления ассоциативными выходными устройствами. В системе также использовано сочетание ассоциативных и матричных методов для получения наиболее высокой производительности.

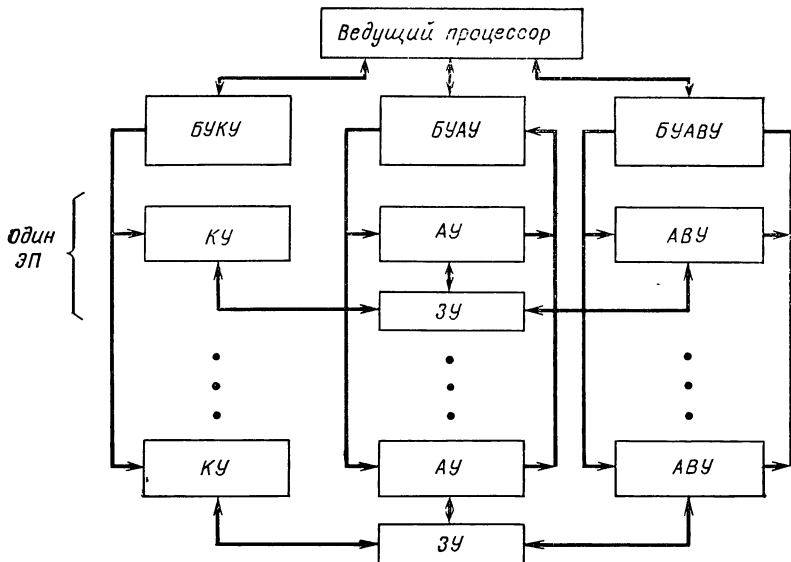


Рис. 2.16. Организация системы РЕРЕ.

ЭП — элементарный процессор, АУ — арифметическое устройство, КУ — корреляционное устройство, АВУ — ассоциативное выходное устройство, БУАУ — блок управления арифметическим устройством, БУКУ — блок управления корреляционным устройством, БУАВУ — блок управления ассоциативным выходным устройством.

OMEN-60 (ортогональная ЭВМ) фирмы Sanders Associates. Управляющим устройством в системе является ЭВМ PDP-11 фирмы DEC. Доступ к памяти может осуществляться по горизонтали (к 16-разрядным словам) или по вертикали (к срезам разрядов). ЭВМ PDP-11 используется в качестве горизонтального арифметического устройства, а специальный процессор — в качестве вертикального. Подобная организация особенно хорошо приспособлена для решения задач линейной алгебры, нахождения быстрых преобразований Фурье и т. д.

СИСТЕМНЫЕ ОРГАНИЗАЦИИ, ОРИЕНТИРОВАННЫЕ НА УСТОЙЧИВОСТЬ К СБОЯМ И ОТКАЗАМ

Потенциальные возможности системных организаций особенно хорошо проявляются в области специальных вычислительных систем, от которых требуется очень высокая устойчивость против сбоев и

отказов. Такая устойчивость у любого мультипроцессора во много раз выше, чем у однопроцессорных систем. Тем не менее можно уделить особое внимание именно этому аспекту эффективности системы, используя избыточные элементы, которые при необходимости могут дублироваться и перегруппироваться так, как это делается в мультипроцессорах. Для определения надежности любой заданной конфигурации приходится затрачивать значительные усилия на анализ конструкции системы. Однако форму исследуемой конфигурации

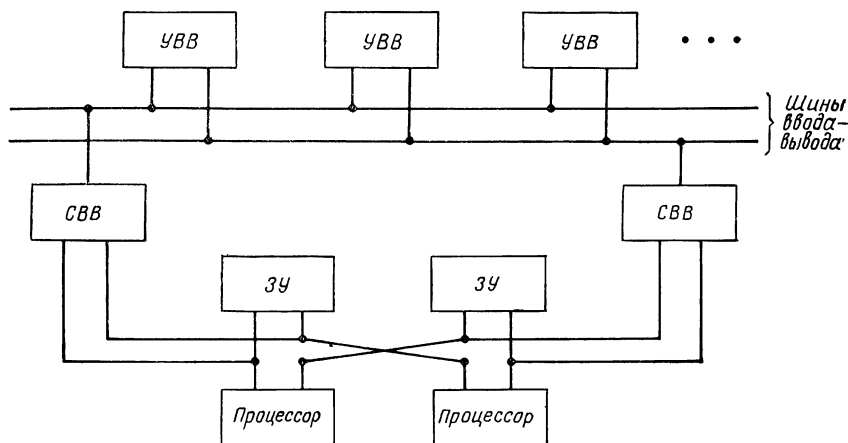


Рис. 2.17. Система, устойчивая к сбоям (ЭВМ SIRU лаборатории приборостроения Массачусетского технологического института).

УВВ — устройство ввода-вывода, СВВ — сенсер ввода-вывода.

обычно определяют как раз не конструктивные соображения. Более подробное обсуждение этого вопроса можно найти в работе [31].

Одна из систем, которая не может считаться ни мультипроцессорной, ни даже многомашинной, была разработана в лаборатории приборостроения Массачусетского технологического института и известна под названием ЭВМ SIRU (часть инерциального опорного блока)¹; ее схема представлена на рис. 2.17. Хотя она содержит два процессора и устройство, которое может быть отнесено к устройствам связи мультипроцессоров, она не является истинной «мультисистемой». Оба процессора работают по одной и той же программе, выполняя синхронно одну и ту же последовательность команд над одним и тем же потоком данных. Один из процессоров является «активным» в том смысле, что вырабатывает команды «запись». Оба запоминающих устройства имеют одинаковое содержимое, по-

¹ Крисп, Джилмор и Гопкинс, «SIRU — новый принцип конструкции инерциальной системы для обеспечения надежности и ремонтнопригодности в полете». Отчет E-2407 (май 1969 г.) лаборатории приборостроения Массачусетского технологического института.

сколько они реагируют на все команды записи. Однако запросы «чтение» выполняет только память первичного процессора. Система внутренней связи обеспечивает непрерывный контроль исправности различных модулей системы и немедленную перестройку ее конфигурации для назначения новых активных или первичных блоков в зависимости от конкретных условий. Очевидно, основным назначением подобной структуры является достижение высокой готовности при минимальных потерях времени.

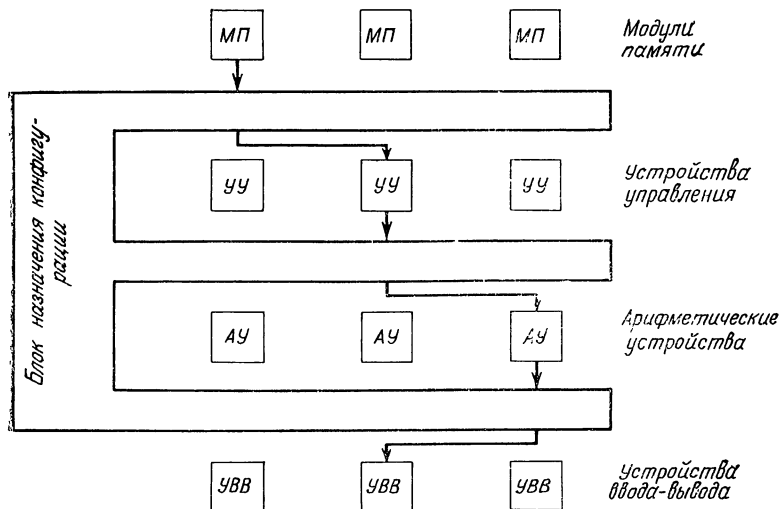


Рис. 2.18. Система, устойчивая к сбоям (модульная ЭВМ фирмы Hamilton Standard).

Национальное управление по авиации и исследованию космического пространства (NASA) также разработало в Исследовательском центре по электронике систему с высокой готовностью (рис. 2.18) [40, 43, 44]. Работа была выполнена фирмой Hamilton Standard. Система включает три группы элементов, каждая из которых содержит запоминающее устройство, устройство управления, арифметическое устройство и устройство ввода-вывода. Кроме того, имеется блок назначения конфигурации, осуществляющий соединение функциональных блоков, необходимое для поддержания работоспособности системы в целом. Блок назначения конфигурации может также подсоединить несколько блоков памяти к одному процессору. Однако система в целом не является мультипроцессором, так как в ней не применяется разделение памяти. Здесь просто осуществлен один из способов сохранения высокой готовности двух или трех однопроцессорных систем.

Возможны следующие режимы работы системы:

1. Три ЭВМ решают одну и ту же задачу и сравнивают свои результаты в те периоды времени, когда требуется особенно высокая надежность, например при запуске баллистической ракеты. Окончательный результат определяется методом голосования; иначе говоря, если одно из независимо полученных решений отличается от двух других, то за верный результат принимают одно из двух совпавших решений.

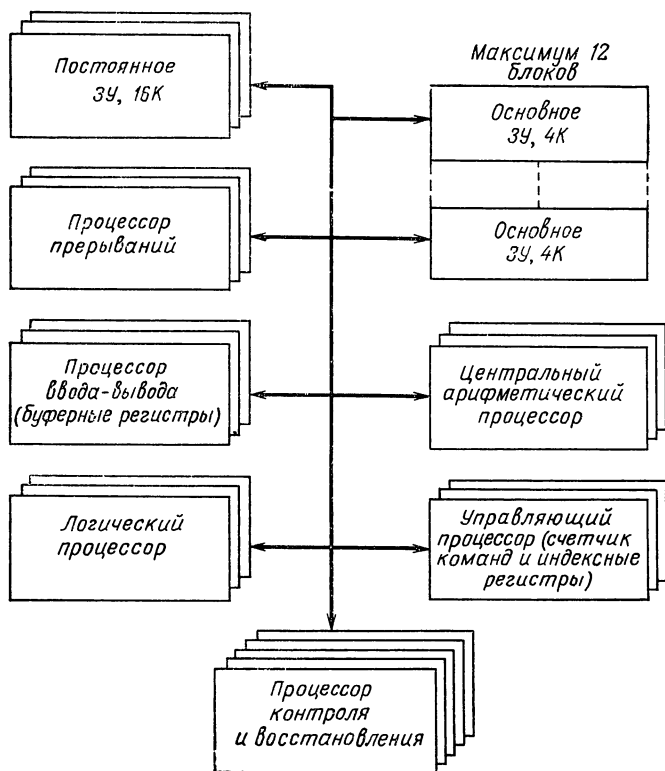


Рис. 2.19. Организация системы STAR фирмы Jet Propulsion Laboratory [41].

2. Три ЭВМ решают независимые задачи.

3. Реализуется один из методов организации работы нескольких ЭВМ в режиме on-line при минимальном резервировании.

Система обладает большими потенциальными возможностями, но очень сложна из-за большого количества коммутационного оборудования. Для достижения высокой надежности системы необходим безотказный блок назначения конфигурации. Поэтому ее преимущества по сравнению с эквивалентным мультипроцессором сомнительны [31].

Блок назначения конфигурации в этой системе является по существу перекрестной коммутационной матрицей, и поэтому здесь возникают все проблемы, связанные со сложностью структуры, кото-

рые присущи и другим системам с коммутационными матрицами.

Другой системой, устойчивой к сбоям и отказам, в которой также используются множественные блоки, по которой не относится к мультипроцессорам, является система STAR (Self-Testing and Repairing — самоконтроль и самовосстановление) фирмы Jet Propulsion Laboratory [4, 5]. Множественные модули здесь используются только

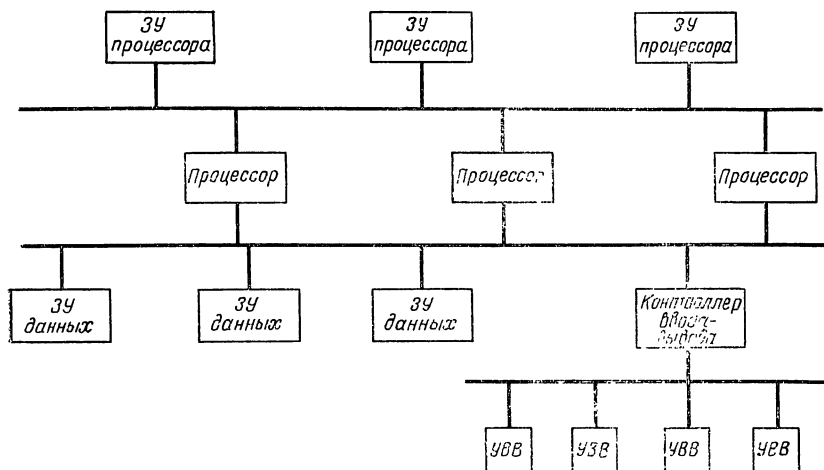


Рис. 2.20. Система, устойчивая к сбоям (ЭВМ ACGN лаборатории приборостроения Массачусетского технологического института).

УВВ — устройство ввода-вывода.

для обнаружения отказа и дублирования. Вышедший из строя модуль устраняется из системы и заменяется исправным (рис. 2.19).

Центральным устройством системы STAR является монитор с ферритовой памятью, получивший название контрольно-восстановительного процессора. Если обнаружена ошибка, он возвращает систему к выполнению программы с заранее установленной контрольной точки. В состав каждой 32-разрядной команды входят четыре контрольных разряда для проверки по модулю 15. Иначе говоря, каждое 32-разрядное слово, рассматриваемое как двоичное число, должно делиться без остатка на 15. Если это условие не выполняется, то контрольно-восстановительный процессор предполагает наличие случайной ошибки и запускает программу, начиная с контрольной точки. Если ошибка вновь появляется в том же месте, то контрольно-восстановительный процессор делает заключение о том, что ошибка является постоянной, а не случайной. Тогда он отключает питание от неисправного блока, подает питание на резервный блок и запускает программу с той же контрольной точки [41].

Любой мультипроцессор обладает некоторой ограниченной устойчивостью к отказам. Одной из главных проблем в стандартных системах является идентификация появления ошибки и принятие решения о перестройке системы для ликвидации последствий отказа.

Изучив опыт выполнения программы космических полетов Аполлон, лаборатория приборостроения Массачусетского технологического института разработала конструкцию совершенной мультипроцессорной системы, обладающей очень высокой надежностью и устойчивостью к сбоям и отказам. Предусматривается включение в систему дополнительных функциональных блоков, обеспечивающих продолжение работы даже после отказов в отдельных устройствах, и механизм голосования или другие методы обнаружения ошибок для идентификации отказов. Общая схема системы ACGN приведена на рис. 2.20.

Надежность нескольких мультипроцессоров, предложенных в качестве кандидатов для системы управления воздушным движением, обсуждена Дэнси [17]. В этой работе представлены некоторые действительные числовые оценки зависимости надежности системы от надежности ее элементов и от типа системной организации. Проведено сравнение системы РЕРЕ, ассоциативного процессора STARAN фирмы Goodyear, ЭВМ IBM 9020, ЭВМ ARTS III фирмы UNIVAC и ЭВМ IBM S/370.

ПУТИ СОЕДИНЕНИЯ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

Рассмотрение различных методов взаимного соединения функциональных блоков показало, что следует принимать во внимание некоторые соображения относительно конструкции самих путей передачи информации. Основное противоречие возникает между скоростью передачи, или общей пропускной способностью, и сложностью интерфейса и управления им. Конструктор системы должен учитывать все перечисленные ниже факторы.

1. Должно быть обеспечено указание места назначения передаваемой информации.

Это требует наличия определенной аппаратуры как для источника, так и для приемника информации.

Возможность указывать большое число мест назначения приводит к увеличению ширины пути, отведенного для управления, т. е. для непроизводительной передачи информации.

Место назначения можно указывать при каждой передаче или же можно устанавливать временный путь, открытый для параллельных передач.

Идентификация источника может подразумеваться, может быть необязательной или может быть выполнена так же, как указание места назначения.

2. Ширина пути оказывает непосредственное влияние на общую скорость передачи и на стоимость.

Пути с большей шириной более сложны по аппаратуре и дороже по стоимости.

Идеальная ширина пути равна длине слова в нормальном блоке, к которому имеется доступ или в котором осуществляется обработка; иначе говоря, это число разрядов, к которым возможен доступ в течение одного цикла обращения к памяти, или число разрядов, содержащихся в одном слове, обрабатываемом процессором. Выбор ширины пути зависит от желаемого соотношения между стоимостью и эффективностью. Ширина пути может быть меньше длины слова, если требуется низкая стоимость, или может составлять несколько слов, если нужно обеспечить высокую эффективность.

Почти всегда существуют расхождения между значениями ширины пути, необходимой для обращения или обработки в запоминающих устройствах, процессорных блоках, устройствах ввода-вывода и для передачи адресов. Длина слова в процессорном блоке фактически может быть переменной, так как часто имеется возможность работать с байтами, полусловами, словами и двойными словами.

Любое расхождение между шириной пути передачи данных и длиной слов в процессорных блоках или запоминающих устройствах требует введения буферных регистров сборки и разборки и, следовательно, дополнительных расходов на интерфейсы в каждом функциональном блоке, где их приходится применять.

3. Частота тактирования пути передачи данных также оказывает непосредственное влияние на общую скорость передачи.

Необходимо учитывать время передачи сигналов по линиям.

Использование более быстросействующей аппаратуры как в интерфейсах, так и точках соединения приводит к увеличению стоимости, сложности системы и эксплуатационных расходов.

4. Тип сигналов, подлежащих передаче (т. е. управляющая информация, данные или адреса).

5. Число отдельных автономных путей.

Очевидно, здесь перечислены далеко не все факторы, которые следует принимать во внимание. Приведенный список, однако, дает некоторое представление относительно тех компромиссов, к которым приходится прибегать при разработке этой чрезвычайно важной части системы. Если в системе не обеспечена достаточно высокая скорость передачи данных, то падает эффективность системы, и применение параллельной структуры для достижения совпадения операций может дать негативный результат.

ВИРТУАЛЬНЫЕ ПРОЦЕССОРЫ

В некоторых системах используется совокупность устройств, которые могут показаться самостоятельными процессорами, но на самом деле представляют собой виртуальные процессоры. Каждый

такой «процессор» обладает собственной памятью и регистрами управления, такими, как накапливающий сумматор и счетчик команд. Однако все они разделяют одно и то же арифметико-логическое

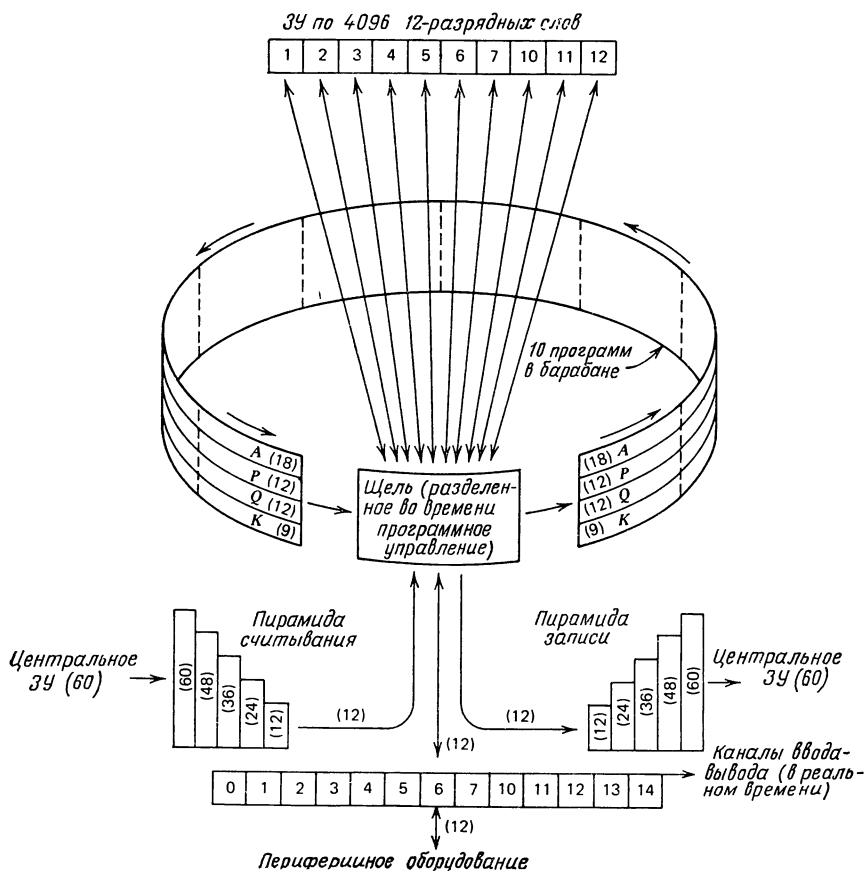


Рис. 2.21. Организация периферийных процессоров CDC 6600.

Показаны соединения с периферийными процессорами, центральной памятью и каналами ввода-вывода.

устройство по методу кругового обхода или каким-либо другим способом. Примерами использования такого метода являются периферийные процессоры ЭВМ CDC 6600 и системы ASC фирмы Texas Instruments, а также центральные процессоры ЭВМ MRX-40 и 50 фирмы Memorex.

В ЭВМ CDC 6600 четыре регистра управления для 10 периферийных процессорных блоков организованы в логический «барабан»

со «целью», характеризующей логическое местоположение арифметической системы и вращающейся вокруг барабана со скоростью 1 об/мкс. В каждой позиции цель находится в течение 100 нс. Этого

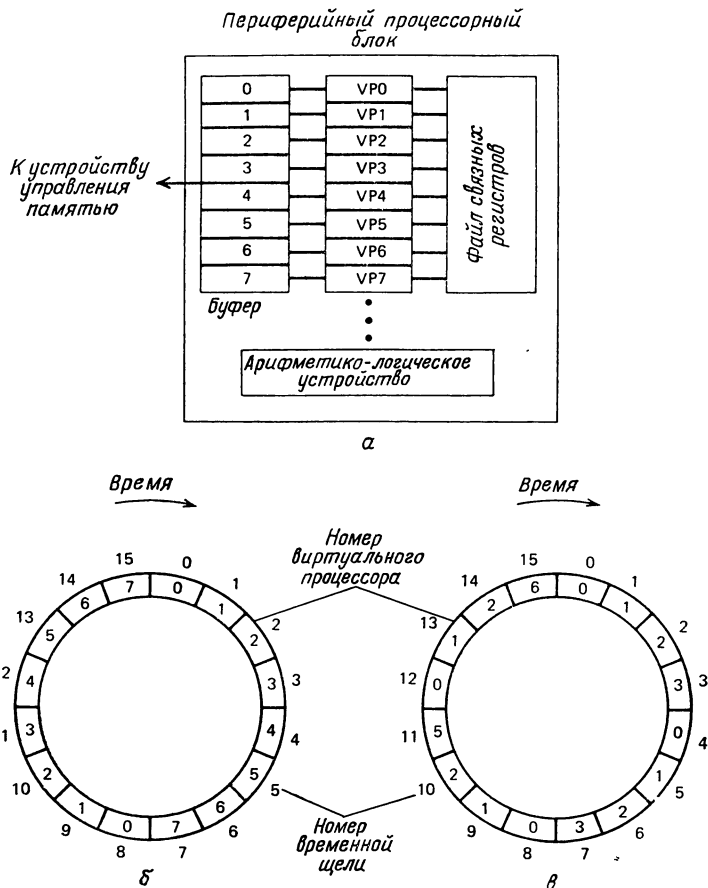


Рис. 2.22. Организация периферийных процессоров системы ASC фирмы Texas Instruments.

а — множественные потоки команд в периферийном процессорном блоке; б — распределение щелей с одинаковым временем; в — распределение щелей со взвешенным временем. VP — виртуальный процессор.

времени достаточно для выполнения большинства операций периферийного процессорного блока, причем полный цикл обращения к памяти составляет 1000 нс (рис. 2.21).

В периферийном процессорном блоке ЭВМ ASC фирмы Texas Instruments имеется аналогичная система. Полный цикл обслуживания всех виртуальных систем составляет 1360 нс. При этом на каж-

дую позицию приходится 85 нс. Максимальное число виртуальных процессоров равно 8. Закрепление виртуальных процессоров за шестнадцатью временными «щелями» может быть изменено так, чтобы какой-либо один из процессоров получал больше времени или же чтобы один или большее число процессоров были полностью исключены из работы (рис. 2.22).

МНОЖЕСТВЕННЫЕ АРИФМЕТИЧЕСКИЕ УСТРОЙСТВА

Во всех рассмотренных до сих пор системных организациях используется набор нескольких операционных элементов на уровне функциональных блоков (процессор, запоминающее устройство, кон-

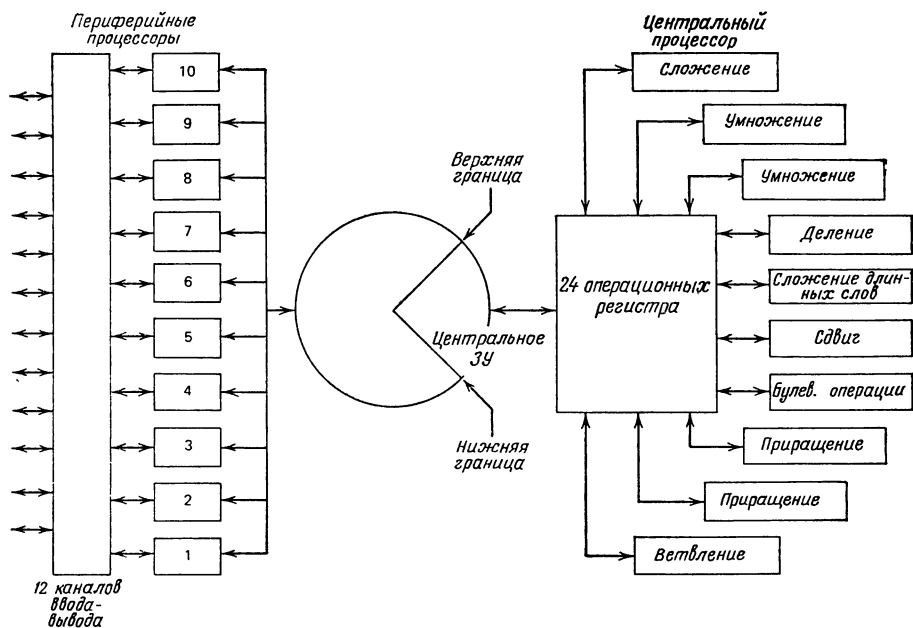


Рис. 2.23. Блок-схема системы 6600. Показаны множественные арифметико-логические устройства в центральном процессоре.

троллер ввода-вывода и т. д.). Режим параллельности или совпадения можно осуществить по желанию и на более низком уровне. Примером может служить центральный процессор ЭВМ CDC 6600. В обычном процессоре одни и те же арифметическо-логические устройства используются для выполнения различных операций: один и тот же накапливающий сумматор применяется для хранения результатов всех арифметических операций, один и тот же накапли-

вающий сумматор используется для всех логических операций и операций сдвига, имеется только один комбинационный сумматор и т. д.

Ключевым понятием для описания работы ЭВМ CDC 6600 является функциональная параллельность. Это понятие означает одновременное выполнение ряда арифметических функций. В машине имеется 10 функциональных блоков, которые работают до некоторой степени независимо друг от друга (рис. 2.23): два блока приращений, один сумматор, два множительных устройства, одно делительное устройство, одно устройство для сложения длинных слов, одно сдвигающее устройство, одно устройство для булевских операций и одно устройство ветвления. Конечно, достигаемая степень параллельности зависит от взаимного расположения команд в программе и от подготовки операндов к обработке.

Очевидно, функции центрального процессора по управлению в этой системе сильно усложнены, так как он должен определять, когда операнды готовы для использования, и контролировать, чтобы они не подверглись изменениям при выполнении другой совпадающей операции.

ОСНОВНАЯ ПАМЯТЬ ДЛЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Во всех рассмотренных конфигурациях мультипроцессорных систем память состояла из нескольких отдельных модулей. Это стандартный способ организации основной памяти и для однопроцессорных систем. Первоначальная цель использования нескольких модулей стандартных размеров состояла в унификации производства и в обеспечении возможности расширения памяти, предназначенной для любой конкретной системы. Учитывались также и экономические соображения, которые приводили к компромиссу между сложностью системы для выборки адресов и считывания, с одной стороны, и возможной выгодой при более широком использовании стандартной аппаратуры, такой, как регистры адреса памяти, выходные регистры памяти, буферы, схемы управления и питания, с другой стороны.

Отдельные модули памяти, которые могут работать автономно и даже асинхронно, предоставляют дополнительные возможности при организации параллельной работы или работы в режиме совпадения. Рассмотрим сначала последовательность явлений, происходящих в течение одного полного цикла работы центрального процессора.

ДОСТУП К ПАМЯТИ С ПЕРЕКРЫТИЕМ

Как для выборочной, так и для исполнительской части цикла необходимо два обращения к памяти (рис. 2.24). Первый возможный способ получения параллельной работы или работы с перекрытием

состоит в обеспечении максимального взаимного перекрытия циклов *I* и *E*, как показано на рис. 2.25. Даже при выполнении этого условия общая производительность системы продолжает зависеть в основном от продолжительности цикла памяти.

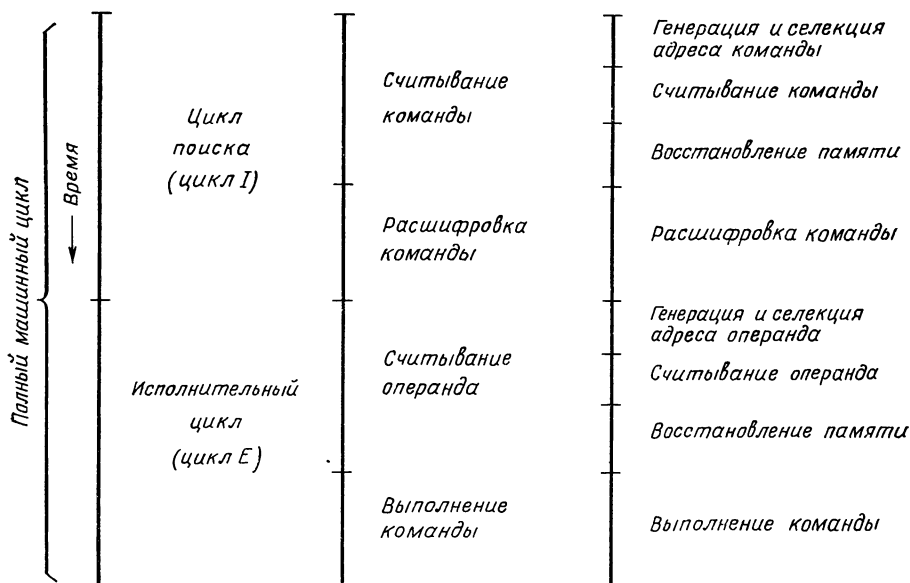


Рис. 2.24. Этапы процесса доступа к памяти в течение полного машинного цикла.

Нормальный цикл для ферритовой памяти состоит из следующих частей (рис. 2.28, а):

- Выборка адреса.
- Импульсы считывания.
- Считывание.
- Период восстановления.
- Импульсы записи.
- Период восстановления.

Если выполняется операция ЗАПИСЬ, то результаты операции СЧИТЫВАНИЕ стираются; если же выполняется операция СЧИТЫВАНИЕ, то результаты считывания должны быть записаны обратно в память, так как при считывании из ферритовой памяти информация разрушается.

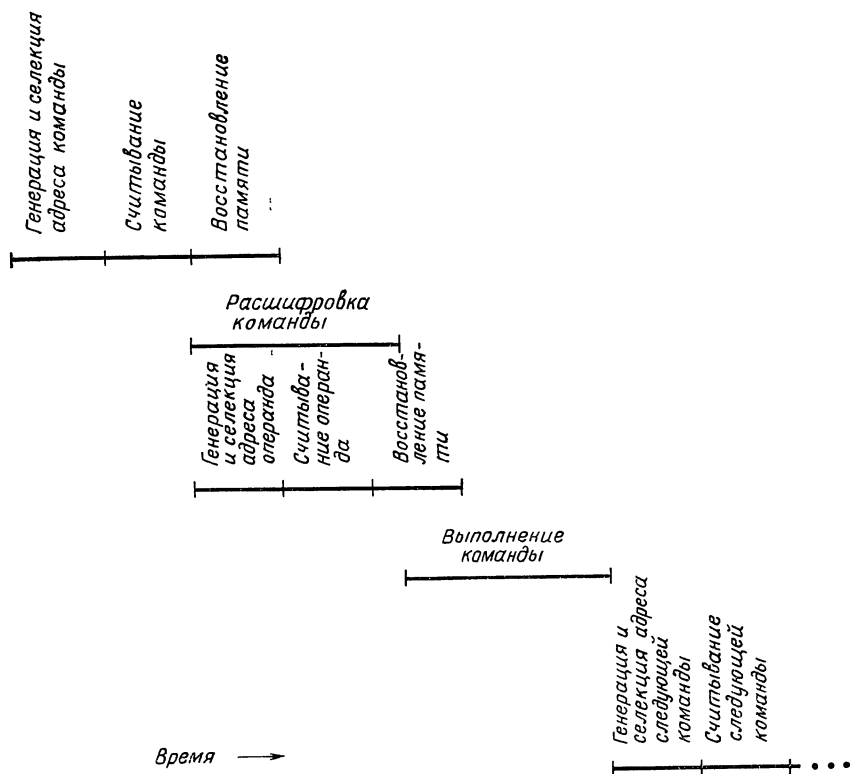


Рис. 2.25. Перекрывание циклов поиска и исполнения.

КОНФЛИКТЫ ПРИ ОБРАЩЕНИИ К ПАМЯТИ

Одним из факторов, снижающих эффективность мультипроцессорных систем, являются конфликты при обращении к памяти. Эти конфликты бывают двух типов — программные и аппаратные. Программный конфликт возникает, когда какой-либо из процессоров пытается использовать таблицу или набор данных, который в данный момент используется другим процессором, причем этим другим процессором включен механизм защиты для предотвращения доступа со стороны какого-либо другого устройства. Такая блокировка необходима для правильного выполнения и сохранения целостности как управляющих, так и прикладных программ. Программные конфликты при обращении к памяти, более известные под названием «блокирование памяти», рассматриваются в разделе гл. 3 о проблемах программного обеспечения. Если процессор встречается с блокирова-

нием памяти, то он может переключиться на какую-либо другую задачу, для которой не требуются данные таблицы или наборы данных. Однако в этом случае он должен большую часть времени находиться в цикле ОЖИДАНИЕ, следя за состоянием блокирования, которое может продолжаться довольно долго.

Аппаратные конфликты возникают при попытке двух или большего числа процессоров или одного устройства ввода-вывода и одного процессора одновременно обратиться к одному и тому же модулю памяти. Используемый здесь термин «одновременно» означает, что конфликтующие запросы сделаны в течение одного и того же цикла памяти. Поскольку за один цикл памяти может быть реали-

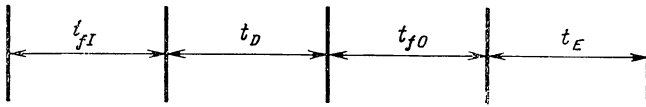


Рис. 2.26. Этапы цикла работы процессора [29].

t_{fI} — время выборки команды из памяти, t_D — время расшифровки адреса команды, t_{fO} — время выборки операнда из памяти, t_E — время выполнения команды.

зовано лишь одно обращение, другой запрос или несколько запросов должны ожидать своей очереди. Однако в этом случае ожидание длится обычно всего лишь один-два цикла. Затруднения возникают, когда число конфликтов становится большим. Такая форма снижения эффективности системы, известная также под названием «межпроцессорных помех», может стать весьма существенной и представлять реальную проблему, если в системе (в ее программной и аппаратной частях) не предусмотрены специальные меры по преодолению этих затруднений.

Действительных измерений этого эффекта проведено очень мало из-за трудностей приборного оснащения испытаний и выбора базы для сравнения. Одна из оценок, выполненная для двухпроцессорной системы IBM S/360, модель 67, работающей в составе системы UMMPs Мичиганского университета, показала, что этот эффект характеризуется величиной «меньше нескольких процентов, а может быть и меньше одного процента» [1].

С помощью моделирования были проведены некоторые исследования для оценки влияния аппаратных конфликтов на эффективность системы с перекрестной коммутацией, работающей в реальном масштабе времени с использованием процедуры страничного распределения памяти.

Число модулей памяти предполагалось равным числу процессорных блоков. Количество конфликтов зависит от длительности цикла памяти и от рабочего быстродействия процессора, которое определяет частоту обращений к памяти. Принятое распределение времени цикла работы процессора показано на рис. 2.26.

Процессоры соревновались за доступ к каждому модулю памяти по методу кругового обхода. Один из модулей памяти на время одного цикла памяти t_{mc} закреплялся за каким-либо одним процессором. Если требуемая память была недоступна, цикл работы процессора (рис. 2.26) соответственно удлинялся. При моделировании были использованы следующие величины:

$$t_{f0} = t_{f1} = t_{mc}/2; t_D = 1 \text{ мкс}; t_E = 1 \text{ мкс (60\%)}, 9 \text{ мкс (40\%)}$$

Было принято предположение, что как команды, так и данные, записанные постранично, распределены случайно между модулями памяти (это, ко-

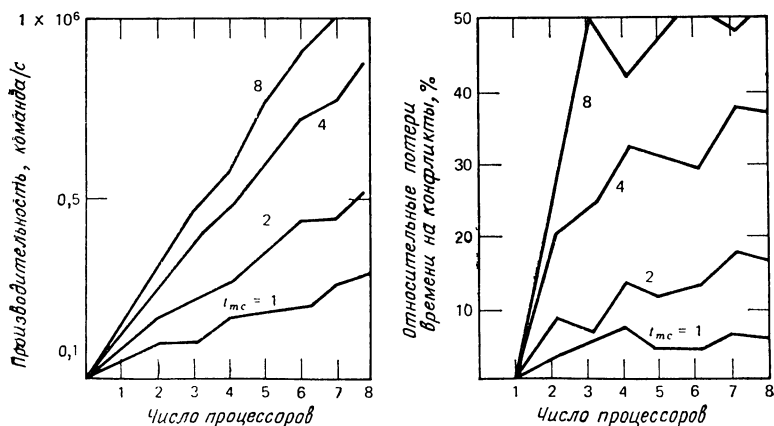


Рис. 2.27. Влияние конфликтов при обращении к памяти [29].

нечно, несколько искусственное условие). Производилось 256 обращений к каждой странице памяти, после чего выбиралась случайным образом следующая страница. Результирующая производительность системы и относительная часть рабочего времени (в процентах), потерянная на конфликты при обращении к памяти, представлены в виде кривых на рис. 2.27. При заданных условиях влияние конфликтных ситуаций оказалось несущественным для запоминающих устройств с длительностью цикла 1 и 2 мкс...

Поскольку диспетчерский алгоритм допускает некоторую степень свободы при распределении общей нагрузки системы между отдельными процессорами, можно было бы предусмотреть такое назначение отдельных заданий процессорам, при котором количество возможных конфликтов при обращении к памяти было бы минимальным. В этом случае возникло бы сложное взаимодействие между функциями операционной системы по планированию заданий и распределению памяти [29].

Моделирование показывает, что в том случае, если длительность цикла памяти равна среднему значению времени выполнения операций (4,2 мкс), может произойти вполне ощутимое снижение эффективности работы системы. Поэтому рассматриваемая проблема имеет очень важное значение для любой системной структуры. Преимущества более быстродействующей памяти очевидны. Вопрос о влиянии увеличения числа модулей памяти (при уменьшении их емкости) остается открытым.

Один из методов, позволяющий уменьшить снижение эффективности из-за конфликтов при обращении к памяти и из-за ожидания в очередях, когда одна и та же управляющая или обслуживающая программа используется несколькими процессорами, состоит в дублировании программы в машинных кодах в различных модулях памяти. Эта проблема была проанализирована количественно в работе [15]; в результате оказалось, что полное дублирование приводит к ненужным затратам. Частичное дублирование «может привести к снижению на 30—40% общей стоимости центрального процессора и памяти программ по сравнению с полным дублированием».

В своей недавно опубликованной заметке Рэви сообщает о результатах исследования аналитическими и теоретико-вероятностными методами вопроса об эффективной емкости множественных модулей памяти с чередованием адресов [35]. В этой работе получено более высокое, чем в предыдущих работах [11, 25], значение оценки числа слов, передаваемых за время одного цикла. Эти результаты, однако, сильно зависят от распределения вероятностей числа обращений на протяжении цикла. Рэви утверждает, что если среднее число случайных обращений больше числа модулей памяти, то эффективность использования памяти повышается.

НАЗНАЧЕНИЕ ФИЗИЧЕСКИХ И ЛОГИЧЕСКИХ АДРЕСОВ ПАМЯТИ

Давно было замечено, что эффективное значение времени считывания может быть значительно сокращено, если следующие один за другим адреса размещены в различных модулях памяти по методу чередования. Это явление иллюстрируется следующей таблицей:

	Модуль памяти № 0	Модуль памяти № 1
Адрес, хранящийся в памяти	0, 2, 4, 6, ...	1, 3, 5, 7, ...
Последовательность операций	Выборка адреса Импульсы считывания Считывание Период восстановления Импульсы записи Период восстановления Выборка адреса Импульсы считывания Считывание Период восстановления » » » » » »	Выборка адреса Импульсы считывания Считывание Период восстановления Импульсы записи Период восстановления Выборка адреса » » » » » »

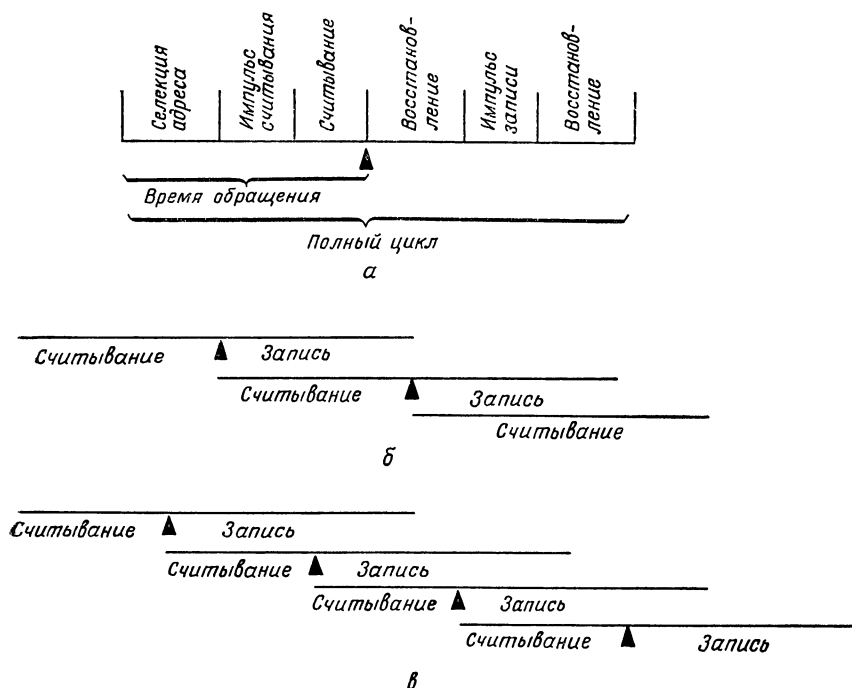


Рис. 2.28. Полный цикл памяти и условия работы с перекрытием.

а — основной полный цикл памяти; б — перекрытие с симметричным циклом; в — перекрытие с укороченным временем обращения.

Применение двух «чередующихся» модулей памяти привело к удвоению эффективной пропускной способности памяти, причем увеличение сложности в основном определяется требованием, чтобы интерфейс каждого модуля различал четные и нечетные адреса (рис. 2.28, б, в). Основные временные соотношения для ферритовой памяти позволяют получить выигрыш и от четырехкратного чередования; однако при дальнейшем увеличении кратности выигрыш становится незначительным.

Описанный метод может быть применен и в мультипроцессорных системах¹. Однако результаты в этом случае могут оказаться отрицательными. Это объясняется возможностью появления взаимных помех или конфликтов между обращениями различных процессоров, так как все процессоры работают одновременно, а метод чередования требует, чтобы они использовали все или по меньшей мере несколько модулей памяти. Эта проблема будет рассмотрена ниже. Другим не-

¹ Для достижения наибольшей эффективности процессор должен иметь несколько путей к памяти, или входов, и некоторый механизм для управления доступом при перекрывающихся обращениях.

достатком метода чередования адресов является снижение готовности системы. Если какой-либо модуль памяти вышел из строя, невозможно даже просто перераспределить память и продолжать работу, так как логическая область памяти распределена между несколькими физическими блоками. По этой причине, а также из-за влияния взаимных помех в мультипроцессорных системах часто распределяют адреса между модулями памяти так, чтобы каждый модуль содержал группу последовательных адресов. Это приводит к снижению эффективной скорости передачи для каждого отдельного процессора, но в результате может повысить общую скорость передачи для системы в целом и способность системы к реконфигурации при выходе из строя какого-либо модуля.

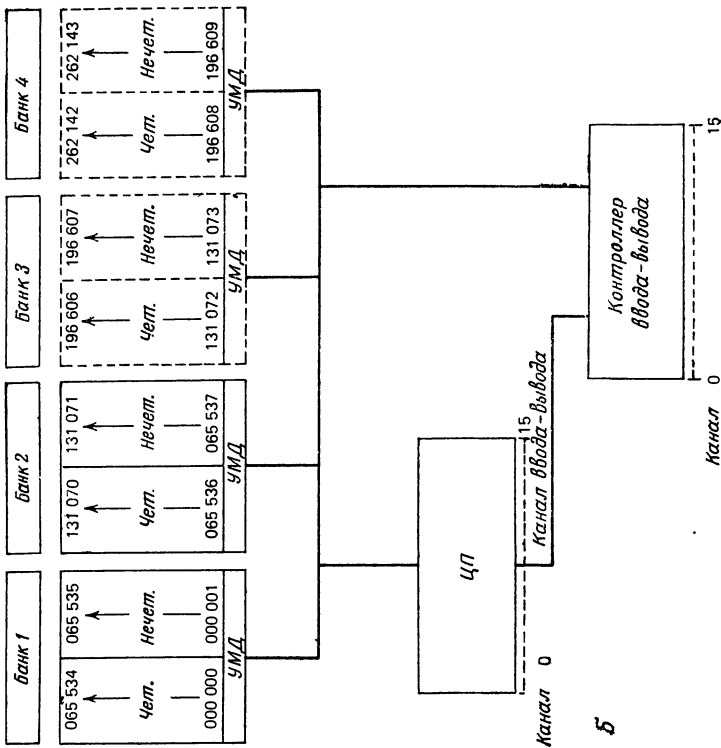
В мультипроцессорной системе с большим числом модулей памяти сравнительно малой емкости можно использовать комбинацию обоих описанных методов, имея в виду, что если один из модулей вышел из строя, то его «соседи» (от одного до трех) также должны быть исключены из работы, пока не будет восстановлена вся эта совокупность модулей.

На рис. 2.29, а, б, в показан пример использования различных методов назначения логических адресов физическим модулям памяти в мультипроцессорной системе UNIVAC 1108. Метод чередования адресов в настоящее время стал общепринятым; в табл. 2.1 перечислены некоторые системы, в которых предусмотрена возможность его применения. Следует отметить, что этот метод используется далеко не всегда; причина этого, как было указано выше, заключается в том, что при отказе одного физического модуля можно потерять логическую память вдвое или вчетверо большего объема.

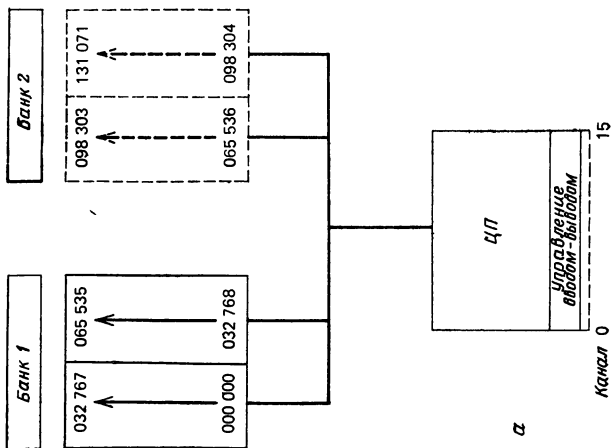
ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА И ИНТЕРФЕЙСЫ

До сих пор при рассмотрении различных структур мультипроцессорных систем основное внимание уделялось организации взаимных соединений между процессорами, запоминающими устройствами, каналами ввода-вывода и контроллерами. Ничего не было сказано о соединениях между отдельными контроллерами устройств ввода-вывода и самими устройствами ввода-вывода, за исключением двух примеров, приведенных в разделе о системах с перекрестной коммутацией. В первом случае речь шла об использовании отдельной коммутационной матрицы между каналами и устройствами ввода-вывода в стандартных системах фирмы Burroughs, а во втором — о непосредственном подсоединении устройств ввода-вывода к основной коммутационной матрице в проектируемой системе Multi-Interpreter фирмы Burroughs.

Подсоединение устройств ввода-вывода непосредственно к основной шине с временным разделением, если использование этой



Б



А

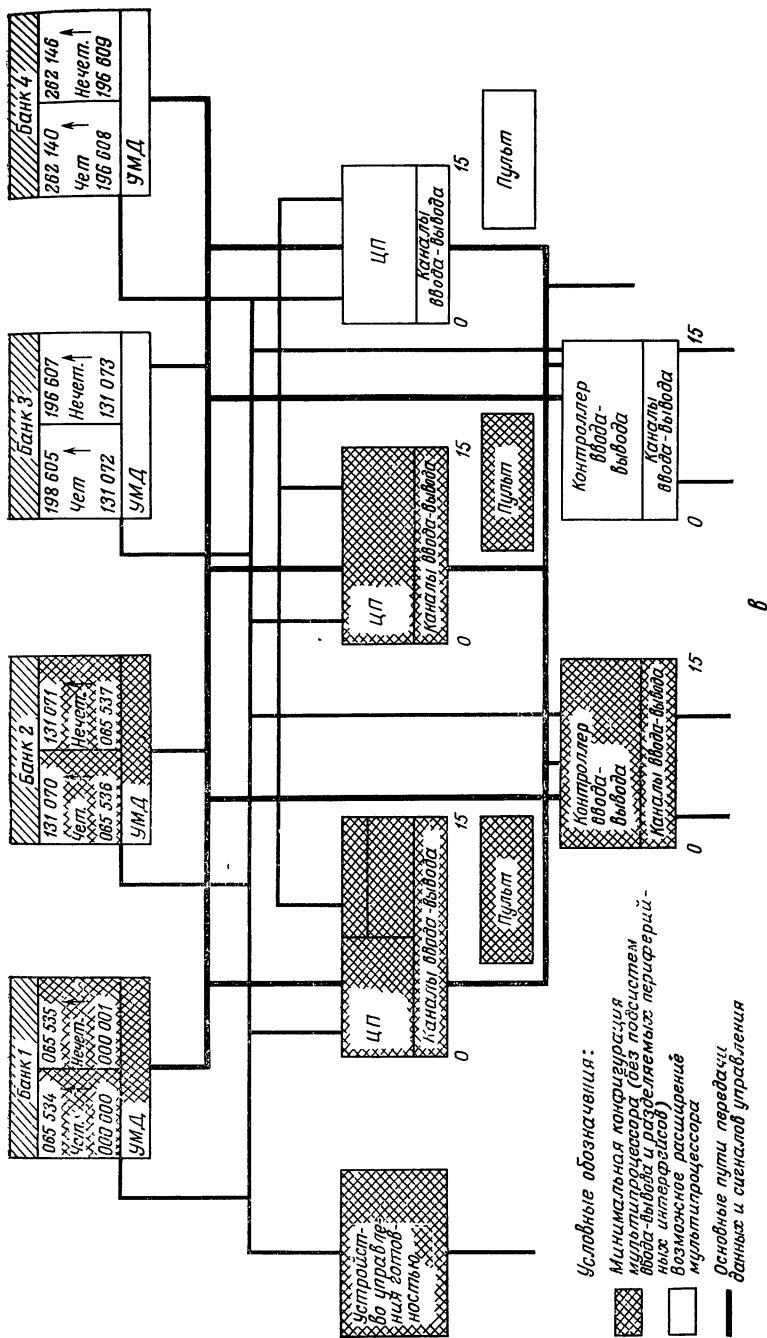


Рис. 2.29. а — память без чередования (однопроцессорная система UNIVAC 1108); б — память с чередованием (однопроцессорная система UNIVAC 1108 с контроллером ввода-вывода); в — память с чередованием (мультипроцессорная система UNIVAC 1108).

УМД — устройство мультимодуля к памяти.

Таблица 2.1

Примеры использования метода чередования адресов в центральной памяти.
(Применение метода в этих системах возможно, но он не всегда реализуется на практике.)

Фирма-изготовитель системы и обозначение модели	Двухкратное чередование	Четырехкратное чередование
Burroughs, B6700 B7700	X X	X
Control Data, Cyber 72/73/74 STAR	(максимум 10 совпадающих доступов) (32-кратное чередование)	
Digital Equipment, System 10	X	X
Honeywell, 6050/60 6070/80	X X	X
IBM, S/360—65/67 75/85 195 S/370—165	X	X X
(16-кратное чередование)		
Texas Instruments, ASC	(8-кратное чередование)	
UNIVAC, 1108 1110	X X	
Xerox, Sigma 6/7/8/9	X	X

шины лежит в основе организации системы, обладает как достоинствами, так и недостатками. Главным достоинством этого метода является гибкость структуры и простота изменения комплекта устройств ввода-вывода. Кроме того, в этом случае любой разрешенный путь между контроллером канала и устройством ввода-вывода может быть легко реализован посредством выбора соответствующего адреса на передающей шине. Основной недостаток этой схемы состоит в увеличении нагрузки на одиночную передающую шину при передаче входной и выходной информации. Этот недостаток можно частично устранить, применяя двойную шину с временным разделением (рис. 2.30). В последнем случае, однако, временное разделение единственного пути по-прежнему является ограничивающим фактором.

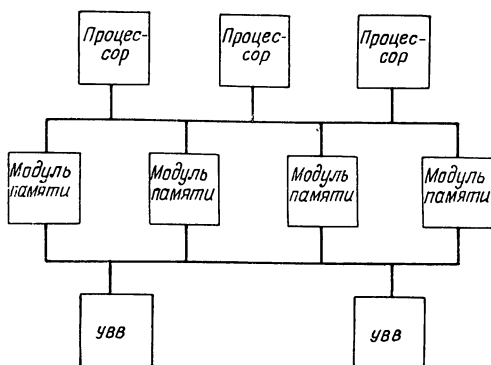


Рис. 2.30. Организация системы с двойной шиной и временным разделением (отдельные шины для процессоров и устройств ввода-вывода).

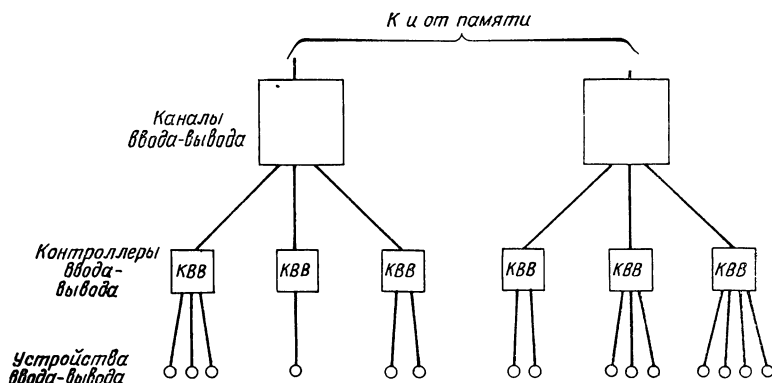


Рис. 2.31. Подсоединение устройств ввода-вывода по древовидной схеме.

Использование для устройств ввода-вывода как перекрестных матриц, так и общих шин с временным разделением часто затрудняется и другим, более существенным обстоятельством, а именно тем, что большинство контроллеров и самих устройств ввода-вывода было первоначально предназначено для применения в системах с древовидной топологией (физической или логической) и не может быть приспособлено к другим структурам (рис. 2.31). Можно было бы создать такую конфигурацию мультипроцессорной системы, в которой каждый процессор имел бы свой собственный набор каналов, контроллеров и устройств ввода-вывода. Это, однако, привело бы к структуре с очень малой гибкостью, потребовало бы подробного знания характеристик неизменяющейся нагрузки и свело бы на нет многие

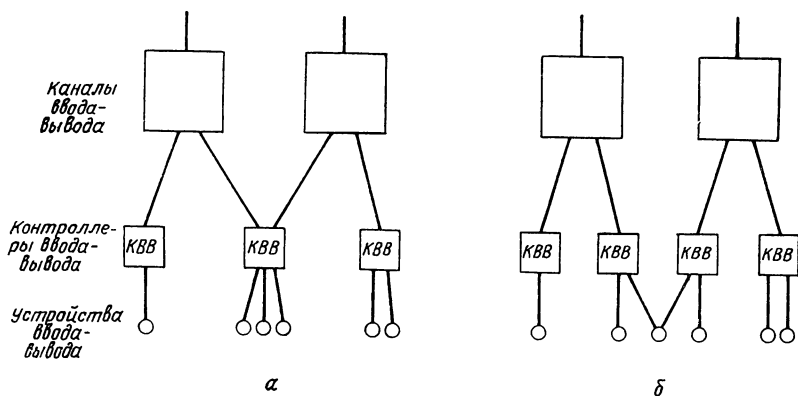


Рис. 2.32. Разделение устройств ввода-вывода при помощи двойного подсоединения на уровне контроллеров (а) и на уровне устройств ввода-вывода (б).

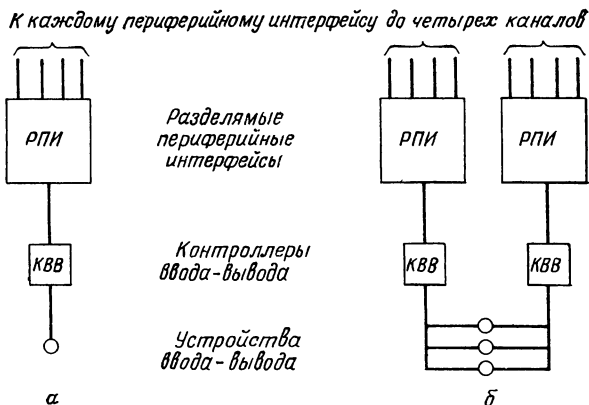


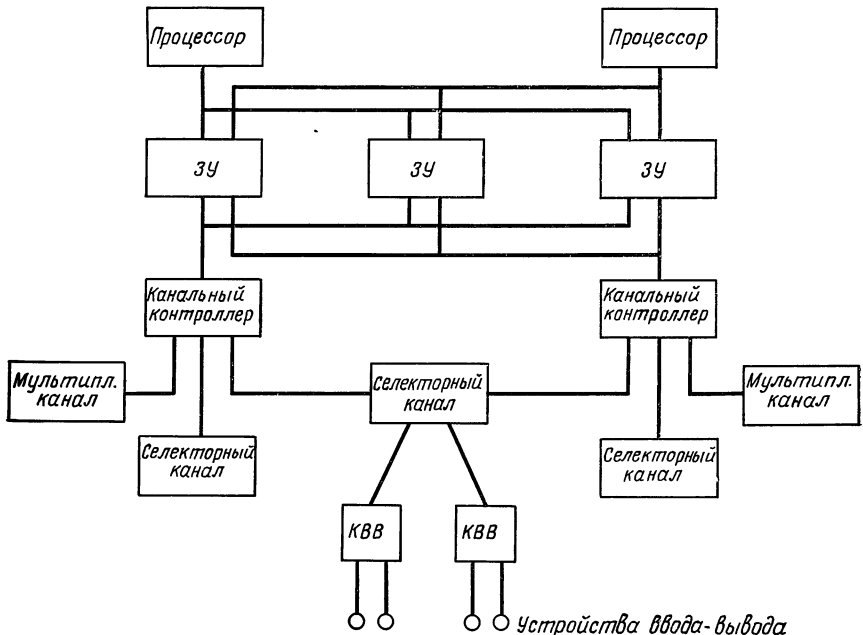
Рис. 2.33. Использование разделяемого периферийного интерфейса в системе UNIVAC 1108:

а — одноканальная система; б — двухканальная система.

преимущества мультипроцессорной организации. Если бы использовались собственные наборы устройств ввода-вывода, то появились бы дополнительные и ненужные расходы на хранение многих копий всего системного программного обеспечения этого уровня. Значительно более эффективная схема соединения контроллеров и устройств ввода-вывода предусматривает применение двухходовой конструкции (которую многие из них уже имеют) или специального интерфейса для осуществления разделения. При этом древовидная топология сохраняется. Двухходовая конструкция может быть реализована на уровне контроллеров или на уровне устройств ввода-вывода (рис. 2.32).

Множественное соединение контроллеров осуществлено в системе UNIVAC 1108 при помощи разделяемого периферийного интерфейса (Shared Peripheral Interface). Этот интерфейс может использоваться с одно- или двухканальными устройствами (рис. 2.33). В системе UNIVAC 1108 применяются оба типа соединений, в зависимости от типа подключаемых устройств, их быстродействия и других характеристик.

ЭВМ IBM 360/67 может иметь конфигурацию двухпроцессорной или однопроцессорной системы. В двухпроцессорном варианте каждый процессор при помощи переключателя подсоединен к специальному устройству, носящему название канального контроллера (рис. 2.34, а). Канальный контроллер сопряжен интерфейсами с 1—6 селекторными каналами и с одним мультиплексным каналом. Эти каналы работают в обычном режиме, т. е. каждый селекторный



а

Рис. 2.34. Схема подсоединения устройств ввода-вывода в мультипроцессорных системах фирмы IBM.
 а — мультипроцессорная система модель 67 для работы в режимах разделения времени;
 б — мультипроцессорная система модель 65MP [39], УВВ — устройство ввода-вывода, КВВ — контроллер ввода-вывода.

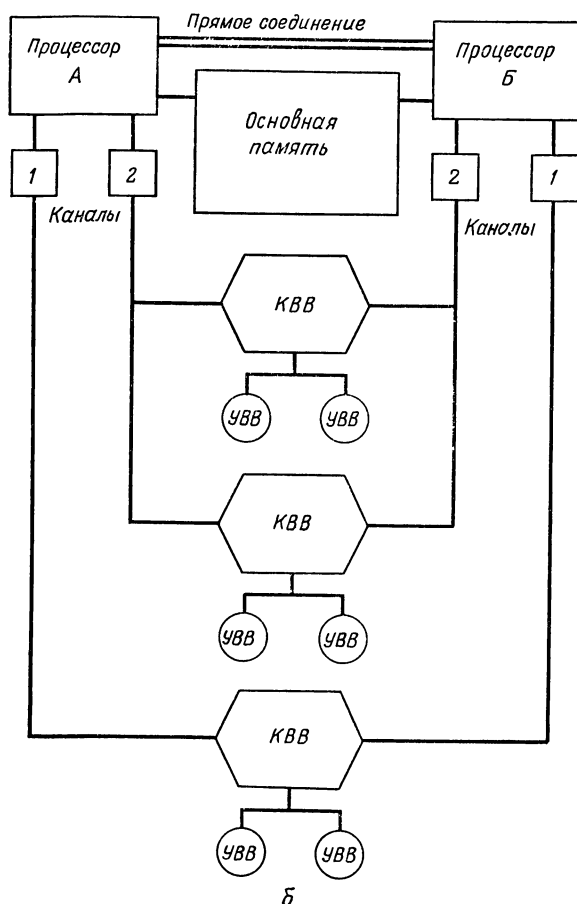


Рис. 2.34. Продолжение

канал имеет быструю связь с любым числом (от 1 до 8) контроллеров и способен адресоваться к 256 внешним устройствам. Селекторные каналы используются для быстродействующих устройств, а мультиплексные каналы — для устройств с меньшим быстродействием, таких, как читающие устройства, перфораторы, печатающие устройства и периферийные устройства управления. Два канальных контроллера придают системе гибкость, в результате чего:

Любой процессор имеет доступ к любому контроллеру или внешнему устройству.

Двухходовые контроллеры могут подсоединяться более чем к одному каналу и обеспечивать различные пути передачи. Аналогично двухходовые внешние устройства могут подсоединяться к двум контроллерам.

В мультипроцессорной системе IBM S/360, модель 65, каналный контроллер не применяется, а ее система ввода-вывода не обладает гибкостью, достигнутой в модели 67 (рис. 2.34, б). Организация модели 65MP базируется на возможности подсоединения каждого контроллера к любому отдельному собственному каналу процессоров (это качество называется двойственностью контроллера). Прямая связь между двумя процессорами позволяет использовать резервный путь, если какой-либо из каналов занят или вышел из строя.

Итак, почти во всех мультипроцессорных системах используется одна из следующих основных схем взаимных соединений:

Матрица перекрестной коммутации.

Шина с временным разделением.

Древовидная структура с двухходовыми элементами.

Могут применяться и комбинации этих основных схем.

НАДЕЖНОСТЬ И ГОТОВНОСТЬ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ

УСТОЙЧИВОСТЬ ПРОТИВ ОТКАЗОВ И ТЕРПИМОСТЬ К ОТКАЗАМ

В начале этой главы были рассмотрены специальные типы организации мультипроцессорной системы, которые были разработаны для придания системе устойчивости против отказов при работе в критических условиях. Такие разработки мотивировались необходимостью обеспечить очень высокую готовность систем и способность выполнять всю нагрузку в любой момент времени независимо от стоимости системы или, во всяком случае, без особых ограничений по стоимости. Столь высокая готовность достигалась применением дублирующих или избыточных функциональных блоков и соединительных схем, при помощи которых при необходимости можно было разделить систему на части. Конечно, такая цель и методы ее достижения находятся на одном конце диапазона достижимой степени надежности и готовности; на другом конце этого диапазона находится самая простая система, содержащая только по одному блоку каждого типа и не обладающая никакой избыточной производительностью.

Безусловно, готовность зависит от надежности системы в целом и ее составных частей; однако эта зависимость не всегда проста. При простом соединении одиночных блоков, упоминавшемся выше,

эта зависимость очевидна: надежность всей системы равна произведению надежностей всех блоков, а готовность системы является простой функцией ее надежности, вероятности выхода из строя и среднего времени восстановления. Если же мы имеем дело с группами одинаковых блоков и возможна некоторая гибкость в отношении изменения конфигурации, свойственная большинству мультипроцессоров, то решить эту задачу не так легко. Один из способов получить желаемые характеристики системы заключается в использовании специального устройства управления. Этот способ применен в системе UNIVAC 1108, в которой имеется устройство управления готовностью (Availability Control Unit), и в системе CLC фирмы Bell Laboratories. Устройство управления готовностью системы UNIVAC 1108 выполняет следующие функции:

Подразделение аппаратуры мультипроцессорной системы на независимые системы.

Отключение блоков для ремонта без нарушения работы остальной части системы.

Защита основной памяти в случае выхода из строя питания центрального процессора или контроллера ввода-вывода.

Автоматический запуск восстановительной процедуры после отказа.

Системы со многими параллельными блоками не являются новинкой в области вычислительной техники, однако их преимущества далеко не всегда используются. Даже в ЭВМ UNIVAC I были предусмотрены два полных комплекта арифметических регистров (напомним, что она разрабатывалась в период, когда преобладали схемы на ненадежных вакуумных электронных лампах). Работа этих регистров непрерывно сравнивалась и машина останавливалась, как только обнаруживалось расхождение. Однако дублирование регистров не способствовало восстановлению работоспособности ЭВМ и поэтому не повышало ее готовность, хотя нельзя отрицать, что обнаружение неправильных результатов само по себе полезно. Таким образом, в этой системе была обеспечена возможность обнаружения ошибки или отказа, что, несомненно, является первым шагом по пути улучшения готовности.

При наличии в системе нескольких одинаковых блоков система может сохранять некоторую функциональную работоспособность в случае выхода из строя одного из них, если отказ блока может быть обнаружен и *если система может быть перестроена* так, чтобы продолжать работу без неисправного блока. Разумеется, возможности этой новой системы будут меньше, чем у исходной системы (если только у нее нет избыточной или неиспользованной вычислительной мощности), так что она не сможет справиться с полной рабочей нагрузкой. Это приводит к расширению понятия устойчивости против отказов и введению понятия «терпимости к отказам» или «постепен-

ной деградации». В гл. 3 рассмотрены различные условия, от которых может зависеть успех процедуры восстановления работоспособности системы после отказов. В этой главе предметом нашего обсуждения будут аппаратные средства, которые обеспечивают возможность реконфигурации системы.

При разработке системы ARTS III фирмы UNIVAC был сделан вывод, что из-за самой природы мультипроцессорной системы изоляция блоков невозможна. Например, невозможно сказать, какой из процессоров вызвал изменение в таблице состояния системы. Поэтому были использованы другие принципы. При обнаружении ошибки разыскиваются те блоки, которые еще исправны, и работа продолжается с этими блоками. Результаты, полученные при разработке этой системы, имеют очень большое значение.

РЕКОНФИГУРАЦИЯ СИСТЕМЫ

Изменение конфигурации, или реконфигурация, системы представляет собой совокупность нескольких действий:

Физические переключения путей передачи или связи.

Действия, выполняемые оператором у пульта при помощи языка управления системой (в системах, работающих в реальном времени, эти действия осуществляются автоматически).

Действия, выполняемые операционной системой.

Изменения состояния управляющих таблиц под воздействием операционной системы.

Любые действия первых трех типов потребуют, по всей вероятности, изменений в таблицах, которые определяют конфигурацию имеющегося в распоряжении оборудования (процессоров, каналов, контроллеров, устройств ввода-вывода и запоминающих устройств), а также функционирующих путей передачи информации. В некоторых операционных системах обеспечена автоматическая чувствительность к изменению рабочих условий при помощи периодического контроля совокупности «битов присутствия», которые вырабатываются каждым блоком, как только он подключается к системе. Этот метод использован в Главной управляющей программе (Master Control Program) фирмы Burroughs. В этом случае отказы блоков или изменения, внесенные оператором в систему при отключении какого-либо блока, быстро обнаруживаются и в контрольные таблицы вносятся соответствующие коррективы. Таблицы в свою очередь приводят в действие подпрограммы операционной системы, распределяющие ресурсы и управляющие ими.

В системе IBM S/360 оператор должен сообщить операционной системе с пульта о выключении блока при помощи команды ИЗМЕНЕНИЕ (VARY). Конечно, если только это не является абсолютно

необходимым, он не будет стремиться немедленно прекратить работу еще действующего блока. Поэтому в мультипроцессоре IBM 360, модель 65, предусмотрена команда управления, осуществляющая «организованный останов» процессора, который требует завершения всех начатых операций ввода-вывода, происходящих под управлением этого процессора, и всех других неоконченных функций, или передачи их одному из оставшихся центральных процессоров системы.

Несколько другой подход осуществлен в системе UNIVAC 1108, где имеется устройство управления готовностью, которое при необходимости изменяет конфигурацию системы и подразделяет ее на части. Кроме того, это устройство позволяет отключать функциональные блоки системы для ремонта, осуществляет защиту центральной памяти при отказах питания, а после отказа автоматически запускает программу восстановления. Выполняя эту последнюю функцию, устройство управления готовностью фактически работает в необычном режиме. А именно, постоянно предполагается, что произошел отказ, и устройство автоматически запускает программу исправления ошибок всей системы, если только его внутренний счетчик времени не будет сброшен исполнительной системой в интервале, продолжительность которого можно задавать в диапазоне от 1 до 15 с. Устройство управления готовностью — это независимо работающее устройство, которое может быть опрошено процессором, когда нужно определить, какие блоки подключены к системе и вполне готовы к работе.

ЭВМ IBM 9020 представляет собой специальную мультипроцессорную систему, предназначенную для управления воздушным движением. Так как она работает в весьма критических условиях и в реальном масштабе времени, все операции по реконфигурации системы должны выполняться автоматически. Метод изменения конфигурации основан на работе регистра контроля конфигурации (Configuration Control Register), который дублирован в собственной памяти (емкостью 32К байтов) каждого контроллера устройства ввода-вывода. Функционирование этой системы реконфигурации описано в работе [45].

Любому специалисту, имеющему опыт работы с цифровой вычислительной аппаратурой, известно, что почти все системы крайне чувствительны к «электрическим шумам», т. е. к посторонним импульсам, проникшим в систему. Поэтому существует своеобразное различие между логической и физической реконфигурацией. Несмотря на то, что какой-либо блок может быть логически исключен из системы путем простой модификации системных таблиц, хотя на самом деле он остается в рабочем состоянии, для физического отключения блока часто требуется пауза в ходе нормальной работы, чтобы предохранить систему от действия шумов. По окончании паузы работа может быть возобновлена без перезапуска системы.

ВЫВОДЫ

В основе проектирования любой системы с высокой готовностью лежат четыре принципа, представленные ниже в сжатой форме:

- 1) модульность — выбор функциональных элементов, которые можно присоединять, соединять и изолировать;
- 2) множественность — дублирование избранных блоков для получения избыточности, достаточной для обеспечения устойчивости к отказам;
- 3) гибкость конфигурации — способность системы к быстрому взаимному переключению элементов при обнаружении ошибки;
- 4) анализируемость и восстанавливаемость — свойство системы обнаруживать ошибки и минимизировать их влияние [45].

Все эти принципы зависят от аппаратного обеспечения — как от отдельных блоков, так и от их системной организации. Последние два принципа требуют высокой степени интеграции между аппаратными и программными средствами.

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Общие руководства для конструкторов мультипроцессорных систем еще не созданы. Нет и четких правил, подобных кулинарным рецептам. Тем не менее проекты мультипроцессорных систем, по-видимому, базируются не только на решениях, продиктованных какими-то частными обстоятельствами, например только стоимостью различных способов технической реализации системы. Можно выбирать между различными компромиссами. Вероятно, наиболее значительный из них — это компромисс между стоимостью и эффективностью в самом общем смысле. Некоторые из факторов, которые необходимо учитывать, приведены в табл. 2.2.

Самым ярким примером такого компромиссного решения может служить выбор системы соединений, или системы передачи данных. На одном конце диапазона возможных решений находится весьма сложная матрица перекрестной коммутации, которая обеспечивает почти полный контроль аппаратных средств на протяжении процесса передачи, но содержит столько же аппаратуры и активных схем, сколько содержат два-три центральных процессора. Противоположный конец этого диапазона занимает абсолютно пассивная общая шина с временным разделением, представляющая собой по существу простой жгут проводов. Конечно, она представляет собой более надежную систему связи — передачи с одним путем. Однако, как было указано выше, скорость передачи данных в ней ограничена, а сложность отчасти просто перенесена на интерфейсы в каждом функциональном блоке, подключаемом к шине. Кроме того, тот факт, что в системе имеется только один путь передачи данных, приводит к катастрофическому отказу системы в целом при выходе из строя какого-либо одного важного элемента или блока.

Таблица 2.2

Факторы, которые необходимо учитывать при сравнении различных организаций аппаратных средств системы

Требуемые физические соединения:

Число проводов, соединяющих функциональные блоки (размеры кабелей).

Сложность механизма управления:

Конкуренция за пути передачи информации.

Разрешение конфликтов в пунктах назначения при передаче информации.

Обеспечение пути передачи информации (соединение).

Надежность:

Устойчивость против отказа отдельного блока из взаимосвязанной аппаратуры.

Способность к постепенной (не резкой) потере эффективности.

Способность восстанавливаться и завершать выполнение задания, начатого отказавшим процессором.

Способность к реконфигурации.

Эффективность:

Общая возможная скорость передачи информации.

Общий показатель эффективности использования аппаратуры.

Возможности аппаратных средств (функциональных блоков):

Число имеющихся входов.

Способность системы работать также в однопроцессорном режиме.

Местонахождение устройства контроля соединений (применение центрального коммутатора или распределение средств контроля по всем блокам).

Предполагаемое применение системы:

Требуемое программное обеспечение (система управления, прикладные программы).

Ниже указаны наиболее важные свойства трех основных типов аппаратных средств для организации мультипроцессорных систем.

Система с общей или разделенной во времени шиной:

Имеет самую низкую стоимость.

Обладает наименьшей сложностью — шина может быть вполне пассивной (хотя мультиплексирование при наличии одной шины может усложнить систему).

Наиболее легко осуществляется физическая реконфигурация системы путем добавления или удаления функциональных блоков.

Производительность системы ограничена быстродействием шины передачи данных.

Отказ шины приводит к катастрофическим результатам для системы в целом.

Расширение системы путем добавления одинаковых блоков может ухудшить общую производительность системы.

Обеспечивает самую низкую эффективность одновременного использования всех имеющихся блоков.

Обычно применяется только в малых системах.

Система с перекрестной коммутационной матрицей:

Наиболее сложна и дорога.

Содержит самые дешевые и простые функциональные блоки, так как ни один из них не требует специальных коммутационных схем.

Конфигурация (и обычно применяемые модули) пригодна только для мультипроцессорных систем.

Обладает потенциальной способностью обеспечить самую высокую общую скорость передачи информации.

Расширение системы обычно повышает общую эффективность системы.

Обладает потенциальной способностью к расширению без изменения программного обеспечения.

Теоретически расширение ограничено только размерами коммутационной матрицы, которые можно увеличивать в пределах технических возможностей, используя модульный принцип.

Надежность системы зависит от надежности одного функционального блока — коммутатора; этот недостаток может быть до некоторой степени ослаблен использованием пассивных схем коммутаторов и (или) разделением коммутатора на автономные блоки или введением избыточности в это ключевое устройство.

Многошинная — многовходовая система:

Для процессора не требуется специальной коммутационной аппаратуры.

Содержит самые дорогие запоминающие устройства.

Позволяет работать в режиме сравнительно дешевой однопроцессорной конфигурации, состоящей из тех же функциональных блоков, без изменения их конструкции и без снижения производительности.

Обладает потенциальной способностью обеспечить высокую скорость передачи информации.

Ограничена по размерам и числу возможных конфигураций из-за ограниченности числа и типов имеющихся входов в память.

Требует большое число кабелей и разъемов.

РАССМОТРЕНИЕ ОРГАНИЗАЦИИ МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМЫ С ФУНКЦИОНАЛЬНОЙ ТОЧКИ ЗРЕНИЯ

Прежде чем завершить обсуждение различных типов организации мультипроцессорных систем, следует сказать, что существует совершенно иной подход к рассматриваемой проблеме. Основной темой этой главы была топология размещения функциональных блоков и способ их соединения в единую систему. Существует дру-

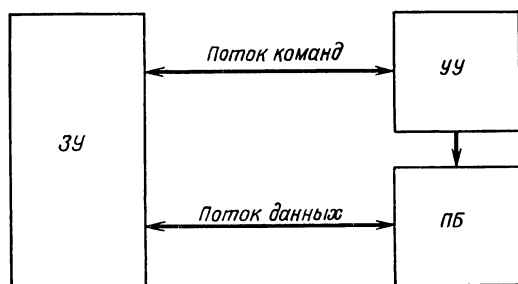


Рис. 2.35. Организация системы с одиночным потоком команд и одиночным потоком данных (ОПК — ОПД). Пример — стандартные однопроцессорные системы.

УУ — устройство управления; ПБ — процессорный блок.

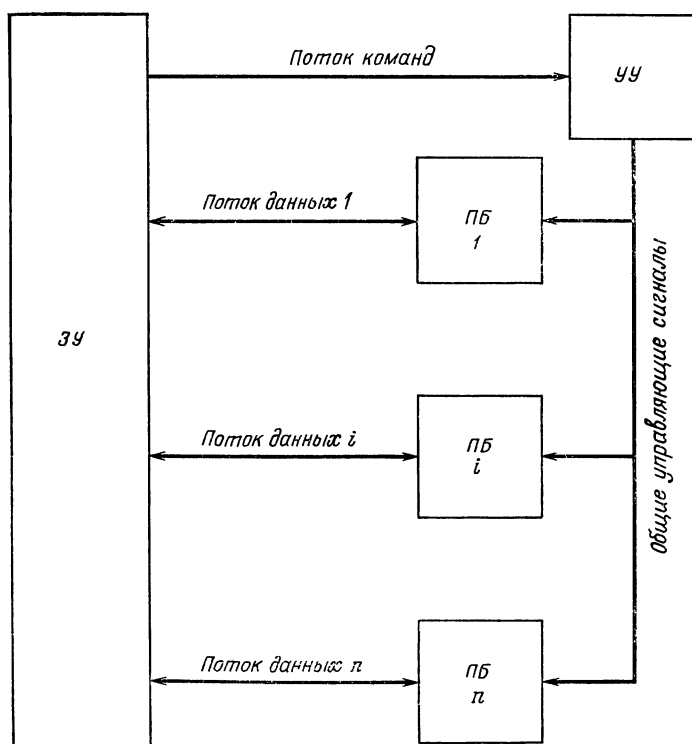


Рис. 2.36. Организация системы с одиночным потоком команд и множественным потоком данных (ОПК — МПД). Примеры — матричные и векторные процессоры и другие параллельные системы.

УУ — устройство управления; ПБ — процессорный блок.

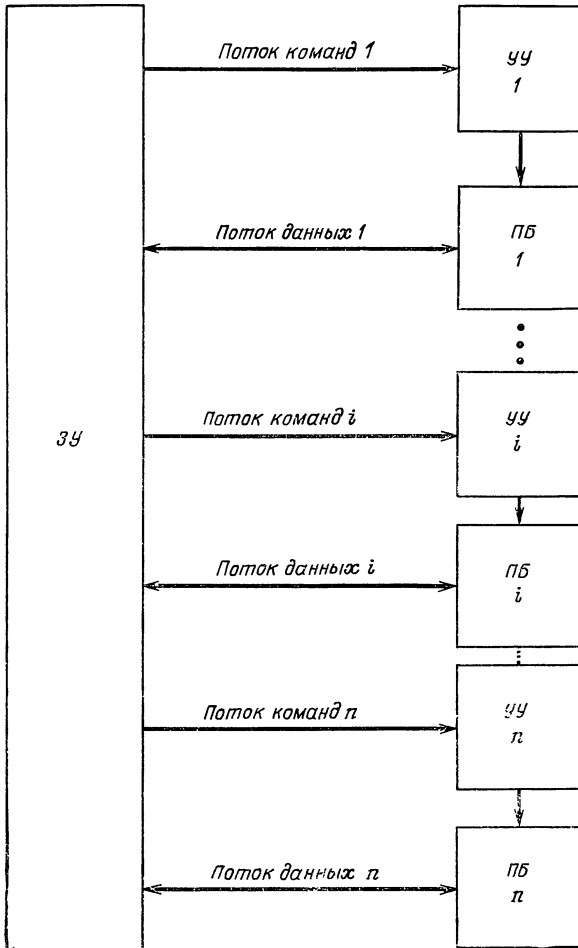


Рис. 2.37. Организация системы с множественным потоком команд и множественным потоком данных (МПК — МПД). Примеры — мультипроцессорные системы по определению, принятому в этой книге.

УУ — устройство управления; ПБ — процессорный блок.

гой способ классификации систем, основанный на учете числа потоков команд и потоков данных и их взаимодействия. Основы этой классификации предложены несколько лет тому назад в работах Флинна (M. Flynn). Наиболее важными представителями этой функциональной классификации системы являются:

СИСТЕМА «ОПК — ОПД» (ОДИНОЧНЫЙ ПОТОК КОМАНД —
ОДИНОЧНЫЙ ПОТОК ДАННЫХ)

Это базовая однопроцессорная система (рис. 2.35).

СИСТЕМА «ОПК — МПД» (ОДИНОЧНЫЙ ПОТОК КОМАНД —
МНОЖЕСТВЕННЫЙ ПОТОК ДАННЫХ)

К этому классу относятся матричные, параллельные и ассоциативные процессоры (рис. 2.36). Наиболее яркими представителями этого класса могут служить системы ILLIAC IV, REPE и STARAN. Системы CDC 6600 и 7600 и IBM S/360, модель 85, также принадлежат к этому классу, хотя их исполнительные блоки не синхронизированы.

СИСТЕМА «МПК — ОПД» (МНОЖЕСТВЕННЫЙ ПОТОК КОМАНД —
ОДИНОЧНЫЙ ПОТОК ДАННЫХ)

Этот класс систем приведен здесь лишь для полноты классификации. Примером могла бы служить многопроцессорная система с несегментированной памятью. Периферийные процессоры систем CDC 6000 и 7000 и система ASC фирмы Texas Instruments также принадлежат к этому классу.

СИСТЕМА «МПК — МПД» (МНОЖЕСТВЕННЫЙ ПОТОК КОМАНД —
МНОЖЕСТВЕННЫЙ ПОТОК ДАННЫХ)

В системе выполняется одновременно несколько программ, каждая со своими данными. Система содержит несколько процессоров и несколько запоминающих устройств. В ней применяется мультипрограммирование. Это классическая мультипроцессорная система, соответствующая определению, данному в этой книге (рис. 2.37). Однако системы, состоящие из нескольких независимых процессоров, также можно отнести к этому классу.

ОПЕРАЦИОННЫЕ СИСТЕМЫ И ДРУГОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

ВВЕДЕНИЕ

Различия между основными характеристиками и свойствами операционных систем мультипроцессора и однопроцессорной ЭВМ третьего поколения внешне не очень заметны. Однако внутренние они различаются довольно сильно. По существу, именно развитие операционных систем часто способствовало дальнейшему совершенствованию мультипроцессорных конфигураций и повышению их производительности. Одним из важных аспектов программного обеспечения систем, который отличает мультипроцессорную систему от однопроцессорной, является принципиальная необходимость развитой операционной системы для эффективного использования ресурсов мультипроцессора. Опытный программист может работать с однопроцессорной системой вообще без какого-либо системного программного обеспечения, хотя эффективность такого использования ЭВМ третьего поколения будет очень низка. Однако такой способ работы просто невозможен даже для простейшей мультипроцессорной системы. Обычно операционная система необходима уже на первых этапах создания системы. Действительно, в процессе разработки первых мультипроцессорных систем, таких, как NBS PILOT, Burroughs D825 (AN/GYK-3) и Ramo-Wooldridge RW-400 (AN/FSO-27), пришлось сразу же сосредоточить внимание на решении многих основных проблем, которые до сих пор встречаются при организации и использовании операционных систем даже в однопроцессорных системах. «Программа автоматической работы и планирования» (AOSP) в системе D825 фактически явилась первой важной вехой в развитии операционных систем. Необходимо иметь в виду, что мультипрограммирование (т. е. выполнение в одном процессоре нескольких резидентных программ в режиме совпадения) уже является важным свойством мультипроцессорной операционной системы и по сути дела может рассматриваться как подсистема общей мультипроцессорной операционной системы.

В настоящей главе не ставится цель детально изложить все сведения, которые необходимы для полного понимания хотя бы основ мультипроцессорных операционных систем. Поскольку они очень

сходны с операционными системами однопроцессорных ЭВМ, читателю предлагается обратиться к уже известным материалам по этому вопросу. Основное внимание в этой главе обращено на те требования, характеристики и функции, которые специфичны для мультипроцессорных систем.

Несмотря на то что технические средства мультипроцессорных систем могут быть более сложными, чем однопроцессорных, не следует думать, что их программное обеспечение также должно быть обязательно сложнее. Наличие дополнительных, избыточных функциональных блоков часто позволяет найти лучшие решения тех основных проблем, которые возникают при создании однопроцессорных ЭВМ, в частности:

Адаптируемость к изменяющимся требованиям в отношении характера и количественных характеристик потока заданий.

Наличие по крайней мере минимальной операционной системы для выполнения наиболее важных функций.

Возможность расширения системы без изменения существующих операционных программ.

Требуемое время ответа и т. д.

С другой стороны, присутствие этих добавочных элементов усложняет реализацию некоторых других важных функций:

Максимально возможное использование всех ресурсов системы.

Обнаружение и исправление ошибок без остановки работы системы.

Синхронизация вычислительных процессоров.

Структуризация прикладных программ для эффективного их выполнения.

Повторная входимость.

Следует отметить, что это далеко не полный список.

ОРГАНИЗАЦИЯ МУЛЬТИПРОЦЕССОРНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

В настоящее время имеется три основных типа организации и способа функционирования мультипроцессорных операционных систем (хотя возможны разные их варианты с небольшими различиями):

Ведущий-ведомый.

Раздельное выполнение заданий в каждом процессоре.

Симметричная, или однородная обработка во всех процессорах.

Основные характеристики этих типов организации систем представлены в табл. 3.1 и подробно обсуждены ниже.

СИСТЕМА ТИПА «ВЕДУЩИЙ-ВЕДОМЫЙ»

Выбор метода обработки «ведущий-ведомый» может диктоваться различием характеристик процессоров в системе; например, система NBS PILOT включает процессор, специально разработанный для управления и выполнения супервизорных функций. Этот метод используется также и в других приложениях ввиду его простоты и возможности максимального использования уже имеющегося программного обеспечения, разработанного для однопроцессорных ЭВМ. Метод был применен в одной из первых систем TRW-400, а в настоящее время применяется в системе SEL 86 (хотя имеются планы по замене ее другими системами, которые сейчас рекламируются как «полностью оперившиеся» мультипроцессоры). Свежим примером такого типа организации и функционирования операционной системы является система RCA-215, которая содержит единую исполнительную программу в резервной части блока памяти 1 и ядро восстановления программы в блоке памяти 2; надежность достигается за счет того, что вероятность отказа обоих блоков в одно и то же время очень мала.

Ниже представлены в обобщенном виде основные характеристики работы систем типа «ведущий-ведомый»:

Супервизорные функции реализуются только в одном из процессоров, который выбран оператором. Этот процессор может иметь специально разработанную конфигурацию только для выполнения супервизорных программ или же может иметь такую же конфигурацию, как все другие процессоры в системе. Специально разработанный супервизорный процессор, вероятно, будет использоваться исключительно для выполнения своих функций; если же для этой цели применяется процессор универсального типа, то он может быть использован и для расчетов по прикладным программам, когда супервизорные процедуры не выполняются.

Супервизорные программы не обязательно должны быть записаны при помощи процедур с повторной входимостью, поскольку они будут выполняться только одним процессором. Кодирование с повторной входимостью потребуется для некоторых общих программ, которые используются рекурсивно или подвергаются многократной загрузке в оперативную память.

Поскольку только один процессор имеет доступ к исполнительным таблицам, в системе не возникают конфликтные ситуации или блокировки, связанные с таким доступом.

Система подвержена отказам в работе при сбоях в ведущем процессоре. Если последний не является специализированным, то оператор может заново запустить систему, используя при этом в качестве ведущего другой процессор.

Система имеет малую гибкость и всегда состоит из ведущего процессора и одного или большего числа ведомых процессоров.

Таблица 3.1

Характеристики трех типов организации мультипроцессорных операционных систем

Ведущий-ведомый	Раздельное выполнение заданий в каждом процессоре	Симметричная, или однородная, обработка во всех процессорах
<p>Исполнительные программы всегда реализуются одним и тем же процессором.</p>	<p>Каждый процессор обслуживает только свои собственные потребности.</p>	<p>Функции ведущего передаются с одного процессора на другой.</p>
<p>Исполнительные программы не обязательно должны обладать повторной входимостью.</p>	<p>Супервизорные программы должны обладать повторной входимостью или дублироваться.</p>	<p>Лучшая балансировка загрузки.</p> <p>Конфликтные ситуации при обслуживании запросов разрешаются с помощью приоритетов.</p>
<p>Для исполнительных программ не возникает проблем, связанных с конфликтами при доступе к системным таблицам или блокировкой (для прикладных программ возможны конфликты при обращении к системным таблицам)</p>	<p>Каждый процессор имеет свой собственный набор таблиц.</p> <p>Одиночные отказы не приводят к полной потере работоспособности системы; однако повторный запуск отказавшего процессора может оказаться очень сложным.</p>	<p>Программные модули должны обладать повторной входимостью. Чтобы несколько процессоров могли выполнять одни и те же сервисные программы.</p>
<p>Сбои могут привести к полной потере работоспособности системы.</p>	<p>Практически каждый процессор имеет свои устройства ввода-вывода, файлы и т. д.</p>	<p>Поскольку несколько процессоров могут одновременно работать в супервизорном режиме, возможны конфликты, связанные с доступом к системным таблицам.</p>
<p>Сравнительно малая гибкость.</p>	<p>Реконфигурация системы ввода-вывода обычно связана с ручными переключениями.</p>	<p>Преимущества системы: возможность «постепенной деградации»; более высокая степень готовности; наличие реального резервирования; максимальная эффективность использования ресурсов.</p>
<p>Суммарное время простоя ведомых процессоров может быть значительным, если ведущий процессор имеет недостаточное быстродействие.</p>		
<p>Максимальная эффективность достигается для специальных приложений и в асимметричных системах</p>		
<p>Сравнительно простое аппаратное и программное обеспечение.</p>		

Ведущий процессор должен обеспечивать выполнение супервизорных функций достаточно быстро, чтобы с опережением удовлетворять выдвигаемым требованиям. Если это условие не выполняется, то эффективность использования системы может оказаться очень низкой. Даже когда «ведущий» представляет собой специализированную супервизорную ЭВМ, общая производительность системы может быть весьма низкой, если он будет перегружен выполнением супервизорных функций. Ведомый процессор, который оказывается свободным в то время, когда ведущий занят, вынужден ждать, пока ему будет назначена новая задача. Преобладание коротких задач в потоке заданий будет создавать сложные проблемы для систем такого типа в связи с тем, что такие задачи предъявляют высокие требования к исполнительным функциям.

Время простоя, накапливающееся в ведомых процессорах, делает рассматриваемый способ функционирования наиболее пригодным для специальных применений, допускающих простои. Он эффективен также для неоднородных систем, которые состоят из процессоров с существенно различающейся мощностью.

Примерами операционных систем типа «ведущий-ведомый» могут служить IBM S/360 TSS для модели 67, Burroughs B5500 Master Control Program (MCP) и Texas Instruments Advanced Scientific Computer (ASC). С точки зрения пользователя в эту категорию попадает также система CDC 6600, хотя внутренне она и не полностью соответствует методу «ведущий-ведомый».

«Как правило, системы «ведущий-ведомый» могут иметь простые структуры технических средств и программного обеспечения, но это достигается за счет потери гибкости» [13].

СИСТЕМА С РАЗДЕЛЬНЫМ ВЫПОЛНЕНИЕМ ЗАДАНИЙ В КАЖДОМ ПРОЦЕССОРЕ

Поскольку оперативная память используется в режиме разделения времени, в рассматриваемых системах отсутствует необходимость в отдельных полных наборах копий модулей операционной системы для каждого процессора. Однако для этого типа организации операционных систем могут возникать такие ситуации, когда каждый процессор должен работать автономно и выполнять все относящиеся к нему супервизорные, исполнительные и вспомогательные функции точно так же, как если бы он был однопроцессорной ЭВМ: каждая задача предназначается для конкретного процессора и полностью выполняется на этом устройстве. Ниже представлены в обобщенном виде характеристики организации и функционирования операционных систем этого типа.

Супервизорные функции выполняются каждым процессором в соответствии с *его собственными* потребностями и требованиями

со стороны программ, *предназначенных для этого процессора*. Поскольку модули супервизора выполняются несколькими процессорами, должна быть предусмотрена их повторная входимость или их копии должны быть загружены в каждый процессор. Иногда используется комбинация этих двух процедур.

Число конфликтов, связанных с блокировкой системных таблиц, невелико, так как каждый процессор может иметь свой собственный набор. При этом число общих исполнительных таблиц не будет большим.

При отказе какого-либо одного процессора система в целом не теряет работоспособности; однако восстановление и повторный запуск программ, которые выполнялись отказавшим устройством, обычно связаны с большими трудностями.

Все операции ввода-вывода для данной задачи выполняет процессор, на который эта задача назначена.

Прерывания по вводу-выводу поступают на процессор, который иницирует операции ввода-вывода.

По существу каждый процессор имеет свой собственный набор устройств ввода-вывода, файлы и т. д.

Совместное использование внешней памяти невозможно без специального кодирования.

Такая система может иметь невысокую эффективность, если один процессор загружен большими заданиями, в то время как другие простаивают.

Реконфигурация системы ввода-вывода требует ручного переключения.

СИСТЕМА С СИММЕТРИЧНОЙ, ИЛИ ОДНОРОДНОЙ ОБРАБОТКОЙ ВО ВСЕХ ПРОЦЕССОРАХ

Наиболее «чистой» конфигурацией аппаратных средств для мультипроцессорной системы является совокупность одинаковых процессорных блоков. Поскольку такая группа идентичных процессоров совместно использует общую память, а также каналы и устройства ввода-вывода, нет причин, по которым их нельзя было бы использовать симметрично. Каждый из процессоров может в равной степени эффективно выполнять супервизорные функции, что в действительности и имеет место. Исполнительные функции «перетекают» из одного процессора в другой. Целый ряд исполнительных функций весьма тесно связан с решаемой задачей и наилучшим образом реализуется тем же процессором, который решает данную задачу; однако имеется много и других функций, таких, как обработка прерываний для асинхронных операций ввода-вывода, которые могут быть выполнены любым процессором. Основной причиной использования этого способа работы мультипроцессора является стремление достичь высокой эффективности всей системы несмотря на трудности, обсуж-

даемые ниже. Описываемый тип системы, вероятно, распространен наиболее широко.

Ниже указаны характерные свойства симметричных систем.

Каждый процессор выполняет те супервизорные функции, которые неразрывно связаны с решаемой им задачей, и те функции, которые необходимы для новой задачи в случае, когда текущая будет прервана или полностью завершена. Вместе с тем любой процессор может выполнять все или большинство общесистемных функций.

Ввиду того, что процессоры однородны и могут быть использованы одинаково, любая задача в процессе своего выполнения может обрабатываться различными процессорными блоками системы. Для успешного ее выполнения могут быть использованы различные наборы процессоров.

Общесистемное управление непрерывно перераспределяется между процессорами:

- один из процессоров, обеспечивающий управление системными таблицами и выполнение таких функций, как планирование, называется исполнительным;

- в каждый момент времени только один процессор может быть исполнительным (для предотвращения конфликтных ситуаций);

- для процессоров может быть установлен определенный приоритет, во-первых, для разрешения конфликтных ситуаций и, во-вторых, для ранжирования управляющих функций; последнее оказывается наиболее эффективным в системах, работающих с большим числом коротких задач. Если бы задачи не были ранжированы, то потребовалось бы много времени на разрешение конфликтных ситуаций и выход из критических положений [28].

Лучшая балансировка загрузки и более эффективное использование системы возможны за счет того, что неспецифические функции, такие, как прерывание для ввода-вывода, могут выполняться теми блоками, которые наименее загружены в данный период времени.

Несмотря на то что в «общесистемном управлении» только один процессор является исполнительным, несколько других могут одновременно выполнять те же самые супервизорные процедуры; поэтому эти процедуры должны обладать повторной входимостью или должны быть обеспечены отдельные копии для каждого случая запуска.

Поскольку в каждый момент времени несколько процессоров могут находиться в супервизорном состоянии, очень важной проблемой являются конфликты при доступе к системным таблицам и наборам данных:

частая блокировка системных управляющих таблиц может сильно повлиять на общую эффективность системы; существенное значение имеют блокировки каждого набора данных и допустимые временные задержки, так как один процессор может попытаться обратиться к тем записям, которые в этот момент модифицируются другим.

Система обладает несколькими важными преимуществами:

при выходе из строя отдельных устройств система частично сохраняет работоспособность (т. е. происходит постепенная деградация);

при правильной организации системы машинное время используется лучше, чем в системах со специально предусмотренным дублированием аппаратуры;

реализуется единственный способ эффективного резервирования для обеспечения надежности;

наиболее эффективно используются ресурсы.

Хотя рассматриваемая организация представляет собой наиболее совершенное воплощение идеи мультипроцессора, большинство систем терпят неудачу, когда начинается их эксплуатация (например, система IBM 9020 для FAA). Тем не менее один проект с самого начала был посвящен разработке программного обеспечения и аппаратных средств для симметричной системы — Carnegie Multi-Mini-Processor [13].

ОСНОВНЫЕ ТРЕБОВАНИЯ К ФУНКЦИОНАЛЬНЫМ ХАРАКТЕРИСТИКАМ

Несмотря на то что функциональные характеристики операционных систем мультипроцессоров и мультипрограммных однопроцессорных ЭВМ весьма сходны, между ними существуют и значительные различия. Одно из наиболее важных различий касается надежности программного обеспечения. Сбои программного обеспечения (ошибки, которые не могут быть автоматически исправлены с помощью программ восстановления) в однопроцессорной системе, как правило, не выводят систему из строя на длительный период времени. Напротив, в мультипроцессоре, когда несколько центральных процессорных блоков могут выполнять одни и те же кодовые операции или различные их части, ошибка в программном обеспечении или временный отказ аппаратуры в одном из центральных процессоров могут быстро распространиться по всей системе, мгновенно создавая такие сложные ситуации, что разобраться в них и внести исправление бывает очень трудно, или, по крайней мере, это связано с большими затратами времени.

Операционная система мультипроцессора должна делать все то, что и операционная система обычной ЭВМ, но, кроме того, она дол-

жна дополнительно выполнять ряд важных функций, обусловленных конфигурацией аппаратных средств мультипроцессорной системы. Эти функции обсуждены ниже.

РАСПРЕДЕЛЕНИЕ И УПРАВЛЕНИЕ РЕСУРСАМИ

Основные задачи распределения и управления ресурсами можно классифицировать по тем же признакам, что и в однопроцессорных системах.

Планирование. Выбор заданий из входной очереди для создания набора активных заданий.

Управление памятью. Распределение, перераспределение и управление центральной памятью системы как для выполнения прикладных программ, так и для функционирования системного программного обеспечения.

Диспетчеризация. Назначение процессорного блока для выполнения данной задачи. (Диспетчеризацию в мультипроцессорной системе можно было бы рассматривать как назначение задачи для определенного процессора, однако такая точка зрения может привести к некоторой путанице, поскольку в большинстве существующих систем задача до своего завершения проходит обработку на нескольких процессорах.)

Сходство с однопроцессорными системами особенно велико, если имеются в виду некоторые наиболее сложные и совершенные однопроцессорные операционные системы, разрабатываемые в настоящее время. Традиционной проблемой здесь остается конфликт между несколькими одновременно выполняемыми процессами, требующими одних и тех же системных ресурсов, включая память, процессоры, данные, системные таблицы, процедуры, устройства ввода-вывода и т. д. Однако при работе с мультипроцессорными системами возникает ряд специфических аспектов этой проблемы.

РАСПРЕДЕЛЕНИЕ И УПРАВЛЕНИЕ ПАМЯТЬЮ

По отношению к пользователю или прикладной программе проблемы, возникающие при работе с мультипроцессором, почти идентичны проблемам однопроцессорных систем; поэтому и решения этих проблем аналогичны — они предусматривают разбиение памяти на участки, сегменты, страницы.

Для системных программ (например, супервизорных и вспомогательных) приходится решать одну и ту же проблему формирования резидентной и транзитной областей памяти. Кроме того, в зависимости от типа операционного управления, используемого в мультипроцессорной системе (см. предыдущий раздел), для каждого процессора может потребоваться собственная память (т. е. память, фи-

зически недоступная для других процессоров). Эта собственная память особенно важна для хранения программ восстановления, собственных системных таблиц и т. д.

Из всех проблем, связанных с организацией распределения памяти и ее управлением в мультипроцессорах, две главные проблемы касаются специальных вопросов технического и программного обеспечения разбиения памяти на страницы и управления доступом к командам или данным, которые могут использоваться более чем одним пользователем. По-видимому, наиболее сложное и совершенное решение этой проблемы для мультипроцессорных систем достигнуто в системе MULTICS.

Поскольку эффективность функционирования любой системы, имеющей постраничную организацию памяти, в значительной степени зависит от аппаратной реализации «автоматического» преобразования адресов запоминающих устройств, возможность разбиения памяти на страницы всегда тесно связана с технической организацией системы (т. е. наличием многих входов, общей шины) и характеристиками самих функциональных блоков.

В системах с постраничной памятью ключевым вопросом в отношении преобразования адресов является вопрос о том, где осуществляется это преобразование — в памяти или в процессоре. Эта характеристика относится только к логическим операциям, хотя физическое расположение предназначенных для этого аппаратных средств также имеет важное значение.

Возможны три способа расположения таблиц переадресации страниц, или регистров, применяемых для отображения логических адресов в физические: в оперативных запоминающих устройствах, в процессорных блоках и в специальном устройстве отображения, или преобразования адресов памяти, которое имеет очень высокую скорость логических операций и в некоторых случаях использует ассоциативную организацию памяти.

Если регистры расположены в блоках оперативной памяти или по крайней мере логически связаны со специальными модулями памяти, то это значит, что используется лишь один набор регистров и запоминающее устройство имеет только один вход. Это в свою очередь ограничивает выбор организации для таких систем либо конфигурацией с одной общей шиной, либо конфигурацией с перекрестной коммутацией. Конечно, если адреса различных блоков памяти организованы так, что возможно их чередование в процессе работы системы, то связь регистров переадресации страниц только с одним блоком памяти невозможна, так как в зависимости от характера чередования они должны использоваться двумя или четырьмя различными блоками (см. раздел гл. 2 о назначении адресов памяти).

Если регистры переадресации расположены в процессорных блоках, то схемы переадресации для различных программ должны быть

определенным образом «сцеплены» одна с другой. Когда в системе производится обработка различного рода прерываний или вызовов супервизорных или исполнительных программ, для процессора, реагирующего на вызов, может потребоваться доступ к схеме распределения памяти запрашивающей программы, которая не обязательно находится в обслуживаемом процессоре.

Во всяком случае, если схема распределения памяти изменяется динамически, то плохая синхронизация может привести к ошибкам.

При исследовании языков и компиляторов для мультипроцессорной системы необходимо иметь в виду, что система управления памятью, по всей вероятности, должна также иметь возможность обработки общих наборов данных (COMPOOLS).

ПЛАНИРОВАНИЕ И ДИСПЕТЧЕРИЗАЦИЯ

Даже в тех случаях, когда операционная система управляет всеми процессорами симметрично, алгоритмы планирования и диспетчеризации не всегда достаточно эффективно используют наличие в системе нескольких процессорных блоков. Чаще всего этот недостаток проявляется тогда, когда планировщик назначает определенное задание или задачу для некоторого определенного процессора, а диспетчер с этого момента времени считает, что только данный процессор может выполнять это задание. Очевидно, при этих условиях в некоторые периоды времени очередь готовых задач будет выстраиваться к одному процессору, в то время как другие процессоры простаивают. Рассмотренная ситуация представляет собой «проблему первого уровня» в системах планирования и диспетчеризации мультипроцессоров.

Для решения проблемы первого уровня необходимо, чтобы диспетчер, так же как и планировщик, для мультипроцессорной системы были значительно более совершенными, чем подобные им элементы в однопроцессорных системах. Планировщик должен четко определять взаимосвязи между различными задачами, составляющими сложное задание, а диспетчер должен затем управлять как сравнительно простыми операциями, связанными с выполнением независимых задач, так и более сложными действиями по координации выполнения комплекса задач, зависящих одна от другой, например при обработке одного набора данных последовательностью различных операций. Определение перечня задач и их частей, которые могут выполняться параллельно, является наиболее важной проблемой, решение которой необходимо для наиболее полного использования возможностей мультипроцессорной системы. Эта «проблема второго уровня» является предметом многочисленных исследований; она рассмотрена в разделе, специально посвященном этому вопросу.

«Проблема третьего уровня» тесно связана с необходимостью равномерной загрузки всех процессоров в каждый период времени. Эта

проблема, известная также как проблема «аномалий диспетчеризации», хорошо иллюстрируется примером из работы [29]:

«Техническая литература... содержит множество примеров, когда попытки сократить время выполнения одной или большего числа задач приводят в результате к увеличению общего времени выполнения всего набора задач. Этот результат, противоречащий интуиции, может иметь место тогда, когда сокращение времени решения задачи изменяет очередность обработки всех задач, вызывая тем самым необходимость сложных изменений порядка выполнения

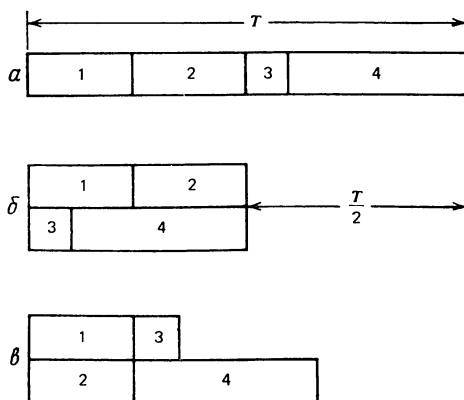


Рис. 3.1. Аномалии диспетчеризации в мультипроцессорных системах [29].

каждой задачи в связанном наборе... Уменьшение реального времени решения конкретной задачи, ослабление упорядоченности по старшинству между задачами и добавление новых процессоров может действительно увеличить общее время, требуемое для обработки последовательного набора задач... Эта проблема иллюстрируется на рис. 3.1.

На рис. 3.1,а представлен мультипроцессор с одним процессорным блоком. Если к «идеальной» мультипроцессорной системе добавить второй процессор, то для обработки данного списка задач потребуются лишь половина затраченного ранее времени. Однако в реальных мультипроцессорных системах требуемое для обработки время зависит от порядка, в котором решаются задачи; это ясно видно из схем на рис. 3.1,б и в. В данном случае необходимо, чтобы каждой отдельной последовательности задач (рис. 3.1,а) соответствовало время выполнения T/N . Вообще еще нет ни оптимальной, ни какой-либо другой методики для реализации этого условия. Для некоторых последовательностей задач оно может оказаться в принципе невозможным, например если набор (а) состоит из одной задачи, которая не может выполняться параллельно».

Решение проблемы третьего уровня для неделимых задач может потребовать применения сложных методов прогнозирования с целью оценки времени решения задач и других требований к ресурсам системы.

Некоторые авторы исследовали граничные, или наихудшие условия, которые возможны в решениях этой проблемы. В частности,

Грэхем (Graham) пришел в некотором отношении к пессимистическим заключениям. Однако всегда полезно исследовать наихудший случай, чтобы оценить необходимость более детальных исследований в данной области. Ряд других эмпирических результатов, приведенных Грэхемом, также интересен с точки зрения определения вероятности достижения «оптимальной» производительности системы. Одна из ситуаций, представляющих практический интерес, была смоделирована Мэнахером (Manacher) в 1967 г., в результате чего было найдено, что при использовании оптимальных списков планирования мультипроцессорная система все же работает неоптимально в течение 80% времени, если время выполнения каждого задания имеет небольшие случайные колебания (что обычно случается на практике).

Мультипроцессор должен рассматриваться как многоканальная система обслуживания с очередями, в противоположность однопроцессорной системе, которая даже при использовании в ней мультипрограммирования является более простой одноканальной обслуживающей системой. Реджис (Regis) изучал влияние методов обслуживания очередей на эффективность использования процессоров и среднее время ответа для каждой задачи. Хотя полученные им результаты позволяют несколько глубже понять эту проблему, все же планирование в мультипроцессорной системе все еще остается «слабо оптимизируемой» процедурой.

ВЗАИМОСВЯЗЬ ПРОЦЕССОРОВ

Если два или большее число процессоров должны совместно выполнять одно или несколько заданий, им необходима определенная взаимосвязь друг с другом. Эта связь между двумя процессорами может быть запланированной, как, например, в случае, когда один из процессоров выполняет независимые задачи большого задания, назначенного в целом для другого процессора, или незапланированной, т. е. возникающей как реакция на сигнал о возникновении ошибки в одном из процессоров или при обслуживании прерывания для ввода-вывода.

Эти совершенно необходимые связи обычно обеспечиваются комбинацией аппаратных средств и программного обеспечения. Один из методов, называемый иногда методом бокового ответвления, предусматривает возможность прерывания одного процессора другим, как, например, в системе IBM S/360, модель 65 MP. Переданное сообщение в этом случае помещается в «почтовом ящике» — специальной области оперативной памяти, выделенной для управляющей системы. Другой, также часто используемый метод заключается в том, что каждый процессор периодически сам опрашивает свой почтовый ящик по сигналу прерывания, поступающему от часов реального времени со сравнительно небольшим интервалом — порядка

200 мс. В любом случае адресат изменяет содержимое почтового ящика, сигнализируя о том, что он получил сообщение и уже выполнил или выполнит предписанные ему действия. Источник сообщений обязан контролировать работу почтового ящика при помощи соответствующих подтверждений о получении сообщения и выполнять необходимые действия на основе результатов такой проверки.

НЕНОРМАЛЬНОЕ ОКОНЧАНИЕ ОБРАБОТКИ ЗАДАНИЙ

Принципы организации мультипроцессорных систем предполагают наличие постоянной взаимной связи между процессорами. Отсутствие этой связи делает затруднительным, если вообще возможным, выполнение в полном объеме сложного задания, состоящего из многих отдельных задач, при его обработке одновременно на нескольких процессорах. Однако требования, вытекающие из необходимости нормального окончания такого задания, не являются какими-то уникальными, присущими только мультипроцессорной системе; скорее это более сложные требования, предъявляемые к однопроцессорным системам.

БАЛАНСИРОВКА НАГРУЗКИ ПРОЦЕССОРОВ

Выполнение этой функции в значительной степени определяется способностью нескольких процессоров к осуществлению взаимной связи и является ключевым фактором для достижения максимальной производительности мультипроцессорной конфигурации. Рассмотрим следующий пример:

Задание 1

высокий приоритет;

две задачи — *A* и *B*.

Задание 2

низкий приоритет;

одна задача *C*.

Процессор *X* выполняет задачу *A*.

Задача *B* находится в состоянии ожидания, так как не завершена операция ввода-вывода.

Процессор *Y* выполняет задачу *C*.

Если в процессоре *X* возникает прерывание, указывающее на окончание операций ввода-вывода для задачи *B*, то *X* будет единственным процессором, которому известно, что задача с высоким приоритетом *B* готова для обработки. Эта информация должна быть передана процессору *Y*, с тем чтобы он мог временно отложить обработку задачи *C* и переключиться на выполнение более приоритетной задачи *B*.

ЗАЩИТА СИСТЕМНЫХ ТАБЛИЦ И НАБОРОВ ДАННЫХ

Хотя защита системных таблиц и наборов данных от неправомерного доступа необходима для любой, в том числе и однопроцессорной системы, наличие нескольких процессоров, каждый из которых может выполнять одни и те же или тесно связанные части супервизорных программ, требует использования очень сложных процедур для обеспечения такой защиты. Дело в том, что неправомерный доступ к таблице в мультипроцессорной системе может произойти в пределах цикла работы памяти, следующего непосредственно за правомерным доступом, прежде чем соответствующие программы смогут эффективно заблокировать таблицу. Поэтому в мультипроцессорной системе необходимо предусматривать некоторые аппаратные средства для обеспечения защиты.

БАЛАНСИРОВКА ЗАГРУЗКИ СИСТЕМЫ ВВОДА-ВЫВОДА

Обеспечение балансировки загрузки ввода-вывода в рассматриваемых конфигурациях вычислительных средств и в трактах ввода-вывода мультипроцессорных систем может оказаться чрезвычайно трудным делом. Если все каналы и оборудование ввода-вывода образуют однородный ансамбль, то можно применить простой приоритетный принцип для формирования очередей запросов на ввод-вывод с подробным планированием ресурсов примерно тем же способом, как это делается в однопроцессорной системе. Основные трудности возникают в тех случаях, когда некоторые тракты или оборудование ввода-вывода назначаются исключительно (без права передачи) одному процессору. Это имеет место в некоторых многошинных конфигурациях, где проблема планирования и управления вводом-выводом должна решаться при помощи механизма межпроцессорной связи.

РЕКОНФИГУРАЦИЯ

Многие однопроцессорные системы в ограниченных пределах способны распознавать изменения в оборудовании, которое в данный момент готово к использованию. Однако в случае мультипроцессора способность осуществлять это автоматически и вносить необходимые коррективы во все системные таблицы и программы является наиболее важным требованием к его операционной системе. Очевидно, такие возможности наиболее существенны и вместе с тем наиболее просто реализуемы в системах, которые создаются для работы в режиме симметричного выполнения потока задания.

КЛИНЧ¹

Проблему клинча, или тупиковых ситуаций в системе иногда образно называют «проблемой смертельных объятий». Она возникает тогда, когда каждая из двух активно выполняемых задач требует в одно и то же время определенные ресурсы, назначенные для другой задачи, причем ни одна из них не может ни продолжать обработку, ни отказаться от захваченных ресурсов; этот спор не может быть разрешен, пока одна из задач не получит все требуемые ресурсы. Очень часто, когда возникает такое безвыходное положение, существует только один путь восстановить работоспособность системы. А именно, необходимо полностью снять одно из заданий и высвободить все назначенные ему ресурсы, попытавшись восстановить то состояние, в котором они находились до использования их отвергнутым заданием. Если высвобожденные ресурсы представляют собой набор данных, который обновлялся или модифицировался в процессе выполнения снятого задания, то задача «восстановления» может оказаться настолько сложной, что вряд ли удастся решить ее успешно. Если оба вида ресурсов, за обладание которыми ведется спор, были только «частично изменены» их первым пользователем, то может возникнуть необходимость прекратить работу обеих программ. В любом случае большая часть завершенной работы оказывается проделанной впустую, и требуются огромные дополнительные усилия, чтобы спасти то немногое, что может быть спасено. Очевидно, оператор за пультом должен обладать наивысшей квалификацией, чтобы определять, что и как надо делать в этих ситуациях. Менее очевиден тот факт, что даже только для того, чтобы заметить появление клинча в системе, необходимы очень большое внимание и опыт.

Возникновение явления клинча может быть связано с использованием всех видов ресурсов:

Устройства ввода-вывода (как специальные устройства, так и устройства обычного типа).

Наборы данных.

Программы без повторной входимости.

Основная память.

Поскольку в действительности нет какого-либо средства для облегчения последствий клинча, важной проблемой является его предупреждение. Обычно с этой целью все «уникальные» ресурсы, такие, как специфические наборы данных или специальное оборудование, назначаются для определенной задачи до того, как ее выполнение запланировано и начата ее реализация. Однако в от-

¹ Полное и систематическое обсуждение этого явления содержится в работе [12].

ношении остальных ресурсов невозможно точно сказать, какие из них будут причиной клинча, чтобы предпринять необходимые меры для его предотвращения.

Характерный пример клинча приводится в одной из основополагающих работ по этому вопросу [26]. Средства системного ввода OS/360 или ASP фирмы IBM, которые используют программы READER и WRITER, полностью отделены от основных программ для ввода и вывода и распределяют только ту область памяти на диске, которая необходима для хранения наборов входных и выходных данных. Так как программа READER функционирует совершенно независимо от обрабатывающих программ, она непрерывно записывает на диск новые наборы данных до тех пор, пока они находятся на устройстве считывания перфокарт или другом источнике входных данных. Критическая ситуация возникает тогда, когда вся область заполнена наборами входных данных для заданий, которые только еще будут выполняться, и наборами выходных данных для заданий, которые выполнены лишь частично. К сожалению, ни одна из двух указанных выше операционных систем не может обеспечить какие-либо процедуры для очистки областей системного ввода от частично выполненных заданий, вследствие чего неизбежно возникает клинч. После этого приходится систему запускать повторно, потеряв при этом результаты уже завершенной работы, не выданные системой. Холт [26] замечает: «...грубое решение этой проблемы... заключается в запрещении ввода новых заданий, когда коэффициент использования области системного ввода становится слишком высоким, скажем, выше 80 %».

В общем исследовании [12], посвященном этому вопросу, указаны четыре условия, которые приводят к возникновению клинча; авторы утверждают, что эти условия являются необходимыми и достаточными:

1. Задачи претендуют на исключительное право распоряжаться ресурсами, которые они затребовали (условие «взаимного исключения»).
2. Задачи удерживают уже выделенные для них ресурсы, ожидая в то же время назначения дополнительных ресурсов (условие «ожидания»).
3. Ресурсы не могут быть принудительно изъяты у тех задач, которые удерживают и используют их, до полного завершения обработки этих задач (условие «отсутствия предпочтения»).
4. Существует замкнутая цепь задач, в которой каждая задача удерживает один или больше видов ресурсов, которые требуются следующей задаче в цепи (условие «циклического ожидания»).

В этой работе также обращается внимание на то, что предупредить или избежать возникновения первого условия для всех ресурсов практически невозможно. Очевидно, необходимо предоставлять возможность только одиночного доступа к файлам или системным таблицам при их модификации. Однако другие три условия могут быть исключены.

Хейвендер (Havender) предлагает следующий подход, который практически позволяет избежать любое из этих трех условий:

1. Каждая задача должна запрашивать сразу все необходимые ей ресурсы, и ее обработка не может быть запущена, пока ей не будут предоставлены все эти ресурсы (устранение условия «ожидания»).

2. Если задаче, удерживающей определенные ресурсы, отказано в последующих запросах, то она должна высвободить уже назначенные ей ресурсы и, если потребуются, вновь запросить их вместе с необходимыми дополнительными ресурсами (устранение условия «отсутствия предпочтения»).

3. Необходимо применение для всех задач линейного ранжирования типов ресурсов; это означает, что если для задачи были назначены ресурсы типа g , то она может последовательно запрашивать ресурсы только тех типов, которые по порядку следуют за g . В том случае, когда каждый тип ресурсов содержит только один элемент, линейно упорядочиваются эти элементы. (Можно показать, что при этом граф состояния не имеет циклов, вследствие чего не может возникнуть условие «циклического ожидания» [12].)

Важно отметить, что реальное присутствие в системе нескольких активных программ всегда создает опасность возникновения условий клинча, которой подвержены как однопроцессорные, так и мультипроцессорные системы.

СПЕЦИАЛЬНЫЕ ПРОБЛЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Как было отмечено выше, при работе с операционными системами мультипроцессоров возникают почти такие же проблемы, как и при использовании однопроцессорных систем. При этом часто оказываются приемлемыми одни и те же решения. Основное различие заключается в глубине и сложности всестороннего анализа проблем в мультипроцессорной системе и в степени уверенности, что полученное «решение» действительно решает данную проблему.

Ниже рассматриваются задачи, которые уже привлекли или должны привлечь самое серьезное внимание при разработке мультипроцессорных систем. Представленный ниже перечень никоим образом не может считаться исчерпывающим как в отношении задач, общих для однопроцессорных и мультипроцессорных систем, так и в отношении задач, характерных только для мультипроцессорных систем. Он содержит лишь немногие из тех специальных проблем, которые привлекли наибольшее внимание специалистов в самых последних разработках и научно-исследовательских проектах.

РАЗДЕЛЕНИЕ ПАМЯТИ И ОРГАНИЗАЦИЯ ДОСТУПА

Здесь необходимо рассмотреть две проблемы, связанные с конфликтами при доступе к разделенной памяти. Первая из них определяется организацией аппаратного обеспечения памяти и механизмом доступа к каждому ее блоку. Возможности, или, лучше сказать,

ограничения, обусловленные доступом в режиме совпадения и совместным использованием несколькими процессорами одних и тех же блоков центральной памяти, могут значительно снизить общую производительность системы из-за соперничества при организации доступа к памяти. «Задержки, вызванные соперничеством процессоров при обращении к памяти, могут достигать по некоторым оценкам 20 %» [45]. Однако эту величину очень трудно и получить, и проверить.

Один из подходов к решению (по крайней мере частично) этой проблемы заключается в распределении последовательных адресов между следующими друг за другом блоками памяти. Этот метод кажется на первый взгляд многообещающим; однако при его использовании все же может происходить большое число конфликтов, хотя в этих условиях они будут более случайными по своему характеру и не приведут к быстрому распространению конфликта по всей системе, как это могло бы произойти в случае, когда два процессора обрабатывают программы в одном и том же блоке памяти. Подобная проблема, связанная с конфликтами при доступе к памяти, может возникать и в однопроцессорной системе; однако здесь она ограничена конфликтом между процессором и контроллерами ввода-вывода при организации их совместного доступа к основной памяти. Конечно, конфликты, связанные с доступом к памяти устройств ввода-вывода, могут происходить также и в мультипроцессорных системах.

Возможен другой подход к решению рассматриваемой проблемы, требующий применения более сложных аппаратных средств, чем распределение и чередование адресов. Он дает значительно больший выигрыш, так как позволяет организовать действительно одновременный доступ к блокам памяти через два отдельных входа. Этот подход использован в системе SEL 85 для создания прямого и независимого канала ввода-вывода; два процессорных блока имеют доступ к памяти через отдельные входы, причем конфликты разрешаются на приоритетной основе.

Еще один способ организации аппаратных средств, используемый также в некоторых однопроцессорных системах, связан с созданием быстродействующей буферной памяти, объединенной с процессорным устройством (например, в системе IBM S/360, модели 85 и 95, и в системе IBM S/370, модели 155, 158, 165, 168, 195). Эта буферная память является собственной памятью процессора и обычно имеет существенно большее быстродействие, чем основная память. Например, центральный процессор системы IBM 370/165 может получить информацию объемом 8 байтов из буферной памяти за два цикла, или за время 160 нс, причем запрос может генерироваться на каждом цикле. Для получения 8 байтов информации непосредственно из основной памяти процессору потребуется 1,44 мкс (или 18 циклов)¹. Буферная память загружается блоками, а не отдельными

¹ «A guide to the IBM System/370 Model 165», IBM Form GC20—1730.

словами, причем используется предварительный просмотр ее содержимого; поэтому, если программа не содержит слишком много переходов и в буфер за одно обращение может загружаться большое количество последовательных ячеек памяти, то число обращений процессора к основной памяти можно значительно сократить. Для системы IBM 360/85 с буферной памятью объемом 16 384 байтов подробный анализ 19 контрольных лент (содержащих приблизительно по 250 000 команд) показал, что вероятность обнаружения требуемых данных в буфере составляет 0,968 [46]. Следует отметить, что если бы адреса, выдаваемые программой, имели чисто случайный характер, то эта вероятность могла бы уменьшиться до 0,01. Поэтому можно считать, что эффективная работа буферной памяти в реальных условиях определяется тем, что программы не применяют случайную адресацию. Общее время, затраченное на доступ, будет также уменьшаться, так как буферная память загружается блоками. Очевидно, наибольший выигрыш при такой организации будет получен в том случае, когда объем локальной памяти настолько велик, что она может вместить коды полного цикла программы, который выполняется последовательно несколько раз. Изучение характеристик системы IBM 360, модель 85, с буферной памятью показало, что при выполнении репрезентативного набора заданий средняя производительность системы составила 81% от производительности, которая могла бы быть достигнута, если бы вся центральная память с быстродействием 1,04 мкс работала со скоростью буферной памяти. В системе IBM 360, модель 85, центральный процессор имеет рабочий цикл длительностью 80 нс и может получать 8 байтов информации из буферной памяти за каждый цикл.

Было проведено также моделирование буферной памяти на ЭВМ [45]. В такой модели компилятор должен был просматривать коды программы и, обнаружив «полезные» блоки, формировать соответствующую таблицу, которая загружалась в специальный банк памяти. Основным фактор, препятствующий широкому использованию локальной, или буферной, памяти,— это высокая стоимость ее аппаратной реализации, которая должна обеспечивать необходимое быстродействие и достаточно большой объем. Однако быстрое снижение стоимости быстродействующих запоминающих устройств на монолитных схемах сделало возможным создание буферов с объемом памяти от 8К до 32К байтов для системы IBM S/370, модели 155—195.

ДОСТУП К СИСТЕМНЫМ ТАБЛИЦАМ И НАБОРАМ ДАННЫХ И ИХ ЗАЩИТА

Конфликты второго рода, связанные с организацией доступа, относятся к проблемам программного обеспечения системы, хотя для их разрешения могут использоваться и некоторые аппаратные сред-

ства. Эта проблема возникает при доступе в режиме совпадения двух или большего числа процессоров к таблицам или наборам данных, когда один из процессоров может внести изменения в эту информацию при ее использовании. Эти изменения, не согласованные с логикой использования данной информации другими процессорами, могут привести к непоправимым ошибкам в результатах или действиях других процессоров. Так как одновременность доступов сама по себе еще не создает проблемы (она возникает, когда один процессор получает доступ, в то время как другой еще использует данный блок информации), точно такая же проблема может иметь место и действительно довольно часто встречается в однопроцессорных системах, хотя здесь речь идет об отдельных процессах или задачах, выполняемых в одной машине.

Поскольку с подобной проблемой впервые столкнулись несколько лет назад в однопроцессорных системах, она стала предметом интенсивного изучения. Наиболее известным исследованием по этому вопросу является работа, посвященная методу «семафоров» [19]. Другие работы хорошо характеризуются их названиями, такими, как «синхронизация», «блокировка таблиц», или просто «блокировка». (Следует отметить, что эта проблема принципиально отличается от «системного клинча», рассмотренного выше, хотя клинч может явиться результатом выполнения неверных операций или использования ошибочных данных, полученных при доступе к управляющей таблице, которая оказалась частично модифицированной.) Наиболее важными с точки зрения решения обсуждаемой проблемы являются управляющие таблицы, такие, как таблицы очередей центрального процессора и системы ввода-вывода, и таблицы распределения ресурсов, поскольку они во многом определяют работу системы и к ним наиболее часто требуется доступ.

Принципы решения рассматриваемой проблемы весьма просты. Если какой-либо набор данных открыт для одной из программ, которая может внести в него изменения, то для этого набора устанавливается «замок» защиты, который должен проверяться любой другой программой, прежде чем она также сможет выдать команду OPEN (открыто) для данного набора. Новая программа не может использовать набор данных до тех пор, пока защита не будет снята предыдущим пользователем при помощи команды CLOSE (закрыто) для данного набора. Подобная процедура используется и при организации доступа к таблицам, хотя здесь и отсутствует команда OPEN. Приведенное объяснение создает обманчивое впечатление, что рассматриваемая проблема решается довольно легко; но даже небольшой опыт написания и отладки реальных контрольных и блокирующих программ быстро обнаруживает ошибочность такого мнения.

Для облегчения создания эффективных программ установки и проверки замков защиты в большинстве существующих процессоров сейчас имеются базовые команды TEST и SET. При помощи этих

команд проверяется контрольный разряд в памяти, выделенной для хранения специальных наборов данных и таблиц. Если этот разряд содержит нуль, то он заменяется на единицу, и программа получает разрешение на доступ к требуемой информации. Если же в нем уже установлена единица, то содержание разряда не изменяется, и программа, запросившая доступ, должна приступить к выполнению другой подпрограммы или перейти в состояние ожидания, если нет каких-либо работ, которые можно было бы выполнить в это время. Последний случай является наиболее распространенным, поскольку запрашиваемая информация часто относится к управляющим таблицам, используемым операционной системой. Именно такие ситуации обуславливают большую часть суммарного времени ожидания. В особенности это справедливо для тех систем, в которых процессор может блокироваться диспетчером.

Встречаются ситуации, когда второй пользователь не может получить доступ к системной таблице, даже если его запрос связан только с чтением. Ниже приведен конкретный пример такой ситуации [основанный отчасти на работах Мэдника (Madnick)].

Исходные условия:

Все задания, готовые к обработке, включены в очередь.

Каждое следующее задание выбирается для обработки из условия «первый пришел — первый обслужен» или в циклическом порядке и идентифицируется с помощью указателя его места в очереди.

Освободившийся процессор обращается к очереди заданий для того, чтобы определить, какое следующее задание нужно обрабатывать.

Процедура SELECT ROUTINE выбирает «следующее задание для выполнения», находящееся в начале очереди.

Процедура ADVANCE ROUTINE перемещает указатель в очереди для идентификации следующего задания.

Операционная система распределена между различными процессорами, по крайней мере до такой степени, что каждый процессор выбирает свое собственное «следующее задание для выполнения».

Последовательные действия, приводящие к конфликтной ситуации:

1. Два процессора *A* и *B* освободились почти одновременно.
2. В процессоре *A* процедура SELECT ROUTINE отмечает в качестве «следующего задания для выполнения» задание 12.
3. В процессоре *B* процедура SELECT ROUTINE отмечает в качестве «следующего задания для выполнения» задание 12.
4. В процессоре *A* процедура ADVANCE ROUTINE перемещает указатель «следующего задания для выполнения» на задание 13.

5. В процессоре *B* процедура ADVANCE ROUTINE перемещает указатель «следующего задания для выполнения» на задание 14.

6. Процессор *A* начинает выполнять задание 12.

7. Процессор *B* начинает выполнять задание 12.

Результат:

Оба процессора выполняют задание 12.

Задание 13 оказалось пропущенным.

Один из методов, позволяющих исключить такие ситуации, предусматривает использование взаимной блокировки диспетчерских таблиц и таблиц очередей [45].

ИСПРАВЛЕНИЕ ОШИБОК

Поскольку одной из характерных черт мультипроцессорной системы является ее устойчивость к аппаратным сбоям в отдельных устройствах и способность к реконфигурации с целью сохранения непрерывности работы, может показаться, что исправление ошибок не является самостоятельной задачей для операционной системы мультипроцессора. На самом деле это утверждение не соответствует действительности. Все потенциальные возможности реконфигурации, «постепенной деградации» системы и др. были бы бесполезны при отсутствии специального программного обеспечения, необходимого для их практической реализации.

Ниже приведены некоторые специальные методы и приемы для исправления ошибок.

Все системные таблицы и таблицы управления задачами формируются в двойном количестве, дублированные экземпляры размещаются в разных банках памяти. В качестве примера законченной реализации этого метода можно привести систему Burroughs Multi-Interpreter [18].

Операционная система должна быть составлена таким образом, чтобы ее можно было приспособить и эффективно использовать как в системе с максимальной конфигурацией, так и в любой возможной подсистеме. Реальные условия работы должны определяться автоматически функциональными блоками при помощи таких процедур, как установка индикаторного разряда в управляющем слове основной памяти, который проверяется операционной системой для определения текущего состояния системы. (Примером может служить система Burroughs B 6500 и другие вычислительные системы этой фирмы.)

Обнаружение и исправление ошибок, обусловленных неисправностью аппаратных средств, должны быть по возможности всеобъемлющими. Необходимо контролировать все важнейшие точки системы, причем этот контроль не должен влиять на эффектив-

ность работы системы. Это достигается, например, использованием контроля по четности при обращении к памяти, контроля по четности при передаче всех данных, применением самоконтролирующей аппаратуры в арифметико-логических устройствах, контроля устройств ввода-вывода путем чтения после записки или перфорации и т. д.

Должны быть подготовлены эффективные программы контроля, которые обрабатываются на процессорах, не занятых в данный момент прикладными программами; это позволяет обнаружить ошибки еще до того, как они проявятся в процессе реализации прикладных программ.

Процессор должен допускать возможность таких ситуаций, когда задача, обрабатываемая неисправным устройством, имеет доступ ко всем системным таблицам и управляющей информации, используемой в этом устройстве. Устройство — приемник, которому передается результат решения, должно «чувствовать» состояние, в котором находится неисправное устройство, с тем чтобы оценить момент, когда следует взять на себя выполнение задачи.

Если в системе произошла ошибка или сбой аппаратуры, что обычно обнаруживается по прерываниям, то для восстановления процесса обработки наиболее важны последствия сбоя, хотя для ликвидации неисправности существенно также определение причин этого сбоя. В этом разделе мы рассматриваем в основном только вопросы восстановления процесса обработки в связи с прерываниями по сбою. Когда анализируются сбои, результаты их воздействия на систему и условия управления реконфигурацией, необходимо дать ответы на два указанных ниже вопроса, что в общем случае приводит к четырем возможным состояниям восстановления [17]:

		Осталась ли ненарушенной база данных?	
		Да	Нет
Возможна ли реконфигурация системы, при которой сохраняется некоторая минимально допустимая производительность?	Да	I	III
	Нет	II	IV

Состояние I. Система может быстро изменить конфигурацию: управление возвращается к диспетчеру для продолжения работы с обычной загрузкой системы.

Состояние II. Даже после реконфигурации системы необходимо полностью исключить из обработки все несущественные задачи,

а также сократить число или временно отложить выполнение задач с низким приоритетом.

Состояние III. После реконфигурации и восстановления исходной мощности системы потребуется ее новая загрузка и перезапуск (желательно, чтобы эти операции начинались с последней контрольной точки).

Состояние IV. После реконфигурации и повторной загрузки производительность системы будет все же ниже нормального уровня.

РАСПОЗНАВАНИЕ ПАРАЛЛЕЛЬНЫХ УЧАСТКОВ В ЗАДАНИИ И ИСПОЛЬЗОВАНИЕ РЕЖИМА ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ

Эта тема занимает настолько важное место среди вопросов, связанных с созданием высокопроизводительных мультипроцессорных систем, что она могла бы стать предметом рассмотрения для целой главы или даже целой книги; кое-что по этой теме уже опубликовано [45]. Проблема может быть исследована по частям, хотя, очевидно, существует тесная взаимосвязь между различными частями проблемы и способами их решения.

Для идентификации задач и подзадач, которые могут выполняться параллельно, в языки программирования должны быть включены специальные операторы и другие средства, которые снабжают операционную систему информацией, необходимой для управления параллельным режимом работы системы:

Спецификация начальных и конечных точек участков программ, которые могут выполняться параллельно.

Задание специальных условий (если они существуют) для управления или ограничения параллельного выполнения программ.

Задание условий, которые должны выполняться для синхронизации параллельно выполняемых процессов друг с другом и с другими частями программы.

Супервизор должен обладать способностью создавать, планировать, диспетчировать и перераспределять «независимо параллельные» задачи, порожденные одной программой.

Распознавание параллельных участков программ и определение их протяженности может быть выполнено:

Самим программистом.

Автоматически — транслятором языка.

Проблеме распознавания параллельных участков программ были посвящены многочисленные исследования, однако удовлетворительное решение еще не получено, а исследования продолжаются [24, 34, 42]. Одним из наиболее вероятных кандидатов на параллельную обработку является программный цикл, если при каждом следующем прогоне в нем используются разные элементы данных, как это обыч-

но имеет место при операциях с векторами и матрицами или при многократном обращении к базе данных. По существу оператор DO в языке ФОРТРАН часто служит для выражения метода параллельных вычислений. Во многих других случаях также имеется возможность эффективно использовать несколько процессоров для решения одной и той же задачи. Не всегда удается легко выделить участки, которые могут обрабатываться независимо, даже если для этой цели используются блок-схемы программ. Решение такой задачи в общем случае связано со сложным и искусным применением теории графов, поэтому наибольшее число исследований концентрируется именно вокруг использования методов этой теории. Но даже применение теории графов не освобождает программиста от трудностей выделения параллельных участков в программе, поскольку для этого требуется проверка большого числа всевозможных комбинаций кодов. По-видимому, в конечном итоге нужно будет осуществлять этот процесс автоматически.

Большинство мультипроцессорных систем может работать довольно эффективно, даже не используя возможности параллельного выполнения отдельных частей обрабатываемых программ. В этих условиях высокая загрузка каждого процессора обеспечивается за счет наличия общего большого пула программ, каждая из которых выполняется последовательно; программы из пула могут назначаться диспетчером на любой процессорный блок. Пул должен содержать достаточное количество активных программ, для того чтобы всегда имелось необходимое число заданий в состоянии готовности, которыми можно загрузить каждый процессор. Очевидно, даже очень большой набор заданий, ограниченный возможностями системы ввода-вывода, будет мало пригоден для такой работы, вследствие относительно большого времени пребывания системы в состоянии ожидания ввода-вывода.

В противоположность «стандартному» мультипроцессору с последовательной загрузкой заданий при использовании векторных, матричных или магистральных мультипроцессорных систем возникают многочисленные специальные проблемы. Дело в том, что если структура программы в принципе не пригодна для параллельного выполнения, то эти системы становятся малоэффективными. Министерством обороны США была проведена значительная исследовательская работа по определению областей применения систем ILLIAC IV и CDC STAR, в которых они могут дать наибольший выигрыш. Идеальным условием использования матричного процессора является, очевидно, наличие большого числа матричных операций такого типа, как в расчетах, связанных с исследованием атомной энергии или моделированием погоды по данным, собранным от большой сети станций наблюдения. Менее очевиден тот факт, что даже программы расчетов платежных ведомостей содержат большое число параллельных вычислений. Для каждого работника выполняются одни и те же

расчетные операции, в которых различаются только такие элементы исходных данных, как количество отработанных часов и тарифные ставки. Так как каждый элементарный процессор системы ILLIAC IV имеет свою собственную оперативную память небольшого объема для хранения текущих операндов, эта система может легко решать задачи по расчету платежной ведомости, обеспечивая занятость всех элементарных процессоров. Однако при этом, вероятно, возможности системы ILLIAC IV не будут использоваться с максимальной эффективностью.

В качестве примера операторов, которые могут быть использованы для обеспечения параллельной обработки информации, можно привести пару команд FORK-JOIN из языка Gamma-60. Оператор FORK (ветвление) имеет следующую форму:

$$L1 \text{ FORK } L2, J, N.$$

Эта запись устанавливает наличие N независимых задач, готовых для параллельной обработки. Коды этих задач начинаются с ячеек $L1+1, \dots$ и т. д. Все эти задачи должны объединиться до оператора с меткой J , причем программа находится в состоянии ожидания, пока не будет закончено выполнение всех N задач. Оператор JOIN (объединение) может быть определен аналогичным образом. В языке PL/1 использование атрибута TASK в процедуре вызова приводит к возникновению двух ветвей, а объединение их управляется оператором WAIT (ожидание). Этот оператор, однако, не очень подходит для мультипроцессорной работы, поскольку он накладывает ограничения на многократные ветвления в программе. Фактически он предназначен для использования в однопроцессорной системе при организации мультипрограммирования и позволяет осуществлять перекрытие операций центрального процессора и системы ввода-вывода.

Даже в одной из первых мультипроцессорных систем — системе Burroughs D825 — программисту предоставлялась возможность при помощи операционной системы AOSP разбивать свою программу на сегменты и идентифицировать те из них, которые могут обрабатываться параллельно [2]. На программисте также лежала ответственность за назначение точек объединения, следующих за параллельными участками.

Подводя итог вопросам, рассмотренным в этом разделе, отметим, что распознавание в программе участков, подлежащих параллельной обработке, и организация такой обработки могут выполняться:

Непосредственно пользователем.

При помощи анализа программы в ЭВМ.

С использованием комбинации двух указанных способов.

РАЗРАБОТКА И КОНТРОЛЬ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ

Программное обеспечение представляет собой комплекс управляющих и обслуживающих программ, необходимых программисту и оператору ЭВМ для эффективного использования и управления аппаратными средствами системы. Основной частью системного программного обеспечения является операционная система. Конечно, все трудности, связанные с созданием и отладкой операционной системы для однопроцессорной системы, характерны и для мультипроцессорных систем, однако в системах последнего типа все проблемы становятся значительно сложнее. Это касается не только создания логических схем и их анализа, но и вопросов тестирования; исчерпывающая проверка всей системы программного обеспечения становится практически невозможной из-за трудности воспроизведения полной картины всех возможных событий, которые могут произойти в системе, с точным заданием их порядка и времени. Конечно же, число различных ситуаций и их комбинаций, возникающих в мультипроцессоре, является практически бесконечным, так что любой контроль системы будет всегда неполным.

Подлинный контроль всех кодов большой программы представляет собой невероятно трудную задачу, поэтому в процессе разработки программы внимание должно быть сосредоточено на скрупулезном и постоянном соблюдении точности и надежности кодирования. Этот подход может привести к большим успехам, о чем свидетельствует описание разработки и усовершенствования операционной системы Т. Н. Е. [Dijkstra, 1968].

Для того чтобы максимально ослабить или устранить операционные ограничения, необходимо как можно шире использовать форму загрузочных модулей с повторной входимостью. Такой подход не только облегчает процесс контроля программного обеспечения, но и обеспечивает максимальную гибкость при выборе метода организации и режима работы системы в целом.

Наконец, поскольку система всегда является результатом работы целого коллектива, очень важно четко определить функции отдельных модулей, из которых, как из строительных блоков, будет собрана вся система программного обеспечения. При этом недостаточно только указать, например, что «пятый модуль будет осуществлять планирование работ». Спецификация должна быть сделана очень подробно, чтобы можно было точно определить задачи, выполняемые модулем, и всю совокупность его взаимодействий с другими модулями. На последнюю проблему следует обратить особое внимание. В настоящее время большая часть программ создается на модульной основе; однако методы «модульного программирования» еще не всегда применяются правильно. Часто сбой в программе вызывается некоторыми изменениями внутри какого-либо модуля, в связи с тем, что к си-

стеме был добавлен другой модуль, имеющий новую организацию. При использовании методики «чистого модульного программирования» такие случаи вряд ли смогут возникнуть, поскольку все модули связываются друг с другом *только через полностью унифицированный интерфейс* и эта унификация является *общей для всех интерфейсов*. Это означает, что один модуль никогда не сможет войти внутрь другого, чтобы получить нужный ему элемент данных или модифицировать его. Любая передача информации должна осуществляться через интерфейс, где она может быть полностью проконтролирована любым человеком или группой лиц, исследующих потоки программ и данных. Применение рассмотренных здесь процедур дает многосторонний выигрыш, как при разработке программ и их контроле, так и в дальнейшем, когда необходимо внести определенные изменения в какой-либо из модулей или добавить к системе новый модуль.

Приведенные выше рассуждения в основном были сосредоточены на вопросах разработки и контроля программ. Это, однако, не означает, что задачи системного анализа и проектирования отодвинуты на второй план. Как было указано выше, можно сравнительно просто определить, что должна делать операционная система, и, вероятно, не слишком сложно разработать логические схемы, реализующие необходимые действия. Однако по-настоящему сложная проблема заключается в оценке того, насколько работа этих схем соответствует поставленным целям и насколько эффективно выполняется эта работа. Даже после того, как программа полностью подготовлена и пропущена через систему, ответы на поставленные выше вопросы, по всей вероятности, еще не будут получены. Единственный способ решения этой проблемы, известный автору, заключается в использовании метода имитационного моделирования и проведении экспериментов, которые могут быть повторены много раз для одних и тех же условий, с тем чтобы получить статистически достоверные результаты. Для этого необходимо иметь точную модель событий, или входов системы, и модель, задающую время их появления. Можно воспользоваться универсальным языком моделирования типа GPSS или какой-либо другой системой моделирования, ориентированной на события. Однако сейчас существует несколько специализированных языков и систем моделирования, которые предназначены для исследования проблем такого типа. В качестве примеров можно указать Имитационную модель для исследования систем и программного обеспечения (System and Software Simulator, S3), машину для системного анализа (System Analysis Machine, SAM) и язык описания операционной системы (Operating System Language, OSL). Моделирование программного обеспечения и аппаратных средств вычислительных систем очень полезно для проверки таких функций операционной системы, как исправление ошибок, управление реконфигурацией, обработка прерываний, а также других функций, которые

в значительной степени зависят от времени или выполняются только при возникновении особых ситуаций, поскольку при моделировании можно полностью управлять как временным фактором, так и условиями работы системы.

Технологический процесс, который в общих чертах представлен ниже, позволил бы, по всей вероятности, создать хорошую операционную систему для мультипроцессора в максимально сжатые сроки и с минимальными общими затратами:

1. Определить рабочую конфигурацию аппаратных средств.
2. Установить общие требования к рабочим характеристикам всей системы и ее производительности.
3. Определить состав модулей, их функции, требования к ним и единый стандартный интерфейс.
4. Определить детальную логическую схему каждого модуля.
5. Разработать имитационные модели отдельных модулей и всей системы управления программой на логическом уровне.
6. Используя имитационную модель, осуществить подробную проверку работы системы и оценку ее характеристик для того, чтобы убедиться, что все поставленные требования выполняются.
7. Внести необходимые изменения в функциональное описание каждого модуля и его логическую схему.
8. Повторять шаги с 5 по 7, пока все требования к системе не будут удовлетворены.
9. Закодировать программу для каждого модуля и произвести проверку ее точности с максимальной тщательностью.
10. Собрать модули в единую систему и подвергнуть ее контрольным тестам.

ЗАКЛЮЧЕНИЕ

Подводя итог этому весьма краткому обсуждению программного обеспечения мультипроцессорных систем, трудно не поддаваться искушению повторить еще раз те же замечания, которые неоднократно делались в каждом разделе этой главы: программное обеспечение мультипроцессорных систем имеет точно такие же цели, проблемы и трудности, как и операционные системы мультипрограммных однопроцессорных систем, но при этом является значительно более сложным. Однако такое утверждение все же сильно упрощает реальную ситуацию.

Конечно, оставаясь в рамках общих положений и определений, можно сказать, что такие функции, как управление и распределение ресурсов в системах, относящихся к двум указанным классам, очень сходны. Однако реальные различия между ними заключаются не только в разной степени их сложности; они значительно глубже. Поэтому полезно будет специально остановиться на анализе этих различий.

Первое и наиболее важное различие обусловлено тем, что в мультипрограммной однопроцессорной системе нескольких программ выполняются *в режиме совпадения с перекрытием*, а в мультипроцессорной системе несколько программ выполняются *одновременно и параллельно*. Одна характерная особенность здесь уже очевидна: при плохом планировании и диспетчеризации эффективность использования ресурсов сильнее снизится именно в мультипроцессоре.

Второе и, возможно, единственно важное из остальных различий заключается в характере взаимодействия программного обеспечения с операционной обстановкой, в которой функционируют аппаратные средства системы. Программное обеспечение однопроцессорной системы обычно предназначено для заданного состава аппаратных средств, объединенных в фиксированную конфигурацию. Конечно, это относится к системе, которая установлена у потребителя, приспособлена к специфическим условиям эксплуатации и удовлетворяет соответствующим требованиям. Что касается мультипроцессорной системы, то *все* элементы ее программного обеспечения должны предназначаться для работы в любой меняющейся обстановке, причем программное обеспечение должно не только нормально работать в таких условиях, но и быть способным управлять устройствами, функционирующими в настоящий момент времени, так же как и всеми аппаратными средствами, представляющими систему при ее максимальной конфигурации.

Эти два различия выявляют смысл и более емкое содержание общего заключения, неоднократно упомянутого в этой главе при сравнении программного обеспечения мультипроцессорных систем с программным обеспечением однопроцессорных систем: «похожее, но более сложное».

ГЛАВА 4

НАСТОЯЩЕЕ И БУДУЩЕЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Мы просмотрели несколько групп сравнительных таблиц, цель которых — показать преимущества одной системной организации перед другой по качеству или соотношению эффективности и стоимости. К сожалению, каждая группа таблиц была составлена приверженцами той или иной конкретной системы. Поэтому сравнительная оценка всех условий и предположений, которыми пользовались их составители для того, чтобы показать, что данная система имеет самые высокие показатели, является почти неразрешимой задачей. Мы не сомневаемся в достоверности представленной информации, хотя ее восприятие и обобщенный анализ затруднительны, особенно если учесть, что почти все результаты противоречивы. Причины этого очевидны. Эффективность любой системы, которая была спроектирована специально для обработки данных в режиме совпадения и параллельными методами, почти полностью зависит от весьма тонких факторов описания рабочей нагрузки. Этим можно объяснить то, что относительно малое число мультипроцессорных систем способно к универсальному решению задач, хотя, безусловно, некоторые из этих систем могут работать и в таких условиях.

Вследствие этих затруднений и сильной зависимости работы мультипроцессоров от специфических характеристик рабочей нагрузки оказалось невозможным в столь кратком введении дать четкие рекомендации относительно того, где и почему должны применяться мультипроцессорные системы. К сожалению, наши замечания по этим вопросам не выходят за пределы общих сведений о достоинствах и недостатках и о некоторых других неспециальных характеристиках этих систем.

СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ СОВРЕМЕННЫХ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ И СИСТЕМ С ПАРАЛЛЕЛЬНОЙ ОБРАБОТКОЙ

Возможно, лучший способ оценить какой-либо конкретный метод или технологию в столь динамичной области, как вычислительная техника, — это определить ее популярность. Несмотря на многие не-

достатки и нерешенные проблемы, объединение функциональных блоков в мультипроцессорные системы становится общепринятой практикой. В настоящее время существуют такие системы всевозможных размеров. Они могут быть поставлены любым из ведущих производителей вычислительной техники (табл. 4.1).

В табл. 4.1 представлены аппаратные характеристики более 50 мультипроцессорных и параллельных систем обработки данных. Некоторые из этих систем больше не производятся, некоторые еще только разрабатываются. Все они были включены в таблицу, чтобы дать возможно более полное представление о состоянии дел в этой области на сегодняшний день. В таблице указан тип системной организации или схемы взаимных соединений как между памятью и процессорами, так и между памятью и устройствами ввода-вывода. Приведены *максимальные* возможности аппаратуры для каждой из систем. Следует отметить, что аппаратура часто предоставляет значительно большие возможности по расширению системы, чем имеющаяся операционная система. К сожалению, нам оказалось не под силу найти аналогичные подходы к оценке возможностей системного программного обеспечения для систем. Если операционная система обеспечивает только уменьшенную конфигурацию оборудования, то это отмечено специальным примечанием.

СТЕПЕНЬ ДОСТИЖЕНИЯ МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМОЙ ЦЕЛЕЙ, ПОСТАВЛЕННЫХ ПРИ ЕЕ ПРОЕКТИРОВАНИИ

ЦЕЛИ И ПРЕИМУЩЕСТВА

Три основные цели, преследуемые при проектировании мультипроцессорных систем, почти в точности совпадают с тремя преимуществами, которыми по замыслу должны обладать эти системы:

Высокая готовность и надежность.

Гибкость функционирования системы.

Более высокие характеристики качества (эффективность и производительность).

ГОТОВНОСТЬ

Утверждение о том, что мультипроцессорная системная организация является наилучшим средством достижения высокой готовности любой вычислительной системы, функционирующей в условиях большой рабочей нагрузки, почти не встречает возражений, так как оно основано на учете аппаратных и топологических особенностей такой организации. Единственной альтернативой может быть полное

Характеристики мультипроцессорных и параллельных систем

Фирма-изготовитель и название системы	Системная организация ¹		Максимальное число функциональных блоков		Память ³
	процессоры	система ввода-вывода	процессоры	система ввода-вывода ²	
Мини-ЭВМ Microdata MICRO 1600D	PB (раздельные)	PB	2	Одна шина с PB на ЦП и один канал прямого доступа к памяти на сис- тему ⁴	$\{16 \times 4K\} \times 8 \text{ бит}$ $\{8 \times 8K\}$ (максимум 64 К)
Малые ЭВМ UNIVAC EMR 6155 Series-60 Xerox SIGMA — 57	PB ⁶ MB (общие)	PB ⁶ MB	2 5 ⁸	—/6 на ЦП ⁶ / 63 на канал 5 ⁸ /8 на ПВВ/ 32 на канал	$6 \times 8K \times 16 \text{ бит}$ $\{32 \times 4K\} \times 32 \text{ бит}$ $\{16 \times 8K\}$ (максимум 131 К)
Средние ЭВМ Digital equipment DECsystem-10/1055 Honeywell HIS 2088 Memorex MRX/40 и 50 RCA model 215	PB (множественные) MB PB (общие)	PB MB PB ПК	2 2 7 ¹⁰ 4	—/4 ⁹ /126 —/16 ⁵ /96 ⁵ 7 ¹⁰ /—/ —/ 4 / 1 мпк или 2 слк на контрол- лер ввода-вывода /—	$16 \times 16K \times 36 \text{ бит}$ $4 \times 131K \times 8 \text{ бит}$ (максимум 64 К байтов) $8 \times 32K \times 36 \text{ бит}$
Systems Engineering Labs SEL 88 Xerox SIGMA — 6 и 77	MB MB	MB MB	3и 7 ¹²	3и /—/ 7 ¹² /8 на ПВВ/32 на канал	$16 \times 8K \times 32 \text{ бит}$ $8 \times 16K \times 32 \text{ бит}$

Умеренно большие ЭВМ Burroughs B5700 (B5500) ¹³ Digital Equipment DECsystem-10/1077 UNIVAC model 498 Xerox SIGMA — 8 ⁷ Xerox SIGMA — 9 ⁷	ПК	ПК (общие)	2	4/4/—	8 × 4К × 48 бит
	РВ	РВ (множественные)	2	—/4 ⁹ /126	64 × 64К × 36 бит
	МВ	МВ	11 ¹⁴	11 ¹⁴ /8 на ПЭВ/32 на ка- нал	8 × 16К × 32 бит
	МВ	МВ	11 ¹⁴	11 ¹⁴ /8 на ПЭВ/32 на ка- нал	32 × 16К × 32 бит
	ПК	ПК (общие)	3	3/36 ¹⁵ /256	$\begin{cases} 64 \times 16К \\ 16 \times 64К \end{cases} \times 48 \text{ бит}$ (максимум 1048 К)
	ПК	РВ	2	20/24/—	32 × 4К × 60 бит
	ПК	РВ	2	20/24/—	32 × 4К × 60 бит
	МВ	МВ	4	4/96/—	16 × 32К × 36 бит
	МВ	МВ	4	4/96/—	16 × 64К × 36 бит
	МВ	МВ	4	4/96/—	32 × 64К × 36 бит
Большие ЭВМ Burroughs B6700 (B6500) Control Data CYBER72 (эквивалентна CDC 6200) Control Data CYBER73 (эквивалентна CDC 6400 и 6500) Honeywell HIS 6050/6060 Honeywell HIS 6070/6080 Honeywell HIS 6180 Honeywell HIS 635 Honeywell HIS 645 (MULTICS) IBM S/360, Model 65 MP IBM S/360, Model 67 UNIVAC 1108	МВ	МВ	4	4/96/—	16 × 32К × 36 бит
	МВ	МВ	4	4/96/—	16 × 64К × 36 бит
	МВ	МВ	4	4/96/—	32 × 64К × 36 бит
	МВ	МВ	4	4/64/—	8 × 32К × 36 бит
	МВ	МВ	4	4/64/—	8 × 32К × 36 бит
	МВ	МВ	2 ¹⁶	—/12 слк /— 2 МПК /—	8 × 256К × 8 бит
	МВ	МВ	2	2/12 слк /— 2 МПК /—	8 × 256К × 8 бит
	МВ	МВ	3	2/32/—	4 × 65К × 36 бит
	МВ	МВ			
	МВ	МВ			

Продолжение таблицы 4.1

Фирма-изготовитель и название системы	Системная организация ¹		Максимальное число функциональных блоков		Память ³
	процессоры	система ввода-вывода	процессоры	система ввода-вывода ²	
Очень большие ЭВМ Burroughs B7700	ПК	ПК	7 ¹⁰	7 ¹⁰ /224/255	$\begin{Bmatrix} 8 \times 13K \\ 4 \times 262K \end{Bmatrix} \times 48 \text{ бит}$ (максимум 1048К) 32×4К×60 бит
Control Data CDC 6500	ПК	PB	2	20/24/—	32×4К×60 бит
Control Data CDC 6600	ПК	PB	1	20/24/—	32×4К×60 бит
Control Data CDC 6700	ПК	PB	2	20/24/—	32×4К×60 бит
Control Data CYBER—74 (эквивалентна CDC 6600 и 6700)	ПК	PB	2	20/24/—	32×4К×60 бит
IBM S/370 model 158MP	MB	MB	2	—/10 бл-мпк /— 4 мпк	8×1024К×8 бит
Сверхбольшие ЭВМ					
Control Data CDC 7600	ПК	PB	1	13/15/—	МЗУ ¹⁷ 32×2К×60 бит БЗУ 8×64К×60 бит
Control Data CYBER—76 (эквивалентна CDC 7600)	ПК	PB	1	13/15/—	МЗУ 32×2К×60 бит БЗУ 8×64К×60 бит
IBM S/370 model 168 MP	MB	MB	2	—/24 (максимум 22 бл-мпк) /—	16×1024К×8 бит
UNIVAC 1110	MB	MB	6 ¹⁸	4/96/—	8×32К×36 бит ¹⁹
Гигантские и специальные системы U. S. Army Adv. Ballistic Missile De-	M—B	MB	288 ²¹	3 ²² /—/—	Память данных ЭП 1×1К×32 бит

fense Agency PERE ²⁰												Память программ	БУАУ 1×4К×32 бит БУКУ 1×2К×32 бит БУАВУ 1×2К×32 бит
Bell Labs CLC (SAFEGUARD)	MB	MB	10	5/80/80								БУАУ 2×32К×32 бит БУКУ 1×2К×32 бит БУАВУ 1×2К×32 бит	63×4К×64 бит ²³ 32×4К×64 бит ²⁴ 16×4К×48 бит
Burroughs D825 Control Data STAR- 100	ПК —	ПК —	4	2/20/128	—								
Goodyear Aerospace STARAN «S»	М—В	М—В	8192	—/2 ³⁵ /—									1×32К×32 бит ²³ 32×256К×256 бит ²⁴ 1×2К×64 бит (на процессоре) 16×16К×32 бит
Burroughs ILLIACIV	М—В	РВ	64 ²⁶	27									
Hughes H 4400	ПК	ПК (общие)	7 ²⁸	7 ²⁸ /8 на ПВВ/—									
IBM 9020A (360/50) (специальная сис- тема FAA)	МВ	МВ	4	3/—/—									
IBM 9020D (360/65) (специальная сис- тема FAA)	МВ	МВ	4	3/—/—									
IBM 4-pi, EP/MP	МВ	МВ	3	2 (пп)/(пп)									8×16К×32 бит 11×16К×32 бит
IBM 4-pi, CC-1 (AWACS)	МВ	МВ	2	2/2 на / (пп)									
Texas Instruments ASC (Advanced Scientific Com- puter)	Мг	РВ	2 ³⁰	8 ³¹ /— ³² /— ³²									8×512К×32 бит ³³ 8×256К×32 бит ³⁴
Ramo-Woolbridge RW-400	ПК	ПК	—	—									
U.S. Navy AADC (Advanced Airborne Digital Computer)	27	27	27	27									

Фирма-изготовитель и название системы	Системная организация ¹		Максимальное число функциональных блоков		Память ³
	процессоры	система ввода-вывода	процессоры	система ввода-вывода ²	
UNIVAC ARTS III (Air Traffic Control)					
UNIVAC AN/UUK-7	MB	MB	3	2 / 2/18 /—	16×16K×32 бит
UNIVAC Model 1832 (авиационная ЭВМ)	MB	MB	2		3×32K×32 бит

¹ Сокращения, используемые в таблице:

РВ — разделение времени (общая шина);
ПК — перекрестная коммутация;
МВ — многоходовая память;
М—В — матрица — вектор;
Мг — матричная обработка.

² Информация в этом столбце расположена в следующем по рядке: процессоры ввода-вывода/каналы ввода-вывода/устройств ввода-вывода (контроллеры ввода-вывода). Тип имеют следующие два значения:

В позиции процессоры ввода-вывода — в системе нет отдельных явно различных процессоров ввода-вывода; каналы сообщаются непосредственно с памятью.

В позиции каналы ввода-вывода и устройств ввода-вывода — максимальное число этих элементов точно не установлено и не ограничено основной конструктивной схемой системы.

³ Число в этом столбце обозначает следующие характеристики: число модулей/размеры модуля/длина слова в памяти.

Для подсистемной каждой процессорной системы к общей системной шине требуется адаптер мультипроцессорной связи. Все передачи по системной шине должны быть обобщенными, т. е. на передающем и на приемном конце должны осуществляться явно выделенные программные операции.

⁴ На одну процессорную систему.

⁵ Отдельная шина на каждую систему. Допускается работа с собственными и разделенными запоминающими устройствами на магнитных сердечниках, а также с разделенным дисковым запоминающим устройством и с шестью собственными мультиплексными каналами ввода-вывода. Имеется также обходная шина, соединяющая медленное устройство ввода-вывода непосредственно с процессором.

⁶ В ЭВМ SIGMA-5 и в более крупных моделях допускается подключение нескольких центральных процессоров и процессоров ввода-вывода, число которых ограничено емкостью имеющихся входов в память; однако стандартное программное обеспечение, поставленное фирмой Хьюз, предназначено только для однопроцессорных конфигураций.

⁷ Общее число центральных процессоров и процессоров ввода-вывода не может превышать шести из-за ограничений по числу входов в память.

⁸ Полос выделенный канал передачи данных и медленный канал, подсоединенный непосредственно к одному из процессоров. Общее число центральных процессоров и процессоров ввода-вывода не должно превышать восьми.

⁹ Любая комбинация центральных процессоров и каналов прямого доступа к памяти общим числом не больше четырех.

¹⁰ Общее число центральных процессоров и процессоров ввода-вывода не должно превышать восьми из-за ограничений по числу входов в память.

¹³ ЭВМ В5700 является развитием ЭВМ В5500 с разделенной дисковой памятью.

¹⁴ Общее число центральных процессоров и процессоров ввода-вывода не должно превышать двенадцати из-за ограничений по числу входов в память.

¹⁵ Плавающие каналы переключения данных.

¹⁶ «Конструкция контролеров ввода-вывода не допускает подconnections их более чем к двум процессорам. Кроме того, из-за особенностей прямого управления между процессорами система не может содержать более двух процессоров. Таким образом, аппаратная структура системы не может быть расширена за пределы двухпроцессорной конфигурации, хотя принципы, на которых основана система, не налагают таких ограничений» (Mil-let).

¹⁷ Основная память разделена на две части: малое запоминающее устройство на магнитных сердечниках (МЗУ), которое является исполнительным, и большое запоминающее устройство на магнитных сердечниках (БЗУ), которое используется в качестве массовой памяти для очень быстрых операций при работе в реальном времени, например в режиме диалога.

¹⁸ В настоящее время имеется только четыре.

¹⁹ Может иметь также расширенную воспроизводящую память на магнитных сердечниках емкостью $16 \times 65K \times 36$ бит.

²⁰ Архитектура системы RERE предназначена для усиления универсальных ЭВМ, таких, как CDC 7600.

²¹ Модель MSI системы RERE рассчитана на 288 элементарных процессоров. Архитектура системы не ограничивает число элементарных процессоров.

²² В системе RERE подconnection к ведущей ЭВМ осуществляется через три устройства управления. Ни одно периферийное устройство не соединено с ведущей ЭВМ непосредственно.

²³ Память программ.

²⁴ Память данных.

²⁵ Один канал емкостью 1 Гбит и один канал емкостью 1 Мбит.

²⁶ Для одного квадранта, имеющегося в настоящее время.

²⁷ В настоящее время характеристики не установлены.

²⁸ Общее число центральных процессоров и процессоров ввода-вывода не должно превышать восьми.

²⁹ Один быстрый канал и один медленный канал.

³⁰ Каждый центральный процессор может иметь 1, 2, 3 или 4 арифметических устройств, которыми управляет один процессорный блок.

³¹ Периферийные процессоры являются виртуальными процессорами (операционная система осуществляет исполнительные операции полностью в периферийном процессоре).

³² Определяется не системной конструкцией, а возможным числом входов в память.

³³ Центральная память.

³⁴ Расширение центральной памяти (может быть осуществлено немедленно).

Некоторые сокращенные обозначения

ЦП	— центральный процессор;
ПВВ	— процессор ввода-вывода;
мпк	— мультиплексный канал;
слк	— секторный канал;
бл-мпк	— блок-мультиплексный канал;
нп	— не производится;
ЭП	— элементарный процессор;
БУАУ	— блок управления арифметическими устройствами;
БУКУ	— блок управления корреляционными устройствами;
БУАВУ	— блок управления ассоциативными выходными устройствами.

дублирование некоторого минимального комплекта аппаратуры и использование дополнительного оборудования в качестве автономной системы или просто как резерва. Какова относительная эффективность такого подхода?

Рассмотрим некоторое практическое задание, для выполнения которого требуется в минимальной конфигурации системы 1 процессор, 6 модулей памяти и 2 подсистемы ввода-вывода. Ниже приводятся иллюстративные значения характеристик готовности и аппаратных требований для различных конфигураций, которые соответствуют этим характеристикам.

	Готовность единицы оборудования	Требуется для одной системы	Требуется для двух автономных или дублированных систем	Требуется для мультипроцессорной системы	Требуется для трех автономных систем
Процессор	0,98	1	2	2	3
Память	0,97	6	12	8	18
Ввод-вывод	0,9	2	4	3	6
Готовность системы выполнять основное задание		0,66	0,88	0,951	0,95

Здесь общая готовность одиночной системы равняется 0,66. Добавление идентичной системы в качестве резерва приведет к повышению готовности системы до 0,88. Мультипроцессорная система представленной в таблице конфигурации обладает готовностью $\approx 0,95$. Чтобы получить такую же готовность с помощью независимых систем, понадобится еще одна полная резервная система. К этим соображениям, сосредоточенным главным образом на аппаратуре, необходимо добавить два замечания. Первое состоит в том, что значение готовности мультипроцессорной системы не было скорректировано на влияние надежности ее системы взаимных соединений, которая, несомненно, более сложна и менее надежна, чем в однопроцессорной системе. Второе касается стоимости дополнительной вычислительной системы по сравнению со стоимостью системы взаимных соединений (последняя, как уже отмечалось, может быть значительной).

Системное программное обеспечение, требуемое для управления таким надежным комплексом аппаратуры, будет сложнее, дороже и менее надежно, чем программное обеспечение для однопроцессорной системы. Для таких систем трудно найти единые общие критерии оценки. Однако опыт показал, что влияние системного программного обеспечения настолько велико, что может свести на нет то повышение надежности, о котором говорилось выше.

По-видимому, этот последний фактор совместно с возможностью катастрофического отказа интегрированной памяти привел некоторых исследователей [41] к заключению, что несколько соединенных между собой однопроцессорных систем с независимыми запоминающими устройствами обладают более высокой надежностью. Возможно, это соответствует действительности, если рассматривается только надежность. Если же учесть другие факторы, такие, как возможность доступа всех процессоров ко всей памяти, т. е. ко всем хранящимся в ней программам и данным, и способность к быстрой реконфигурации при работе в реальном времени, то оценка готовности системы выполнять минимальный набор наиболее важных задач по-прежнему указывает на некоторое преимущество мультипроцессорной конфигурации с точки зрения соотношения эффективности и стоимости.

ГИБКОСТЬ

Когда говорят о гибкости, первое определение, которое приходит в голову,— это способность системы перестраивать свою конфигурацию. Нельзя забывать, однако, что в действительности гибкость — это функциональное свойство системы, обеспечивающее высокую готовность и надежность системы при минимальном ее составе. При наличии соответствующего программного обеспечения система в любой момент времени обладает способностью, например, выполнять заданные работы в условиях необычно большой нагрузки.

Работа системы значительно улучшается, если ее гибкость обеспечивает использование всех системных ресурсов каждым заданием. Особую важность имеет возможность доступа нескольких программ к большим базам данных.

ЭФФЕКТИВНОСТЬ

Мультипроцессорная система может использоваться с учетом только ее эффективности и высокой производительности лишь по одной причине — если ни одна однопроцессорная система не в состоянии справиться с какой-то заданной большой задачей в течение заданного времени. Все другие требования к производительности можно удовлетворить при помощи комбинации однопроцессорных систем.

К большому сожалению, а может быть, и к удивлению разработчиков мультипроцессорных систем, следует констатировать, что введение дополнительных процессоров вместе с соответствующим увеличением памяти и мощности устройств ввода-вывода не дает линейного возрастания эффективности¹. Принято считать, что увеличение эффективности при введении в большую систему второго процессора

¹ Эти данные не относятся к матричным или векторным системам.

составляет 60—80%¹. При добавлении третьего процессора это увеличение составляет только 30—50%. В одном случае автор лично наблюдал повышение эффективности при введении второго процессора на 80%. Однако, когда к этой же системе был добавлен третий процессор, общая эффективность (производительность) составила всего лишь 210% от производительности однопроцессорной конфигурации, состоящей из тех же аппаратных элементов².

Очевидно, такие результаты объясняются влиянием программного обеспечения системы, ибо операционная система не может использовать эффективно имеющиеся функциональные блоки. Обычно один, а часто и два процессора простаивают в бездействии, ожидая назначения задания.

Интересно, что во время моделирования мультипроцессорной системы для проекта Аполлон эффективность системы оказалась выше при «большом числе медленных процессоров», чем при «небольшом числе быстрых процессоров» [30]. Так как основными функциями операционной системы, которые требуют наибольших затрат времени и которые сильнее всего влияют на эффективность использования процессоров, являются планирование работы и диспетчеризация,

¹ Во время одного из специальных 10-часовых прогонов эталонного потока заданий через систему В6700 наблюдалось возрастание эффективности на 90%.

² Из работы [31]: фирма UNIVAC предложила формулу для оценки повышения эффективности в результате добавления процессоров к системе и опубликовала числовые данные для системы 1108. Эта формула имеет вид

$$N = \frac{P \times 10^6}{C + Q + D + E},$$

где N — скорость выполнения команд, P — число процессоров, C — длительность цикла памяти, Q — задержка из-за ожидания в очередях к модулям памяти, D — аппаратная задержка (в мультимодульных адаптерах и т. д.), E — время, добавляемое при использовании длинных команд.

Для одного процессора

$$N = \frac{1 \times 10^6}{0,75} = 1,33 \times 10^6.$$

С длинными командами

$$N = \frac{1 \times 10^6}{0,75 + 0,30} = 0,95 \times 10^6.$$

Для двух процессоров

$$N = \frac{2 \times 10^6}{0,75 + 0,05 + 0,125 + 0,30} = 1,63 \times 10^6.$$

Таким образом, выигрыш при добавлении второго процессора составляет:

$$\frac{1,63 - 0,95}{0,95} = 0,71.$$

полученный результат может натолкнуть на мысль, что эти операции следует производить реже. Однако при этом возникает дополнительное требование, чтобы большие работы разбивались на короткие независимые участки, а эта проблема еще не решена ни в одной из известных операционных систем.

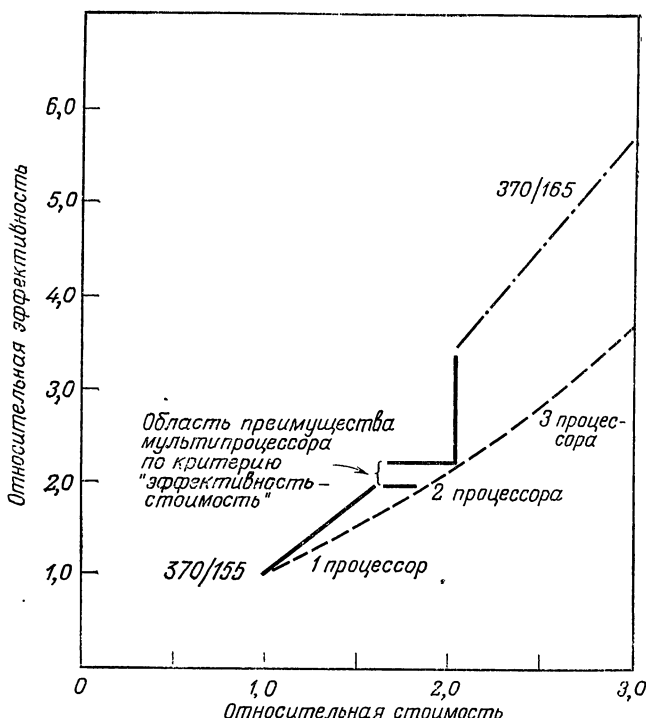


Рис. 4.1. Сравнение вычислительных систем по соотношению «стоимость — эффективность». В качестве мультипроцессора рассматривается система IBM 370/155 с дополнительной стоимостью мультипроцессорного оборудования.

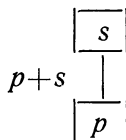
Существует особая область, в которой могут быть использованы преимущества мультипроцессорных систем по эффективности, даже при наличии вышеупомянутых ограничений. Это область разрыва кривых зависимости эффективности от стоимости для различных систем, выпускаемых одним и тем же производителем. Рассмотрим в качестве примера семейство систем IBM S/370. Если принять базовую конфигурацию модели 155 за начало отсчета и считать ее относительную стоимость и относительную эффективность равными единице, то различные конфигурации модели 155 дают зависимость «стоимость — эффективность», показанную на рис. 4.1 сплошной ли-

нией. Такую же зависимость можно построить для модели 165. Базовая конфигурация модели 155 по-прежнему остается основой для сравнения. Предположим, что имеется возможность создать мультипроцессорную систему с использованием составных элементов модели 155. Такая система будет стоить дороже, чем простое объединение моделей 155, а ее эффективность не будет возрастать линейно. Однако мультипроцессорная система на базе модели 155 все же может быть использована для заполнения разрыва между кривыми, причем она будет более выгодна, чем переход к модели 165, до тех пор, пока коэффициент загрузки не превысит значения 2,2. Этот пример весьма гипотетичен, но он хорошо иллюстрирует утверждение, что часто существуют области, в которых мультипроцессорная система представляется наиболее выгодным вариантом по показателю «стоимость — эффективность». Само собой разумеется, что, рассматривая эту сравнительно небольшую область, где выгодно применение мультипроцессорных систем, необходимо также учитывать и другие преимущества этих систем.

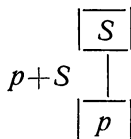
СРАВНИТЕЛЬНАЯ ОЦЕНКА ВИТТА

Б. Витт — один из разработчиков системы IBM S/360, модель 65, в своей статье [39], посвященной этой системе и мультипроцессорам вообще, дал сравнительную оценку способности нескольких системных структур выполнять некоторую заданную совокупность конструктивных требований. Он сравнил пять конфигураций:

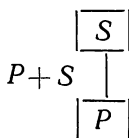
Стандартный процессор со стандартной памятью.



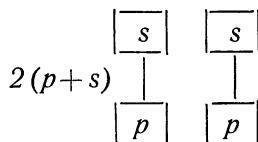
Стандартный процессор с удвоенной стандартной памятью.



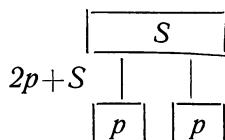
Процессор с удвоенной мощностью и с удвоенной стандартной памятью.



Две автономных конфигурации, каждая из стандартного процессора со стандартной памятью.



Мультипроцессор, состоящий из двух стандартных процессоров, разделяющих удвоенную память.



На этих схемах буквами p и P обозначены два различных типа процессоров. Первый — процессор, обладающий производительностью стандартной ЭВМ, а второй — теоретический процессор, ровно вдвое производительнее стандартного. Буквами s и S обозначены два варианта памяти и каналов, причем второй вариант обладает вдвое большей емкостью, чем первый.

Результаты сравнительной оценки Витта приведены в таблице, помещенной ниже. В таблице применяются следующие оценки:

0 — стандарт, относительно которого оцениваются все конфигурации;

1 — лучше;

2 — самый лучший вариант.

Если оценки двух систем совпадают, то системы считаются одинаково эффективными.

	$2p + S$	$p + S$	$2(p + s)$	$P + S$
Защита наиболее важной информации от потерь	0	0	2 (если системы дублированы)	0
Частота перезапусков системы и заданий	0	2	1	Неизвестно
Критическая готовность системы	2	0	1	0
Необычные задания и пиковые нагрузки	2	1	0	2
Системные операции	2	2	0	2
Эффективность	1 ¹	0 ¹	1 ¹	2

¹ Результаты измерений; остальные данные — теоретические оценки.

Все критерии понятны сами по себе, за исключением критерия «системные операции». Различия в оценках по этому критерию определяются проблемой отсутствия единой базы данных. Следует обратить внимание на то, что в конфигурации 2 ($p+s$) обе памяти совершенно автономны.

Читатель не должен приписывать оценкам, помещенным в таблице, смысл конкретных чисел. Хотя мультипроцессорная конфигурация обладает некоторыми преимуществами перед другими конфигурациями в некоторых областях, тем не менее ее пиковая производительность никогда не достигает удвоенной производительности однопроцессорной системы. Более реалистические оценки этого фактора приведены в других местах книги. Здесь следует обратить внимание на то, что мультипроцессорная система равноценна другим конфигурациям или превосходит их по всем критериям, за исключением критерия относительной частоты перезапусков системы и заданий. Этот недостаток мультипроцессорной конфигурации можно объяснить большей сложностью ее операционной системы и других управляющих механизмов.

Для повышения эффективности мультипроцессорной системы можно применить те же методы и приемы, что и для однопроцессорных систем:

Применение автономных групп управляющих и операционных регистров для облегчения перехода от одной программы к другой, пока каждый процессор работает в мультипрограммном режиме. Применение банков памяти для осуществления доступа с перекрытием (см. выше).

Использование микропрограммирования и постоянного запоминающего устройства для подгонки системы команд к определенной специфической области применения.

Введение специальных команд для повторяющихся операций, таких, как вычислительные циклы.

Использование иерархической структуры построения памяти с применением быстродействующей буферной памяти на твердом теле в качестве быстродействующего упреждающего блочного буфера. (Применение такой памяти может создать новые трудности и проблемы, связанные с сохранением целостности содержимого этой памяти, если хранящийся в ней массив данных используется более чем одним процессором.)

НЕДОСТАТКИ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Хотя некоторые недостатки мультипроцессорных систем уже упоминались, приведенный ниже список дает полный их перечень. Системное программное обеспечение очень сложно, требует больших затрат сил и средств при проектировании, производстве и эксплуатации.

Принципы, положенные в основу программного и аппаратного обеспечения, и их сложное взаимодействие делают контроль системных программ весьма затруднительным и требующим больших затрат времени.

Возрастание эффективности при введении дополнительных элементов или функциональных блоков связано нелинейной зависимостью с повышением стоимости системы. Для некоторых видов нагрузки, состоящей из большого числа мелких задач, однопроцессорные системы позволяют при тех же расходах получить больший прирост эффективности. Теоретический анализ подтверждает это заключение.

Возможные конфигурации сильно ограничены аппаратными возможностями функциональных блоков:

- наличием временного разделения общей шины;
- числом входов в память;
- техническими требованиями со стороны интерфейса;
- длиной кабелей между блоками (из-за дополнительных задержек передачи сигналов);
- наличием подходящего механизма коммутации.

Потенциальные преимущества аппаратного обеспечения и организации системы могут оказаться нереализованными из-за ограничений, налагаемых системным программным обеспечением.

НЕКОТОРЫЕ ДОПОЛНИТЕЛЬНЫЕ СООБРАЖЕНИЯ ОТНОСИТЕЛЬНО РАБОТЫ В РЕЖИМЕ СОВПАДЕНИЯ И ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ

ДРУГИЕ ТИПЫ КЛАССИФИКАЦИИ

Весьма общий принцип классификации структуры и способа функционирования систем, состоящих из большого числа функциональных блоков, был предложен Креншоу [16] в докладе на конференции НАТО. Хотя автор рассматривал главным образом космические и авиационные бортовые системы, его терминология оказалась применимой к любым системам. Он разделил все вычислительные системы на два класса — федеративные и интегрированные.

ФЕДЕРАТИВНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Федеративная система состоит из нескольких ЭВМ, каждая из которых предназначена для выполнения какой-либо частной задачи. Для нормальной работы ЭВМ связываются друг с другом при помощи своих каналов ввода-вывода. В системе может быть предусмотрена избыточность, как в случае дублированной федеративной системы, когда каждая ЭВМ дополнительно к своим нормальным функциям может выполнять функции резервирования в случае неисправности другой ЭВМ. Отдельные элементы федеративной сис-

темы могут обладать всеми характеристиками интегрированной системы. Федеративные системы могут быть стандартизованными, когда все процессоры одинаковы, или могут быть специализированными, когда каждый процессор предназначен для своей особой цели. Во втором случае достигается большая вычислительная эффективность. В первом случае, однако, имеются преимущества в части обучения персонала, использования контрольно-измерительного оборудования, эксплуатации и снабжения запасными частями.

ИНТЕГРИРОВАННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Интегрированной называется система, которая выполняет не связанные между собой задачи в мультипрограммном режиме работы. Система может содержать один центральный процессор (симплексная система) или же два центральных процессора, разделяющих общую основную память и работающих в мультипроцессорном режиме. Федеративные системы отличаются от интегрированных числом очередей заданий. В федеративной системе предусмотрены отдельные очереди заданий и исполнительные программы для каждого процессора, в то время как в интегрированной системе имеется только одна очередь заданий и одна исполнительная программа, хранящаяся в общей памяти.

Креншоу также представил свои оценки сравнительных преимуществ этих двух классов систем. За исключением замечаний, касающихся боевых повреждений, эти оценки также применимы к любым вычислительным системам.

ПРЕИМУЩЕСТВА ФЕДЕРАТИВНЫХ СИСТЕМ:

Менее сложное программное обеспечение, включающее операционную программу и исполнительную программу.

Простота реконфигурации.

Меньшая уязвимость при боевых действиях.

Отсутствие помех при одновременном обращении к памяти.

Более высокая надежность при выполнении критических функций.

ПРЕИМУЩЕСТВА ИНТЕГРИРОВАННЫХ СИСТЕМ:

Более эффективное распределение нагрузки.

Большая гибкость.

Более эффективная связь между процессорами.

Меньшая избыточность памяти.

Более высокая общая производительность.

Более высокая системная надежность.

Очевидно, мультипроцессорные системы и системы с параллельной обработкой, соответствующие данному в этой монографии определению, представляют собой интегрированные вычислительные системы, а все другие многомашинные конфигурации принадлежат к классу федеративных систем. Название каждого класса систем подчеркивает замечания, сделанные нами относительно характеристик аппаратного и программного обеспечения, по которым они отличаются друг от друга.

Шор [37] отказался от попытки разъяснить такие термины, как параллельный, ортогональный, ассоциативный, матрица, вектор и т. п. Он изложил свои соображения относительно рассматриваемой проблемы путем введения шести типов ЭВМ:

Машина I. Обычный последовательный процессор (последовательная обработка слов, параллельная обработка разрядов).

Машина II. Параллельный процессор, который одновременно работает над одним и тем же разрядом во всех информационных словах. Пример — система STARAN фирмы Goodyear (параллельная обработка слов, последовательная обработка разрядов).

Машина III. Работает в режиме машины I или машины II по желанию.

Машина IV. Множественные запоминающие устройства данных и процессоры, одно устройство управления. Пример — система PERE.

Машина V. Машина IV с линейным матричным соединением процессоров и запоминающих устройств данных. Пример — система ILLIAC IV.

Машина VI. Несколько ассоциативных запоминающих устройств и процессоров, использующих матрицы памяти с встроенной логикой (LIMA).

Шор заканчивает свою короткую заметку о параллельной обработке следующими замечаниями:

1. Большинство современных (проектируемых или уже созданных) высокопроизводительных организаций вычислительных систем целесообразно рассматривать как реализации различных способов увеличения отношения объема аппаратуры процессора к объему аппаратуры памяти, т. е. способов повышения «коэффициента процессирования».

2. При сравнении различных способов повышения коэффициента процессирования полезным критерием может быть величина рабочего цикла аппаратуры процессора, т. е. относительное время, которое расходуется на арифметические и логические действия, имеющие смысл.

3. Чем выше коэффициент процессирования, тем для меньшего числа задач и для более специальных приложений можно использовать этот показатель с высоким значением рабочего цикла.

4. При выборе между аппаратной и программной реализацией последовательностей параллельных команд важным критерием является относительное число осмысленных арифметических и логических операций, обусловленных этими последовательностями.

5. Если невозможно найти какое-либо другое решение, кроме аппаратной реализации «параллельных» процессоров с низким значением рабочего цикла, то это свидетельствует о плохом выборе алгоритмов и о возможности получить ту же производительность при меньшем объеме аппаратуры при помощи машины I и соответствующего программного обеспечения.

Трудно сказать, каким образом обычные мультипроцессорные системы, рассмотренные нами, можно было бы связать с классификацией Шора, но его замечания относительно влияния различных структур на эффективность обработки не лишены интереса.

СТЕПЕНЬ ПАРАЛЛЕЛЬНОСТИ РАБОТЫ

Профессор Це Юн Фань во вступительном слове на специальной сессии, посвященной системам с параллельной обработкой, сделал обзор основных параллельных систем, обращая особое внимание на степень параллельности их работы, а не на их общую производи-

тельность (которая, конечно, зависит и от частоты синхронизации) [20]. Каждый элементарный процессор может выполнять последовательные арифметико-логические операции (т. е. обрабатывать разрядный «срез» всех слов, подлежащих обработке) или полностью параллельные арифметико-логические операции. В первом случае требуется только один запоминающий регистр (для хранения одного слова или разрядного среза нескольких слов); во втором случае полностью параллельный процессор требует применения нескольких регистров. Возможности режима максимального параллелизма можно представить графически, сравнивая размер основного машинного слова с размером разрядного среза, обрабатываемого одновременно (рис. 4.2).

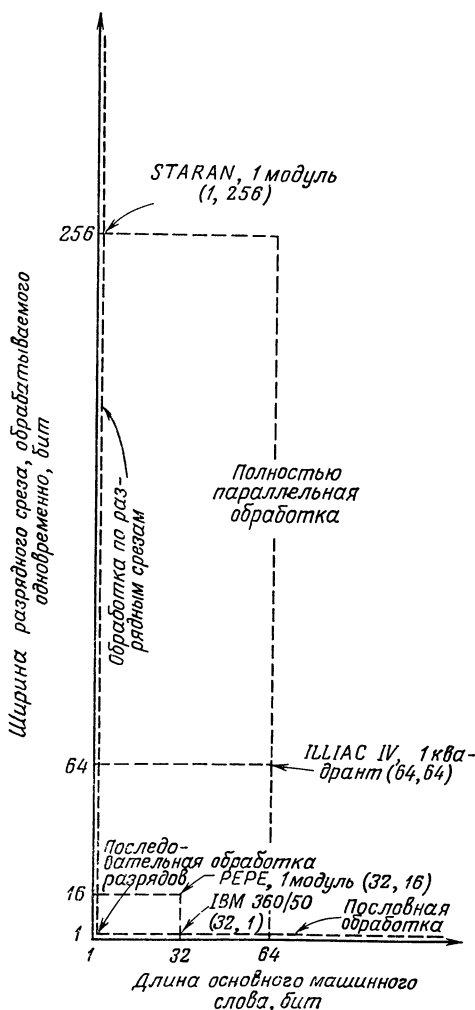


Рис. 4.2. Максимально возможная степень параллельности обработки для различных вычислительных систем [20].

БУДУЩЕЕ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Нет сомнений, что применение мультипроцессорных систем будет продолжаться в тех областях, где главное значение имеют высокая готовность и высокая надежность. Разработка небольших и легких блоков позволяет надеяться, что такие системы будут использоваться в качестве бортовых ЭВМ для боевых самолетов. Вспомним,

что всего лишь несколько лет тому назад технология в этой области позволила создать однопроцессорные бортовые системы.

Можно было бы ожидать, что основной спрос на мультипроцессоры будет со стороны таких применений, как управление технологическими процессами, в которых готовность и надежность имеют решающее значение. Однако ограниченный рынок сбыта таких систем узкого назначения не стимулировал разработку стандартных систем для приложений такого рода. Экономическими стимулами во всех областях применения всегда были стоимость и эффективность, и мультипроцессоры в этом отношении обладают большими преимуществами. Основная характеристика мультипроцессора, которая соответствует указанным требованиям, — это его модульность, обеспечивающая лучшее согласование ресурсов с поставленными заданиями.

В мультипроцессорных системах можно добиться лучшего использования ресурсов, оплачиваемых пользователем, ибо система может быть точно «подогнана» к его специфическим требованиям. Гибкий характер модульности всей аппаратуры, выполняющей основные функции (обработка, хранение, ввод-вывод), предоставляет пользователю возможность оптимального выбора системы с точки зрения соотношения эффективности и стоимости.

Расходы на проектирование, разработку, производство и эксплуатацию могут быть значительно снижены, так как небольшое число моделей каждого функционального блока обеспечивает широкий диапазон рабочих характеристик системы.

Для специальных видов заданий, характеризующихся особенно большим объемом работ, мультипроцессорные системы представляются единственным средством сосредоточения достаточной вычислительной мощности в заданный период времени.

Так как стоимость аппаратуры со временем падает, а стоимость программного обеспечения возрастает, одним из немногих перспективных подходов является более высокая степень стандартизации в очень широком диапазоне. Мультипроцессорные и параллельные системы позволяют реализовать этот подход. Это справедливо и в отношении отдельных систем, физически сосредоточенных в одном месте, и в отношении кооперированных сетей функциональных блоков, соединенных между собой линиями связи. Производители должны найти более удовлетворительные решения с точки зрения показателя «стоимость—эффективность».

Хотя мультипроцессорные системы при выполнении некоторых классов требований к работе, несомненно, обладают экономическими преимуществами, весьма возможно, что основное внимание в будущем будет обращено не на стоимость, а на другие показатели. Мультипроцессорные системы наверняка завоюют признание благодаря надежности и гибкости, а также возможности наращивания, которые обеспечиваются их системной организацией.

1. *Alexander M. T.*, Time Sharing Supervisor Programs, Class Notes, Computing Center, University of Michigan, Ann Arbor, Mich., May 1969, revised May 1970, 62 pp.
2. *Anderson J. P., Hoffman S. A. Shifman J. and Williams R. J.*, D825-A Multiple Computer System for Command and Control, Proceedings of the Fall Joint Computer Conference, 1962, v. 22, p. 86—96.
3. *Aschenbrenner R. A., Flynn M. J. and Robinson G. A.*, Intrinsic Multi-Processing, Proceedings of the Spring Joint Computer Conference, 1967, v. 30, p. 81—86.
4. *Avizienis A.*, Design of Fault-Tolerant Computers, Proceedings of the Fall Joint Computer Conference, 1967.
5. *Avizienis A., Mathur F. P. and Rennels D. A.*, Automatic Maintenance of Aerospace Computers and Spacecraft Information and Control Systems, AIAA Paper 69—966, AIAA Aerospace Computer Systems Conference, Los Angeles, Calif., September 1969.
6. *Baer J.-L.*, Large Scale Systems, ch. 5 in Cardenor A. F., Presser L. and Martin M. A., eds., Computer Science, New York, Wiley-Interscience, 1972.
7. *Baer J.-L. and Bovet D. P.*, Compilation of Arithmetic Expressions for Parallel Computations, Proceeding of the IFIP Congress, Booklet B. 4—10, 1968.
8. *Barnes G., Brown R., Kato M., Kuck D., Slotnick D. and Stokes R.*, The Illiac IV Computer, IEEE Transactions on Computers, 1968, v. C—17, No 8, p. 746—757.
9. *Baskin H. B., Borgenon B. R., Roberts R.*, PRIME — A Modular Architecture for Terminal-Oriented System, Proceedings of the Spring Joint Computer Conference, 1972, p. 431—437.
10. *Bell C. G. and Newell A.* Computer Structures: Readings and Examples, New York, McGraw-Hill, 1971.
11. *Burnett G. J. and Coffman E. G., Jr.*, A Study of Interleaved Memory Systems, Proceedings of the Spring Joint Computer Conference, 1970, v. 36, p. 367—374.
12. *Coffman E. G., Jr., Elphick M. J. and Soshani A.*, System Deadlocks, ACM Computing Survey, 1971, June, v. 3(2), p. 67—78.
13. *Cohen E.*, Symmetric Multi-Mini-Processors: A Better Way to Goé, Computing Decision, 1973, January, p. 16—20.
14. *COMPTRE Corp., Sayers A. P.*, ed., Operating Systems Survey, Princeton, Auerbach, 1971.
15. *Covo A. A.*, Analysis of Multiprocessor Control Organizations with Partial Program Memory Replication, в печати. Сокращенный вариант опубликован в Proceedings of the IEEE COMPCON 1972.
16. *Crenshaw J. H.*, Federated versus Integrated Computer Systems, in Leondes C. T., ed. Computers in the Guidance and Control Aerospace Vehicles, NATO, AGARDograph 158, February 1972. p. 9—22.

17. *Dancy Ch. A.*, III. System Reliability and Recovery, Department of Transportation Systems Center, Cambridge, Mass. 02142, Report DOT-TSC-FAA-71-16, June 15, 1971, AD 733760.
18. *Davis R. L. Tucker S. and Campbell C. M.*, A Building Block Approach to Multiprocessing, Proceedings of the Spring Joint Computer Conference, 1972, p. 685—703.
19. *Dijkstra E. W.*, Solution of a Problem in Concurrent Programming, Communications of the ACM, September 1965.
20. *Feng, Tse-yun*, An Overview of Parallel Processing Systems, 1972 WESCON Technical Papers, Session 1—Parallel Processing Systems, September 19—22, 1972.
21. *Filene R. J. and Weinstien W.*, The Regional Computer, Section 2.2 in STS Data Management System Design (Task 2), Charles Stark Draper Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., E-2529, June 1970.
22. *Flynn M. J.*, Some Computer Organizations and Their Effectiveness, IEEE Transactions on Computers, 1972, v. C—21(9), p. 948—960.
23. *Flores I.*, Computer Organization, Prentice-Hall, Englewood Cliffs, N. J., 1969; Организация вычислительных машин, изд-во «Мир», 1972.
24. *Gonzalez Jr., Mario J. and Ramamoorthy C. V.*, Program Suitability for Parallel Processing, IEEE Transaction on Computers, 1971, v. C—20(6), No 6, p. 647—654; AD 730 240.
25. *Hellerman H.*, Digital Computer System Principles, New York, McGraw-Hill, 1967, p. 228—229.
26. *Holt R. C.*, Comments on Prevention of System Deadlocks, Communications of the ACM, 1971, January, v. 14(1), No 1, p. 36—38.
27. *Hopkins A. L., Jr.*, A Fault-Tolerant Information Processing Concept for Space Vehicles, Charles Stark Draper Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., R-682, December 1970.
28. *Irwin J. D., Thorington J. M., Jr., and Blanco V. S.*, Multiprocessing Computer Systems, Project Themis, Digital Systems Laboratory, Auburn University, Auburn, Ala., Technical Report AU-T-12, July 1970, 97 pp.
29. *Jordan J. W.*, Task Scheduling For a Real Time Multiprocessor Electronics Research Center, Cambridge, Mass., NASA TN-D-5786.
30. *Mallach E. G.*, Analysis of a Multiprocessor Guidance Computer, Instrumentation Laboratories, MIT, T-515, June 1969, 113pp. (докторская диссертация).
31. *Miller J. S., Lickly D. J., Kosmala A. L. and Saponaro J. A.*, Multiprocessor Computer System Study—Final Report, Intermetrics, Inc., Cambridge, Mass., March 1970, 161 p.
32. *Murtha J. C.*, Highly Parallel Information Processing Systems, Advances in Computers, 1966, v. 7, p. 2—116.
33. *Quaise J. T., Gaulene P. and Dodge D.*, The External Access Network of a Modular Computer System, Proceedings of the Spring Joint Computer Conference, 1972, p. 783—790.
34. *Ramamoorthy C. V. and Gonzalez M. J.*, A Survey of Techniques for Recognizing Parallel Processable Streams in Computer Programs, Proceedings of the Fall Joint Computer Conference, 1969, v. 35.
35. *Ravi C. V.*, On the Bandwidth and Interference in Interleaved Memory Systems, IEEE Transactions on Computers, 1972, v. C—21, No 8, p. 899—901.
36. *Seligman L.*, LSI and Minicomputer System Architecture, Proceedings of the Spring Joint Computer Conference, 1972, p. 767—773.
37. *Shore J. E.*, Second Thoughts on Parallel Processing, Proceedings of the 1972 IEEE INTERCON, p. 358—359.
38. *Slotnick D. L., Borck W. C. and McReynolds R. C.*, The SOLOMON Computer, Proceedings of the Fall Joint Computer Conference, 1962.
39. *Witt B. J.*, M65: An Experiment in OS/360 Multiprocessing, presented at Information Systems Symposium, September 4—6, 1968, Washington, D. C., also presented 1968 ACM Conference.

40. Wang G. Y., An In-House Experimental Air Space Multiprocessor — EXAM, ERC Memo KC-T-031, September 20, 1967.
41. Weitzman C., Aerospace Computer Technology Catches Up With Ground Gear, Electronics, September 11, 1972, p. 112—119.
42. Wilson D. E., The PEPE Support Software System, COMPCON 72 IEEE Computer Society International Conference, p. 61—64.
43. Wood P. E. Jr., Interconnection of Processors and Memory in the Multiprocessor System, ERC Memo KC-T-041, February 5, 1968.
44. Wood P. E. Jr., Input/Output System for An Aerospace Multiprocessor, ERC Memo RC-T-062, May 19, 1969.
45. Lorin H., Parallelizm in Hardware and Software: Real and Apparent Concurrency, Prentice Hall, Inc., Englewood Cliffs, N. Y., 1971.
46. Liptay I. S., Structural Aspects of the System/360 Model 85, IBM Systems Journal, 1968, v. 7(1).
47. Curtin W. A., Multiple Computer Systems, Advances in Computers, 1963, v. 4, p. 245—303.

СЛОВАРЬ ТЕРМИНОВ

Этот словарь приведен с единственной целью — помочь читателю усвоить материал, изложенный в книге. Такое разъяснение необходимо, ибо многие из приводимых терминов могут иметь и другие значения, не имеющие отношения к нашему тексту. Поэтому авторы даже не пытались придать словарю сколько-нибудь универсальный характер.

Алгоритм (algorithm) — предписанная совокупность четко определенных правил или процедур для решения какой-либо задачи за конечное число шагов, например, полное описание арифметической процедуры определения значения функции $\sin x$ с заданной точностью или полное описание действий, выполняемых аппаратурой при реализации какой-либо элементарной арифметической или логической операции.

Арифметико-логическое устройство (arithmetic and logic unit) — функциональный блок вычислительной системы, содержащий схемы, которые выполняют арифметические и логические операции.

Архитектура системы (systems architecture) — характеристики конструкции вычислительной системы, определяющие аппаратную реализацию рабочих алгоритмов в каждом из функциональных блоков.

Асимметричная система (asymmetrical system) — многомашинальная система, в которой составляющие ее процессорные блоки сильно различаются по своим возможностям.

Ассоциативный (associative) — обычно относится к памяти, доступ к которой определяется указанием смысла ее содержимого, а не местоположения ячейки. Поэтому ячейки идентифицируются не именем или физическим положением, а содержимым. Применяется также для обозначения режима работы ЭВМ, при котором логические решения принимаются на основании анализа содержимого группы ячеек памяти. Соответствует памяти с адресацией по содержанию (ПАС).

Блокировка данных (data-interlock) защита набора данных посредством запрещения доступа к ним от всех пользователей, за исключением одного.

Блокировка памяти (memory lockout) — то же самое, что и *блокировка данных*.

Векторный процессор (vector processor) вычислительная система, в которой несколько *арифметико-логических устройств* организованы таким образом, что операции обработки компонент вектора или элементов данных могут выполняться одновременно (отличается от матричного процессора).

Вход (port) — точка соединения или интерфейс *функционального блока*.

Готовность (availability) — степень способности системы выполнять свои функции благодаря тому, что она находится в рабочем состоянии.

Двусторонняя шина (bi-directional bus) — шина, позволяющая передавать информационные сигналы в обоих направлениях.

Дуплексная система (duplexed system) — система с двумя отдельными комплектами средств, каждый из которых может принять на себя выполнение функций системы, в то время как второй комплект находится в резерве. Обычно эти комплекты идентичны.

Канал (channel) — элемент системы, выполняющий функции независимой ЭВМ по управлению всеми устройствами ввода-вывода (с перфокартами, магнитными лентами, дисками, барабанами). Канал обеспечивает одновременную работу устройств ввода-вывода и основной ЭВМ.

Канал ввода-вывода (input/output channel) — устройство, позволяющее осуществлять независимую связь между *центральной памятью* и устройствами ввода-вывода. Управляет всеми периферийными устройствами и выполняет все проверки при передаче информации.

Контроллер ввода-вывода (input/output controller) — устройство, которое выполняет те же функции, что и *канал ввода-вывода*, однако может обладать большими логическими возможностями для работы в составе мультипроцессорной системы.

Конфликты при обращении к памяти (memory access conflicts) — ухудшение работы системы в целом, вызванное одновременным обращением двух или большего числа *процессоров* к одному и тому же *модулю памяти* в пределах одного и того же цикла памяти. В этих случаях необходимо, чтобы один *процессор* «подождал» до следующего цикла. То же самое, что и межпроцессорные помехи.

Магистральная обработка данных (pipelining) — разделение арифметической или логической операции на отдельные шаги, выполняемые различными аппаратными модулями одного процессора.

Магистральная система (pipeline system) — система, обеспечивающая возможность параллельной обработки с использованием магистрального принципа, когда каждый блок, входящий в систему, выполняет одну и ту же операцию над элементами совокупности данных.

Матричный процессор (array processor) — вычислительная система с большим числом арифметико-логических устройств, называемых иногда элементарными процессорами, которые организованы так, что могут одновременно выполнять операции над элементами матрицы данных.

Машина фон Неймана (von Neumann machine) — термин, применяемый в данной книге к структуре системы из пяти самостоятельных функциональных блоков: арифметико-логического устройства, устройства управления, устройства ввода, устройства вывода и центральной памяти.

Межпроцессорные помехи (interprocessor interference) — снижение эффективного быстродействия процессора из-за задержек доступа к центральной памяти, вызванных тем, что память используется другими процессорами или устройствами ввода-вывода, входящими в состав системы.

Многошинная — многовходовая система (multiple-bus/multiport system) — мультипроцессорная система, в которой взаимные соединения выполняются с помощью индивидуальных шин, подключающих каждый процессор и контроллер ввода-вывода к отдельному входу центральной памяти.

Многомашинная система (multicomputer system) — система, содержащая больше одного процессора, но не удовлетворяющая всем признакам, характеризующим мультипроцессорную систему.

Модуль памяти (memory module) — физический блок или узел центральной памяти. Типичный модуль памяти может выполнять одновременно лишь одну операцию обращения (считывание или запись).

Мультипрограммирование (multiprogramming) — совпадающее выполнение двух или большего числа программ одним процессором. Отдельные независимые программы (или по крайней мере части каждой из программ) должны одновременно находиться в центральной памяти, а управление обработкой должно динамически переключаться с одной программы на другую.

Мультипроцессорная система (multiprocessor) — система, состоящая из двух или большего числа *процессорных блоков* с примерно равными возможностями, которые:

имеют доступ к общей *центральной памяти*;

имеют общий доступ по крайней мере к части устройств ввода-вывода;

управляются одной общей операционной системой, которая обеспечивает требуемое взаимодействие между процессорами и выполняемыми ими программами на уровнях задания, задачи, шага работы, *набора данных*, элемента данных и аппаратного *прерывания*.

Набор данных (data set) — основная совокупность данных, используемая при организации хранения и поиска информации с помощью операционной системы. Она представляет собой группу данных, сосредоточенных в одном или в нескольких заранее установленных местах и описываемых управляющей информацией, которая передается в систему. (Термин был впервые применен фирмой IBM. Вместо него часто используют слово *файл*.)

Надежность (reliability) — степень способности системы работать без отказов.

Неоднородная система (nonhomogeneous system) — *многомашинная* или *мультипроцессорная система*, состоящая из неодинаковых процессоров.

Нормализация (normalization) — приведение порядка (характеристики) и мантиссы числа с плавающей запятой к такому виду, чтобы мантисса находилась в заданном стандартном диапазоне. Эта операция обычно включает сдвиг мантиссы влево до исключения всех нулей в старших разрядах и соответствующее изменение порядка.

Объединение в блоки (blocking) — группирование нескольких логических записей и регистрация их в виде одной физической записи (одного блока) на магнитной ленте, диске или барабане для лучшего использования объема запоминающего устройства и повышения эффективной скорости ввода-вывода.

Одновременная работа (simultaneity) — способность ЭВМ выполнять последовательность операций ввода-вывода в *периферийных устройствах* параллельно с выполнением других операций в *центральной процессоре*.

Одновременный (simultaneous) — относится к случаю существования или возникновения двух или большего числа событий в один и тот же момент времени (отличается по смыслу от понятия *совпадающий*).

Одновременная работа устройств ввода-вывода (simultaneous input/output capabilities) — то же самое, что и *совмещение операций ввода-вывода*.

Однонаправленная шина (unidirectional bus) — соединительная шина, в которой управляющие схемы и усилители (если они предусмотрены) обеспечивают передачу сигналов только в одном направлении. Изменение направления потока информации невозможно даже при использовании режима разделения времени.

Однопроцессорная система (uniprocessor system) — цифровая вычислительная система, имеющая только один *центральный процессор*.

Организация системы (systems organization) — характеристики конструкции вычислительной системы, определяющие выбор типов и числа *функциональных блоков* и способа их взаимного соединения.

Память с чередованием адресов («interleaved» memory) — 1) память с такой организацией различных частей последовательности адресов памяти, что они чередуются с частями одной или большего числа других последовательностей адресов, причем каждая последовательность сохраняет свою идентичность; 2) память, организованная в отдельные блоки с независимыми *шинами*, чтобы в течение заданного периода времени последовательные обращения к памяти перекрывали друг друга.

Параллельное выполнение задач (parallel executive of tasks) — выполнение больше одной программы в один и тот же период времени параллельным способом, когда одновременно работают два или несколько *процессоров*.

Периферийные устройства (peripheral systems) — общий термин, обозначающий устройства, которые не входят в основную структуру ЭВМ (устройства чтения перфокарт, печатающие устройства, ленточные устройства).

Периферийные процессоры (peripheral processors) — встречаются в вычислительных системах CDC 6000 и CDC 7000.

Переключательная матрица (switch matrix) — совокупность вентилей и других логических схем, позволяющих осуществлять взаимосвязь *функциональных блоков* и других рабочих элементов системы.

Полиморфная система (Polymorphic System) — одна из ранних систем, разработанных фирмой Ramo-Wooldridge для ВВС США; система обладает многими свойствами современных *мультипроцессорных систем* с перекрестной связью.

Прерывания (interrupts) — остановки нормальной последовательности работы системы или программы, выполняемые таким образом, что эта последовательность позднее может быть восстановлена начиная с места прерывания. Прерывания инициируются сигналами следующих двух типов: 1) сигналами, генерируемыми самой вычислительной системой для синхронизации работы различных ее частей, и 2) сигналами, поступающими извне для синхронизации совместной работы вычислительной системы с внешними объектами, например с человеком-оператором или с каким-либо физическим процессом.

Проект МАС (Project MAC) — вычислительная система с удаленными терминалами, разработанная в Массачусетском технологическом институте. Фактически обозначение МАС относится к первоначальному проекту «Machine Aided Cognition».

Процессор (processor) — то же самое, что и процессорный блок.

Процессорный блок (processor unit) — функциональный блок ЭВМ, выполняющий запрограммированные операции. Состоит из арифметико-логического устройства и устройства управления.

Процессор ввода-вывода (input/output processor) — то же самое, что и контроллер ввода-вывода.

Прямой доступ к памяти (direct memory access) — способность системы передавать данные непосредственно от устройств ввода-вывода к центральной памяти без использования арифметико-логического устройства. В этом процессе может принимать участие весь канал или только какой-либо интерфейс с более ограниченными возможностями.

Распаковка блоков информации (deblocking) — расчленение сложной физической записи на составляющие ее логические записи.

Реконфигурация (reconfiguration) — изменения в организации системы, выполняемые вручную или автоматически.

Система с вспомогательным процессором (attached support processor, ASP) — многомашинная система, в которой одна машина работает главным образом как процессор ввода-вывода для другого процессора, выполняющего вычисления. (Термин был впервые использован фирмой IBM.)

Система с общей шиной (common-bus system) — система, организованная так, что все функциональные блоки подсоединены к одной общей соединительной шине. Термин применим к любой организации системы даже при наличии нескольких шин, если в одной шине недостаточно проводов для одновременной работы всех функциональных блоков.

Системное программное обеспечение (system software) — математические средства, предназначенные прежде всего для управления аппаратными средствами системы и обеспечения взаимодействия «человек—ЭВМ».

Система с косвенной связью (indirectly coupled system) — система, в которой в качестве общей памяти при передаче данных используется устройство, работающее в режиме on-line, такое как ЗУ на магнитной ленте, барабан или диск.

Система с перекрестной коммутацией (crossbar switch system) — мультипроцессорная система, в которой связь осуществляется при помощи прямоугольной решетки соединительных шин, которые могут контактировать в любой точке их пересечения. Перекрестный коммутатор является «неблокирующимся» в том смысле, что передача через него не может быть запрещена из-за отсутствия путей передачи (но он может быть заблокирован, если одно из соединяемых устройств уже занято).

Система с прямой связью (directly coupled system) — система, в которой в качестве общей памяти при передаче данных используется быстродействующая центральная память.

Система с спутниковыми ЭВМ (satellite computers) — многомашинная система, в которой основному процессору подчинены спутниковые ЭВМ.

Система управления воздушным движением (The Air Traffic Control System, ATC) — установлена и эксплуатируется Федеральным авиационным управлением США (Federal Aviation Authority, FAA).

Система, устойчивая к сбоям и отказам (fault-tolerant system) — система, организованная так, что использование избыточных функциональных блоков и системы связи позволяет исключить влияние ошибок, возникающих в любом элементе связанной системы. Часто она удовлетворяет всем условиям, характеризующим мультипроцессорную систему. Однако основная цель создания таких систем отличается от цели создания других мультипроцессорных систем.

Система с шиной, работающей в режиме разделения времени (time-shared-bus system) — мультипроцессорная система, в которой соединительные шины распределяются путем разделения времени.

Собственная память (private memory) — такая организация памяти, когда система соединений обеспечивает доступ к определенной части центральной памяти только какому-либо одному процессору.

Совпадение (concurrency) — событие, состоящее в том, что два или более явлений или процессов происходят в одном и том же определенном интервале времени. (Отличается по смыслу от понятий «последовательность», «поочередность», «одновременность».)

Совмещение операций ввода-вывода (overlapped input/output capabilities) — применение канала обмена данными, который допускает асинхронное выполнение в *центральной процессоре* операций ввода-вывода и рабочей программы.

Устройства ввода-вывода единичных записей (unit-record I/O) — устройства ввода-вывода типа устройства чтения перфокарт, карточного перфоратора, печатающего устройства.

Устройство управления (control unit) — аппаратный элемент ЭВМ, который управляет выполнением последовательности операций, дешифрует коды команд и вырабатывает нужные сигналы для посылки в цепи ЭВМ, реализующие эти команды.

Функциональные блоки (functional units) — основные операционные устройства ЭВМ, например *центральная память*, *арифметико-логическое устройство*, *устройство управления*, контроллер, или *процессор ввода-вывода*.

Центральная память (central memory) — *функциональный блок*, содержащий ячейки памяти с непосредственным доступом от *арифметико-логического устройства* и от *устройства управления*.

Центральный процессор (central processor unit) — обычно применяется к комплексу, состоящему из *арифметико-логического устройства*, *устройства управления* и *центральной памяти*.

Шина (bus) — один или несколько проводов, используемых для передачи информационных сигналов или энергии. Часто играет роль соединительного звена между несколькими пунктами.

ЭВМ с посимвольной обработкой данных (character-oriented machine) — ЭВМ, в которой нормальная длина единицы элемента информации, хранимой в памяти и обрабатываемой в *арифметико-логическом устройстве*, равна длине символа. Соответствует ЭВМ с побайтной обработкой.

ЭВМ с пословной обработкой (word-oriented machine) — ЭВМ, в которой нормальный формат единицы информации, хранимой в памяти или обрабатываемой в *арифметико-логическом устройстве*, равен полному слову.

Элементарный процессор (processing element) — обычное название *арифметико-логического устройства* с ограниченными возможностями, используемого в *матричных* и *векторных процессорах*.

BMEWS — Система раннего обнаружения баллистических ракет ВВС США.

SAGE — наземная полуавтоматическая система управления средствами ПВО. Первая из больших систем противовоздушной обороны ВВС США, широко использующая цифровые ЭВМ.

ПРИЛОЖЕНИЕ I

СИСТЕМА РЕРЕ УПРАВЛЕНИЯ ПЕРСПЕКТИВНЫХ РАЗРАБОТОК ПО ПРОТИВОРАКЕТНОЙ ОБОРОНЕ АРМИИ США¹

ОРГАНИЗАЦИЯ СИСТЕМЫ

Система РЕРЕ (Parallel Element Processing Ensemble — ансамбль параллельных элементарных процессоров) представляет собой быстродействующую вычислительную систему, в архитектуре которой использованы оригинальные идеи, направленные на усиление возможностей универсальных ЭВМ с последовательной обработкой в применении к обработке радиолокационных данных системы противоракетной обороны. Система является программно управляемым комплексом, использующим процедурно-ориентированные языки и работающим под управлением ведущей универсальной ЭВМ. Эффективность системы обусловлена применением методов параллельной и ассоциативной обработки, реализованных при помощи гибкой структуры, которая состоит из неограниченного числа элементарных процессоров, контролируемых общими глобальными блоками управления.

Связь между системой РЕРЕ и системой противоракетной обороны при обработке радиолокационных данных показана на рис. П. I. 1. Система РЕРЕ состоит из следующих функциональных подсистем:

1. Блок управления арифметическими устройствами (БУАУ), один на систему.
2. Блок управления корреляционными устройствами (БУКУ), один на систему.
3. Блок управления ассоциативными выходными устройствами (БУАВУ), один на систему.
4. Элементарные процессоры (количество не ограничено); в состав каждого из них входят:

¹ В работе по программе РЕРЕ принимали участие следующие организации: фирма Bell Telephone Laboratories (разработала основные принципы системы и модель для проверки осуществимости системы), фирма System Development Corporation (главный подрядчик по программному обеспечению и конструкции системы на этапах проектирования и практической реализации), фирма Honeywell (субподрядчик по аппаратной части на этапе проектирования), фирма Burroughs (субподрядчик по аппаратной части на этапе практической реализации).

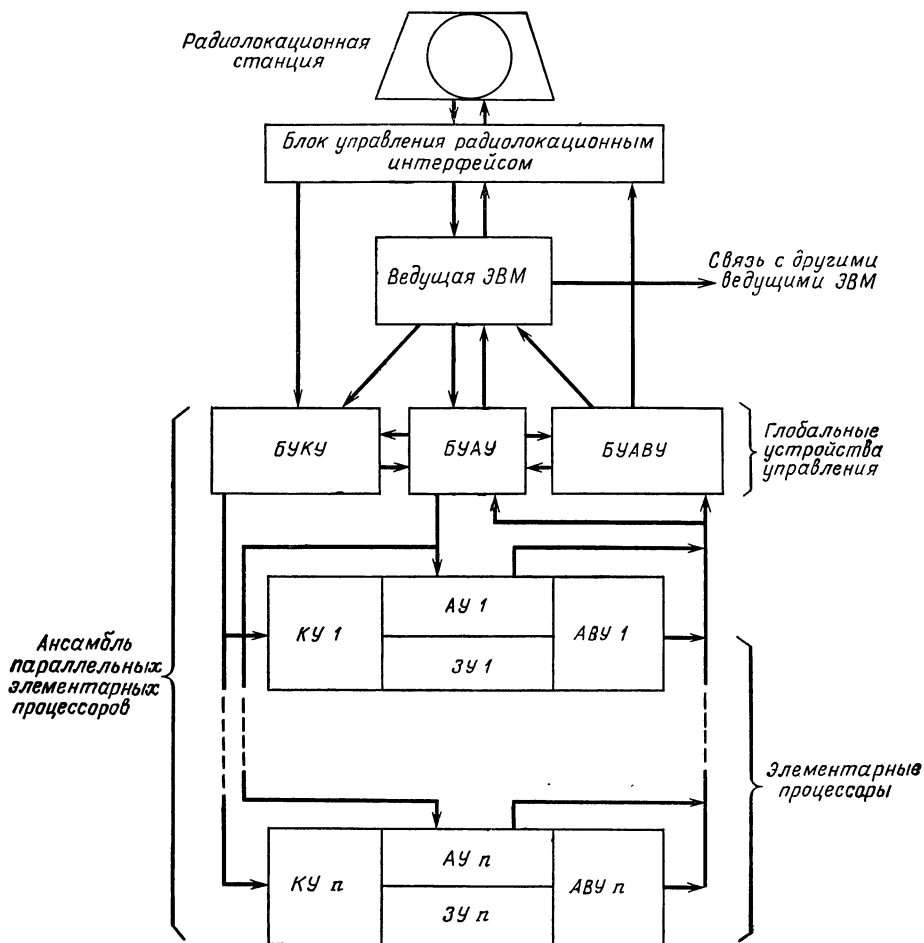


Рис. П.1.1. Организация системы РЕРЕ и ее связь с системой противоракетной обороны.

БУКУ — блок управления корреляционными устройствами, БУАУ — блок управления арифметическими устройствами, БУАВУ — блок управления ассоциативными выходными устройствами, КУ + КУ_n — корреляционные устройства, АУ₁ + АУ_n — арифметические устройства, ЗУ₁ + ЗУ_n — запоминающие устройства, АВУ₁ + АВУ_n — ассоциативные выходные устройства.

арифметическое устройство;
корреляционное устройство;
ассоциативное выходное устройство;
запоминающее устройство.

Кроме того, имеются распределительные подсистемы питания и сигналов, предназначенные для преобразования и распределения питающих напряжений, а также управляющих и информационных сигналов, циркулирующих между различными функциональными подсистемами.

Элементарные процессоры являются основными вычислительными элементами системы РЕРЕ. Универсальный ведущий процессор передает каждому из них определенную часть вычислительной нагрузки по обработке данных системы противоракетной обороны. Процесс распределения нагрузки, передаваемой в элементарные процессоры, зависит от присущей задаче возможности параллельного решения и способности системы РЕРЕ выполнять такую работу более эффективно, чем при использовании только ведущей ЭВМ. Каждый элементарный процессор «обслуживает» один определенный объект, наблюдаемый радиолокационной системой. Он организует в своей памяти файл данных, относящихся к определенным объектам, и использует свои арифметические возможности для непрерывного обновления этих файлов.

Число элементарных процессоров в системе РЕРЕ можно увеличивать или уменьшать в зависимости от конкретного приложения. Такая изменчивость структуры не оказывает влияния на эффективность всей системы в целом при условии, что всегда имеется достаточное число элементарных процессоров, чтобы справиться с вычислительной нагрузкой. Работа всех элементарных процессоров контролируется глобальными блоками управления (блоком управления арифметическими устройствами, блоком управления корреляционными устройствами и блоком управления ассоциативными выходными устройствами).

Такие арифметические процедуры, как коррекция траектории, прогнозирование траектории, идентификация траектории и наведение ракеты-перехватчика, выполняются параллельно несколькими арифметическими устройствами. Все эти устройства или выбранная их группа одновременно выполняют команды, поступающие от БУАУ, в зависимости от состояний триггеров активности элементарных процессоров.

Ввод новой информации в элементарный процессор осуществляется корреляционным устройством под управлением БУКУ. Процедура, выполняемая корреляционным устройством, состоит в сравнении вновь поступившей информации о местоположении объекта (полученной из отраженных радиолокационных импульсов) с предсказанной информацией о будущем его местоположении, выработанной арифметическим устройством элементарного процессора и переданной соответствующему корреляционному устройству. Из БУКУ в каждый текущий момент времени передается всем корреляционным устройствам одновременно информация об одном объекте, и все корреляционные устройства (или выбранная группа)

сравнивают хранящиеся данные с вновь поступившими. Информация об объекте вводится в то корреляционное устройство (или устройства), где имеет место корреляция между этими данными, или в первый свободный элементарный процессор, если корреляция не обнаружена ни в одном корреляционном устройстве.

Выходная информация, передаваемая обратно радиолокационной системе, обрабатывается ассоциативным выходным устройством под управлением БУАВУ. Распределение импульсов для радиолокационной системы состоит в упорядочении запросов, вырабатываемых элементарными процессорами в результате обновления содержимого файлов информации о радиолокационных объектах, вычисленной арифметическими устройствами и записанной в соответствующих запоминающих устройствах для выборки ее ассоциативными выходными устройствами. Упорядоченным поиском запросов на импульсы от ассоциативных выходных устройств управляет БУАВУ для одного объекта в каждый момент времени с использованием метода поиска максимально-минимального элемента в массиве ассоциативно адресуемых данных.

Арифметические, корреляционные и ассоциативные выходные устройства и блоки управления ими могут работать независимо в режиме совпадения. Запоминающие устройства программ, имеющиеся в каждом блоке управления, позволяют всему комплексу системы РЕРЕ работать в режиме совпадения с ведущей ЭВМ, причем требуется лишь самый минимальный контроль со стороны ведущей ЭВМ для целей планирования.

Программы системы РЕРЕ загружаются от ведущей ЭВМ в БУАУ, а отсюда поступают в память программ или данных каждого блока управления. Программы и значения большинства параметров загружаются во время запуска системы. После этого от ведущей ЭВМ время от времени поступают только изменения параметров.

Передача радиолокационной информации к (от) системы РЕРЕ производится при помощи двух соединительных устройств. БУКУ и БУАВУ передают данные к (от) ведущей ЭВМ или к (от) блока управления радиолокационным интерфейсом (БУРИ). На начальном этапе создания системы БУКУ и БУАВУ будут подсоединены к ведущей ЭВМ CDC 7600, которая будет имитировать функции БУРИ. В дальнейшем предполагается подсоединить БУКУ и БУАВУ к ведущей ЭВМ и к БУРИ. Радиолокационные данные и приказы будут передаваться через интерфейс «РЕРЕ — радиолокационная система», а сетевая информация и другие данные о коммутации и о состоянии системы — через интерфейс «Ведущая ЭВМ—РЕРЕ».

В последующих параграфах подробно описаны функции БУАУ, БУКУ, БУАВУ, арифметического устройства, корреляционного устройства, ассоциативного выходного устройства и запоминающего устройства.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

БЛОК УПРАВЛЕНИЯ АРИФМЕТИЧЕСКИМИ УСТРОЙСТВАМИ

БУАУ выполняет следующие функции:

1. Обеспечивает хранение программ, общей базы данных и данных, требуемых для работы БУАУ и всех арифметических устройств.

2. Декодирует команды программы параллельной работы арифметических устройств, получаемые из памяти программ, и посылает результирующие последовательности сигналов микропрограммного управления ко всем арифметическим устройствам.

3. Обеспечивает доступ системы РЕРЕ к ведущей ЭВМ для передачи обобщенной информации о состоянии системы, которая необходима для правильного планирования будущих работ для самой ведущей ЭВМ и для системы РЕРЕ под управлением ведущей ЭВМ.

4. Обеспечивает выполнение арифметических и логических операций, необходимых для эффективного управления параллельным выполнением команд во всех арифметических устройствах.

5. Обеспечивает передачу информации от ведущей ЭВМ к запоминающим устройствам программ и данных БУАУ, БУКУ и БУАВУ при их загрузке и инициации.

6. Выполняет супервизорные операции для прерываний от ведущей ЭВМ, БУКУ, БУАВУ, таймеров (часов) и при определенных аварийных ситуациях в системе.

БУАУ содержит память программ емкостью 32К 32-разрядных слов и память данных емкостью 2К 32-разрядных слов, которые загружаются от ведущей ЭВМ при инициации системы. В БУАУ имеется логическая схема последовательного управления, которая осуществляет поиск команд в памяти программ и либо выполняет их сама для программногo управления, либо посылает их в блок управления параллельным выполнением команд, где они преобразуются в микропрограммные последовательности и направляются во все арифметические устройства для параллельного выполнения. Это позволяет выполнять вычисления в элементарных процессорах при минимальном вмешательстве и контроле со стороны ведущей ЭВМ. Если для БУКУ и БУАВУ понадобятся операции, требующие участия арифметических устройств, то они прерывают нормальную работу БУАУ и иницируют выполнение в арифметическом устройстве программ, хранящихся в их блоке управления, а по завершении выполнения этих программ возвращают управление нормальной программе БУАУ.

БУАУ является глобальным блоком управления для всех арифметических устройств. Все три глобальных блока управления сходны друг с другом по архитектуре и по своим функциям. Поэтому в настоящем параграфе подробно рассматриваются элементы БУАУ, а при обсуждении других блоков управления делаются ссылки на этот параграф. На рис. П.1.2 представлена функциональная блок-схема

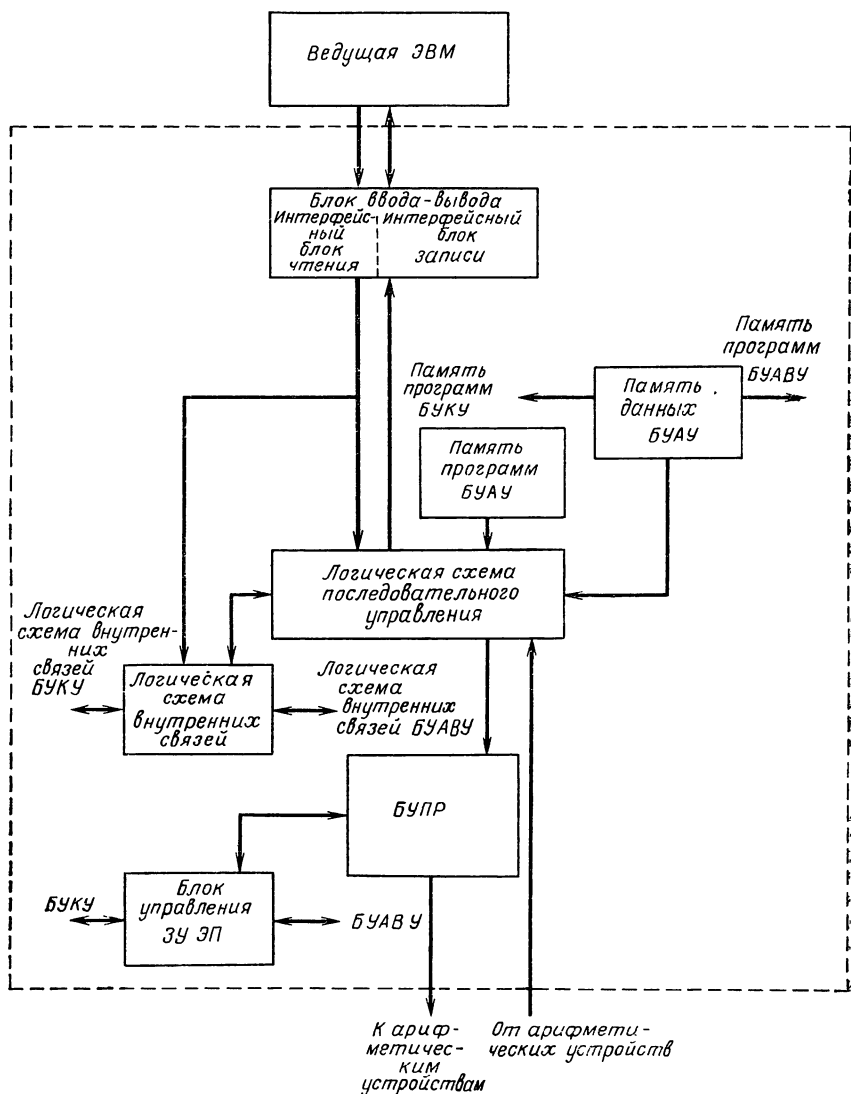


Рис. П.1.2. Блок управления арифметическими устройствами.

БУПР — блок управления параллельным выполнением команд и регистр Q параллельных команд, ЭП — элементарный процессор.

БУАУ. Взаимосвязь между всеми тремя глобальными блоками управления показана схематически на рис. П.1.3. Остальные блоки рассматриваются ниже.

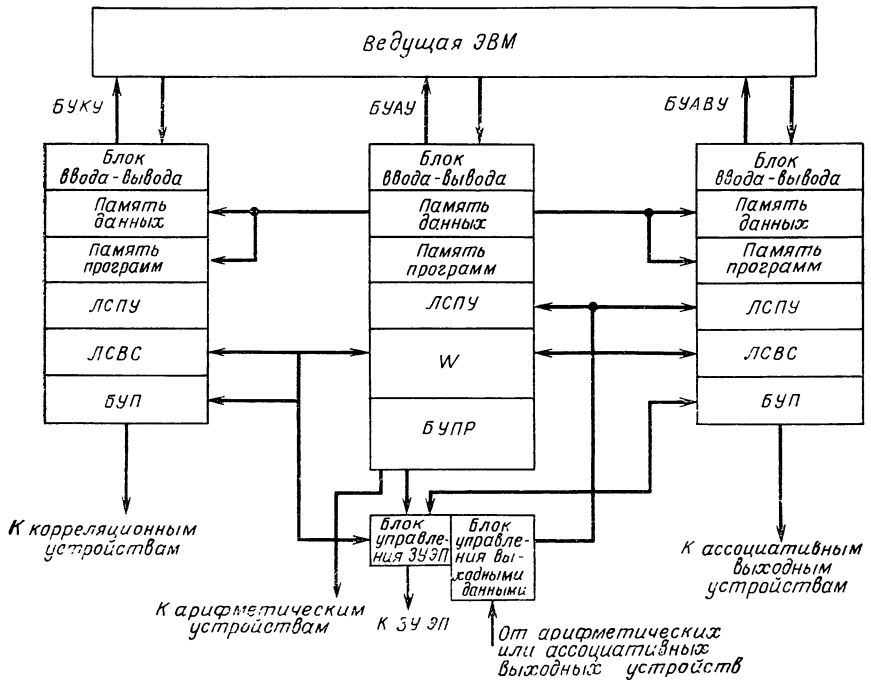


Рис. П.1.3. Взаимосвязь глобальных блоков управления.

ЛСПУ — логическая схема последовательного управления; ЛСВС — логическая схема внутренних связей, W — ЛСВС, устройство обработки прерываний, инициация программ, работа в реальном времени, трансляция команд; БУП — блок управления параллельным выполнением команд.

БЛОК ВВОДА-ВЫВОДА

Блок ввода-вывода обеспечивает сопряжение между БУАУ и мультиплексным каналом ввода-вывода ведущей ЭВМ. Блок ввода-вывода представляет собой 12-разрядный дуплексный канал, схема связи которого совместима с ведущей ЭВМ CDC 7600. Он выполняет все операции, необходимые для устранения различий в форматах слов ведущей ЭВМ и системы РЕРЕ. При этом программист должен только указать характер преобразований, сами же преобразования выполняются аппаратными средствами. Блок делится на две части — интерфейсный блок чтения и интерфейсный блок записи. Оба блока управляются набором команд из логической схемы последователь-

ного управления, общим для всех блоков управления. Эти команды определяют начальные функции при преобразовании и передаче, найденные в результате выполнения команд ввода-вывода логической схемой последовательного управления.

ЛОГИЧЕСКАЯ СХЕМА ПОСЛЕДОВАТЕЛЬНОГО УПРАВЛЕНИЯ

Логическая схема последовательного управления осуществляет управление всеми операциями БУАУ и самих арифметических устройств. Схема получает параллельные команды из памяти программ и посылает их в блок управления параллельным выполнением команд, который вырабатывает соответствующие микропрограммные последовательности для арифметических устройств элементарных процессоров. Кроме того, логическая схема получает последовательные команды из памяти программ для внутреннего выполнения.

БЛОК УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМ ВЫПОЛНЕНИЕМ КОМАНД

Блок управления параллельным выполнением команд получает от логической схемы последовательного управления те команды, которые должны быть выполнены во всех арифметических устройствах. Этот блок генерирует микропрограммные последовательности управления, необходимые для параллельного выполнения команд.

ПАМЯТЬ ПРОГРАММ

Память программ БУАУ представляет собой запоминающее устройство с непосредственной адресацией, имеющее емкость 32К 32-разрядных слов и длительность цикла запись — чтение меньше 600 нс. Конструкция БУАУ позволяет в дальнейшем добавлять идентичные блоки памяти емкостью 32К, не изменяя имеющуюся аппаратуру. Доступ к памяти программ возможен только со стороны логической схемы последовательного управления. Назначение памяти программ состоит в хранении всех программ, необходимых для арифметических устройств, находящихся под управлением БУАУ и реализующих соответствующие алгоритмы.

ПАМЯТЬ ДАННЫХ

Память данных БУАУ представляет собой запоминающее устройство с непосредственной адресацией, имеющее емкость 2048 32-разрядных слов и длительность цикла запись — чтение меньше 100 нс. БУКУ и БУАВУ имеют доступ к памяти данных при помощи расширенного адресного формата. Схема соединений запоминающих устройств блоков управления показана на рис. П.1.4. Память

данных предназначена для хранения глобальных переменных и параметров, необходимых для выполнения программ, хранящихся в памяти программ.

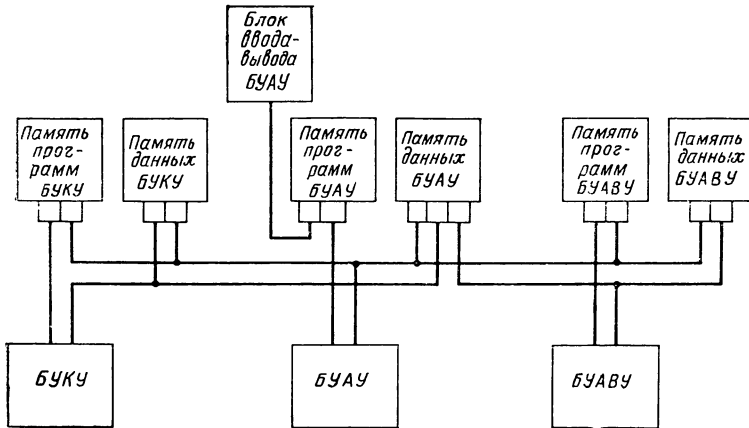


Рис. П.1.4. Связь между запоминающими устройствами глобальных блоков управления.

ЛОГИЧЕСКАЯ СХЕМА ВНУТРЕННИХ СВЯЗЕЙ

Логическая схема внутренних связей осуществляет контроль отклонений от нормальных условий в БУАУ. К этим отклонениям относятся загрузка и инициация системы для нормального выполнения программ, установление связей между блоками управления для обработки прерываний, обработка прерываний со стороны ведущей ЭВМ, синхронизация и сбор статистических данных. Так как БУАУ является блоком супервизорного типа, он обрабатывает все прерывания. Следует заметить, что при нормальной работе БУКУ и БУАВУ могут прервать БУАУ, но не наоборот. Единственный вид прерывания, который БУАУ может вызвать в двух других блоках управления,— это специальная операция прокрутки, используемая для отладки аппаратуры и для программной диагностики.

БЛОК УПРАВЛЕНИЯ ЗАПОМИНАЮЩИМИ УСТРОЙСТВАМИ ЭЛЕМЕНТАРНЫХ ПРОЦЕССОРОВ

Блок управления запоминающими устройствами разрешает приоритетные конфликты, которые возникают, когда отдельные блоки элементарных процессоров (арифметическое устройство, корреляционное устройство и ассоциативное выходное устройство) требуют одновременного доступа к памяти элементарного процессора.

БЛОК УПРАВЛЕНИЯ ВЫХОДНЫМИ ДАННЫМИ

Блок управления выходными данными подключает 32 линии связи, выходящие из элементарных процессоров, к накапливающему сумматору БУАВУ или БУАУ после выполнения команды вывода, поступающей из блока управления. Первый из этих блоков имеет более высокий приоритет при доступе к выходной информации.

БЛОК УПРАВЛЕНИЯ КОРРЕЛЯЦИОННЫМИ УСТРОЙСТВАМИ

БУКУ имеет память программ емкостью 2К 32-разрядных слов и память данных такой же емкости. Обе памяти загружаются из БУАУ при инициации системы. БУКУ принимает данные от радиолокационного интерфейса, передающего новую информацию о наблюдаемом объекте. Этот блок, используя хранящуюся в нем программу, обеспечивает выполнение корреляционными устройствами программ определения корреляции между новой информацией и предсказанными данными о положении объектов, хранящимися в корреляционных устройствах элементарных процессоров. Для получения обновленной информации осуществляется прерывание БУАУ. Для этой цели БУАУ выполняет программы, хранящиеся в его памяти программ. БУКУ получает радиолокационную информацию также от БУРИ или от ведущей ЭВМ, если используются данные от радиолокационной сети, и обеспечивает выполнение команд ассоциативного сопряжения в корреляционном устройстве для быстрой записи данных.

БЛОК ВВОДА-ВЫВОДА

Блок ввода-вывода БУКУ по конструкции и функциям аналогичен блоку ввода-вывода БУАУ, за исключением того, что он сопрягает БУКУ с ведущей ЭВМ (см. стр. 167).

ЛОГИЧЕСКАЯ СХЕМА ПОСЛЕДОВАТЕЛЬНОГО УПРАВЛЕНИЯ

Логическая схема последовательного управления является той частью БУКУ, которая управляет всеми операциями этого блока и самими корреляционными устройствами (см. стр. 168).

БЛОК УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМ ВЫПОЛНЕНИЕМ КОМАНД

Функции блока управления параллельным выполнением команд БУКУ аналогичны функциям блока управления параллельным выполнением команд БУАУ, за исключением того, что в нем нет регистра Q параллельных команд.

ПАМЯТЬ ПРОГРАММ

Память программ БУКУ представляет собой запоминающее устройство с непосредственной адресацией, имеющее емкость 2048 32-разрядных слов и длительность цикла запись — чтение меньше 100 нс. В конструкции БУКУ предусмотрена возможность подключения дополнительного блока памяти такой же емкости без внесения изменений в аппаратуру. К памяти программ БУКУ имеют доступ БУКУ или БУАУ. Функции памяти программ БУКУ аналогичны функциям памяти программ БУАУ.

ПАМЯТЬ ДАННЫХ

Память данных БУКУ представляет собой запоминающее устройство с непосредственной адресацией, имеющее емкость 2048 32-разрядных слов и идентичное по конструкции и функциям памяти данных БУАУ (см. стр. 168). Доступ к памяти данных БУКУ имеют БУКУ и БУАУ.

ЛОГИЧЕСКАЯ СХЕМА ВНУТРЕННИХ СВЯЗЕЙ

Логическая схема внутренних связей БУКУ выполняет следующие функции:

Управление инициацией программы в БУКУ по команде от ведущей ЭВМ.

Управление инициацией программы в БУКУ по команде из БУАУ для диагностических проверок.

Посылка команд запросного или триггерного прерывания в БУАУ для инициации программ БУАУ.

БЛОК УПРАВЛЕНИЯ АССОЦИАТИВНЫМИ ВЫХОДНЫМИ УСТРОЙСТВАМИ

БУАВУ содержит память программ емкостью 2К 32-разрядных слов и память данных такой же емкости. Они загружаются из БУАУ при инициации системы. Основное назначение БУАВУ состоит в управлении всеми ассоциативными выходными устройствами или выделенной их группы во время выполнения программ, хранящихся в памяти программ БУАВУ. Для ассоциативного поиска и выдачи данных через ассоциативные выходные устройства в упорядоченном виде в БУАВУ используется алгоритм поиска максимально-минимального элемента массива. Получив данные от ассоциативных выходных устройств, БУАВУ пересылает их в БУРИ. БУАВУ также управляет поиском информации о радиолокационных объектах в элементарных процессорах и передачей данных ведущей ЭВМ.

БЛОК ВВОДА-ВЫВОДА

Блок ввода-вывода БУАВУ по конструкции и по функциям аналогичен блоку ввода-вывода БУАУ, за исключением того, что он сопрягает БУАВУ с ведущей ЭВМ.

ЛОГИЧЕСКАЯ СХЕМА ПОСЛЕДОВАТЕЛЬНОГО УПРАВЛЕНИЯ

Буфер команд, индексный сумматор, регистры А, В, Q и сумматор логической схемы последовательного управления БУАВУ идентичны соответствующим устройствам логической схемы последовательного управления БУАУ. Индексные регистры и счетчик команд такие же, как в БУКУ.

БЛОК УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМ ВЫПОЛНЕНИЕМ КОМАНД

Блок управления параллельным выполнением команд БУАВУ аналогичен соответствующему блоку БУКУ.

ПАМЯТЬ ПРОГРАММ

Память программ БУАВУ аналогична памяти программ БУКУ, за исключением того, что к ней имеют доступ БУАВУ или БУАУ.

ПАМЯТЬ ДАННЫХ

Память данных БУАВУ аналогична памяти данных БУКУ, за исключением того, что к ней имеют доступ БУАВУ и БУАУ.

ЛОГИЧЕСКАЯ СХЕМА ВНУТРЕННИХ СВЯЗЕЙ

Логическая схема внутренних связей БУАВУ аналогична по своим функциям логической схеме внутренних связей БУКУ, за исключением того, что она управляет работой БУАВУ.

АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО

Арифметическое устройство осуществляет следующие функции:

Выполняет арифметические (целочисленные и с плавающей запятой) и логические операции над файлами данных о радиолокационных объектах, хранящимися в запоминающем устройстве элементарного процессора, или над операндами, поступающими из памяти данных БУАУ.

Выполняет команды запуска и выбора.

Арифметическое устройство является 32-разрядным блоком, выполняющим вычисления над арифметическими двоичными данными, представленными в форме дроби и в дополнительном коде, и над

логическими данными; оно может выполнять одноадресные команды, поступающие из БУАУ. Цель вычислений, осуществляемых арифметическим устройством, — это обновление файлов радиолокационной информации, хранящихся в памяти. Арифметические операции с плавающей запятой выполняются так, чтобы можно было сделать логические заключения относительно последующих вычислений в элементарном процессоре. По окончании вычислений их результаты заносятся в память элементарного процессора.

КОРРЕЛЯЦИОННОЕ УСТРОЙСТВО

Корреляционное устройство является входной частью элементарного процессора. Архитектура и система команд корреляционного устройства специально выбраны так, чтобы максимально ускорить процесс ввода. Конструкция устройства обеспечивает выполнение команд с максимальной скоростью, допускаемой современной техникой.

АССОЦИАТИВНОЕ ВЫХОДНОЕ УСТРОЙСТВО

Ассоциативное выходное устройство осуществляет следующие функции:

Выполняет операции сложения и вычитания над целыми числами и логические операции над файлами радиолокационных данных, хранящимися в памяти элементарного процессора, или над операндами, поступающими из памяти данных БУАВУ. Выполняет команды запуска и выбора.

Ассоциативное выходное устройство идентично арифметическому устройству по длине и формату слов, используемых в арифметических и логических схемах, за исключением того, что в ассоциативном выходном устройстве операции производятся только над целыми числами. Назначение ассоциативного выходного устройства состоит в выполнении вычислений, необходимых для вывода данных из элементарного процессора.

ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО

Запоминающее устройство осуществляет следующие функции:

Хранит все файлы радиолокационных данных, необходимых арифметическому устройству, корреляционному устройству и ассоциативному выходному устройству для выполнения в элементарных процессорах той вычислительной работы, которая выделена для системы РЕРЕ.

Организует локальную базу данных для минимизации обращений арифметического устройства, корреляционного устройства и ассоциативного выходного устройства к другим элементам системы РЕРЕ (глобальным блокам управления, ведущей ЭВМ или к БУРИ) при работе элементарных процессоров.

Запоминающее устройство имеет непосредственную адресацию, емкость 1024 32-разрядных слов и время цикла обращения меньше 75 нс. Запоминающее устройство используется совместно с арифметическим, корреляционным и ассоциативным выходным устройствами по методу пропуска цикла. Первенство при доступе к запоминающему устройству определяется блоками управления запоминающими устройствами элементарных процессоров в глобальных блоках управления (см. стр. 169). Все ячейки запоминающего устройства доступны для арифметического устройства, корреляционного устройства и ассоциативного выходного устройства.

ПРИЛОЖЕНИЕ II

МУЛЬТИПРОЦЕССОРНАЯ СИСТЕМА D825 ФИРМЫ BURROUGHS CORPORATION

Модульная вычислительная система D825 появилась в результате анализа требований к вычислительному оборудованию для командных и управляющих систем, предпринятого несколько лет назад фирмой Burroughs. Вычислительный комплекс D825 был разработан для военных приложений. Сначала была создана и испытана система под названием AN/GYK-3(V) для Исследовательской лаборатории военно-морских сил США. В этом приложении дан обзор конструкторских идей и критериев, которые были использованы при проектировании и которые привели к выбору структуры системы D825. Приведено также описание практической реализации системы и ее работы. Особый интерес представляют функции по координации работы отдельных частей системы, возложенные на программную операционную систему.

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ОБРАБОТКЕ ДАННЫХ В КОМАНДНЫХ И УПРАВЛЯЮЩИХ СИСТЕМАХ

Под командной и управляющей системой понимается система, обладающая способностью контролировать и направлять все стороны деятельности (функционирования) человека-машинного комплекса. До недавнего времени этот термин применялся исключительно к военным комплексам, но он в такой же степени применим, например, к интегрированной системе управления воздушным движением или даже к большому промышленному комплексу. Функционирование командной и управляющей системы характеризуется очень большим числом различных, но связанных между собой задач, обычно требующих решения в реальном масштабе времени. Лучше всего решать такие задачи при помощи оборудования автоматической обработки данных, причем наибольший эффект достигается при полной централизации средств обработки. Под функциями обработки данных подразумеваются обычные функции такого рода вместе со

специальными функциями, связанными с обслуживанием дисплеев, ручным вводом данных с пультов и взаимодействием со средствами связи. Ниже будут рассмотрены конструкторские решения этих проблем.

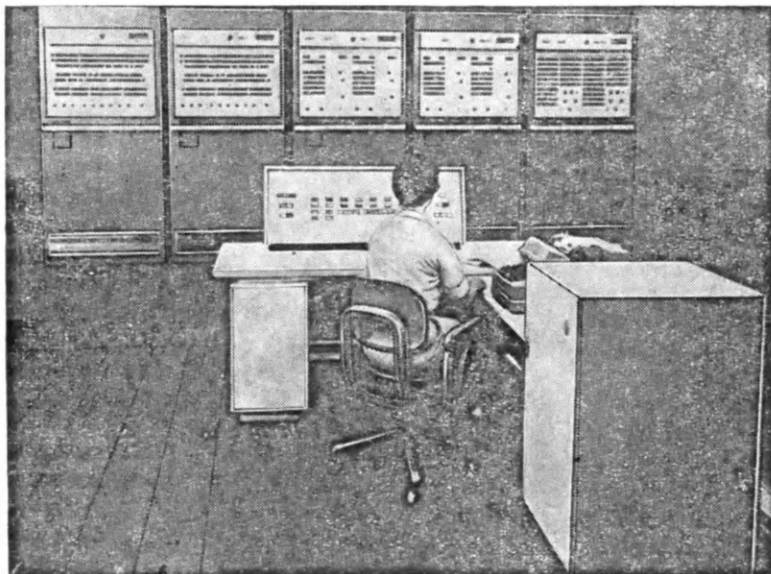


Рис. П.И.1. Общий вид системы D825 (вычислительные модули, модули памяти, контроллеры ввода-вывода, пульт индикации состояния системы и супервизорное печатающее устройство).

КРИТЕРИЙ ГОТОВНОСТИ

Самым важным требованием к средствам обработки данных является их готовность. Будучи по существу функцией надежности и ремонтпригодности аппаратуры, этот критерий представляет собой относительную продолжительность времени, в течение которого система готова к надежной работе в режиме on-line в пределах заданного периода. Каждый системный конструктор должен находить компромисс между стоимостью мер по обеспечению надежности и потерями, которые может вызвать неготовность к работе. Ни в одном из практических приложений, однако, стоимость потерь от неготовности не является столь высокой, как в командных и управляющих системах. В них не только выше требования к надежности аппаратуры, чем в коммерческих системах, но и недопустим даже малейший перерыв в работе всей системы, например для профилактиче-

ского обслуживания. В зависимости от конкретного применения большая или меньшая часть всей системы должна быть *всегда* готова к выполнению главных функций системы, а *вся система в целом* должна быть готова *большую часть времени*.

Средства обработки данных могут быть привлечены для испытания или оценки рабочих характеристик некоторых частей системы или же для имитации функций системы. При этом система должна сохранять некоторую эффективность, быть может, частично и временно пониженную, и способность быстро восстанавливать полную работоспособность. С этим опять же связано требование, которое имеет важнейшее значение при проектировании системы, — чтобы *большая часть системы всегда* была готова к работе. В системе не должно быть элементов, не подкрепленных специальным резервом и выполняющих настолько критические функции, что выход их из строя может поставить под угрозу выполнение системой главных ее функций.

КРИТЕРИЙ АДАПТИВНОСТИ

Следующее требование, которое так же трудно удовлетворить, как и предыдущее, состоит в том, что вычислительная система должна обладать способностью анализировать поступающие запросы и определять на основании этого анализа, каким конкретным задачам из общего набора заданий должно быть уделено особое внимание и отдано преимущество. Рабочая конфигурация системы должна обладать высокой степенью адаптивности и приспосабливаться к различным комбинациям задач. Кроме того, она должна быстро реагировать на все существенные изменения, о которых могут свидетельствовать либо внешние сигналы, либо результаты внутренних вычислений (например, превышение некоторых порогов), а также на изменения в конфигурации аппаратуры из-за выхода из строя некоторых ее элементов или намеренного удаления их из системы. Система должна обладать способностью динамично и автоматически перестраивать свою структуру в соответствии с представленным ей набором заданий.

КРИТЕРИЙ СПОСОБНОСТИ К РАСШИРЕНИЮ

Требование способности к расширению относится не только к области командных и управляющих систем; это свойство желательно в любой области применения средств обработки данных. Однако оно наиболее необходимо в командных и управляющих системах, так как их эффективность сильно зависит от способности удовлетворять быстро меняющимся способам ведения военных действий. Кроме того, должна быть обеспечена возможность вводить в систему новые

функции с минимальными затратами времени на перестройку аппаратного обеспечения или вообще без таких затрат.

Расширение системы должно осуществляться без расходов на обеспечение таких возможностей, которые не нужны в данный момент времени. Система должна быть способна к расширению не только за счет обычно расширяемых средств, таких, как запоминающие устройства и устройства ввода-вывода, но и за счет собственно вычислительных устройств.

ПРОГРАММНЫЙ КРИТЕРИЙ

Расширение возможностей по обработке данных не должно требовать составления новых программ для выполнения старых функций, а программы для выполнения новых функций должны легко подключаться к уже сложившейся системе. Для этого необходимо составлять программы так, чтобы они не зависели от конфигурации системы или от набора заданий. Необходимо даже взаимозаменяемость программ, используемых в разных географических пунктах для выполнения одних и тех же задач. Наконец, вследствие большого числа составляющих программ в программном обеспечении командной и управляющей системы должна быть предусмотрена возможность написания отдельных программ или частей программ разными программистами и последующего объединения их при помощи соответствующей операционной системы.

В связи с последним требованием и требованием независимости программ от конфигурации системы выдвигается пожелание, чтобы конструкция и рабочие принципы системы были пригодны для использования процедурно-ориентированного языка высокого уровня. Такой язык должен обладать всеми качествами обычных алгоритмических языков для решения научных задач, но в нем должна быть предусмотрена возможность обработки больших файлов данных, которые могут быть плохо организованы. Желательно также, чтобы язык отражал специфику области применения системы; это особенно важно в тех случаях, когда язык используется для управления хранением и поиском данных.

ОСНОВНЫЕ ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ ОБОРУДОВАНИЕ ДЛЯ ОБРАБОТКИ ДАННЫХ

В основу разработки конструкции системы D825 были положены требования готовности, адаптивности и способности к расширению. Перед принятием окончательного решения о конструкции системы было рассмотрено несколько существующих и предложенных схем организации таких систем в свете перечисленных выше критериев. Многие из полученных заключений относительно тех или иных схем применения вычислительной техники в командных и управляющих

системах совпали с результатами недавно опубликованной работы [1], выполненной независимо Институтом анализа оборонных проблем для Министерства обороны США.

ОДНОМАШИННАЯ СИСТЕМА

Наиболее очевидной и наименее пригодной схемой для командной и управляющей системы является одномашинная структура. Она не может удовлетворить требованию готовности просто потому, что выход из строя любого ее элемента — процессора, памяти, контроллера устройств ввода-вывода — приводит к отказу всей системы. Поэтому она и не подвергалась сколько-нибудь серьезному рассмотрению.

ОДНОМАШИННЫЕ СИСТЕМЫ С ДУБЛИРОВАНИЕМ

В период, когда обсуждались рассматриваемые проблемы, широкое распространение имела системная организация с дублированием (утроением и т. д.) одномашинной системы для достижения необходимой готовности и более высокой производительности. Такой подход сначала казался привлекательным, поскольку рабочие программы можно было разделить между двумя или тремя независимыми одномашинными системами и выбрать столько таких систем, сколько необходимо для выполнения всей вычислительной работы. Даже требования готовности, казалось, были удовлетворены, так как можно было всегда держать одну избыточную систему в качестве резерва при выполнении главных функций.

Однако при более близком рассмотрении оказалось, что такие системы обладают многими недостатками при использовании их для командных функций и управления. Помимо того, что они требуют значительных усилий со стороны человека по координации работы системы и что в них имеют место существенные потери машинного времени, такие системы весьма малоэффективны именно потому, что при выполнении командных и управляющих функций данные и программы сильно взаимосвязаны между собой. Кроме того, процесс передачи наиболее важных задач избыточной или вспомогательной системе (если в этом возникает необходимость) оказался весьма громоздким и длительным, особенно в тех случаях, когда требуется постоянный контроль за событиями и когда предъявляются особенно жесткие требования к временному фактору.

СХЕМЫ С ЧАСТИЧНО РАЗДЕЛЕННОЙ ПАМЯТЬЮ

Если дублированную систему несколько модифицировать, используя в ней частично разделенную память, то она может приобрести некоторые новые, весьма важные свойства. Частично разделенная память может быть организована по-разному, но в ней обяза-

тельно имеется некоторый объем разделенной памяти и некоторый объем памяти, закрепленный за индивидуальными ЭВМ. Разделенная память может быть реализована на магнитных лентах, на дисках или на сердечниках, но чаще всего для нее используют магнитные сердечники. Такая система, обеспечивающая непосредственную связь между ЭВМ, является большим шагом вперед по пути удовлетворения перечисленных выше требований.

Одним из преимуществ закрепления части памяти за отдельным ЭВМ является защита данных. Это преимущество исчезает, как только возникает необходимость обмениваться данными между ЭВМ, так как при отказе одной из ЭВМ содержимое памяти, закрепленной за этой машиной, будет потеряно для системы. Кроме того, при решении многих командных и управляющих задач нужен доступ к одним и тем же данным. Если, например, потребуется, чтобы данные, хранящиеся в памяти одной из ЭВМ, были доступны для другой ЭВМ или для всех ЭВМ, пользующихся разделенной частью памяти, то на передачу данных будет затрачено значительное время. Ясно также, что в такой системе недостаточна максимальная эффективность использования оборудования, так как некоторая часть памяти, закрепленная за одной из ЭВМ, может не использоваться, в то время как другая ЭВМ может нуждаться в дополнительной памяти и будет вынуждена организовать передачу данных из своей памяти в общую, чтобы освободить место для хранения необходимых данных. Заметим, что при рассмотрении отдельных устройств ввода-вывода данных для ЭВМ также возникают проблемы снижения общей готовности и общей эффективности.

СХЕМЫ С ВЕДУЩИМИ И ВЕДОМЫМИ ЭВМ

Другим аспектом систем с частично разделенной памятью является управление. В ряде таких систем используются схемы типа «ведущий-ведомый». В них одна ЭВМ, называемая ведущей, координирует работу, выполняемую другими машинами. Ведущая ЭВМ может отличаться по типу от остальных ЭВМ, как, например, в системе PILOT, разработанной Национальным бюро стандартов [2], или она может быть того же типа и отличаться от остальных только своей ролью, как, например, в системе TRW-400 (AN/FSQ-27) фирмы Thompson Ramo-Wooldridge [3]. В этих схемах учитывается важность координации вычислительных операций многомашинной системы. Ведущая ЭВМ является эффективным средством такой координации. Однако при реализации этой схемы приходится преодолевать некоторые трудности. Выход из строя ведущей ЭВМ приводит к полной потере работоспособности всей системы, и, таким образом, требования готовности, предъявляемые к командным и управляющим системам, не удовлетворяются. Если даже преодолеть этот недостаток, предусмотрев возможность автоматической передачи

функций ведущей какой-либо из других ЭВМ, все же такая схема по существу не может обеспечить максимальную эффективность. Если, например, вычислительная нагрузка ведущей ЭВМ чрезмерно возрастет, то эта ЭВМ станет узким местом системы, и использование других элементов системы будет малоэффективным. Наоборот, если нагрузка такова, что ведущая ЭВМ недогружена, то и в этом случае происходят непроизводительные потери вычислительной мощности. Отсюда вытекает заключение, что ведущая ЭВМ должна появляться в системе только тогда, когда в этом имеется необходимость. Такой подход был применен при проектировании системы D825.

СХЕМА С ПОЛНОЙ МОДУЛЬНОСТЬЮ

В результате проведенного анализа стали очевидными некоторые выводы. Требование готовности диктует необходимость децентрализации вычислительных функций, т. е. наличие нескольких вычислительных блоков. Однако особенности рассматриваемых приложений системы требуют свободного обмена данными между этими ЭВМ. Поэтому было принято решение о полном разделении памяти между всеми ЭВМ. Кроме того, очевидно, что с точки зрения готовности и эффективности нежелательно, чтобы операции ввода-вывода были жестко связаны с какой-либо одной ЭВМ. Поэтому управление устройствами ввода-вывода было отделено от ЭВМ.

Система с несколькими процессорами, полностью разделенной памятью и с автономной системой ввода-вывода оказалась идеальной структурой, отвечающей требованию адаптивности. Она обеспечивает гибкость управления, необходимую для выполнения динамичных, быстро изменяющихся вычислительных заданий.

Для реализации вычислительного потенциала, сосредоточенного в такой системе, нужно еще решить проблему координации работы всех элементов системы, чтобы она в любой момент времени действовала так, как действовала бы система, специально предназначенная для решения конкретного комплекса задач, который поставлен перед системой в данный момент времени. Вследствие того что имевшееся ранее оборудование обладало ограниченными возможностями, программы операционной системы всегда отождествлялись с аппаратурой, которая реализовала эти программы. Однако в предложенном проекте все модули ЭВМ должны были иметь непосредственный доступ ко всей памяти, и поэтому появилась возможность не связывать операционную систему с каким-либо определенным процессором. Стала возможной координация работы всей системы, при которой каждый процессор реализует программы операционной системы лишь тогда, когда в этом возникает необходимость. Таким образом, ведущая ЭВМ фактически превратилась в программу, хранящуюся в полностью разделенной памяти, что значительно усилило гибкость программирования.

До сих пор ничего не было сказано относительно того, что все процессоры системы должны быть идентичны. Одинаковая ответственность всех вычислительных блоков, позволяющая любому процессору выступать в роли ведущего при реализации программ операционной системы, привела в конце концов к конструктивному требованию идентичности всех вычислительных модулей. При выполнении этого требования они могут свободно подсоединяться к множеству идентичных модулей памяти и к множеству идентичных контроллеров ввода-вывода, а последние в свою очередь могут свободно подключаться к изменяющемуся набору разнообразных устройств ввода-вывода. Совершенно очевидно, что подобная полная модульность всех элементов системы является эффективным решением проблемы ее расширения, которое может быть осуществлено простым добавлением модулей такого же типа, как модули уже существующей системы. Ясно также, что такая система обладает важными преимуществами, в частности экономического характера, связанными с производством и эксплуатацией одинаковых модулей и снабжением их запасными частями. Возможно, самым важным преимуществом полностью модульной организации является то, что избыточность комплекта модулей любого типа, необходимую для обеспечения требуемой надежности, легко получить простым добавлением к системе всего лишь одного модуля данного типа. Кроме того, дополнительные модули не обязательно бездействуют — они могут рассматриваться как активный резерв.

Таким образом, в конце концов была выбрана структура вычислительной системы, основанная на полной модульности. Для того чтобы объединить различные функциональные модули в единую координированную систему, оставалось решить две проблемы: создать устройство электронной связи между модулями и создать операционную систему, выполняющую функции ведущей ЭВМ для координации работы всех модулей.

В системе D825 эти две задачи выполняют соответственно коммутатор и программа автоматической работы и планирования AOSP (ПАРП).

РЕАЛИЗАЦИЯ СИСТЕМЫ

При реализации проекта системы D825 наиболее важными оказались проблемы создания коммутатора и программы автоматической работы и планирования. Процессоры и контроллеры ввода-вывода допускали более привычные решения, хотя и должны были обладать некоторыми специальными характеристиками, в то время как многие из устройств ввода-вывода были просто выбраны из имеющегося стандартного оборудования. Ниже дан краткий обзор всех этих элементов (за исключением устройств ввода-вывода). Характеристики системы D825 приведены в табл. П.11.1.

Характеристики модульной системы обработки данных D825

Число вычислительных модулей	Максимум 4
Тип вычислительного модуля	Цифровой, двоичный, параллельный, полупроводниковый
Длина слова	48 разрядов, включая знак (8 шестизначных символов), и 1 разряд проверки на четность
Число индексных регистров в каждом вычислительном модуле	15
Регистры на магнитных тонких пленках (в каждом вычислительном модуле)	128 шестнадцатиразрядных слов, длительность цикла записи — считывание 0,33 мкс
Часы реального времени (в каждом вычислительном модуле)	Дискретность 10 мс
Среднее время сложения двоичных чисел	1,67 мкс
Среднее время умножения двоичных чисел	36,0 мкс
Среднее время сложения с плавающей запятой	7,0 мкс
Среднее время умножения с плавающей запятой	34,0 мкс
Время логической операции И	0,33 мкс
Тип памяти	Однородная, модульная, с произвольным доступом, с линейной выборкой, на ферритовых сердечниках
Емкость памяти	65 536 слов (максимум 16 модулей по 4096 слов каждый)
Число коммутаторов ввода-вывода в системе	1 или 2
Максимальное число контроллеров ввода-вывода	10 на один коммутатор ввода-вывода
Максимальное число устройств ввода-вывода	64 на один коммутатор ввода-вывода
Доступ к устройствам ввода-вывода	Все устройства ввода-вывода доступны каждому контроллеру, подключенному к коммутатору ввода-вывода.
Скорость передачи через коммутатор ввода-вывода	2 000 000 символов/с
Тип устройства ввода-вывода	Все типы стандартных устройств ввода-вывода, включая запоминающие устройства на магнитных лентах с шириной полосы пропускания 67 кГц, магнитные барабаны и диски, устройства для записи и чтения перфокарт и перфолент, знако- и строчнопечатающие устройства, дисплеи и устройства связи.

КОММУТАТОР

После того, как было принято решение, что только полностью разделенная память отвечает поставленным требованиям, нужно было найти способ доступа каждого процессора к каждому запоминающему устройству и разделения каждого запоминающего устройства между двумя или большим числом процессоров или контроллерами ввода-вывода.

Такая функция, физически распределенная по всем модулям системы D825 и названная обобщенно «коммутацией», представляет собой электронный способ реализации многочисленных кратковременных контактов для передачи всех потоков информации между процессорами, памятью и контроллерами ввода-вывода. Помимо выполнения функций электронного переключения, коммутатор должен обладать способностью обнаруживать и разрешать все конфликты, которые возникают в тех случаях, когда два или больше процессоров пытаются получить доступ к одному и тому же модулю памяти.

Функционально коммутатор состоит из матрицы с перекрестной коммутацией, которая непосредственно осуществляет контакты между шинами, и распределителя шин, который разрешает конфликты, возникающие при одновременном требовании на доступ к одной и той же шине или к одному и тому же системному модулю. Конфликтные запросы выстраиваются в очередь в соответствии с приоритетами, присвоенными запрашивающим устройствам. Приоритеты являются абсолютными в том смысле, что появление запроса с более высоким приоритетом приводит к обслуживанию этого запроса, прежде чем будет обслужен запрос, уже стоящий в очереди, но обладающий более низким приоритетом. Анализ вероятностных характеристик очередей показывает, что очереди, содержащие больше одного запроса, весьма маловероятны.

Функция планирования запросов с учетом их приоритетов выполняется распределителем шин, который представляет собой по существу набор логических матриц. Конфликтная матрица обнаруживает наличие конфликтов среди запросов на соединения. Приоритетная матрица распознает приоритет каждого запроса. Логическое произведение состояний конфликтной и приоритетной матриц определяет состояние матрицы очередей, которая в свою очередь включает соответствующие контакты матрицы перекрестной коммутации, если только запрашиваемый модуль еще не занят.

ПРОГРАММА АВТОМАТИЧЕСКОЙ РАБОТЫ И ПЛАНИРОВАНИЯ (ПАРП)

ПАРП является программой операционной системы, хранящейся в полностью разделенной памяти и поэтому доступной любому процессору. Программа реализуется только в случаях, когда необходимо

соответствующее управление системой. В ПАРП имеется собственная исполнительная подпрограмма, являющаяся «операционной системой для операционной системы» и вызывающая по мере необходимости дополнительные подпрограммы. Таким образом, конфигурация ПАРП позволяет изменять (в зависимости от решаемой задачи) как последовательность и число рабочих подпрограмм, так и размещение ПАРП в памяти.

ПАРП эффективно работает на двух уровнях — на уровне управления системой и на уровне обработки задания.

Функция управления системой включает все операции, необходимые для вызова системных программ и соответствующих данных из определенных периферических устройств, подготовки программ к выполнению путем поиска и назначения зоны в памяти и запуска процедур обработки. Большая часть этой функции управления (точно так же, как и функции обработки задания) состоит в тщательном учете реализуемых программ, активных программ (т. е. программ, занимающих место в памяти), команд ввода-вывода, подлежащих выполнению, других команд ввода-вывода, ожидающих выполнения, массивов внешних данных, подлежащих приему и дешифровке, и в запуске соответствующих программ для обработки внешних данных. Здесь не целесообразно рассматривать огромное множество деталей программы ПАРП. Некоторое представление о ее назначении можно получить из следующего списка ее главных функций:

1. Определение конфигурации.
2. Распределение памяти.
3. Планирование.
4. Подготовка программ и очистка памяти после выполнения задания.
5. Сообщения о ходе работы и регистрация.
6. Диагностика и проверка правильности решения.
7. Обработка внешних прерываний.

Функция ПАРП по обработке заданий состоит в выполнении всех программных запросов на ввод-вывод для обеспечения централизованного планирования и предотвращения возможного разрушения информации неправильно составленными или конфликтующими программами.

ОБРАБОТКА ПРЕРЫВАНИЙ

Работа ПАРП в значительной степени опирается на широкий набор типов прерываний, предусмотренный в системе D825. Сигналы прерываний всех типов передаются всем модулям системы, и каждый модуль может реагировать на прерывание любого типа. Однако для распределения ответственности за прерывания различных типов (как системных, так и локальных) в каждом модуле имеется регистр масок прерываний, который контролирует значения отдельных раз-

рядов регистра прерываний. Поступивший сигнал прерывания какого-либо типа заставляет один из модулей прекратить реализацию текущей программы и переключиться на соответствующую подпрограмму ПАРП, перейдя в *режим контроля*. Режим контроля отличается от режима нормальной работы тем, что в этом режиме блокируется реакция на некоторые низкоприоритетные прерывания (хотя они и регистрируются) и становится возможным выполнение некоторых дополнительных команд, введенных специально для ПАРП (таких, например, как установка регистра масок прерываний или регистров защиты памяти или передача какой-либо команды ввода-вывода одному из контроллеров ввода-вывода).

Реагируя на прерывание, ПАРП передает управление соответствующей подпрограмме обработки прерываний данного типа. После устранения причины прерывания управление снова возвращается рабочей программе, которая выполнялась перед прерыванием. В режиме нормальной работы возможны прерывания, вызванные следующими условиями:

1. Внешние запросы (16 видов).
2. Завершение операции ввода-вывода.
3. Переполнение часов реального времени.
4. Отсутствие массивов данных.
5. Сигналы прерывания от одной ЭВМ к другой.
6. Переход в режим контроля (прекращение работы в нормальном режиме).

Прерывания, связанные с отклонением от нормальных условий в программе или в аппаратуре, могут быть вызваны следующими причинами:

1. Попытка программы записать информацию за пределами граничных адресов.
2. Арифметическое переполнение.
3. Неразрешенная команда.
4. Невозможность получить доступ к памяти или ошибка, выявленная контролем на четность (если ошибка произошла во время выполнения операции ввода-вывода, то эта операция будет завершена, но в ПАРП будет передана соответствующая информация).
5. Выход из строя первичного источника питания.
6. Автоматический перезапуск системы после выхода из строя первичного источника питания.
7. Ненормальное завершение операции ввода-вывода.

Хотя причины учета большинства перечисленных выше прерываний очевидны, тем не менее уместно будет обсудить некоторые из них подробнее.

Прерывание из-за отсутствия массива данных возникает при обращении к данным, которых нет в памяти. Так как все обращения, например, к массиву $A[k]$ выполняются относительно базы масси-

ва, т. е. по адресу первого элемента массива, необходимо получить этот адрес и добавить к нему величину k . При обращении к базе массива A аппаратура, распознающая содержимое «разряда присутствия», либо разрешает продолжать операцию, либо инициирует прерывание из-за отсутствия массива данных. Таким образом удаётся упростить слежение за данными, используемыми взаимодействующими программами, а также задачу распределения памяти.

Прерывание вследствие отказа первичного источника питания имеет наивысший приоритет и всегда выполняется в первую очередь. Это прерывание заставляет все процессоры и контроллеры ввода-вывода прекратить выполнение операций и записать всю текущую информацию либо в модулях памяти, либо в регистрах на магнитных тонких пленках (последние представляют собой интегральные элементы вычислительных модулей). Этот вид прерываний защищает систему от влияния кратковременных отключений питания и выполняется в тех случаях, когда напряжение первичного источника питания падает ниже заранее установленного уровня.

В системе предусмотрен автоматический перезапуск после прерывания из-за отказа первичного источника питания; он обеспечивает восстановление предшествующего состояния системы.

Описание процесса обработки внешнего прерывания может пояснить общую процедуру прерывания. После поступления сигнала внешнего прерывания процессор, ответственный за обработку таких прерываний, автоматически запоминает содержимое тех регистров, которые необходимы для последующего восстановления состояния (например, счетчика программы), переходит в режим контроля и обращается к стандартной (аппаратурно определяемой) ячейке, в которой хранится начало программной ветви, содержащей подпрограмму обработки внешних прерываний. Эта подпрограмма определяет, какая из внешних запросных линий требует обслуживания, и после рассмотрения таблицы внешних устройств (телетайпные буферы, клавишные пульты, дисплей и т. д.), связанных с линиями прерываний, процессор формирует и посылает в запрашивающее устройство команду ввода начального сообщения. Затем процессор обращается к таблице программы ввода-вывода (программы обработки прерываний по вводу-выводу) для активации соответствующей подпрограммы после чтения сообщения. Далее производится проверка на отсутствие других внешних запросов. Наконец, процессор восстанавливает сохраненное содержимое регистров и возвращается к нормальному режиму для выполнения прерванной программы.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Как уже упоминалось, управление вводом-выводом также относится к функциям ПАРП, которая регистрирует состояния и степень готовности всех устройств ввода-вывода. Записываются также ме-

стоположения в вычислительной системе всех файлов независимо от того, где они находятся: на магнитной ленте, на барабанах, на дисках, на перфокартах или представлены информацией, поступающей извне. Сначала рассматривается запрос на вход в файл по его имени и, если устройство, которому присвоено это имя, свободно, то инициируется процедура ввода-вывода. Если по какой-либо причине запрос должен быть отвергнут, то он помещается в программную очередь и ожидает, когда возникнут условия для его реализации. Типичными условиями, вызывающими отказ в выполнении операции ввода-вывода, являются:

1. Отсутствие свободного контроллера ввода-вывода или канала.
2. Занятость устройства, в котором находится запрашиваемый файл.
3. Отсутствие нужного файла в системе.

В последнем случае обычно на супервизорное печатающее устройство выводится сообщение о запросе отсутствующего файла.

Прерывание при завершении операции ввода-вывода сигнализирует об окончании каждой такой операции. Одновременно с этим прерыванием в таблице ПАРП записывается дескриптор результата операции ввода-вывода. Информация, содержащаяся в этом дескрипторе, указывает, успешно ли закончилась данная операция, а если нет, то указывается причина неудачи (ошибка при проверке на четность, обрыв ленты, замятие перфокарты и т. д.), чтобы ПАРП могла предпринять соответствующие действия. Если же операция закончилась успешно, то запускается следующая процедура ввода-вывода, ожидающая своей очереди.

УПРАВЛЕНИЕ ПЛАНИРОВАНИЕМ ПРОГРАММ

Процесс планирования в системе D825 основан на таблице заданий, которую обрабатывает ПАРП. Каждый запрос идентифицируется по имени, приоритету, условиям предшествования и требованиям к оборудованию. Приоритет может быть динамическим, зависящим от времени, внешних запросов, других программ, и может быть функцией многих изменяющихся условий. Каждый раз, когда ПАРП вызывается для выбора следующей рабочей программы в результате завершения предыдущей или в результате прерывания по другим причинам, осуществляется просмотр таблицы заданий. В системах, работающих в реальном времени, могут возникать ситуации, когда на очереди нет системных программ, и машинное время освобождается для других целей; это время может быть использовано для выполнения вспомогательных функций, таких, как реализации закрытых (секретных) программ.

ПАРП предоставляет возможность сегментирования программ по указаниям программиста. Макрокоманды управления, вставленные программистом в рабочую программу, информируют ПАРП о

том, что в данный момент можно начать параллельную обработку данных на двух процессорах. Кроме того, программист должен указывать, где должны сойтись ветви программы после завершения параллельной обработки.

ВЫЧИСЛИТЕЛЬНЫЙ МОДУЛЬ

Вычислительные модули системы D825 представляют собой по существу одинаковые универсальные арифметические устройства и устройства управления. При определении внутренней структуры вычислительных модулей учитывались два главных соображения. Во-первых, все программы и данные должны легко перемещаться в любое место, с тем чтобы упростилась функция распределения памяти программой ПАРП. Во-вторых, программы не должны изменяться во время их исполнения. Последнее необходимо, чтобы свести к минимуму объем работы для завершения программы, так как все, что нужно запомнить для возобновления прерванной программы, должно свестись к исходным данным для этой программы и к содержимому регистров вычислительного модуля в тот момент, когда произошло прерывание его работы.

В вычислительных модулях системы D825 используется переменный по длине формат команд, состоящий из слогов, длина которых равна четверти слова. Одно-, двух-, трехадресные и безадресные слоги могут быть связаны по требованию с любым базисным слогом команды. Совместно с арифметическим устройством работает неявно адресуемый накапливающий стек памяти. Предусмотрена индексация всех адресов команды, а также косвенная адресация данных любого ранга.

Каждый вычислительный модуль содержит память на тонких пленках на 128 позиций, которая применяется в стеке, а также во многих регистрах ЭВМ, таких, как базовый регистр программ, базовый регистр данных, индексные регистры и регистры границ.

В систему команд комплекса D825 входят обычные арифметические операции с фиксированной и плавающей запятой, логические операции и операции с частями поля, которые можно встретить в любом достаточно большом процессоре, предназначенном для решения научных задач.

МОДУЛЬ ПАМЯТИ

Модули памяти состоят из независимых блоков емкостью 4096 48-разрядных слов. Каждый такой блок снабжен своим источником питания и всеми электронными схемами, необходимыми для записи, считывания и передачи данных. Размер модулей памяти выбран на основании компромисса между малыми размерами, минимизирующими конфликты при обращении к одному и тому же модулю

памяти двух или большего числа процессоров или модулей ввода-вывода, и большими размерами, обеспечивающими приемлемую стоимость дублированных источников питания и логических схем адресации.

Следует заметить, что для модульной системы больших размеров эти компромиссные соображения могут привести к выбору модулей памяти емкостью 8192 слова. Использование модулей еще больших размеров — например, емкостью 16 384 или в 32 768 слов — приведет к трудностям при выполнении требования минимизации габаритов аппаратуры. Дополнительные расходы, связанные с использованием более мелких моделей, компенсируются снижением убытков при выходе из строя отдельного модуля.

КОНТРОЛЛЕР ВВОДА-ВЫВОДА

Контроллер ввода-вывода выполняет операции по вводу-выводу, указываемые и инициируемые вычислительным модулем. В соответствии с общесистемными целями контроллеры ввода-вывода не закрепляются за определенными вычислительными модулями, а используются системой почти так же, как модули памяти. В них предусмотрено автоматическое разрешение конфликтов, возникающих при попытках одновременного доступа, при помощи коммутационной системы. Если началось выполнение какой-либо операции ввода-вывода, то она продолжает выполняться до своего полного завершения независимо от других запросов.

Операция ввода-вывода инициируется путем выполнения команды начала ввода-вывода в одном из вычислительных модулей, в результате чего из определенной ячейки памяти передается в один из свободных контроллеров ввода-вывода дескрипторное слово ввода-вывода. Этот дескриптор ввода-вывода представляет собой команду, заставляющую контроллер ввода-вывода выбрать устройство ввода-вывода, определить направление потока данных, адрес первого слова и число слов, подлежащих передаче.

Между контроллерами ввода-вывода и физическими периферийными устройствами включен другой перекрестный коммутатор, называемый коммутатором ввода-вывода. Этот автоматический коммутатор, аналогичный общей системе коммутации, обеспечивает двусторонний поток данных между любым контроллером ввода-вывода и любым устройством ввода-вывода. Такой коммутатор увеличивает гибкость системы, предоставляя столько внешних путей передачи данных, сколько имеется контроллеров ввода-вывода.

КОМПЛЕКТАЦИЯ ОБОРУДОВАНИЯ

Система D825 может быть собрана (или расширена) посредством выбора соответствующих модулей в любой комбинации; можно использовать от 1 до 4 вычислительных модулей, от 1 до 16 модулей

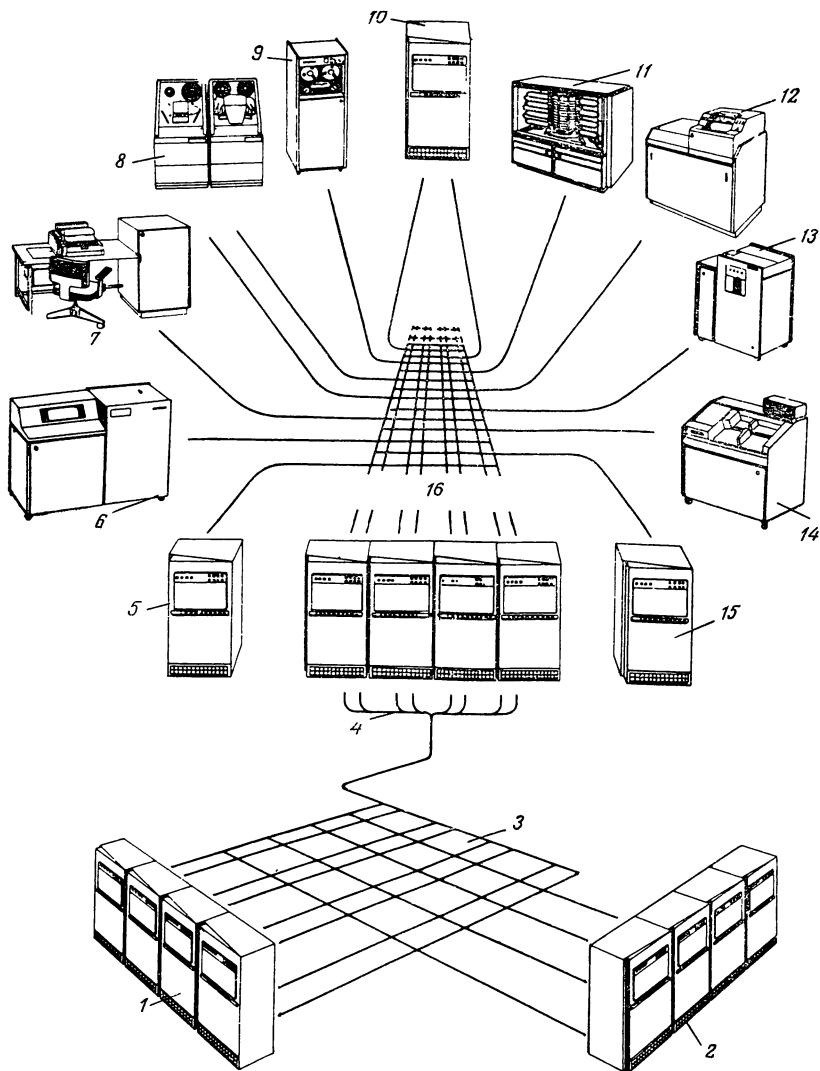


Рис. П.П.2. Организация модульной системы обработки данных D825.

1 — модули памяти, 2 — вычислительные модули, 3 — коммутатор, 4 — контроллеры ввода-вывода, 5 — специальные часы реального времени и преобразователи данных, 6 — высокоскоростное печатающее устройство, 7 — клавишное печатающее устройство, 8 — перфоратор и читающее устройство для бумажной ленты, 9 — ЗУ на магнитной ленте, 10 — магнитные диски (два на шкаф), 11 — магнитные диски, 12 — печатающее устройство S203, 13 — карточный перфоратор, 14 — устройство чтения перфокарт, 15 — устройство для внутрисистемной передачи данных, 16 — автоматический коммутатор ввода-вывода (максимум на 64 устройства).

памяти, от 1 до 10 контроллеров ввода-вывода, 1 или 2 коммутатора ввода-вывода и от 1 до 64 устройств ввода-вывода на один коммутатор ввода-вывода. Допускается любая комбинация устройств ввода-вывода, которая может включать следующее оборудование: рабочие пульты (или пульты индикации состояния), запоминающие устройства на магнитной ленте, магнитные барабаны, магнитные диски, карточные перфораторы и читающие устройства, ленточные перфораторы и читающие устройства, супервизорные печатающие устройства, быстродействующие строчно-печатающие устройства, специальные преобразователи данных, специальные датчики времени и устройства для внутрисистемной передачи данных (рис. П.11.2).

ЗАКЛЮЧЕНИЕ

По убеждению авторов описанной системы, модульные системы (в том смысле, в каком они были рассмотрены выше) представляют собой естественное решение проблемы получения большой вычислительной производительности — более естественное, чем просто создание больших ЭВМ с высокой производительностью. Если же говорить более конкретно, то было показано, что организационная структура системы D825 может служить подходящей базой при создании систем обработки данных для командных и управляющих систем. Хотя исследования, в результате которых была выбрана данная структура, проводились как развернутое наступление на целый ряд разнообразных проблем, теперь стало очевидным, что требования, присущие рассматриваемой области применения, в действительности представляют собой различные аспекты одной и той же характеристики, которую можно было бы назвать «структурной свободой». Более того, теперь ясно, что наиболее уникальная характеристика реализованной структуры — согласованная работа свободно взаимодействующих модульных элементов — обеспечивает достижение структурной свободы.

Например, одно из требований заключается в том, что всегда должна быть обеспечена определенная минимальная способность системы обрабатывать данные, т. е. в любых условиях деградации системы вследствие отказов или выполнения работ по ее обслуживанию оставшееся оборудование должно быть в состоянии выполнять главные функции системы. В системе D825 выход из строя одного модуля приводит к уменьшению рабочей конфигурации, но позволяет продолжать нормальную работу, хотя, возможно, с пониженной производительностью. Отдельные модули системы обладают весьма высокой надежностью и ремонтпригодностью, но высокая готовность системы обусловлена не только этим единственным фактором, как это имеет место у обычных систем. Модульная конфигурация фактически позволяет работать с активным резервом, что исключает необходимость стопроцентной избыточности оборудования.

Вторым требованием является возможность мгновенной перестройки рабочей конфигурации системы в любой данный момент и образования новой структуры, более подходящей для вычислительной нагрузки, которая динамично и непредсказуемо изменяется. В системе D825 все каналы связи общедоступны, все модули функционально независимы друг от друга, все планирование заданий выполняется динамично, а конкретные рабочие конфигурации являются гибкими. Системы прерываний и приоритетов, управляемые ПАРП и системой коммутации, позволяют мгновенно адаптироваться к любой работе без разрушения прерванных программ.

Требование расширяемости сводится просто к способности адаптироваться на более длительные периоды времени. Так как все модули системы D825 функционально развязаны, добавление новых модулей любого типа может быть осуществлено просто подключением их к схеме коммутации или к коммутатору ввода-вывода. Расширение этой системы во всех функциональных областях возможно в значительно больших масштабах, чем для обычных вычислительных систем.

Однако совершенно очевидно, что система D825 не достигла бы всех поставленных перед ней целей, если бы она опиралась только на аппаратные средства. ПАРП является такой же неотъемлемой частью структуры системы, как и аппаратура. Идея «плавающей» ПАРП как орудия объединения модулей аппаратуры в единую систему является очень важной и оказывает влияние далеко за пределами конкретной ее реализации в системе D825. Интересным побочным результатом проектирования системы D825 является изменение сложившихся взглядов: стало совершенно очевидно, что не ЭВМ выполняет программы, а программы управляют работой ЭВМ.

ЛИТЕРАТУРА

1. *Kroger M. G. et al.*, Computers in Command and Control, ITR61-12, prepared for DOD: ARPA by Digital Computer Application Study, Institute for Defense Analyses, Research and Engineering Support Division, November 1961.
2. *Leiner A. L., Notz W. A., Smith J. L. and Weinerger A.* Organizing a Network of computers to Meet Deadlines, Proceedings of the Eastern Joint Computer Conference, December 1957.
3. *Porter R. E.*, The RW-400—A New Polymorphic Data System, Datamation, 1960, v. 6, No (1), p. 8—14.

ПРИЛОЖЕНИЕ III

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ B6700 ФИРМЫ BURROUGHS CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система B6700 характеризуется фирмой Burroughs как большая система обработки информации, обладающая следующими основными особенностями:

Монолитные интегральные схемы.

Память, емкость которой может быть увеличена до 1 048 576 слов (по шесть знаков кода EBCDIC¹ в слове).

Три возможных значения времени цикла обращения к памяти: 1,5 мкс, 1,2 мкс и 500 нс.

Число периферийных устройств может достигать 256.

Система из трех процессоров ввода-вывода позволяет выполнять одновременно до 36 операций ввода-вывода.

Программное обеспечение режима дистанционной обработки данных и работы с файлами.

Дисковая файловая память емкостью больше 72 миллиардов байтов (восьмиразрядных символов).

Возможность организации различных конфигураций оборудования обеспечивается центральной системой обмена типа перекрестного коммутатора. При помощи этой системы группы модулей памяти, центральных процессоров и процессоров ввода-вывода объединяются в виде матрицы, как показано в левой части блок-схемы системы на рис. П.III.1. Эта схема иллюстрирует возможную степень модульности системы:

Один, два или три центральных процессора.

Любые комбинации модулей памяти емкостью 16К или 64К с максимальной общей емкостью 1 048 576 слов.

Один, два или три процессора ввода-вывода.

Каналы ввода-вывода, контроллеры периферийных устройств, периферийные устройства и линии связи, соединенные в соответствии с правилами, приведенными в разделе о процессоре ввода-вывода.

¹ EBCDIC (Extended Binary Coded Decimal Interchange Code) — расширенный двоично-кодированный десятичный код для обмена информацией.

Дальнейшее рассмотрение рис. П.III.1 показывает, что в системе предусмотрен независимый доступ к основной памяти от каждого периферийного устройства. Этот доступ осуществляется не под централизованным управлением с помощью контроллеров ввода-вывода или канального процессора, а при распределенном управлении, когда у каждого периферийного устройства имеется свое собственное устройство управления. Следует обратить внимание на то, что максимальная конфигурация с тремя процессорами ввода-вывода может быть расширена при помощи коммутаторов дисковых файлов и магнитных лент. Эти устройства управления и коммутаторы рассмотрены ниже в разделе о процессоре ввода-вывода.

Центральные процессоры при работе в мультипроцессорной системе логически идентичны. Это позволяет каждому из трех процессоров обрабатывать прерывания и выполнять рабочие функции, предусмотренные главной управляющей программой для одновременного выполнения нескольких заданий, независимо от рабочего состояния любого другого процессора. Центральные процессоры связаны с процессорами ввода-вывода посредством сканируемой шины, состоящей из 20 адресных проводов, 48 информационных проводов, одного провода проверки на четность и 11 проводов управления. Это позволяет процессорам вырабатывать команды ввода-вывода независимо от реальной конфигурации периферийного оборудования. Этот вопрос также рассмотрен в разделе о процессоре ввода-вывода.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР

Каждый центральный процессор состоит из двух основных частей, называемых функциональными ресурсами и операционными алгоритмами. Такая организация позволила реализовать несколько системных принципов, которые отличают систему В6700 от других мультипроцессорных систем:

1. Структура команд процессора основана на бесскочечной польской записи¹. Операции являются результатом выполнения процессором последовательности троек команд, состоящих из строки восьмиразрядных операторных «слов», которые управляют набором аппаратно реализованных операций.

¹ Вид логической записи последовательности арифметических операций, предложенный в 1929 г. польским математиком Я. Лукашевичем. В ней не используется никаких символов группирования, что значительно упрощает обработку записи транслятором. Например, выражение $Z = A(B + C)$ в польской записи выглядит как $BC + A \times Z =$ (запись читается слева направо). — *Прим. перев.*

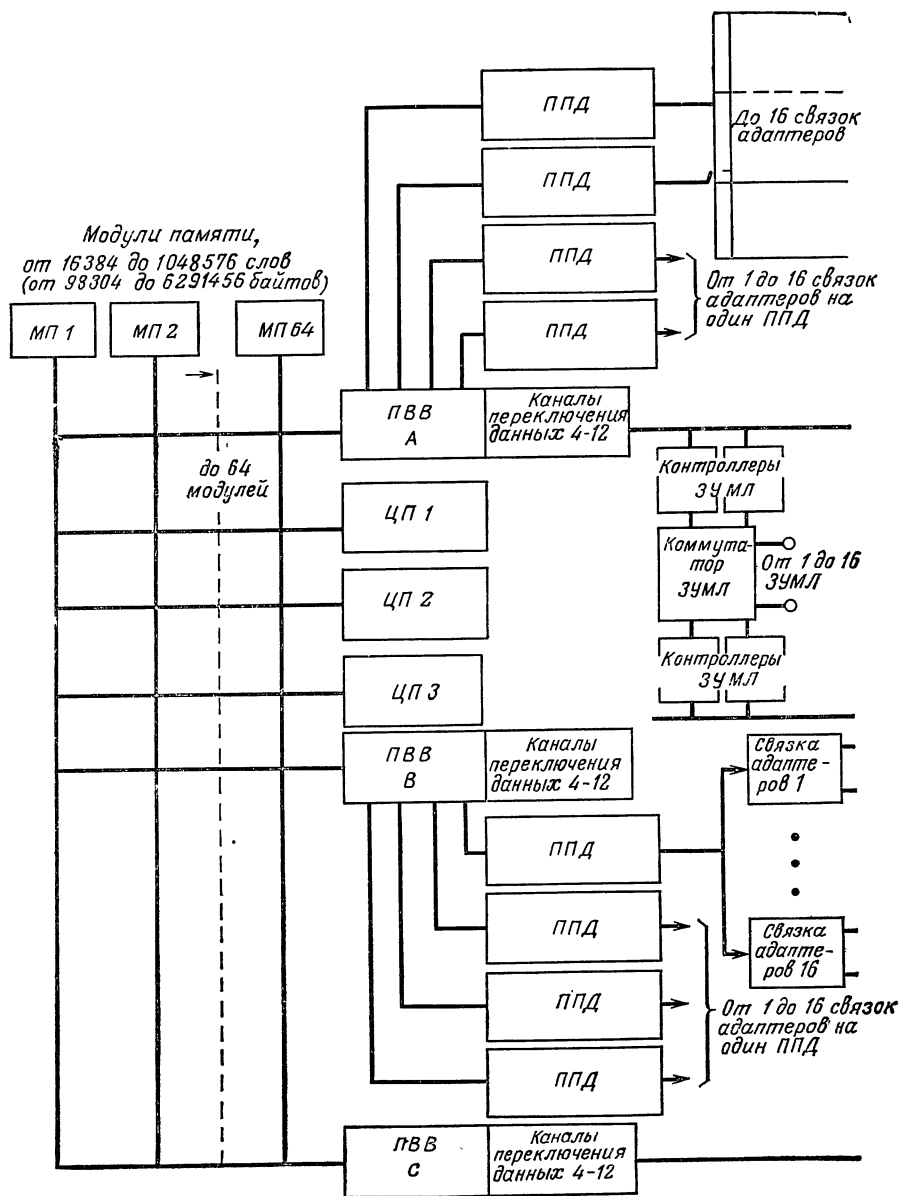
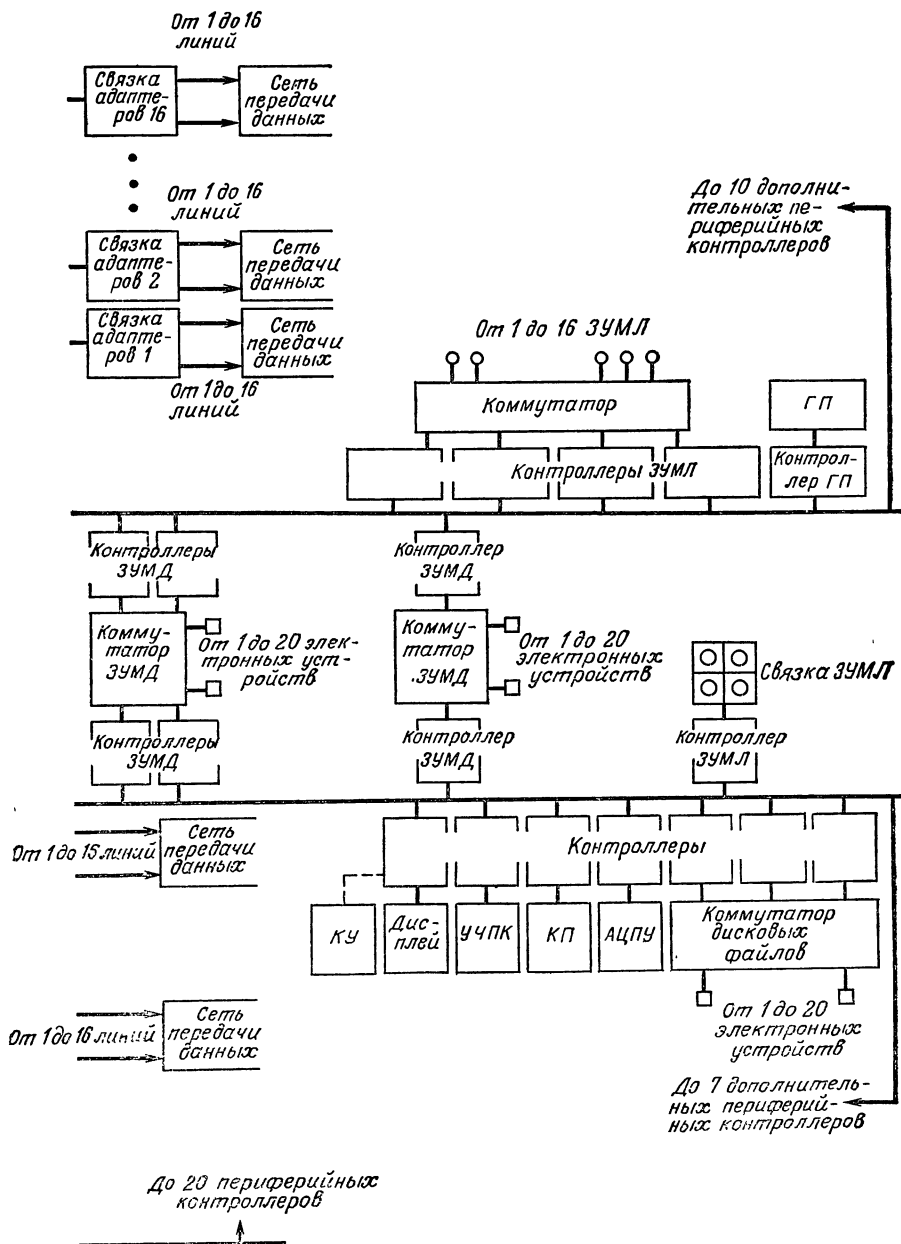


Рис. П.ИИ.1. Блок-схема

МП — модуль памяти, ПВВ — процессор ввода-вывода, ППД — процессор передачи дан
устройство на магнитных дисках, ГП — графопостроитель, УЧПК — устройство чтения
устройство, КУ — кла



системы В 6700.

ных, ЗУМЛ — запоминающее устройство на магнитных лентах, ЗУМД — запоминающее перфокарт, КП — карточный перфоратор, АЦПУ — алфавитно-цифровое печатающее внешнее устройство.

2. Все программы пользователя и операционной системы записываются на процедурно-ориентированных языках высокого уровня, и все операции по компилированию дают в результате программы в объектном коде, которые могут выполняться в режиме совпадения как несколько заданий, называемых «процессами».

3. Неограниченное и в основном автоматическое рекурсивное использование подпрограмм несколькими процессами.

4. Все массивы данных, сегменты программ и строки символов сначала адресуются при помощи дескрипторных слов, которые приводят в действие аппаратный механизм сегментации программ и выполнения таких программ, у которых команды записаны в основной памяти не последовательно.

Стековый метод хранения информации реализован в виде области памяти, подчиняющейся правилу «последний пришел — первый обслужен» и выделенной каждому процессу для программных переменных и для обращения к данным, связанным с этим процессом. Кроме того, он обеспечивает временное хранение информации об истории данных и процесса, например такой, как объявленный — текущий приоритет процесса, разрешение на использование программы, динамическая оценка объема памяти, процедурная глубина (вложенность) процесса и время ожидания активации. Более подробное описание стекового механизма приведено ниже после ознакомления с некоторыми аппаратными регистрами, обеспечивающими управление стеками.

Информационное слово процессора состоит из 51 разряда. Первые три разряда используются для записи признаков различных типов слов (рис. П. III.2). В остальных 48 разрядах записываются данные или управляющая информация. Признаки позволяют адресоваться к данным как к операндам, причем процессор определяет, из одного или двух слов состоит операнд. Они также обеспечивают защиту памяти до уровня основного машинного информационного слова, а не до уровня сегментов памяти фиксированного минимального размера, как, например, базовые регистры или ключи защиты. Двадцатичетырехразрядное адресное поле в дескрипторных словах допускает прямую адресацию 1 048 576 слов.

В процессоре имеется контроллер памяти, в отличие от некоторых других мультипроцессорных систем, в которых управление приоритетом доступа к памяти осуществляется отдельным блоком, таким, как блок распределения сигналов в системе RCA 215 или коммутатор памяти — процессор в системе Hughes 4400. Это не означает, что передача данных по каналу ввода-вывода в какой-либо модуль памяти зависит от надзора со стороны центрального процессора; однако в каждом контроллере памяти процессора должен быть предусмотрен интерфейс с процессором ввода-вывода, содержащий сканируемую шину для асинхронного обмена информацией между процессорами и мультиплексорами. Все это, по-видимому, указыва-

ИНФОРМАЦИОННЫЕ СЛОВА

000	Порядок	Мантисса	Операнд с нормальной точностью
010	Порядок	Мантисса (старшие разряды)	Операнд с двойной точностью (первое слово)
010	Порядок (старшие разряды)	Мантисса (младшие разряды)	Операнд с двойной точностью (второе слово)
	← 6 разрядов →	→ ————— 39 разрядов ————— →	

ДЕСКРИПТОРНЫЕ СЛОВА

101	↑	Длина	Адрес	Дескриптор данных (DD)
	Разряд присутствия			
011	↓	Длина	Адрес	Дескриптор сегмента (SD)
		← 20 разрядов →	← 20 разрядов →	

СПЕЦИАЛЬНЫЕ УПРАВЛЯЮЩИЕ СЛОВА

011	Номер стека	Смещение	↑	DF	Управляющее слово состояния стека (MS CW)
		Лексикографический уровень			
111	Номер стека	Индекс много-слога		Адресная пара	Управляющее слово программы (PCW)
011		Индекс много-слога		Адресная пара	Управляющее слово возврата (RCW)
001				Адресная пара	Слово косвенного обращения (IRW)
001	Номер стека	Смещение		Дельта	Упакованное слово косвенного обращения (SIRW)
011	Состояния FF	DS		DF	Управляющее слово верхушки стека (TOSCW)
	← 10 разрядов →	← 16 разрядов →			
	← 4 разряда →	← 20 разрядов →		← 14 разрядов →	

Рис. П.ИИ.2. Форматы слов системы В 6500.

ет на дублирование аппаратуры управления, а не на отсутствие модульности, если сравнивать эту систему с другими. Помимо интерфейса мультиплексора ввода-вывода контроллер памяти содержит адресный блок, в который входят 32 адресных регистра объектных

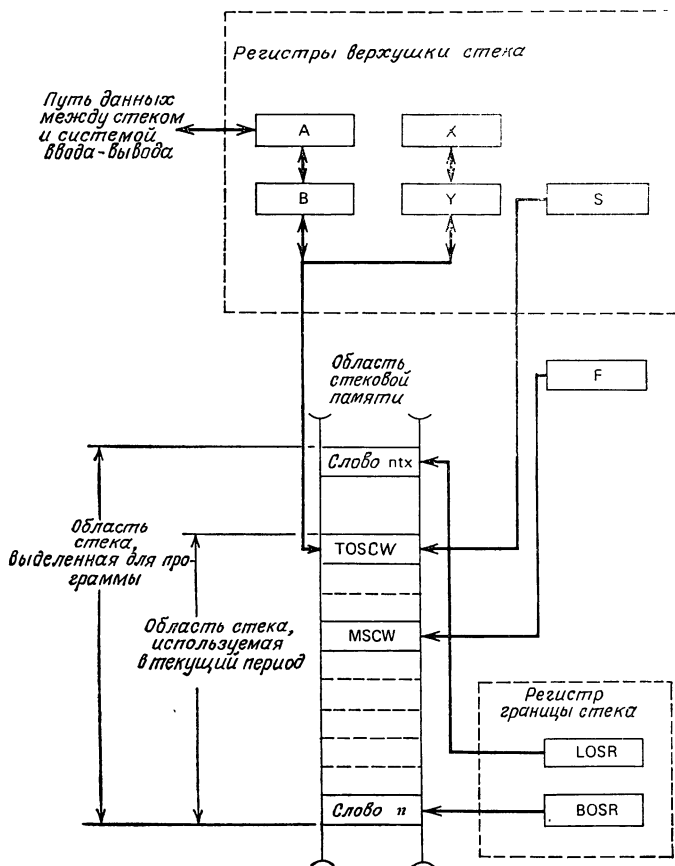


Рис. П.И.3. Регистры верхушки стека и границ стека.

программ (D — регистры) и 16 базовых и индексных регистров, используемых логическими схемами системного оператора и контроллером последовательности программ. В число этих двадцатирядных регистров входят 5 регистров, используемых контроллером стековой памяти для автоматической перестройки стека по требованию системного оператора и других функциональных устройств управления. Эти регистры в известной степени осуществляют аппа-

ратное управление стековым механизмом; они охарактеризованы в следующей таблице:

Обозначение регистра	Функция регистра
S	Регистр — указатель стека, дающий адрес верхушки стека.
SNR	Регистр номера стека, используемый для определения места дескриптора данных для текущего активного стека.
LOSR	Регистр границы стека — один из регистров защиты стековой памяти, определяющий верхний адрес текущего активного стека.
BOSR	Регистр базы стека — один из регистров защиты стековой памяти, определяющий адрес базы текущего активного стека.
F	Указатель верхнего управляющего слова состояния стека в текущем активном стеке, информирующий о том, какая процедура процесса выполняется.

Регистры данных A, B, X и Y называются регистрами верхушки стека; они образуют путь, по которому операнды входят в стек или выходят из него в другие регистры данных в соответствии с указаниями текущего оператора. Этот механизм показан на рис. П.III.3, где буквами MSCW обозначено последнее управляющее слово состояния стека, входящее в стек и определяющее текущую выполняемую процедуру.

Процедура относительной адресации, использованная в стековой памяти системы В6700, соответствует условиям адресации алгольной процедуры, которые устанавливаются в том случае, когда структура программы указана программистом; эту процедуру можно назвать лексикографическим упорядочением процедурных блоков (рис. П. III.4).

Спаренные адресные поля управляющих слов процессора (рис. П. III.2) содержат следующие две составляющие: лексикографический уровень переменной и значение индекса, служащее для определения местоположения данной конкретной переменной в пределах некоторого заданного лексикографического уровня. Так как последовательность процедурных блоков в процессе выполнения программы остается неизменной, к переменной всегда можно обратиться через посредство этой адресной пары. Следует заметить, что управляющее слово состояния стека, входящее в стек, содержит поле лексикографического уровня, указывающее лексикографический уровень входящей процедуры. Если задана структура алгольной программы, показанная на рис. П.III.4, то механизм текущей адресации для определенного процесса будет таким, как представленный на рис. П.III.5, в предположении, что выполняется процедура В. D-регистры указывают адрес базы каждого сегмента одного

адресного уровня; таким образом, локальные переменные всех процедур адресуются относительно адресов, хранящихся в D-регистрах.

—Begin	Лексикографический уровень 2
real V1;	LL=2, 1=2
real V2;	LL=2, 1=3
procedure A;	LL=2, 1=4
—Begin	Лексикографический уровень 3
real V3;	LL=3, 1=2
procedure B;	LL=3, 1=3
—Begin	Лексикографический уровень 4
V3←3;	
V1←V3;	
—End;	
B	
—End;	
Procedure C	LL=2, 1=5
—Begin	Лексикографический уровень 3
real V4;	LL=3, 1=2
procedure D;	LL=3, 1=3
—Begin	Лексикографический уровень 4
real V5;	LL=4, 1=2
V4←4;	
V5←5;	
A;	
V2 V4;	
—End;	
D;	
—End;	
C;	
—End;	

Алгольная программа с указанием лексикографической структуры

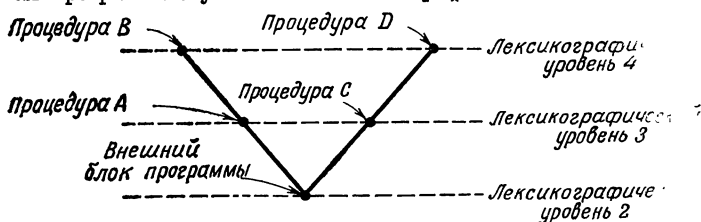


Рис. П.III.4. Дерево адресации при выполнении алгольной программы.

Упакованное слово косвенного обращения (рис. П.III.2) служит для адресации непосредственно за пределами текущих процедур и допускает обращение к переменным в других стеках. Так как все стеки для процессов, известных системе, организованы в древовидную структуру, причем ствол дерева содержит некоторые глобальные переменные операционной системы и дескрипторы сег-

ментов, описывающие различные процедуры внутри операционной системы, код программы с повторной входимостью, вырабатываемый компиляторами, может быть одновременно использован несколькими заданиями, для которых выполняется одна и та же программа.

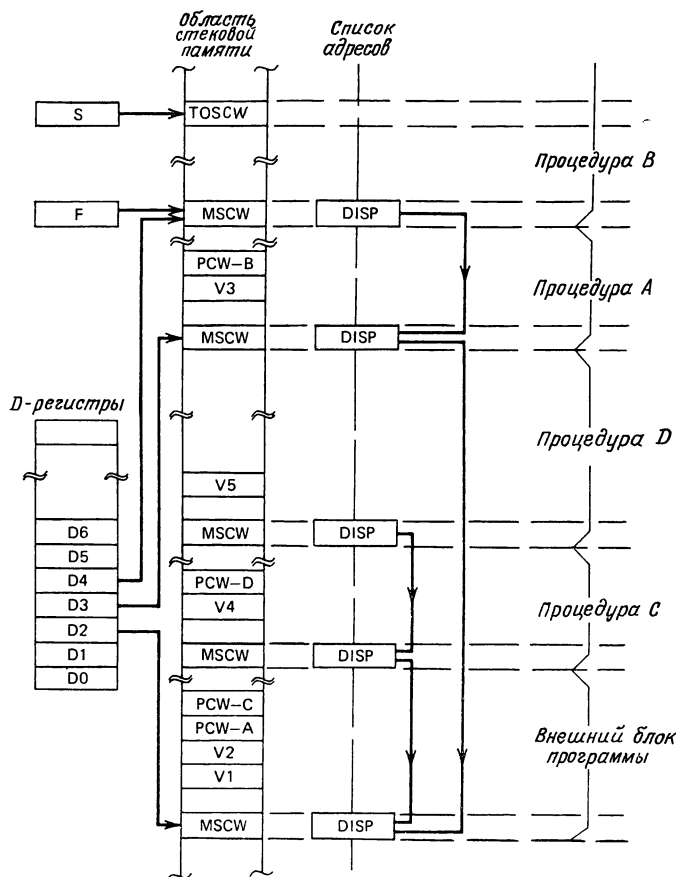


Рис. П.III.5. Схема D-регистров, указывающих условия текущей адресации.

Такая древовидная структура стеков показана на рис. П.III.6, где стековый вектор дескрипторов данных указывает местоположение стековых ветвей дерева. Индикаторные регистры D0 и D1, соответствующие лексикографическим уровням 0 и 1, указывают управляющие слова состояния стека в стеке операционной системы и в стеке сегментных дескрипторов, определяя тем самым размеры и местоположение всех кодов программ, которые предстоит выполнять. Именно отделение немодифицируемого кода программы от

переменных, связанных с процессом, при помощи аппаратно реализованного стекового механизма обеспечивает благоприятные условия для работы с процедурно-ориентированными языками.

В системе предусмотрены три класса прерываний, включающие 25 различных типов прерываний, распознаваемых процессором, в том

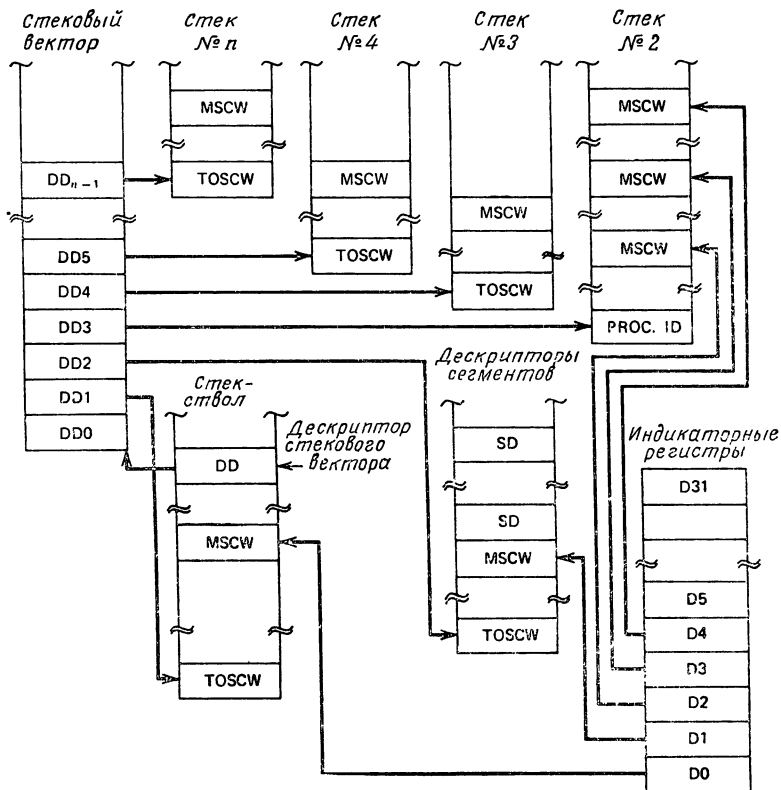


Рис. П.И.И.6. Многостековая организация.

числе 4 типа прерываний, связанных с работой процессора ввода-вывода. Имеются следующие классы прерываний: прерывания, определяемые слогами (командами), аварийные прерывания и внешние прерывания. 12 прерываний первого класса аналогичны программным прерываниям других вычислительных систем, таким, например, как прерывания из-за переполнения порядка, нарушения защиты памяти, неверного операнда. Однако среди них имеются два типа прерываний, используемые для реализации механизма разбиения на страницы и сегменты переменной длины. Они позволяют

передавать сегменты программ или массивы данных в основную память из вспомогательной памяти в тех случаях, когда при обращении к ним во время выполнения процесса они отсутствуют в основной памяти. Эти прерывания осуществляются:

По разряду присутствия. Такое прерывание происходит при обращении процессора к данным, к дескриптору или к дескриптору сегмента, когда разряд присутствия имеет нулевое значение. Это указывает на то, что в памяти нет никаких обращений к дескрипторам. Разряд присутствия в дескрипторных словах показан на рис. П.III.2.

По сегментированному массиву. Такое прерывание указывает на то, что при записи в память массива данных операционная система разбила его строки на сегменты, и после этого была сделана попытка выполнить индексацию за границами текущего сегмента. Процедура обработки прерывания должна в этом случае заменить текущий сегмент нужным сегментом, если таковой существует, и продолжать выполнение процесса.

Имеется 6 типов аварийных прерываний, которые примерно аналогичны прерываниям по ошибкам в других системах. Таковы, например, прерывания при незавершении выполнения команды, по сигналу системы контроля на четность, при неправильном операторе и (или) операнде. Кроме того, поскольку в процессоре содержится контроллер памяти (рис. П.III.1), предусмотрено прерывание по сигналу системы контроля на четность процессора ввода-вывода — внутреннее процессорное прерывание, указывающее на прием данных от процессора ввода-вывода с ошибкой.

Имеется также 7 групп внешних прерываний, в том числе прерывания по сигналу таймера, по переполнению стека процесса, по отказам в системе межпроцессорной связи, по работе процессора ввода-вывода. Прерывание по переполнению стека процесса предусмотрено на тот случай, если процесс так сложен, что для его реализации требуется динамическое расширение размеров стека.

На рис. П.III.7 представлена блок-схема процессора системы В6700, которая наглядно иллюстрирует все сведения, изложенные выше. На ней ясно видны взаимосвязи между основными составными частями процессора (функциональными ресурсами и операционными алгоритмами) и потоки команд и данных внутри процессора.

ОСНОВНАЯ ПАМЯТЬ

Основная память системы В6700 организована так, что любой модуль памяти может принимать или получать информацию от всех процессоров по любой из шести информационных шин (рис. П.III.8).

Модули памяти проверяют каждое слово, появляющееся на шине, чтобы определить, какому из модулей оно адресовано. Если оно адресуется к какому-либо определенному модулю, то он устанавливает связь для приема слова. Это устраняет необходимость в центральном управлении связью для передачи слова к нужному

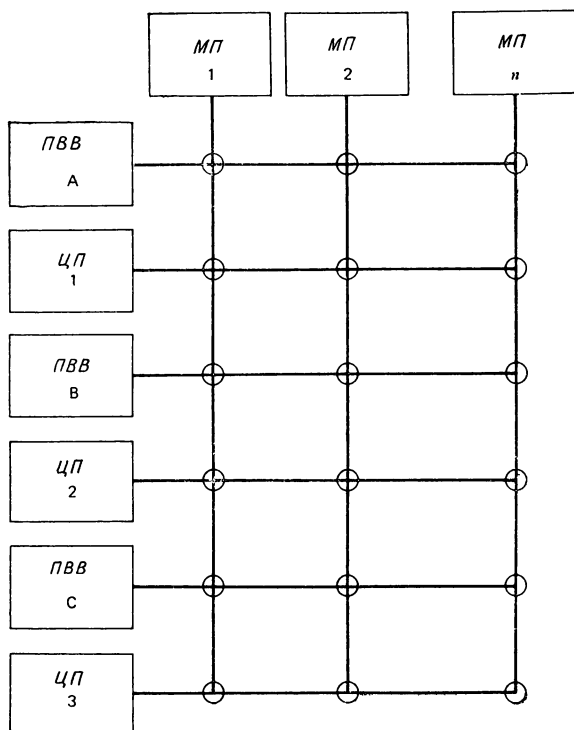


Рис. П.III.8. Организация памяти.

МП—модуль памяти, ПВВ—процессор ввода-вывода

модулю. Через 200 нс после начала цикла обращения к памяти модуль разрешает доступ, еще через 200 нс слово поступает на шину и еще через 200 нс оно находится в регистре центрального процессора или процессора ввода-вывода. Каждый модуль памяти работает независимо от других модулей памяти.

Информационные и управляющие сигналы передаются по 80-проводной двусторонней шине параллельным кодом, показанным ниже.
20-разрядный адрес:

6 разрядов для модулей с номерами от 0 до 63.

14 разрядов для адресов памяти от 0 до 16 383.

6 управляющих разрядов
(чтение, запись, занятость и т. д.).
52 информационных разряда.

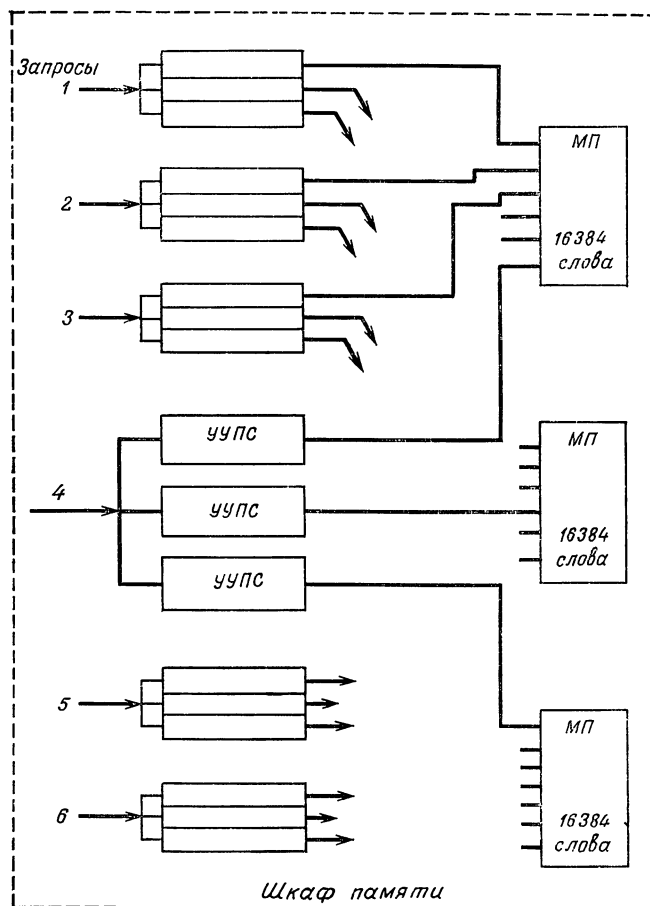


Рис. П.ИИ.9. Выбор модуля памяти.

УУПС — устройство управления перекрестными соединениями, МП — модуль памяти.

Основная память системы В6700 может содержать от 1 до 64 модулей, каждый емкостью 16 384 слова. Хотя на схеме матрица связи расположена вне процессора и памяти, физически она размещается в шкафах памяти; она называется контроллером памяти.

В одном шкафу памяти можно поместить до трех модулей и один контроллер памяти (49 152 слова). Контроллер памяти может

реагировать на запрос о доступе к памяти от любого из шести запрашивающих устройств. Такими устройствами могут быть:

1. Центральные процессоры 1, 2 и 3.
2. Процессоры ввода-вывода А, В и С.
3. Устройство проверки памяти (подключаемое через один из процессоров ввода-вывода).
4. Логические схемы технической диагностики процессоров А или В (подключаемые через один из процессоров ввода-вывода).

Рассмотрим пример, когда один из процессоров запрашивает доступ к нулевому модулю памяти в нулевом шкафу. Процессор передает адрес и информацию на шины. Они воспринимаются всеми контроллерами памяти, но поступают только в нулевой модуль, так как адрес был декодирован только в этом шкафу. Таким образом, каждый контроллер памяти должен обладать способностью принимать адреса от шести различных запрашивающих устройств и подсоединять их к одному из трех модулей памяти. Это осуществляется устройством управления перекрестными соединениями, размещенным в контроллере памяти (рис. П.ІІІ.9). В каждом контроллере памяти имеется три группы устройств управления перекрестными соединениями для каждого из шести возможных запрашивающих устройств. Три запрашивающих устройства могут получить одновременный доступ в один шкаф, если только они адресуются к различным модулям памяти.

Приоритетная система, которая активируется перед срабатыванием устройств управления перекрестными соединениями, разрешает конфликты при адресации более чем одного запрашивающего устройства к одному и тому же модулю памяти. Модули памяти адресуются при помощи 20-разрядного кода, как показано на схеме дескрипторного слова на рис. П.ІІІ.2. Разряды 0—13 используются для выбора слова, а разряды 14—19 — для выбора модуля. Предусмотрены возможности чередования слов между двумя последовательно расположенными модулями памяти.

ПРОЦЕССОР ВВОДА-ВЫВОДА

В системе В6700 процессоры ввода-вывода и связанные с ними контроллеры периферийных устройств используются для передачи данных между модулями памяти и периферийными устройствами (рис. П.ІІІ.1).

Как только процессор ввода-вывода получает от какого-либо центрального процессора команду ввода-вывода, начинается передача данных независимо от центральных процессоров. К системе могут быть подключены один, два или три процессора ввода-вывода.

Любой из центральных процессоров может запускать операцию ввода-вывода в любом из процессоров ввода-вывода.

К каждому процессору ввода-вывода может быть подсоединено до 20 контроллеров периферийных устройств. Периферийным устройствам можно присвоить номера от 0 до 255. В процессор ввода-вывода входят также датчик времени и символьные трансляторы.

В процессорах ввода-вывода применяются каналы переключения данных для передачи данных и буферное устройство емкостью в одно слово для каждого контроллера, передающего данные. Каждый процессор ввода-вывода может содержать от 4 до 12 каналов. Эти каналы переключения данных являются «плавающими», так что любой свободный канал может быть использован для любого из устройств, подсоединенных к процессору ввода-вывода. Таким образом, каждый процессор ввода-вывода может производить одновременно до 12 операций над данными. Назначение каналов переключения данных выполняется процессором ввода-вывода. После запуска какой-либо операции назначение свободного канала для выполнения этой операции производится аппаратным способом. Каждый раз, когда какой-либо контроллер периферийных устройств посылает запрос, затрачивается время одного рабочего цикла процессора ввода-вывода (1,2 мкс). Во время каждого рабочего цикла между процессором ввода-вывода и контроллерами быстродействующих периферийных устройств (например, устройств записи на магнитную ленту или на диски) передается 16 битов (2 байта) информации. Все другие контроллеры передают 8 битов за цикл. В форматах командных слов ввода-вывода предусмотрена возможность ответа на запрос центрального процессора всех активных процессоров ввода-вывода или только одного определенного процессора. Типичные функции по вводу-выводу состоят в запросе о состоянии периферийного устройства, сообщении о готовности пути передачи данных к периферийному устройству, запуске операции ввода-вывода и сообщении об окончании операции.

Особый интерес представляет функция запроса о пути ввода-вывода, так как она увеличивает модульность и расширяет возможности реконфигурации. Реализация этой функции определяет, имеется ли путь (свободный контроллер и канал), по которому в рассматриваемом блоке может быть запущена операция ввода-вывода. Этот путь может быть проверен для всех процессоров ввода-вывода, для периферийного устройства, общего для всех процессоров ввода-вывода, или для отдельного процессора ввода-вывода. В ответе содержится информация о том, в каком процессоре (или процессорах) ввода-вывода имеется такой путь. Конфигурация периферийных устройств может изменяться, не оказывая влияния на исполнительные программы.

ВСПОМОГАТЕЛЬНАЯ ПАМЯТЬ

Массовая память реализована в виде дисковых файлов с отдельной головкой на каждую дорожку, имеющих среднее время обращения от 20 до 40 мс и емкость одного модуля от 10 млн. до 72 млрд. 8-разрядных байтов, и в виде пакетов дисков со средним временем обращения около 30 мс и емкостью одного модуля от 121 млн. до 1,6 млрд. байтов.

ДИСКОВЫЕ ФАЙЛЫ С ОТДЕЛЬНОЙ ГОЛОВКОЙ НА КАЖДУЮ ДОРОЖКУ

Можно сформулировать следующие правила образования различных конфигураций дисковых файлов системы В6700:

К одному электронному блоку дискового файла может быть подсоединено не больше пяти модулей дисковых файлов, что составляет до 100 млн. байтов памяти на один электронный блок.

Все дисковые модули, подсоединенные к одному электронному блоку, должны быть одного типа.

Один электронный блок может быть подсоединен непосредственно к одному контроллеру дисковых файлов.

Используя коммутатор дисковых файлов, можно подсоединять к контроллеру дисковых файлов по желанию от 1 до 20 электронных блоков.

Коммутатор дисковых файлов позволяет при помощи 1—4 контроллеров дисковых файлов обслуживать банк, состоящий не больше чем из 20 дисковых файлов.

Предусмотрен также оптимизатор дискового файла, который доводит среднее время доступа к дискам с фиксированной головкой на каждый тракт до 2—6 мс посредством предварительной подготовки положения диска относительно головок. Такой оптимизатор выстраивает в очередь запросы на доступ к дисковому файлу в соответствии с их положением на диске относительно считывающих-записывающих головок. После того как открыт путь данным, первым предоставляется доступ к запрашиваемой записи, находящейся ближе всего к считывающей головке. По мере вращения диска открывается доступ к другим записям в наиболее эффективной последовательности. Если система должна осуществлять большое число обращений к дисковому файлу за одну секунду, то этот метод позволяет уменьшить время ожидания и заменяет обычную программную очередь, обслуживаемую по правилу «первый пришел — первый обслужен».

ПАКЕТЫ ДИСКОВ

Применяются дисководы для пакетных дисков, подобные по конструкции дисководам IBM 3300 и обладающие емкостью 121 и 242 млн. байтов.

Контроллер пакетного дисковогода с одиночной выборкой может быть использован с 8 осями пакетов дисков (4 двойных привода) в конфигурации «1×8» или с двумя группами по 8 осей пакетов дисков (8 двойных приводов) в конфигурации «1×16». Выбор каждой группы определяется вариантом дескриптора ввода-вывода. Контроллер пакетного дисковогода с двойной выборкой может быть использован в конфигурации «2×8», в которой контроллер дисковогода содержит два внутренних устройства управления. Это позволяет процессору ввода-вывода одновременно выполнять две операции (два считывания, две записи или одно считывание и одну запись). Эта конфигурация может быть расширена до конфигурации «2×16».

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Главная управляющая программа системы B6700 может управлять выполнением заданий или процессов в мультипрограммном режиме при условии достаточного объема основной памяти. Эта программа составлена полностью на языке высокого уровня Эспол. Особое внимание уделено динамическому распределению памяти, кодированию объектных программ с повторной входимостью и параллельному выполнению независимых частей объектной программы. В сочетании с аппаратным стековым механизмом, описанным ранее, главная управляющая программа может реализовать все процессы, включая свою собственную резидентную исполнительную программу, так, как если бы они представляли собой одну большую алгольную программу, содержащую локальные и глобальные переменные, блоки и процедуры в рамках унифицированной программной структуры.

ДИСПЕТЧЕРИЗАЦИЯ И РАСПРЕДЕЛЕНИЕ РЕСУРСОВ

При распределении ресурсов, необходимом для эффективного планирования, используются «сегменты» программ на машинном языке (объектные программы), вырабатываемые всеми компиляторами, функционирующими под управлением главной управляющей программы. Длина программного сегмента изменяется в зависимости от логики программы и используемого языка. Сегментация алгольных программ основана на блочной структуре исходных программ — каждый блок исходной программы транслируется в сегмент программы на машинном языке. Кобольные программы сегментируются на уровне секций, если только программист не дает других указаний. Сегментация фортранных программ производится на уровне программ (подпрограмм или основных программ), и эти программы, если необходимо, подвергаются дальнейшей сегментации до опти-

мальных размеров сегмента. Кроме того, у программиста имеется возможность продолжать сегментацию объектных программ в предположении, что наименьшие полученные им сегменты можно использовать как переходные.

Главная управляющая программа планирует выполнение процессов в соответствии с алгоритмом динамического планирования, который решает приоритетные уравнения для выстраивания процессов в активную очередь, если все ресурсы системы, требуемые процессами, доступны, или в пассивную очередь, если текущих ресурсов недостаточно. При освобождении ресурсов (например, основной памяти или периферийных устройств) процесс перемещается из пассивной очереди в активную и выбирается для выполнения, как только его приоритет становится наивысшим, согласно алгоритму, который учитывает следующие факторы:

1. Объявленный приоритет пользователя.
2. Время ожидания в активной очереди.
3. Общее время ожидания в пассивной и активной очередях.
4. Общее время ожидания во всех системных очередях.
5. Время центрального процессора, требуемое для реализации процесса.
6. Текущее значение используемого объема основной памяти.
7. Разность между планируемым временем начала реализации процесса и текущим временем.

Каждому из этих факторов приписан свой весовой коэффициент, который может изменяться пользователем в соответствии с его требованиями. Например, если пользователь желает, чтобы задание, связанное с ресурсами процессора, не заняло слишком много имеющегося времени процессора, он может придать фактору 5 соответствующий вес. Этот вес в дальнейшем может быть изменен при помощи сообщения, передаваемого с пульта главной управляющей программе, если расчеты пользователя оказались неверными. Пользователь может также ввести совершенно другой алгоритм планирования, заменив приоритетные уравнения в главной управляющей программе; для облегчения модификации эти уравнения записаны в отличие от остальной программы в машинных кодах.

Когда какой-либо процесс окончательно выбран для выполнения, в основной памяти находится только первый сегмент этого процесса. Каждый выполняемый процесс управляется стеком процесса, как было подробно описано выше. Это позволяет обрабатывать прерывания по биту присутствия каждый раз, когда текущий сегмент сообщает об отсутствии объектного сегмента или массива данных. Такой подход обеспечивает реализацию одного из видов разбиения на переменные страницы, осуществляя динамическое распределение основной памяти для объектных сегментов и массивов данных переменных размеров по мере обращения к ним во время выполнения программы. Так как объектные сегменты одного процесса не обяза-

тельно должны располагаться в памяти последовательно один за другим, память распределяется с максимальной плотностью.

Процесс может планироваться также другим процессом, использующим специальную конструкцию ZIP, при этом указывается местоположение управляющих карт, которые определяют задание. Предусмотрена также конструкция EXECUTE, которая может быть вызвана некоторым процессом, чтобы загрузить и запустить в работу объектный файл. Эти конструкции представляют собой по существу макрокоманды вызова высокого уровня, включаемые в исходные программы пользователя вместе с соответствующими аргументами, определяющими вводимое задание.

СВЯЗЬ ОПЕРАТОРА С СИСТЕМОЙ

Связь оператора с главной управляющей программой осуществляется при помощи модифицированного дисплейного терминала BIDS, который позволяет создать конфигурацию из нескольких (до 7) дисплеев, и при помощи постоянно подключенного к системе устройства чтения перфокарт, в которое можно вводить системные управляющие карты.

В нормальном режиме на экране дисплея отображается сводная таблица, указывающая состояние каждого задания, известного главной управляющей программе. Если задание находится в работе, то на экране указывается в сокращенном виде индекс смеси (номер задания), название задания, приоритет, код транслятора, обозначение используемого блока основной памяти, значение использованного времени процессора и состояние задания (начало выполнения, выполнение, конец выполнения, прерванное выполнение). По требованию оператора вместо сводной таблицы на экран могут быть выведены различные другие таблицы:

Таблица расписания, указывающая состояние каждого задания, ожидающего выполнения.

Таблица периферийных устройств, указывающая состояние и режим работы каждого периферийного устройства, входящего в конфигурацию.

Таблица идентификаторов файлов, указывающая, какие файлы используются на заданном периферийном устройстве и какие задания в данное время используют это устройство.

Таблица каталога дисковой памяти, которая содержит все идентификаторы файлов, помещенные в каталог диска и входящие в состав группы, указанной во входном сообщении.

Таблица заданий, содержащая подробную информацию о каждом выполняемом (активном) задании,— управляющие карты, относящиеся к этому заданию, и связь между физическими устройствами и именами файлов, используемых при выполнении задания.

Кроме этих таблиц, определен набор сообщений, позволяющий оператору снимать задания с обработки, повышать или понижать приоритет заданий, задерживать выполнение заданий и т. д.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

ПЛАНИРОВАНИЕ ВВОДА-ВЫВОДА И ПЕРЕДАЧА ДАННЫХ

Все операции ввода-вывода выполняются системой подпрограмм ввода-вывода, входящей в состав главной управляющей программы. Подпрограммы ввода-вывода ориентированы либо на файлы, либо на периферийные устройства, в зависимости от того, на каком логическом уровне пользователь желает иметь доступ к периферийным устройствам. Чтобы выполнить операции ввода-вывода по расписанию, выработанному центральной процедурой ввода-вывода главной управляющей программы, трансляторы, генерирующие объектные программы, обращаются к набору внутренних подпрограмм главной управляющей программы или к общим процедурам, доступным всем активным процессам на основе повторной входимости.

Эта центральная процедура, носящая название `IOREQUEST`, принимает все запросы на операции ввода-вывода и планирует их в соответствии со следующими общими требованиями:

1. Каждая операция ввода-вывода должна быть связана со специальной буферной зоной определенной программы, так как подпрограмма `IOREQUEST` планирует выполнение операций ввода-вывода для всех программ.

2. Упор должен быть сделан на построение последовательности операций ввода-вывода и на быстрое возвращение управления вызывающей программе, даже если запрос на операцию ввода-вывода обращен к устройству, которое не может быть запущено.

3. Процедура планирования должна быть способна блокировать буфер ввода-вывода, а после завершения операции ввода-вывода разблокировать его. Эта операция блокирования должна быть доступна программисту и должна разрешать запуск и останов программы только в том случае, когда программа попытается обрабатывать данные в буфере, на которые был сделан, но еще не выполнен запрос по вводу-выводу.

С планированием ввода-вывода связаны два типа очередей — очередь ожидания устройства (одна для каждого периферийного устройства) и очередь ожидания канала (одна для каждого канала ввода-вывода). Эти очереди подчиняются правилу «первый пришел — первый обслужен», причем очередь ожидания устройства убывает, как только образуется путь к устройству (по какому-либо каналу). Как упоминалось раньше, при рассмотрении модуля ввода-вывода используется плавающая связь каналов с процессором ввода-вывода,

так что любой свободный канал может быть использован для передачи данных к любому из устройств, подсоединенных к процессору ввода-вывода. Стратегия планирования и передачи данных, заложенная в главную управляющую программу, должна использовать это свойство с максимальной эффективностью.

После запуска подпрограммы IOREQUEST другой процедурой главной управляющей программы (INITIATEIO) канал выполняет ввод-вывод, как указано выше, и по завершении этой операции генерирует дескриптор результата. После окончания операции происходит прерывание, которое запускает процедуру проверки ввода-вывода на отсутствие ошибок, входящую в состав главной управляющей программы. Если ошибок не обнаружено и имеются дополнительные запросы на ввод-вывод, то новые операции ввода-вывода запускаются в последнем использованном канале, причем сначала принимаются запросы из очереди ожидания канала, а затем снимаются запросы из очереди ожидания устройства и передаются в очередь ожидания канала для оптимизации работы канала при заданной работе устройства.

РАБОТА С ФАЙЛАМИ

Все трансляторы системы B6500 позволяют использовать «символические файлы», или логические массивы данных; обращение к ним производится по имени логического файла, которое может содержать до 17 символов. Для установления связи между именем символического файла и именем реального (физического) файла в заголовке файла предусматриваются специальные управляющие операторы уравнивания меток. Система управления файлами обеспечивает следующие возможности:

1. Может быть использована перманентная файловая система, управляемая каталогом диска с древовидной структурой, который указывает связь имен символических файлов с файловыми каталогами пользователей, и тремя классами защиты; эти классы контролируют доступ к файлам в промежутки времени, когда файлы открыты, или доступ к конкретным записям.

2. Может быть реализован принцип «тома», допускающий размещение нескольких файлов на одном физическом носителе, таком, как магнитная лента или пакет дисков.

3. Возможна реализация поколений файлов, допускающая обращение к «генеалогии» файлов при помощи тех же символических имен, уточненных номерами поколений.

4. Файлы могут быть помечены в соответствии со стандартом, рекомендованным Американским национальным институтом стандартов (ANSI), или со стандартом данной системы, чтобы все помеченные файлы автоматически приписывались к объектным программам, запрашивающим символические файлы.

5. Возможны файловые записи пяти типов: с нормальной точностью, с двойной точностью, в упакованном десятичном коде, в коде BCL (двоично-кодированные литералы), в коде EBCDIC (расширенный двоично-кодированный десятичный код для обмена информацией) и в стандартном американском коде для обмена информацией ASCII.

6. Возможны форматы файлов восьми типов — формат с фиксированной длиной, двоичные или десятичные числа с переменной длиной, неопределенный формат, формат с переменной длиной, определяемой в записи, фортранные записи и др.

7. Атрибуты файлов во время выполнения программы состояются из управляющих операторов уравнения меток, сообщений дискового каталога и главной управляющей программы, образуя блок файловой информации переменной длины (до 39 слов).

Примером имени символического файла может служить имя D/E/F, где D — идентификатор каталога файла, E — идентификатор тома и F — идентификатор файла. Пункт 4 следует расширить, если принять во внимание, что главная управляющая программа распознает до 12 различных типов меток, включая метки ANSI систем В5700 и В3700.

ПРИЛОЖЕНИЕ IV

СИСТЕМЫ CDC 6500, CYBER-70, МОДЕЛИ 72-2X, 73-2X И 74-2X, ФИРМЫ CONTROL DATA CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Фирма Control Data Corporation создала четыре системы, которые удовлетворяют тому определению мультипроцессорной системы, которое было сформулировано в начале этой книги. В 1966 г. появилась система CDC 6500, за которой в 1972 г. последовало семейство систем CYBER-70. В этом семействе имеется 7 основных моделей, но из них только три содержат по два центральных процессора — модели 72 и 73 (они почти идентичны системе CDC 6500) и модель 74 (она по существу является системой из двух ЭВМ CDC 6600). Все четыре модели совместимы на уровне языка ассемблера и, если не считать некоторых потерь в эффективности, совместимы также на уровне языка компилятора.

Один путь связи между процессорами, имеющийся в семействе CYBER-70, а именно распределительный путь передачи данных, в системе CDC 6500 отсутствует. На рис. П.IV.1 представлена схема организации, одинаковая для всех четырех систем. Системы отличаются друг от друга только некоторыми особенностями центрального процессора. Скорости выполнения команд в центральном процессоре различны для всех трех моделей. Центральный процессор модели 74 по своей организации, а следовательно, и функционированию отличается от центрального процессора модели 72 и системы CDC 6500.

Каждая система содержит два больших центральных процессора. Обращение к ним возможно через быстродействующую центральную память и группу из 10, 14, 17 или 20 периферийных процессоров. Каждый из таких периферийных процессоров снабжен автономной памятью и может выполнять программы независимо от других периферийных или центральных процессоров. При помощи специальной процедуры прерывания и связи через центральную память эти периферийные процессоры управляют центральными процессорами. Между собой они связаны через центральную память, каналы ввода-вывода и регистр блокировки (которого нет в ЭВМ 6500).

Основные функции центральных процессоров вычислительной системы состоят в выполнении арифметических и логических опе-

раций; так как их два, имеется возможность одновременно выполнять две программы, хранящиеся в центральной памяти. Под управлением операционной системы SCOPE можно в режиме совпадения с использованием мультипрограммирования выполнять до 15 заданий, хранящихся в центральной памяти.

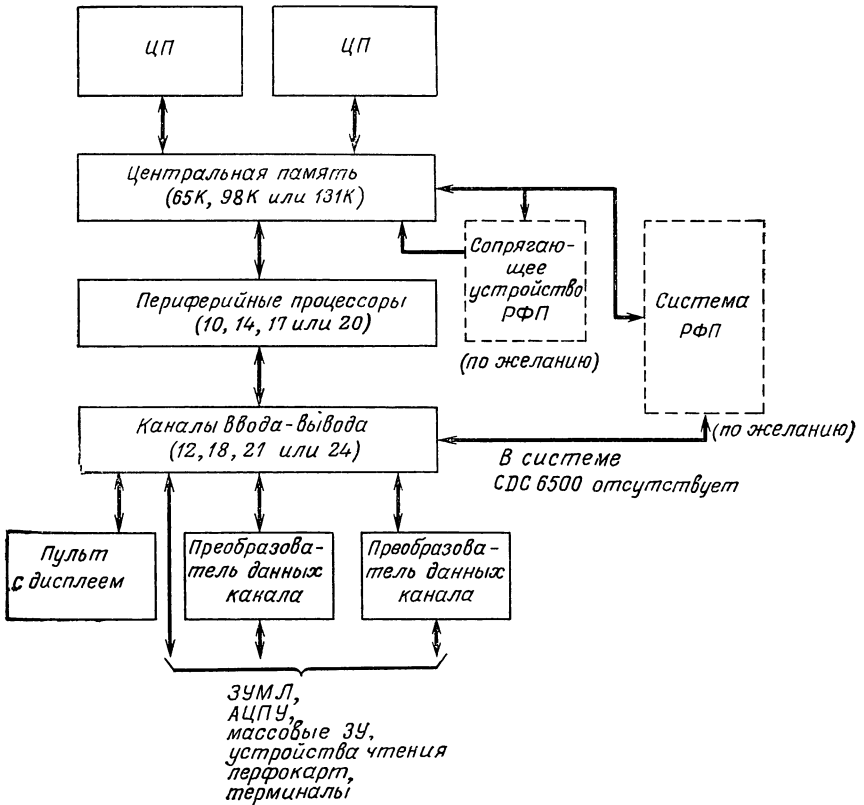


Рис. П.IV.1. Организация мультипроцессорной системы фирмы CDC.
РФП — расширенная ферритовая память.

Функции периферийных и управляющих процессоров вычислительной системы состоят в реализации операций ввода-вывода для программ, выполняемых в центральных процессорах, а также в организации исходных данных (операндов, адресов, констант, сведений о длине программ, относительных начальных адресов, способов вывода) и их хранении в центральной памяти. Для реализации функций операционной системы SCOPE используются два периферийных процессора: периферийный процессор 0 работает как мо-

нитель вычислительной системы и как управляющий процессор; периферийный процессор 1 — как процессор связи между дисплейным пультом оператора и системой. Остальные периферийные процессоры могут быть использованы для обслуживания стандартных периферийных устройств (устройств записи на магнитную ленту, устройств чтения перфокарт, карточных перфораторов, построено печатающих устройств, дисковых файлов и т. д.), систем связи, дисплеев и графопостроителей.

Все периферийные процессоры обмениваются данными с периферийными устройствами по 12 независимым дуплексным каналам. Канал имеет емкость 12 разрядов (плюс контрольный разряд); каждый из каналов может быть подсоединен к одному или нескольким периферийным устройствам. Однако в любой данный момент времени каждый канал может быть использован только одним устройством, хотя все 12 каналов могут быть одновременно связаны с разными устройствами.

Сигналы датчика реального времени передаются по отдельному каналу. Датчик времени начинает работать, как только подключается питание, и работает непрерывно, причем его показания нельзя установить заранее или изменить. Такой датчик может быть при необходимости использован для определения времени выполнения программы или для указания времени суток.

Центральная память доступна для всех процессоров системы и может состоять из 65К, 98К или 131К 60-разрядных слов, заключенных в 16, 24 или 32 куба (по 4096 слов в каждом). Кубы памяти логически независимы друг от друга, и последовательные адреса программ размещены в различных кубах, что уменьшает вероятность конфликта между несколькими процессорами при обращении к памяти, а также позволяет вводить кубы в работу поочередно через каждые 100 нс, обеспечивая тем самым весьма высокое быстродействие памяти в целом: через каждые 100 нс может заноситься в память или извлекаться из нее одно 60-разрядное слово.

Управление потоком данных в системе осуществляется периферийным процессором 0, который содержит в своей памяти системный монитор. Этот периферийный процессор имеет доступ к резидентным программам других периферийных процессоров и к активным программам в центральных процессорах через ячейки связи центральной памяти или же, как в моделях семейства CYBER-70, через регистры блокировки. Работа аппаратной части при функционировании системы иллюстрируется схемой, представленной на рис. П.IV.2.

Все обращения к центральной памяти от всех частей системы (от центральных и периферийных процессоров) поступают в общий блок модификации адресов (линии В и С на рис. П.IV.2), а отсюда пересылаются ко всем кубам центральной памяти (линия Е). Если по переданному адресу должно размещаться информационное слово, то это слово пересылается в центральный

К периферийным устройствам

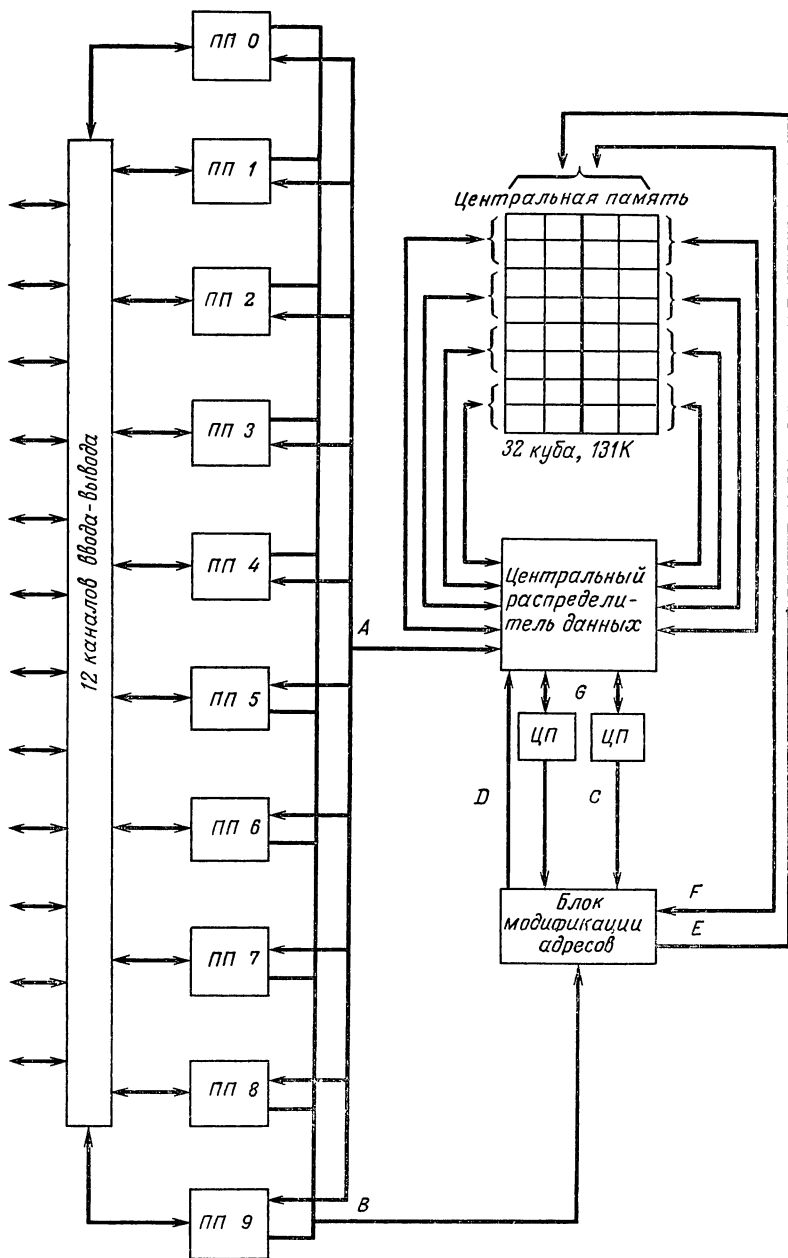


Рис. П.И.2. Схема распределения потоков данных в мультипроцессорной системе.

ПП — периферийный процессор.

распределитель данных по дуплексной линии емкостью 24 разряда (линия А), состоящей из 12 проводов для записи и 12 проводов для считывания. Блок модификации адресов принимает адреса из различных источников на приоритетной основе с максимальной скоростью 1 адрес за 100 нс. Адрес посылается во все кубы, и затребованный куб, если он свободен, принимает этот адрес и сообщает центральному распределителю данных, что нужный куб свободен; затем соответствующее информационное слово посылается в центральный распределитель данных или принимается от него по одной из восьми линий емкостью 60 разрядов, каждая из которых обслуживает четырехкубовый сегмент памяти. Обращение к центральной памяти со стороны центральных процессоров производится точно так же, как со стороны периферийных процессоров в том смысле, что адрес также направляется в блок модификации адресов (линия С), а затем информационные слова проходят через центральный распределитель данных в направлении к центральному процессору или от него (линия G).

Блок модификации адресов при помощи бункерного механизма сохраняет каждый адрес, который он посылает в центральную память, и в случае, если он не будет принят из-за конфликта за доступ к какому-либо кубу памяти, снова посылает его (и снова запоминает) на приоритетной основе. Такой процесс посылки — сохранения адреса повторяется до тех пор, пока адрес не будет принят; после этого адрес выпадает из бункера и начинается распределение записываемого или считываемого информационного слова. Адрес, который хранится в бункере (т. е. не принятый раньше), имеет самый высокий приоритет при посылке адресов в центральную память. Далее следуют по порядку центральные процессоры и периферийные процессоры.

Через распределитель данных, общий для всех процессоров, проходят все информационные слова в направлении к центральной памяти и от нее. Группа буферных регистров в распределителе данных обеспечивает кратковременное хранение информационных слов, подлежащих записи в центральную память, но задержанных из-за отказа в приеме адресов при конфликте на входе куба. Хотя каждая группа из четырех кубов, как уже упоминалось выше, общается с распределителем данных по отдельным каналам записи и считывания емкостью 60 разрядов, в каждый момент времени по этим каналам может передаваться только одно слово. Однако слова могут передаваться между распределителем данных и центральной памятью или между источником адреса и распределителем данных с интервалом 100 нс.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР

Центральный процессор представляет собой быстродействующее арифметическое устройство, общающееся только с центральной памятью. Он изолирован от периферийных процессоров и может свободно выполнять вычисления без каких-либо ограничений со стороны ввода-вывода.

Каждый процессор функционально состоит из арифметического устройства и устройства управления. В арифметическом устройстве содержатся все логические схемы, необходимые для выполнения арифметических, логических операций и операций управления. Устройство управления осуществляет управление всеми арифметическими операциями и обеспечивает сопряжение между арифметическим устройством и центральной памятью. Оно выполняет также поиск команд, подготовку адресов, защиту памяти, а также поиск и хранение данных.

Центральный процессор модели 74 по своей организации отличается от центральных процессоров остальных моделей семейства CYBER. По существу он идентичен центральному процессору системы CDC 6600. Он состоит из 10 отдельных функциональных процессоров, работающих параллельно. Десять арифметико-логических устройств выполняют арифметические, логические операции и операции управления (сложение, сложение длинных слов, двоичное приращение, сдвиг, двоичное умножение, деление, булевские операции и условный переход). Устройство управления осуществляет управление арифметическими операциями и обеспечивает сопряжение между функциональными блоками и центральной памятью.

Арифметические устройства системы CDC 6500 и систем CYBER, модели 72 и 73, выполняют команды последовательно и с небольшой степенью совпадения. Время выполнения команд уменьшено за счет использования перекрытия, при котором во время реализации одной команды считывается следующая за ней команда.

В набор команд центрального процессора входят арифметические и логические операции, операции индексации и переходов, а также операции нормальной и двойной точности с плавающей запятой, по желанию с округлением и нормализацией. Формат данных состоит из 48-разрядной мантиссы, 11-разрядного порядка и одnorазрядного знака мантиссы. Возможны арифметические операции с фиксированной запятой — сложение и вычитание полных 60-разрядных слов.

Для обнаружения и обработки прерываний используются как аппаратные, так и программные средства. Специальная команда прерывания, вырабатываемая периферийным процессором, инициирует действия аппаратуры центрального процессора по прерыванию

текущей выполняемой программы и по замене ее программой, параметры которой определены в специальной команде прерывания. Эта же команда используется для запуска центрального процессора после его останова. Аппаратура центрального процессора обеспечивает три типа остановов по ошибкам:

При переполнении адреса.

При переполнении операнда.

При неопределенном результате.

Останов любого из этих типов может быть предусмотрен по желанию. После прерывания центрального процессора специальной командой прерывания происходит следующий четырехэтапный процесс, предназначенный для сохранения прерванной программы в таком состоянии, чтобы можно было продолжать ее реализацию:

1. После извлечения всех команд из текущего командного слова, хранящегося в стековой памяти команд, извлечение команд прерывается.

2. В регистре адреса команды устанавливается адрес следующего командного слова, подлежащего выполнению.

3. Выполняются извлеченные команды.

4. Параметры для двух программ взаимно меняются.

Теперь при последующем переключении программ можно вновь войти в прерванную программу в точке прерывания.

СВОДКА ХАРАКТЕРИСТИК ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

По 24 рабочих регистра на каждый центральный процессор (для уменьшения числа обращений к памяти):

восемь 60-разрядных регистров операнда,
восемь 18-разрядных регистров адреса,
восемь 18-разрядных регистров приращений.

Скорость выполнения команд (в млн. команд/с)	Модель 72	Модель 73	Модель 74	Модель 6500
Один процессор	0,9	1,2	3,0	—
Два процессора	1,5	2,0	3,7	2,0

Целочисленное умножение (кроме модели 6500).

Специальная команда прерывания.

Команды move (пересылка) и compare (сравнение) (кроме моделей 6500 и 74).

ПЕРИФЕРИЙНЫЕ ПРОЦЕССОРЫ

Все периферийные процессоры одинаковы и работают независимо и одновременно как вычислительные машины с запоминаемыми программами. Возможно одновременное выполнение нескольких программ или же несколько процессоров могут решать одну задачу, требующую участия нескольких устройств ввода-вывода, а также центральной памяти и центрального процессора (процессоров).

Периферийные процессоры работают в качестве ЭВМ системного управления и процессоров ввода-вывода. Это позволяет центральному процессору выполнять вычисления, в то время как периферийные процессоры выполняют более медленные операции ввода-вывода и супервизорные операции.

Каждый процессор снабжен памятью с произвольным доступом, емкостью 4096 12-разрядных слов и временем цикла 1000 нс. Время выполнения команд процессора зависит от времени цикла памяти.

Каждый периферийный процессор может выполнять приблизительно 500 000 команд в 1 с. Все периферийные процессоры разделяют общие блоки операций сложения-вычитания, ввода-вывода, передачи данных к (от) центральной памяти и других необходимых операций. Это означает, что программы внутри группы периферийных процессоров выполняются по методу мультиплексирования с использованием принципа разделения времени. Мультиплексор состоит из 10-позиционного барабана, в котором хранится информация о командах в каждой из 10 программ, и общего блока управления командами. 10 программных команд перемещаются последовательно вокруг барабана, и каждая команда предоставляется блоку управления. Таким образом выполняются одновременно до 10 программ, причем каждая программа выполняется шагами через 1 мкс. Команды в барабане интерпретируются во время критических интервалов, благодаря чему к моменту входа команды в блок управления необходимая информация уже находится в этом блоке. Следовательно, обращение к памяти осуществляется с упреждением, и информационное слово уже готово к использованию в момент прихода команды. Аналогично интерпретация команд осуществляется до того, как они поступают в блок управления, так что к их приходу уже устанавливаются пути передачи сигналов управления.

Для команд ввода-вывода или для связи с центральной памятью блок управления за один цикл передает одно 12-разрядное слово к (от) периферийной памяти. Таким образом, для передачи блока данных требуется несколько оборотов вокруг барабана.

Каждый периферийный процессор обменивается с центральной памятью данными, упакованными в блоки по N слов. Пять последовательных 12-разрядных слов периферийного процессора собираются в одно 60-разрядное слово центральной памяти, и, наоборот,

одно 60-разрядное слово центральной памяти разбивается на пять 12-разрядных слов для пересылки в пять последовательных ячеек памяти периферийного процессора. Отдельные пути передачи данных к центральной памяти со сборкой (записью) и разборкой (чтением) разделяются всеми периферийными процессорами. До четырех периферийных процессоров могут записывать информацию в центральную память и одновременно до четырех периферийных процессоров могут считывать информацию из центральной памяти.

В периферийных процессорах имеется аппаратная сигнализация, указывающая различные состояния каналов передачи данных, например «полностью занят — свободен», «активный — пассивный» и т. д. При помощи команд периферийных процессоров можно также проверить состояние каналов и аппаратуры, например получить сообщения «готов», «не работает», «конец файла» и т. д.

В периферийных процессорах моделей CYBER-70 имеются регистры блокировки, состоящие из 64 или 128 разрядов. Они доступны всем периферийным процессорам при организации взаимных связей; благодаря этому сильно сокращается число обращений к памяти, требуемых для выполнения этой функции.

Сводка характеристик периферийного процессора:

Возможны конфигурации из 10, 14, 17 или 20 периферийных процессоров.

Емкость ферритовой памяти 4096 12-разрядных слов.

Все каналы являются общими для всех процессоров.

Максимальная скорость передачи данных в одном канале — 1 слово/мкс.

Все каналы могут работать одновременно.

Все каналы дуплексные, с емкостью 12 разрядов.

Датчик реального времени работает с периодом 4096 мс.

Арифметические действия с фиксированной запятой.

Доступ к центральной памяти с разделением времени.

Связь между процессорами через регистр блокировки (в системе CDC 6500 отсутствует).

Приоритетность доступа к центральной памяти.

Возможность адресации до $131K$ слов (прямой, косвенной и с модификацией).

ЦЕНТРАЛЬНАЯ ПАМЯТЬ

Центральная память состоит из кубов ферритовых запоминающих устройств емкостью по 4096 60-разрядных слов. Полный цикл для одного куба составляет 1 мкс. Кубы организованы так, что последовательные адреса размещены в различных кубах, благодаря чему центральная память может работать с более высокой скоростью, чем допускает основной цикл. Максимальная скорость передачи данных составляет одно 60-ти разрядное слово за 100 нс.

Имеется пять путей доступа к центральной памяти:

Два пути между центральным процессором и центральной памятью.

Путь между расширенной ферритовой памятью и центральной памятью.

Путь между одной или двумя группами периферийных процессоров и центральной памятью.

Центральная память содержит управляющую часть, которая обеспечивает обслуживание каждого из этих путей доступа на приоритетной основе, выстраивает в очереди запросы на доступ, если это необходимо, и разрешает конфликты при обращении к памяти.

Приоритетный механизм доступа к центральной памяти позволяет определенным периферийным процессорам иметь преимущество перед другими, неприоритетными периферийными процессорами при считывании или записи в центральную память. Это обеспечивает также максимальную (для заданной конфигурации) скорость передачи между центральной памятью и расширенной ферритовой памятью.

Защита памяти основана на установлении верхней и нижней границ для каждой рабочей программы и включена в каждую программу как ее часть. Все обращения центрального процессора к центральной памяти за новыми командами или для считывания и записи данных выполняются относительно опорного адреса, который определяет нижнюю границу программы в центральной памяти. Изменяя опорный адрес, можно легко перемещать программы в пределах центральной памяти.

Сводка характеристик центральной памяти:

Память состоит из логически независимых кубов по 4096 слов каждый, работающих в многофазном режиме (память содержит не больше 32 кубов).

Скорость передачи составляет до 1 слова за 100 нс при многофазном режиме

Возможны конфигурации общей емкостью 65K (18 модулей), 98K (24 модуля) или 131K (32 модуля).

Аппаратура обеспечивает прием адресов от 12 процессоров через интервалы 100 нс, разрешение конфликтов при обращении к центральной памяти и предварительный просмотр команд во время обращения.

СИСТЕМА ВВОДА-ВЫВОДА

Все периферийные процессоры сообщаются с периферийными устройствами и друг с другом через независимые дуплексные каналы ввода-вывода. Число каналов зависит от числа периферийных процессоров в системе. Все каналы имеют емкость 12 разрядов (плюс

контрольный разряд), и каждый из них может соединяться с одним или с большим числом периферийных устройств. Один канал в один и тот же момент времени может быть использован только одним периферийным устройством, однако все каналы могут работать одновременно. Данные передаются в систему или из нее 12-разрядными словами с максимальной скоростью 1 слово/мкс. К одному каналу ввода-вывода можно подсоединить до восьми различных периферийных устройств (контроллеры устройств записи на магнитную ленту, контроллеры устройств чтения перфокарт и карточных перфораторов и т. д.).

РАСШИРЕННАЯ ФЕРРИТОВАЯ ПАМЯТЬ

Подсистема расширенной ферритовой памяти, которая может использоваться в системе по желанию, состоит из собственно расширенной памяти, ее контроллера и (только для моделей CYBER-70) одного или большего числа распределительных путей передачи данных, которые подключаются к каналам ввода-вывода.

Расширенная ферритовая память состоит из кубов по 125 952 60-разрядных слов. В 488-разрядном физическом слове расширенной памяти содержится восемь 60-разрядных слов. Каждое 60-разрядное слово имеет соответствующий разряд контроля на четность в слове расширенной памяти. Полный цикл обращения для одного куба составляет 3,2 мкс на одно 488-разрядное слово расширенной памяти.

Кубы расширенной памяти организованы так, что передача информации производится последовательными записями по восемь слов, поступающими из различных кубов. Такая организация вместе с широким каналом доступа (8 слов) позволяет производить передачу информации в расширенную память и из нее с очень большой скоростью. После начального обращения, длящегося 3,2 мкс, расширенная память может передавать информацию со скоростью одно 60-разрядное слово за 100 нс. Это дает минимальную скорость 600 млн. битов в 1 с.

Расширенная ферритовая память поставляется комплектами от одного куба (125 952 слова) до 16 кубов (2 015 232 слова). Высокая скорость передачи и малое время доступа делают расширенную память идеальной для таких применений, как буферизация между центральной памятью и вращающимися массовыми запоминающими устройствами (например, быстродействующими устройствами с переменным обращением к разным накопителям), хранение больших массивов данных и хранение часто используемых программ и системных подпрограмм.

Имеются два пути доступа к расширенной ферритовой памяти:

Путь между центральной памятью и расширенной памятью.

Распределительный путь передачи данных между каналом (каналами) ввода-вывода и расширенной памятью (только в моделях семейства CYBER-70).

Распределительный путь передачи данных обеспечивает непосредственное движение потока данных между расширенной памятью и периферийными процессорами в моделях CYBER-70. Он позволяет быстрый доступ периферийного процессора к данным в расширенной памяти через канал ввода-вывода и значительно уменьшает плотность потока данных через центральную память. Это в свою очередь сокращает число конфликтов при обращении к центральной памяти и повышает эффективность работы операционной системы.

МАССОВАЯ, ИЛИ ВСПОМОГАТЕЛЬНАЯ, ПАМЯТЬ

В дополнение к расширенной ферритовой памяти, рассмотренной выше, в системе применяется несколько типов запоминающих устройств на магнитных дисках и барабанах. Диапазон емкости этих устройств — от 8 млн. до 167 млн. символов на одну ось, а среднее время доступа — от 17 до 75 мс.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

На рис. П.IV.3 показана схема связей и управления системы CYBER-70.

Составляющие части операционной системы SCOPE распределены между центральной памятью, памятью периферийных процессоров, расширенной ферритовой памятью и системной памятью на дисках. Один из периферийных процессоров содержит монитор и непрерывно управляет системой¹. Второй периферийный процессор, находящийся под управлением монитора, постоянно обслуживает клавиатуры и дисплей на пульте оператора. Остальные процессоры могут назначаться для выполнения операционных функций по мере необходимости.

Монитор распределяет работу между периферийными процессорами, общаясь с ними через область центральной памяти, специально выделенную для этой цели. В периферийных процессах хранятся подпрограммы, которые постоянно опрашивают эту область. При

¹ В версии 3.4 операционной системы SCOPE, которая имеет наибольшее распространение и является основой для дальнейших модификаций, предусмотрено, что некоторые функции выполняются дополнительным монитором — резидентом центрального процессора. В число этих функций входит отчасти распределение памяти и управление заданиями. В более ранних версиях операционной системы все супервизорные функции выполнялись одним монитором — резидентом периферийного процессора.

появлении в ней запроса периферийный процессор оповещает об этом монитор и возвращается к выполнению подпрограммы ожидания до тех пор, пока не поступит новый запрос. Подобное использование независимых периферийных процессоров обеспечивает минимальную загрузку центральной памяти и центрального процессора, повышая тем самым эффективность системы.

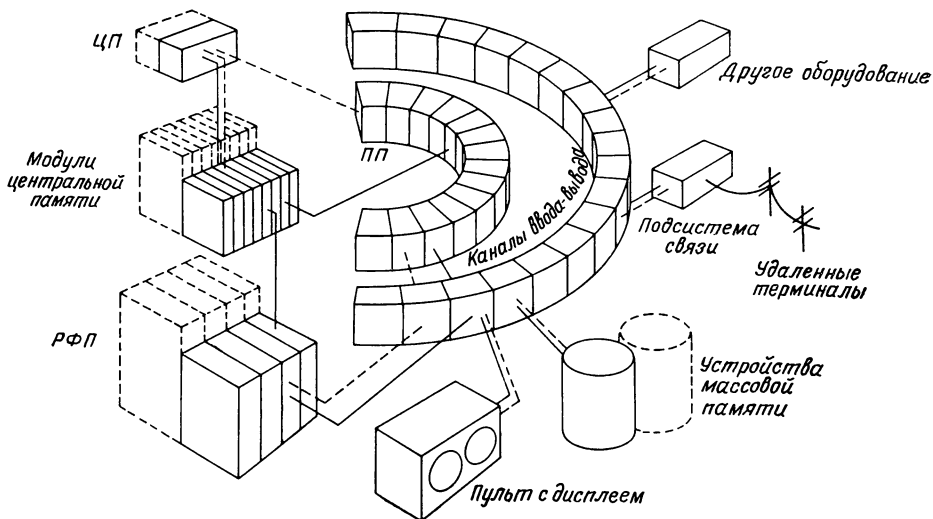


Рис. П.IV.3. Потоки данных и управляющих сигналов.

РФП — расширенная ферритовая память, ПП — периферийный процессор.
— — — передача данных; — — — передача управляющих сигналов.

УПРАВЛЕНИЕ ЗАДАНИЯМИ¹

В центральной памяти могут одновременно размещаться до 15 отдельных программ. Каждой из них приписан номер контрольной точки и длина поля в центральной памяти. Если задание активно, т. е. находится в центральной памяти, то область контрольной точки в резидентной части центральной памяти содержит следующую информацию: имя задания, длину задания, начальный адрес в центральной памяти, суммарное использованное время, номера устройств ввода-вывода, назначенных для этого задания, и управляющие опе-

¹ Программы управления заданиями и планирования центральных процессоров претерпели большие изменения в версии 3.4 операционной системы SCOPE. Приведенное здесь описание фактически относится к более ранним версиям.

раторы. Область контрольной точки содержит также обменный блок информации — массив из 16 слов, включающий содержимое всех регистров центрального процессора, используемых при выполнении программы. Эта информация необходима для запуска или возобновления выполнения программы.

Задания, закрепленные за контрольными точками и ожидающие выполнения, помещаются в стек в порядке их приоритета. В верхушке стека находится задание, использующее центральный процессор. Если происходит прерывание (например, когда ожидается завершение выполнения функций периферийного процессора), верхушкой стека становится следующее задание, а первое задание временно выходит из стека. Когда оно вновь возвращается в стек, то задание, использующее центральный процессор, и все задания, находящиеся под ним, смещаются вниз.

Задание состоит из одной или нескольких программ, которым предшествуют управляющие карты, указывающие имя задания, приоритет, предельные значения времени выполнения, инструкции оператора и другую необходимую информацию. Операционная система SCOPE начинает обработку с чтения первой управляющей карты. Она переписывает файл задания в дисковое запоминающее устройство и заносит имя задания в список входных файлов. Если контрольная точка доступна и имеется требуемый объем свободной памяти, то задание передается к контрольной точке и начинается выполнение задания.

Блоки центральной памяти, приписанные контрольным точкам, занимают в центральной памяти места, определяемые относительно номеров соответствующих контрольных точек. Когда задания в контрольных точках запрашивают память или освобождают ее, блоки памяти, приписанные контрольным точкам, при необходимости перемещаются вверх или вниз, с тем чтобы каждое задание занимало непрерывный блок памяти. Такие перемещения возможны благодаря тому, что все обращения к центральной памяти производятся относительно опорного адреса.

Все операции по планированию заданий и распределению ресурсов унифицированы. Задания от всех источников (при пакетной, дистанционной и диалоговой обработке) выстраиваются в очереди, соответствующие разбиению на пять классов. Управление каждой очередью определяется установочными параметрами, указывающими начальный приоритет, степень старения, максимальный приоритет, квантовый интервал времени при передаче задания в контрольную точку и квантовый приоритет (приоритет класса задания, когда оно находится в контрольной точке). Сказанное относится только к версии 3.4 операционной системы SCOPE.

В системе предусмотрены средства контроля со стороны оператора, необходимые для регулирования системного планирования в соответствии с текущим потоком заданий.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Операционная система SCOPE является файло-ориентированной — вся информация в системе рассматривается как файлы или части файлов. Имеются следующие активные файлы, т. е. файлы, доступные для системы в любой момент времени:

Все задания, ожидающие выполнения (каждое задание является файлом); они составляют стек заданий, или входную очередь.

Выходные файлы, ожидающие вывода на печать, перфорацию и т. д.

Задания (файлы), находящиеся в данный момент в той или иной стадии выполнения.

Местные файлы пользователей, часто используемые при выполнении заданий.

Файлы пользователей можно объявить постоянными. Для таких файлов имя, местонахождение и пароли допуска, назначаемые пользователям, хранятся в каталоге. Допуск к постоянным файлам или обновление информации в них разрешаются только тем пользователям, которые могут предъявить соответствующий пароль. При необходимости можно сохранить копию первоначального файла.

Местные файлы, или постоянные файлы пользователей, могут быть определены как файлы с последовательным или с произвольным доступом. Обращение к файлам с произвольным доступом осуществляется при помощи индекса, вырабатываемого стандартными подпрограммами операционной системы SCOPE или одной из подпрограмм пользователя. Подсистема связи INTERCOM, находящаяся под управлением операционной системы SCOPE, обеспечивает общение пользователя с центральной вычислительной системой в режиме диалога при помощи удаленных терминалов. Эта подсистема связи позволяет также пользователю включать свои задания в пакетные очереди операционной системы SCOPE и получать результаты выполнения этих заданий. Подсистема связи INTERCOM управляет потоком данных между удаленными терминалами и центральной вычислительной системой. Она содержит все подпрограммы, необходимые для взаимодействия пользователя с операционной системой SCOPE.

ПРИЛОЖЕНИЕ V

СИСТЕМЫ DEC 1055 И 1077 ФИРМЫ DIGITAL EQUIPMENT CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система 10 фирмы Digital Equipment Corporation (сокращенное обозначение DECsystem-10) может быть однопроцессорной или двухпроцессорной, содержащей первичный и вторичный процессоры. Имеются две модели двухпроцессорных систем — модульные системы 1055 и 1077 (рис. П.V.1).

В двухпроцессорной системе каждый процессор выполняет программы пользователей, диспетчирует свою работу и обрабатывает программные прерывания. Помимо этих задач, первичный процессор управляет всеми устройствами ввода-вывода и обрабатывает все обращения к операционной системе. Он завершает работы, которые не смог закончить вторичный процессор из-за обращения к операционной системе. Оба процессора подсоединены к одной и той же памяти и используют одну и ту же копию операционной системы. Этим достигается экономия ферритовой памяти по сравнению с мультипроцессорными системами, в которых каждый процессор имеет собственную копию операционной системы. Основной целью двухпроцессорной модели системы 10 является получение большей вычислительной мощности, чем в однопроцессорной модели той же системы. Это означает, что с добавлением второго процессора можно за одно и то же время выполнять большее число программ пользователей. Если же число этих программ не увеличилось, то введение второго процессора уменьшает время, требуемое для выполнения имеющихся программ.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Система приоритетных прерываний центрального процессора КА10 системы 1055 имеет семь уровней прерываний для устройств, подключенных к шине ввода-вывода. Вся система приоритетных прерываний является программируемой. При помощи программного управления можно связать любое число устройств с любым уровнем, причем отдельные уровни или вся система приоритетных прерыва-

Сводка основных характеристик процессора

	Система	
	1055	1077
Тип процессора	KA10	KI10
Число аппаратно реализованных команд	366	378
Аппаратная реализация операций с плавающей запятой и с двойной точностью	Нет	Есть
Предварительный просмотр команд	Нет	Есть
Накапливающие сумматоры	16	4×16
Индексные регистры	15	4×15
Время обработки прерывания	6 мкс	3 мкс
Максимальная задержка прерывания	40 мкс	10 мкс

ний может быть выключена, а затем снова включена. Прерывания могут запрашиваться на любом уровне.

В соответствии с исполнительной управляющей логикой процессор KA10 может работать в одном из следующих трех режимов: а) в исполнительном режиме, когда могут выполняться все команды, а перемещение программ не допускается; б) в режиме пользователя, когда некоторые команды запрещены (например, команды ввода-вывода), а перемещение программ и защита памяти допускаются; в) в режиме пользователя с вводом-выводом, когда все команды разрешены и допускаются перемещение программ и защита памяти.

Центральный процессор KI10, используемый в системе 1077, обладает примерно вдвое большим быстродействием, чем процессор KA10. Это достигается в результате применения другой архитектуры, более быстродействующих схем, более сложного сумматора, усовершенствованных алгоритмов и логики предварительного просмотра команд, которая позволяет вызывать следующую команду во время выполнения текущей.

Центральный процессор KI10 работает в одном из двух режимов: в режиме пользователя или в исполнительном режиме. В каждом из этих режимов имеется два подрежима: а) открытый и закрытый подрежимы в режиме пользователя; б) супервизорный подрежим и подрежим ядра в исполнительном режиме.

Рабочие программы выполняются в режиме пользователя. В этом режиме все программы имеют возможность использовать память емкостью до 256К слов. Разрешены все команды, за исключением тех, которые нарушают обработку программ других пользователей или целостность системы. Программа открытого подрежима может перейти в закрытый подрежим только в результате передачи ее в ячейки, имеющие команды ENTRY. Программа в закрытом подрежиме может считывать и записывать информацию (если эти опе-

рации разрешены), выполнять действия и обращаться в любой адрес, который обозначен как открытый. Закрытый подрежим позволяет вместе с программами и данными пользователя вводить в систему запатентованное программное обеспечение, но исключает возможность вносить в него изменения или копировать его. Это обеспечивает прямое взаимодействие между пользователем и запатентованным программным обеспечением по существу без дополнительных затрат.

Прерывание процесса KI10 заставляет процессор и прерывающее устройство немедленно начать одно из нескольких возможных действий. В ответ на сигнал разрешения прерывания от процессора устройство может выработать 36-разрядное слово, включающее 18-разрядный адрес, 12-разрядный элемент данных, 3-разрядный уровень прерываний и 3-разрядную функцию. После этого процессор выполняет одно из следующих трех действий:

Выполняет команду, соответствующую указанному 18-разрядному адресу.

Передает слово в адресуемую ячейку или из нее.

Прибавляет к адресу 12-разрядное число со знаком.

Периферийные устройства, не снабженные декодирующей логикой, выполняют прерывание и передачу управления (как в случае процессора KA10) одной из стандартных ячеек прерывания.

ЦЕНТРАЛЬНАЯ ПАМЯТЬ

Время цикла обращения к памяти как для системы 1055, так и для системы 1077 равно 1 мкс для одного 36-разрядного слова. В системе 1055 могут быть операнды с длиной, равной половине слова или полному слову. Система 1077 может, кроме того, обрабатывать двойные слова.

Запоминающие устройства обеих систем могут иметь двух- или четырехкратное чередование адресов. В системе 1077 предусмотрено также управление перекрытием памяти и возможность разбиения на страницы. В системе 1055 нет механизма виртуальной памяти с разбиением на страницы, но имеются регистры перемещения программ и защиты.

В системе 1077 ферритовой памятью управляет система разбиения на страницы процессора KI10. Эта система позволяет программам пользователей работать с эффективным адресным полем емкостью до 256K слов. Это поле сегментировано на 512 страниц по 512 смежных слов каждая, причем страницы не обязательно должны быть физически смежными в ферритовой памяти.

Процессор KI10 обеспечивает также отображение адреса памяти из эффективного адресного поля в физическое адресное поле посредством замены старших разрядов эффективного адреса. Такое отобра-

жение обеспечивает доступ ко всему физическому полю памяти, которое в 16 раз больше эффективного адресного поля (эффективное адресное поле программы имеет емкость 256К 18-разрядных слов, а физическое поле — 4096К 22-разрядных слов). При отображении памяти используется страничная таблица.

ВВОД-ВЫВОД

Сводка основных характеристик

	Система	
	1055	1077
Скорость работы медленного канала ввода-вывода (мультиплексный канал)	222К слов/с (1110К байтов/с)	370К слов/с (1850К байтов/с)
Скорость работы быстрого канала ввода-вывода (4 селекторных канала по 8 уровней каждый)	4000К слов/с (20 000К байтов/с)	4000К слов/с (20 000К байтов/с)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

Резидентная операционная система составлена из нескольких отдельных и до некоторой степени независимых частей, или подпрограмм. Некоторые из этих подпрограмм имеют циклический характер и повторяются при каждом системном контрольном прерывании; это необходимо для подтверждения, что каждый пользователь вычислительной системы получает затребованное им обслуживание. К этим циклическим подпрограммам относятся:

1. Процессор команд, или дешифратор.
2. Планировщик.
3. Обменник.

Дешифратор команд интерпретирует команды, набранные пользователем на пульте, и пересылает их в соответствующие системные программы или подпрограммы. Планировщик решает, какую программу пользователя следует пропустить во время интервала между прерываниями по сигналу датчика времени, распределяет разделяемые системные ресурсы, сохраняет и восстанавливает условия, необходимые для возобновления выполнения программы, прерванной датчиком времени. Обменник переносит программы пользователя между вспомогательной памятью (обычно на дисках или на барабане) и центральной памятью, после того как принято решение, какие задания должны находиться в центральной памяти, но от-

сутствуют там в данный момент. Эти три подпрограммы являются частью операционной системы, которая позволяет одновременно выполнять несколько заданий.

Нециклические подпрограммы операционной системы вызываются только программами пользователей и обеспечивают эти программы теми видами обслуживания, которые предусмотрены в операционной системе. Имеется три вида таких подпрограмм:

1. Подпрограмма управления обслуживанием программ пользователей.
2. Подпрограммы ввода-вывода.
3. Подпрограмма управления файлами.

Первая из этих подпрограмм является средством общения между программой пользователя и операционной системой, обеспечивающим получение запрашиваемого обслуживания. Связь осуществляется при помощи программируемых операторов, содержащихся в программе пользователя и поступающих в операционную систему для обработки. Подпрограммы ввода-вывода обеспечивают передачу данных между программами периферийных устройств и программами пользователей в центральной памяти. Эти подпрограммы вызываются программой управления обслуживанием программ пользователей, благодаря чему сокращается объем подробного программирования, который должен выполнять пользователь для управления периферийными устройствами. Подпрограмма управления файлами расширяет постоянную память пользователя, позволяя ему хранить в виде файлов программы и данные, снабженные именами.

Рабочие характеристики системы

Число одновременно выполняемых заданий	127
Число потоков пакетов программ в режиме совпадения	14
Максимальный размер файла пользователя	Нет операционных ограничений в пределах всего доступного поля файла
Минимальный размер файла пользователя	128 слов (768 символов)
Максимальный размер задания, хранящегося в центральной памяти (без размера монитора в системе 1055)	256 слов (1028К байтов).

ПРИЛОЖЕНИЕ VI

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА STARAN ФИРМЫ GOODYEAR AEROSPACE SYSTEMS

ОРГАНИЗАЦИЯ СИСТЕМЫ

Система STARAN представляет собой цифровую вычислительную систему, которая одновременно выполняет арифметические, логические и поисковые операции над всеми или избранными словами, хранящимися в ее памяти. Система отличается от обычной ЭВМ следующими основными особенностями:

Наличием матричной памяти с многомерным доступом.

Наличием памяти, адресуемой по содержанию (ассоциативной памяти).

Наличием простого процессорного блока для каждого слова памяти.

Наличием уникальной схемы перестановок для сдвига и перегруппировки данных в памяти.

В системе STARAN каждая многомерная матрица состоит из квадрата ($256 \text{ разрядов} \times 256 \text{ слов}$), содержащего в общей сложности 65 536 разрядов. В каждом матричном модуле к данным можно обращаться либо в направлении расположения разрядов, либо в направлении расположения слов. Иначе говоря, процессорные блоки и каналы ввода-вывода могут получать доступ или к разрядному срезу (n -й разряд всех 256 слов), или к полному слову (256 разрядов). Обычная память с адресуемыми ячейками заменена в системе STARAN памятью, адресуемой по содержанию. Для того чтобы определить местонахождение заданного элемента информации, система учитывает степень его совпадения с некоторым входным элементом информации. Все слова в памяти, которые удовлетворяют этому критерию поиска, идентифицируются в течение одного цикла памяти. Следовательно, любой элемент информации может быть найден за одно обращение к памяти.

В системе STARAN общий процессорный блок, характерный для обычных ЭВМ, заменен процессорными блоками для каждого слова. Каждый арифметический блок обрабатывает последовательно (разряд за разрядом) слово памяти, к которому приписан этот блок. Все арифметические блоки выполняют операции одновременно по указаниям устройства управления. Поэтому при одной реализации ко-

манды все избранные слова памяти обрабатываются одновременно процессорными блоками этих слов. В системе используется схема перестановок, которая позволяет сдвигать и перегруппировывать данные так, чтобы над словами, хранящимися в матричной памяти, можно было выполнять параллельные арифметические и поисковые операции.

Существенный выигрыш в производительности можно получить, если использовать систему для выполнения заданий со следующими характеристиками:

Требования к скорости обработки чрезвычайно высоки.

Данные имеют весьма динамичный характер.

Одновременно имеется большое количество данных, требующих одинаковой обработки.

Требуются немедленные ответы на разнообразные запросы.

Наиболее выгодно использовать систему STARAN для решения задач, допускающих высокую степень параллельности обработки. Чем выше возможная степень параллельности, тем эффективнее применение системы.

Исследования, проведенные фирмой Goodyear Aerospace, показали, что такие применения, как обработка информации от многих датчиков, слежение за движущимися объектами, организация обработки и хранения данных, составление прогнозов погоды и решение других задач с матричной структурой, могут обеспечить эффективное использование возможностей системы по параллельной обработке. Фирма пришла к заключению, что в большинстве применений, изученных к настоящему времени, вычислительная система для решения сложных задач по обработке информации должна представлять собой гибридную систему, состоящую из процессоров параллельного и последовательного действия. Каждый процессор выполняет те задачи, которые лучше всего согласуются с его возможностями.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

Схема организации системы STARAN показана на рис. П.VI.1, а общие сведения о функциональных возможностях системы приведены в табл. П.VI.1. Ключевыми элементами системы являются ассоциативные матрицы и устройства управления ассоциативным процессором. Управляющая память ассоциативного процессора используется в основном для хранения программ. Страничное устройство, последовательный контроллер и логическая схема внешних функций представляют собой вспомогательные элементы системы, используемые для выполнения разнообразных внутренних функций по контролю и учету данных.

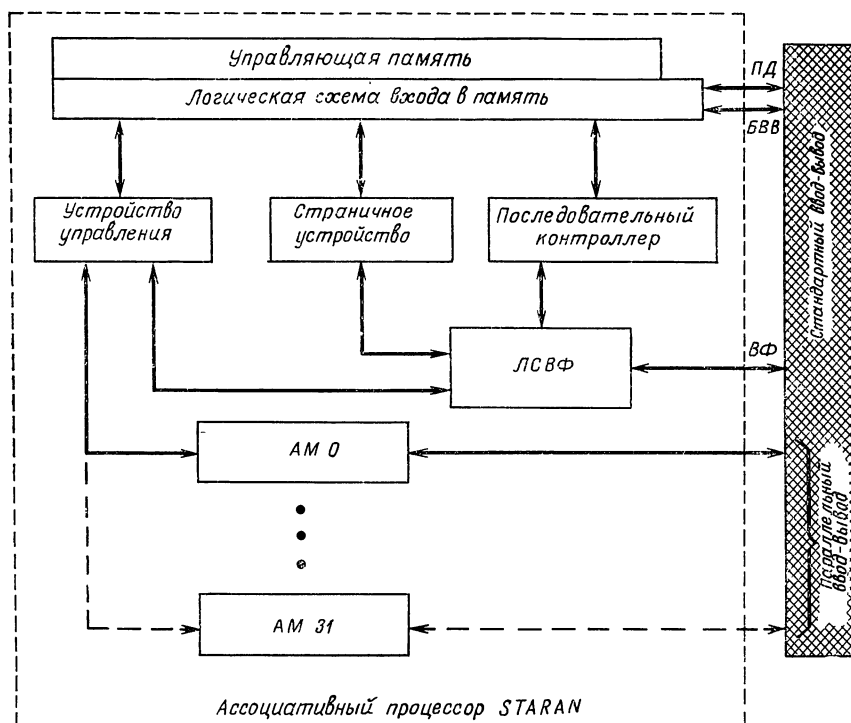


Рис. П. VI.1. Блок-схема системы STARAN.

АМ — ассоциативная матрица, ЛСВФ — логическая схема внешних функций, ПД — прямой доступ, БВВ — буферизованный ввод-вывод, ВФ — внешние функции.

АССОЦИАТИВНАЯ МАТРИЦА

Самым важным элементом системы STARAN является ассоциативная матрица, которая обеспечивает адресацию по содержанию и параллельность выполнения команд. Большая часть вычислительной работы системы осуществляется внутри одного слова ассоциативной матричной памяти, а не между словами памяти, как в обычном процессоре. Обычно слово ассоциативной матрицы разбивается программистом на поля переменной длины так, чтобы удовлетворялись требования данной программы. Содержимое этих полей можно затем складывать, вычитать, умножать и делить в рамках одного слова. По существу система может выполнять те же операции, что и последовательный процессор, однако она способна выполнять эти операции одновременно буквально над тысячами слов в матрицах ассоциативного процессора.

Общие сведения об элементах системы STARAN

Элемент	Функция
Ассоциативная матрица	Обеспечивает работу памяти с многомерным доступом и адресацией по содержанию емкостью 65 536 (2 ¹⁶) разрядов и работу 256 элементарных процессоров; допускает параллельное выполнение арифметических, логических и поисковых операций.
Устройство управления	Производит обработку данных в ассоциативных матрицах под управлением программы, хранящейся в управляющей памяти ассоциативного процессора.
Управляющая память	Хранит команды управления ассоциативного процессора; может также хранить данные и служить буфером между устройством управления ассоциативным процессором и другими элементами системы.
Страничное устройство	Перемещает сегменты программ в быстродействующие страничные запоминающие устройства.
Последовательный контроллер и память	Выполняет функции контроля исправности, управления периферийными устройствами и управления заданиями; предоставляет оператору возможность общения с различными элементами системы.
Устройство внешних функций	Передает управляющую информацию между элементами системы.

Базовая конфигурация системы STARAN содержит одну ассоциативную матрицу. Однако число матриц может быть доведено до 32. Эти матрицы по выбору могут работать в параллельном или последовательном режиме. Кроме того, если к матрице добавить вторую управляющую часть, то ее можно динамически переключать между двумя устройствами управления ассоциативным процессором. Это позволяет системе работать в мультипрограммном режиме. Как уже упоминалось выше, каждая матрица содержит 65 536 разрядов, организованных в виде квадрата (256 слов \times 256 разрядов); матрица состоит из элементов памяти на твердом теле. Параллельный доступ может быть осуществлен либо в направлении расположения разрядов, либо в направлении расположения слов. Можно обращаться к полному 256-разрядному слову или к разрядному срезу, т. е. к n -му разряду всех 256 слов. Ассоциативный процессор содержит 256 эле-

ментарных процессоров. Каждый элементарный процессор обеспечивает одновременную обработку многих наборов данных и хранение результатов выполнения арифметических, логических и поисковых операций.

Элементарные процессоры могут использоваться для временного хранения данных, извлеченных из матрицы, или данных, подлежащих записи в матрицу. Элементарный процессор используется также для составления логических комбинаций данных, полученных при последовательных операциях считывания, и для генерации адреса первого ответившего слова при поиске.

УСТРОЙСТВО УПРАВЛЕНИЯ АССОЦИАТИВНЫМ ПРОЦЕССОРОМ

Устройство управления ассоциативным процессором выполняет операции над данными, хранящимися в многомерной матричной памяти, по командам, хранящимся в управляющей памяти ассоциативного процессора. Это устройство может реализовать параллельные операции поиска для идентификации данных по содержанию, а затем выполнять над ними арифметические и (или) логические операции. Устройство управления ассоциативным процессором обладает следующими наиболее важными свойствами:

Выбор многих рабочих подмножеств из общего множества данных, хранящихся в памяти.

Выполнение операций над этими подмножествами без нарушения информации, хранящейся в остальных ячейках.

УПРАВЛЯЮЩАЯ ПАМЯТЬ АССОЦИАТИВНОГО ПРОЦЕССОРА

Организация управляющей памяти ассоциативного процессора предоставляет пользователю значительные возможности эффективной работы. Это достигается разделением памяти на несколько секций, как показано на рис. П.VI.2. Слово управляющей памяти имеет длину 32 разряда. Каждое слово содержит 16-разрядный адрес.

Страница 0	Страница 1	Страница 2	Быстродействующий буфер данных	Основная ферритовая память	Область памяти для прямого доступа
Память библиотекки подпрограмм (512 слов)	Память команд (512 слов)	Память команд (512 слов)	(512 слов)	(16 384 слова)	(30 720 слов)

Рис. П.VI.2. Организация управляющей памяти ассоциативного процессора.

В трехстраничных запоминающих устройствах используются твердотельные элементы. Цикл памяти составляет меньше 200 нс. Каждая страничная память может быть по желанию расширена до двойной емкости, т. е. до 1024 слов. Страница 0 предназначена для хранения библиотеки подпрограмм. Страницы 1 и 2 используются попеременно (как при игре в пинг-понг); считывание команд устройством управления ассоциативным процессором на одной странице производится одновременно с загрузкой другой страницы от страничного устройства. Память имеет входные вентили, которые предотвращают преждевременное использование страницы до окончания ее загрузки.

Быстродействующий буфер данных входит в состав управляющей памяти ассоциативного процессора; в нем также использованы элементы памяти на твердом теле. В стандартной конфигурации системы STARAN емкость буфера составляет 512 слов; по желанию ее можно удвоить.

В быстродействующем буфере данных очень удобно хранить данные и группы команд, к которым может потребоваться быстрый доступ со стороны различных элементов системы.

Основная память выполнена на ферритовых сердечниках и имеет время цикла меньше 1 мкс; в ней обеспечивается сохранение информации при выключении питания. В стандартной конфигурации эта память содержит 16 384 слова, но может быть расширена до объема 32 768 слов. Основная память используется для хранения программ.

Для обеспечения прямого доступа к внешней памяти в управляющей памяти ассоциативного процессора резервируется область адресов, которая в стандартной конфигурации составляет 30 720 слов.

Страничное устройство загружает любое из трех страничных запоминающих устройств словами из основной ферритовой памяти, быстродействующего буфера данных или канала прямого доступа к памяти. Это устройство работает под программным управлением.

ПОСЛЕДОВАТЕЛЬНЫЙ КОНТРОЛЛЕР

Последовательный контроллер обеспечивает:

Работу в режиме off-line для трансляции и отладки программ системы STARAN.

Средства первоначальной загрузки управляющей памяти ассоциативного процессора.

Связь между оператором и системой STARAN для управления системой в режиме on-line и контроля ее работы.

Управление программами обработки прерываний по ошибкам, технической диагностике и обслуживанию.

Средства последовательного выполнения арифметических операций и функций по учету работы системы.

Последовательный контроллер снабжен обычной памятью емкостью 8К слов, клавишным печатающим устройством, устройством записи-считания перфоленты и логической схемой интерфейса для подсоединения последовательного контроллера к другим элементам системы. Для хранения программ последовательного управления и данных используются 8192 слова памяти.

ВВОД-ВЫВОД

Система ввода-вывода имеет стандартную конструкцию, позволяющую подключать к системе STARAN другие вычислительные системы (и периферийные устройства). Предусмотрена возможность выбора из четырех видов интерфейса:

Прямой доступ к памяти.

Буферизованный ввод-вывод.

Логическая схема внешних функций.

Параллельный ввод-вывод.

В результате прямого доступа к памяти несистемной (внешней) ЭВМ эта память может использоваться как часть управляющей памяти системы STARAN. Информация, хранимая в памяти несистемной ЭВМ, одинаково доступна как для этой ЭВМ, так и для системы STARAN. Поэтому исключается необходимость в буферизации передач по системе ввода-вывода между системой STARAN и несистемной ЭВМ. В интерфейсе прямого доступа к памяти может потребоваться модификация адресов для сопряжения адресов системы STARAN и несистемной ЭВМ. Она выполняется в стойке стандартного устройства ввода-вывода. Блок адресов в интерфейсе прямого доступа к памяти может использоваться не только для доступа к памяти несистемной ЭВМ, но и для других целей, например для доступа во внешнюю память, которая может быть доступна или недоступна для других устройств, или для доступа к специальным устройствам ввода-вывода.

Буферизованный ввод-вывод используется для связи различных периферийных устройств с управляющей памятью системы STARAN. Кроме того, он может применяться для передачи блоков данных и (или) программ между управляющей памятью системы STARAN и памятью несистемной ЭВМ. Вообще применение интерфейса прямого доступа к памяти при работе с несистемной ЭВМ предпочтительнее, чем применение интерфейса буферизованного ввода-вывода, так как цикл последнего обычно длиннее, чем цикл первого. Кроме того, интерфейс буферизованного ввода-вывода обязывает программиста группировать данные в блоки, в то время как интерфейс прямого доступа к памяти может работать с неупорядоченными

данными. Исходное значение ширины интерфейса буферизованного ввода-вывода равно 32 разрядам плюс разряд проверки на четность. Стандартная стойка ввода-вывода может содержать буферные устройства, допускающие широкий диапазон способов распаковки; поэтому имеется возможность составлять каналы ввода-вывода любой ширины.

Логическая схема внешних функций облегчает координацию работы различных элементов системы STARAN при выполнении специальных функций и упрощает реализацию таких функций, как учет работы, контроль работоспособности и техническое обслуживание. Вырабатывая коды внешних функций и передавая их в логическую схему, одни элементы системы STARAN могут запрашивать информацию о состоянии других элементов и управлять этими состояниями. Коды функций могут передаваться в логическую схему устройством управления ассоциативным процессором, страничным устройством, последовательным контроллером и несистемной ЭВМ. В системе имеются коды функций переключения страничных входов, блокировок, разбиения программ на страницы, проверки наличия ошибок, прерываний по командам устройства управления ассоциативным процессором и по командам последовательного контроллера. Предусмотрен также резерв для включения дополнительных функций.

Канал логической схемы внешних функций интенсивно используется для непосредственной связи с устройствами, подключенными к стандартной стойке ввода-вывода. Последняя может одновременно принимать до 19 разрядов кода функций и обрабатывать каждую из них последовательно на приоритетной основе до тех пор, пока все коды не будут обработаны. Канал логической схемы внешних функций может использоваться:

В качестве интерфейса с несистемной ЭВМ.

Для выполнения внешних функций буферизованного ввода-вывода.

Для выполнения внешних функций параллельного ввода-вывода.

Система STARAN и несистемная ЭВМ могут общаться друг с другом при помощи механизма внешних прерываний. Если несистемная ЭВМ обладает способностью использовать механизм внешних функций, то она может осуществлять полное управление системой STARAN, вырабатывая любые необходимые коды внешних функций. Работу буферизованного ввода-вывода можно легко инициировать при помощи кодов логической схемы внешних функций. Результаты счета слов, начальные адреса и прочая информация упаковываются в требуемый формат для передачи. Работа интерфейса параллельного ввода-вывода иницируется при помощи кодов логической схемы внешних функций почти так же, как и работа интерфейса буферного ввода-вывода.

Каждая ассоциативная матрица может иметь до 256 входов и 256 выходов в стандартной стойке ввода-вывода. Они могут быть использованы для:

Увеличения скорости передачи данных между матрицами.

Обеспечения связи системы STARAN с широкополосными устройствами ввода-вывода.

Обеспечения непосредственной связи любого устройства с ассоциативными матрицами.

Например, при помощи интерфейса параллельного ввода-вывода можно осуществить эффективную связь между накопителем на магнитном диске с несколькими одновременно работающими головками и системой STARAN. Скорость передачи при такой конфигурации обычно зависит от длительности цикла и от числа используемых головок дискового устройства (табл. П.VI.2). Запрашивая, к при-

Таблица П.VI.2.

Быстродействие системы STARAN по вводу-выводу

Функция	Быстродействие
Прямой доступ к памяти	200 нс на одно 32-разрядное слово плюс время обращения к внешней памяти
Буферизованный ввод-вывод	400 нс на одно 32-разрядное слово
Передача между быстродействующим буфером данных и несистемной ЭВМ	
Передача между ферритовой памятью программ и несистемной ЭВМ	1 мкс на одно 32-разрядное слово
Передача команд внешних функций	
Время приема прерывания и возвращения информации о причине прерывания несистемной ЭВМ	1 мкс
Время генерации прерывания несистемной ЭВМ	1 мкс
Параллельный ввод-вывод	
Передача от памяти системы STARAN к периферийному устройству	200 нс на один разрядный срез или одно полное слово
Передача от периферийного устройства к памяти системы STARAN	400 нс на один разрядный срез или одно полное слово (1 срез равен 256 разрядам)

меру, данные от дискового устройства, система STARAN посылает ему один или несколько кодов внешних функций, которые указывают адрес начального сектора, число секторов и направление передачи данных.

Когда будет достигнут затребованный сектор, дисковое устройство может прервать работу системы STARAN, чтобы начать передачу по каналу параллельного ввода-вывода. Управляющие команды системы STARAN, реализующие запись или чтение по каналу параллельного ввода-вывода, могут быть синхронизированы с дисковым устройством так, что на время передачи тактирование системы STARAN будет подчинено тактированию дискового устройства.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ STARAN

Вместе с базовой конфигурацией системы STARAN поставляется группа автономных системных программ, позволяющих пользователю составлять, загружать и отлаживать свои программы. Системное программное обеспечение состоит из следующих программ:

Ассемблер.

Блок вспомогательных программ.

Блок отладочных программ.

Диагностические программы.

Библиотека подпрограмм.

Все эти программы выполняются последовательным контроллером, за исключением библиотеки подпрограмм и некоторых диагностических программ, которые реализуются устройством управления ассоциативным процессором.

ПРИЛОЖЕНИЕ VII

СИСТЕМА MULTICS 6180 И СИСТЕМЫ СЕРИИ 6000 ФИРМЫ HONEYWELL

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система Multics 6180 и системы серии 6000 фирмы Honeywell базируются на архитектуре, впервые использованной в системах 635 и 645.

Основные особенности архитектуры системы определяются полнотой модульной, асинхронной, обрабатывающей набор прерываний, ориентированной на память, многоходовой, распределенной системой управления. «Ориентация на память» означает, что каждый модуль контроллера памяти служит пассивным координирующим

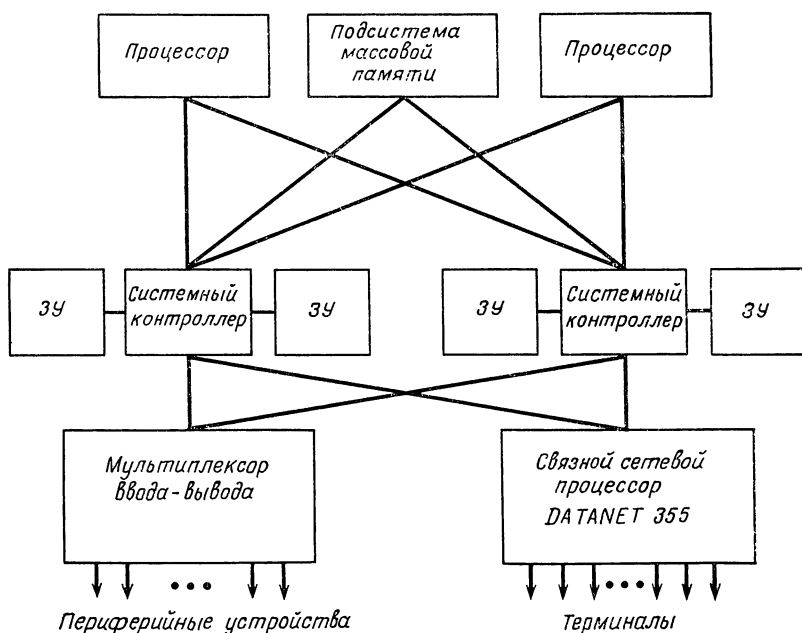


Рис. П.VII.1. Организация мультипроцессорной системы фирмы Honeywell.

Таблица П.VII.1

Характеристики систем 6000 и 6180

	Модель 6050/6060	Модель 6070/6080	Модель 6180
Максимальное число процессоров	4	4	6
Максимальное число связанных сетевых процессоров DATANET	3	3	3
Максимальная емкость памяти (36-разрядных слов)	524 288	1 048 576	2 097 152
Максимальная емкость памяти (9-разрядных байтов)	2 097 152	4 194 304	8 388 608
Длительность цикла памяти (для 8 байтов)	1,2	0,5	0,5
Максимальное число мультиплексоров ввода-вывода	4	4	4
Максимальное быстродействие мультиплексора ввода-вывода (байтов/с)	2,3М	4,0М	4,0М
Максимальное быстродействие мультиплексора ввода-вывода (символов/с)	3,7М	6,0М	6,0М
Число каналов данных на один мультиплексор ввода-вывода	24	24	24
Число периферийных устройств (подсистем)	24	24	24
Число одновременных операций ввода-вывода	24	24	24
Число программируемых регистров	49/57	49/57	
Операции с плавающей запятой	Да	Да	Да
Защита памяти	Да	Да	Да
Аппаратное преобразование системы счисления	Да	Да	Да
Чередование обращений к памяти	Да	Да	Да
Перекрытие операций	Да	Да	Да
Максимальное быстродействие (команд/с)	550 000	1 400 000	1 400 000

элементом, распределяющим обращения к памяти на основе аппаратных приоритетов. Каждое запускаемое по требованию запоминающее устройство содержит 32 ячейки прерываний, доступ к которым осуществляется по линиям связи. Это исключает непосредственные соединения между процессорами и модулями контроллеров ввода-вывода. Приоритеты прерываний можно изменять динамически при помощи программного управления и регистров масок пре-

рываний. Другой особенностью системы является возможность неограниченно расширять управляющую логику прерываний при помощи дополнительного мультиплексирования ячеек памяти, при котором каждый разряд памяти работает как регистр масок.

На рис. П.VII.1 приведена схема общей организации системы, где показаны пути прохождения данных и иллюстрируется главная роль системного контроллера (контроллера памяти). В табл. П.VII.1 представлены характеристики систем серии 6000 и системы 6180.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

Главным фактором, определившим конструкцию системы 6180, было стремление к тому, чтобы любой модуль системы как можно меньше отличался от стандартных модулей серии 6000. Важнейшим и самым заметным исключением является процессор. При разработке процессора была поставлена цель, чтобы логика процессора отличалась от логики обычного процессора системы 645 только при условии, что это приведет к существенному повышению общего показателя стоимость — эффективность системы MULTICS и сохранится совместимость с программным обеспечением систем серии 6000.

В соответствии с этим характеристики системы можно рассматривать как результат увеличения значений соответствующих характеристик системы 6000, за исключением характеристик собственно процессора, которые явились результатом существенной модификации характеристик стандартного процессора системы 645.

ПРОЦЕССОР

Большинство особых возможностей системы 6180, которые заметны для ее пользователей, обеспечиваются процессором. Модуль процессора может совершенно самостоятельно выполнять программы и всю обработку информации в системе. Процессор осуществляет выборку команд, подготовку адресов, защиту памяти, выборку и хранение данных. Эти функции выполняются с перекрытием, чтобы получить возможно более высокую скорость реализации команд.

Центральный процессор системы 6180 обладает всеми основными характеристиками процессора системы 6000, а также следующими дополнительными характеристиками:

Аппаратура для работы с сегментами и страницами.

Аппаратура для выработки 24-разрядных адресов памяти.

Два ассоциативных запоминающих устройства для ускорения выработки адресов (одно для сегментов и одно для страниц).

Программно адресуемые регистры, используемые при адресации сегментов и страниц.

Более широкие возможности модификации адресов.

Команды для управления аппаратурой сегментации и разбиения на страницы и системным датчиком времени.

Несколько уровней разрешения на доступ к памяти.

Обнаружение нескольких новых типов неисправностей процессора.

До восьми входов для подсоединения к модулям системных контроллеров.

Аппаратно реализованный механизм кольцевой защиты.

Процессор системы 6180 снабжен также ручным переключателем режимов; в зависимости от положения этого переключателя он может работать как процессор системы 6180 с расширенной системой команд (режим MULTICS) или как процессор системы 6000 (режим GCOS). Переключатель логически добавляет или исключает специальный дополнительный блок.

ОРГАНИЗАЦИЯ ПРОЦЕССОРА

Процессор состоит из четырех функциональных блоков:

блока управления,
блока операций,
десятичного блока,
дополнительного блока.

Первые три блока составляют процессор системы 6000 и выполняют все функции, которые предусмотрены в этом процессоре. В процессоре системы 6180 они работают как отдельный процессор и одновременно обеспечивают интерфейс и управление дополнительным блоком, который выполняет системные функции по сегментации памяти и разбиению ее на страницы.

ДОПОЛНИТЕЛЬНЫЙ БЛОК

Ниже перечислены основные свойства и функции дополнительного блока.

Обеспечивает 24-разрядную адресацию.

Содержит до 16 дескрипторов сегментов и 16-страничную таблицу.

Содержит дескрипторный регистр базы сегмента.

Содержит 8 регистров указателей сегмента.

Содержит аппаратуру кольцевой защиты.

БЛОК УПРАВЛЕНИЯ

Блок управления процессора системы 6180 работает точно так же, как стандартный блок управления процессора системы 6000, который обеспечивает сопряжение между блоком операций процессора и системным контроллером. Он также осуществляет выборку команд, подготовку адресов, защиту памяти, выборку и хранение

данных и общее тактирование. Ниже приведены некоторые дополнительные сведения о блоке управления системы 6180.

Режимы работы. Блок управления процессора системы 6180 может использовать нормально-абсолютную адресацию либо адресацию с дополнительным словом и может работать либо в режиме выполнения привилегированных команд, либо в режиме выполнения непривилегированных команд (в соответствии с работой процессора). В качестве процессора системы 6000 процессор системы 6180 ограничен двумя режимами работы: режимом ведущего процессора или режимом ведомого процессора.

Декодирование операций. В любом из двух режимов работы процессора (в режиме системы 6180 или системы 6000) все 10-разрядные коды операций, не разрешенные в данном режиме, вызывают появление сигнала ошибки при попытке их выполнения.

Отказы и логика определения отказов. При работе в режиме процессора системы 6180 блок управления распознает до 27 видов отказов. При работе же в режиме процессора системы 6000 он распознает только первые 16 отказов из полного списка отказов.

БЛОК ОПЕРАЦИЙ

Блок операций содержит логические схемы для выполнения арифметических и логических операций.

ДЕСЯТИЧНЫЙ БЛОК (С РАСШИРЕННЫМ СПИСКОМ КОМАНД)

Аппаратура расширенного списка команд добавляет к стандартному набору команд возможность обработки байтов, двоично кодированных десятичных символов, упакованных десятичных данных и разрядных строк. Эта аппаратура позволяет также выполнять второй уровень модификации адресов для всех стандартных команд и для команд расширенного списка.

РЕЖИМЫ РАБОТЫ

В режиме GCOS у процессора имеется два подрежима работы: в качестве ведущего и в качестве ведомого. В режиме MULTICS имеется четыре подрежима работы: абсолютный, привилегированный, непривилегированный и с регистром базового адреса. В абсолютном подрежиме и в подрежиме ведущего разрешены все команды. В непривилегированном подрежиме и в подрежиме ведомого разрешена большая часть команд (но не все). Обычные пользователи ограничены работой в непривилегированном подрежиме и, следовательно, ограждены от выполнения команд, которые могут нарушить другие программы или операционную систему MULTICS.

Привилегированные команды, такие, как команды, воздействующие на дескрипторный базовый регистр, системный датчик времени и устройства ввода-вывода, разрешены только в абсолютном и привилегированном подрежимах.

Полные возможности процессора по сегментации и разбиению на страницы используются в привилегированном и в непривилегированном подрежимах для выборки команд и данных. При адресации в абсолютном подрежиме возможности по сегментации и разбиению на страницы не используются.

ПАМЯТЬ

Система 6180 является системой с ориентацией на память. Память служит пассивным координирующим элементом системы, обеспечивающим как промежуточное хранение информации, так и управление системными связями. Поэтому контроллер, связанный с памятью, является не просто контроллером памяти, а фактически и системным контроллером.

СИСТЕМНЫЙ КОНТРОЛЛЕР

Модули системного контроллера служат центрами связи между другими модулями системы 6180, выполняя следующие функции:

Предоставляют ферритовую память для хранения команд, управляющих слов и данных.

Создают контрольные точки для передачи управляющих сигналов от одного активного модуля к другому.

Обеспечивают работу 52-разрядного системного датчика времени, определяющего время суток и интервалы времени.

Каждый системный контроллер имеет до восьми входов для подсоединения к процессорам, мультиплексорам ввода-вывода и подсистемам массовой памяти.

Системный контроллер может быть снабжен памятью емкостью 32К, 64К, 128К или 256К 36-разрядных слов (плюс разряд контроля четности) с временем цикла 1,2 мкс или 500 нс на одно слово. За одно обращение к памяти может быть записано или считано одно или два слова. Имеется также возможность хранить 6- или 9-разрядные символы в ячейке слова, не нарушая другие символы этого слова.

ФУНКЦИЯ ХРАНЕНИЯ

Для хранения или поиска информации активные модули (процессоры, мультиплексоры ввода-вывода, массовые запоминающие устройства) посылают в соответствующий системный контроллер

команду, адрес и необходимые данные. Системный контроллер выполняет команду и либо обеспечивает хранение полученных данных, либо посылает требуемые данные в запрашивающий модуль.

При связи с системными контроллерами активные модули используют следующие шесть команд:

Считывание — восстановление, с нормальной точностью.

Считывание — восстановление, с двойной точностью.

Очистка — запись, с нормальной точностью.

Очистка — запись, с двойной точностью.

Условное хранение (используемое для ключевых слов).

Считывание — очистка, в режиме MULTICS с регистром базового адреса.

Команда условного хранения позволяет активному модулю считывать и изменять содержимое какой-либо ячейки, не разрешая доступ к этой ячейке другого активного модуля во время этого процесса. Такая возможность, предусмотренная в системе 6180, весьма полезна при мультипроцессорной работе.

Обычно система 6180 содержит несколько системных контроллеров. Запросы на доступ к памяти распределяются между системными контроллерами методом переплетения модулей памяти. Этот метод выравнивает нагрузку системных контроллеров и повышает эффективность системы, уменьшая вероятность конфликтов и длину очередей запросов к одному и тому же физическому системному контроллеру. В системе может применяться двухкратное переплетение, четырехкратное переплетение или же переплетение может вообще отсутствовать. Выбор одного из этих способов распределения запросов является одной из основных характеристик системной конфигурации.

СИСТЕМНЫЙ ДАТЧИК ВРЕМЕНИ

Системный датчик времени состоит из календарных часов, которые представляют собой 52-разрядный двоичный счетчик, работающий с дискретностью 1 мкс и обеспечивающий отсчет времени в течение периода больше 142 лет без переполнения. Программа может считывать содержимое календарных часов с точностью до 1 мкс. 142-летний диапазон работы часов позволяет работать во всемирном времени. Для этого инженер-наладчик фирмы Honeywell должен установить часы на число микросекунд, истекших с полуночи 1 января 1901 года по гринвичскому среднему времени. Хотя любой процесс, происходящий в системе, может считывать показания часов в любое время, предусмотрены меры против попадания часов под управление программы; поэтому программные ошибки и неисправности аппаратуры вне часов не могут нарушить правильный отсчет текущего времени.

МУЛЬТИПЛЕКСОР ВВОДА-ВЫВОДА

Мультиплексор ввода-вывода сопрягает периферийные устройства и предварительный процессор связи DATANET 355 с памятью системы 6180 и может работать с большим числом устройств почти любого типа и быстродействия. Мультиплексор ввода-вывода управляется информацией, хранящейся в памяти, причем доступ к памяти он разделяет с другими активными модулями системы. Передача данных между устройствами ввода-вывода и памятью выполняется мультиплексором ввода-вывода одновременно с работой процессоров по выполнению программ. Управляющие слова ввода-вывода, подготовленные системой MULTICS, хранятся в памяти. Если операция ввода-вывода закончена или обнаружены какие-либо особые условия, то мультиплексор ввода-вывода информирует об этом систему MULTICS, инициируя программное прерывание.

МАССОВАЯ ПАМЯТЬ

Емкость. Один контроллер массовой памяти охватывает диапазон емкости от 1 до 8 млн. 36-разрядных слов.

Быстродействие. Устройство может передавать блок размером 1К слов в основную память или из нее за 382 мкс, не считая затрат времени на операции управления.

Входы. Устройство сопрягается непосредственно с системными контроллерами. В нем может быть до 8 входов—столько же, сколько в мультиплексоре ввода-вывода и процессоре системы 6180.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

СУПЕРВИЗОР

Супервизор системы MULTICS реагирует на любую информацию, которая может оказать воздействие на состояние процесса и (или) процессора. Процесс может находиться в следующих состояниях: рабочем, готовности или заблокированном. Процесс в рабочем состоянии — это процесс, выполняемый в данный момент процессором. Процесс находится в состоянии готовности, если он не выполняется, но ожидает освобождения процессора. Процесс в заблокированном состоянии ожидает некоторого события (не обязательно чрезвычайного) в ходе другого процесса или во внешней среде, например приема входной величины от устройства ввода-вывода. Супервизор должен не только следить за тем, в каком состоянии находится каждый процесс и когда следует изменить его состояние, но должен также обеспечить сообщение данному процессу о значительных событиях в других процессах.

УПРАВЛЕНИЕ ПОТОКОМ ЗАДАНИЙ

Процессы разделяются по принципу мультиплексирования между процессорами, которые входят в «анонимный банк» процессоров. Контроллер потока заданий представляет мультиплексную систему пользователю в таком виде, как если бы вычислительная система была предназначена на все время только для выполнения его заданий. Такой метод работы скрывает от пользователя подробности управления аппаратными средствами и осуществляет мультиплексирование системных ресурсов между пользователями без участия самих пользователей. Система защиты исключает возможность взаимодействия между заданиями пользователей, если на то не было предварительной договоренности.

Так как количество обрабатываемых процессов велико по сравнению с числом процессоров в системе, не все деблокированные процессы могут выполняться немедленно. Поэтому предусмотрено состояние готовности процесса. В системе имеется так называемый список готовности, в котором перечислены все процессы, находящиеся в состоянии готовности.

В списке готовности содержится вся основная информация, необходимая для диспетчеризации работы процессора после его освобождения от реализации какого-либо процесса. По команде активации блокированный процесс включается в список готовности. Если же процессор вырабатывает команду блокировки, то процесс должен временно покинуть процессор, в котором он выполнялся, после чего управление освободившимся процессором передается первому процессу в списке готовности.

Коммутацией процессов называются те процедуры контроллера потока заданий, которые осуществляют распределение процессоров между процессами, диспетчеризацию процессов и передачу процессов. Коммутация процессов инициируется исключительно по сигналам от других процедур супервизора, обычно в результате прерывания.

Основной аппаратный механизм, при помощи которого процессор переключается с одного процесса на другой, — это команда загрузки дескрипторного регистра базы сегмента. Во время перезагрузки этого регистра процессор воспринимает отображение в памяти нового процесса. Однако содержимое регистров процессора временно остается неизменным.

Процедуры коммутации процессов ведут список всех процессов, находящихся в состоянии готовности, в соответствии с заданным порядком реализации процессов. Список готовности состоит по существу из двух параметров: идентификатора процесса и предельного времени выполнения процесса, устанавливаемого программой-планировщиком.

Список готовности составляется и ведется процедурой планирования. Эта процедура оценивает приоритет процесса, запросившего

выполнение, сравнивая его с приоритетами процессов, уже состоящих в данный момент в списке готовности. Далее эта процедура устанавливает предельное время процесса и помещает его в соответствующее место списка. После того как процесс занесен в список готовности, он будет реализован в соответствии с правилом «первый пришел — первый обслужен». Однако если известно, что данный процесс обладает высоким приоритетом, то ему может быть предоставлено преимущественное положение в списке готовности в соответствии с его приоритетом по сравнению с другими процессами, состоящими в списке. Планировщик может применить внесрочное прерывание, чтобы заставить какой-то процесс уступить место в процессоре другому процессу, имеющему более высокий приоритет. Если процесс исчерпал отведенное ему время, но его выполнение еще не завершено, то он возвращается на свой уровень в списке готовности.

Для организации параллельной обработки каждая процедура пользователя и многие модули системы MULTICS выполняются как наборы отдельных процессов, которые называются группами процессов. Передача информации между процессами определяется системой межпроцессных связей.

Процесс, передающий информацию, помещает сообщение в какой-либо сегмент, доступный двум процессам — передающему и принимающему. Сообщение может представлять собой либо данные, либо управляющую информацию, либо процедуру. Принимающий процесс считывает из общего сегмента информацию, заключенную в сообщении; выполнение принимающего процесса может быть приостановлено в ожидании сообщения. При помощи сигнала инициации передающий процесс заставляет принимающий процесс возобновить свою активность, когда сообщение уже поступило в сегмент.

Фундаментальным понятием для межпроцессной связи является понятие события. Событием является все, происшедшее во время реализации одного процесса и представляющее интерес для другого процесса. Например, завершение сборки символов строки, введенной в систему при помощи автоматической пишущей машинки, является событием для рабочего процесса. Событие — это уникальное явление, которое имеет место только один раз. Если процесс управления устройством в приведенном примере распознает несколько последовательных завершений строки, то каждое завершение представляет собой самостоятельное событие. Межпроцессное сообщение является сигналом от одного процесса к другому о том, что произошло какое-то событие.

РАСПРЕДЕЛЕННАЯ ОПЕРАЦИОННАЯ СИСТЕМА

Для выполнения заданий пользователя в системе MULTICS комбинируются необходимые части операционной системы (выбранные сегменты) с сегментами программы пользователя. Таким образом,

реализуемый процесс не является ни программой пользователя, ни собранием сегментов системы MULTICS, а представляет собой их комбинацию. Этот метод получил название «распределенная операционная система», так как функции системы надлежащим образом распределены внутри процесса. Искусный пользователь системы MULTICS может по желанию заменить код системы MULTICS своим системным кодом в большей части системы. Программы, которые пользователь составляет для реализации, становятся неотличимыми от программ в системном коде MULTICS.

ДИНАМИЧЕСКИЕ СВЯЗИ

Так как сегменты перемещаются из вспомогательной памяти в ферритовую память только тогда, когда они нужны для выполнения процесса, установление связи *символических обращений* между сегментами производится тогда, когда такое обращение встречается первый раз во время выполнения процесса. В каждом сегменте программы имеется область связи, содержащая пары слов (пары связи), представляющие все обращения из данного сегмента к другим, внешним, сегментам. До первого выполнения операции связи каждая пара связи содержит модификатор, который вызывает появление сигнала ошибки связи, если внешнее обращение затребовано преждевременно.

ПРИЛОЖЕНИЕ VIII

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА H4400 ФИРМЫ HUGHES AIRCRAFT COMPANY

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система H4400 представляет собой модульный мультипроцессор с автоматической реконфигурацией, снабженный системой программного обеспечения (система SHOC) для управления распределением ресурсов. Система состоит из нескольких автономных модулей, связанных при помощи переключателя память-процессор. В состав модулей системы входят: арифметический-управляющий процессор, процессоры ввода-вывода, процессор специального назначения и модули памяти. Функциональная взаимосвязь между модулями показана на рис. П.VIII.1, который иллюстрирует также возможности расширения конфигурации системы.

В мультипроцессорной конфигурации системы каждый арифметический-управляющий процессор и процессор специального назначения функционируют автономно (с одинаковым системным статусом) при двухстороннем чередующемся доступе к модулям памяти (число которых может достигать 16). Процессор специального назначения обеспечивает работу связанных, дисплейных или других процессоров, находящихся в ведении пользователя и выполняющих функции, которые обычно не возлагаются на арифметический-управляющий процессор, имеющий более общее назначение. Централизованный переключатель память-процессор, помимо организации взаимосвязи между памятью и процессором, выполняет ряд центральных системных функций, таких, как диагностическое управление часами истекшего реального времени (от двух до четырех независимых часов), управление прерываниями и назначение исполнительных систем.

Последние три функции выполняются при помощи 16 системных регистров-переключателей, каждый из которых содержит стандартное 32-разрядное слово. Вероятность выхода из строя всей системы в целом в результате сбоя в переключателе память-процессор сведена до минимума благодаря функциональному логическому разбиению системы на модули и наличию резервных источников питания. Переключатель память-процессор занимает центральное место в реализации функций автоматической реконфигурации системы и диагностического контроля; в переключатель память-процессор входят

системные средства, определяющие последовательность диагностических операций; в переключателе память-процессор может осуществляться изменение логических адресов модулей памяти; с помощью

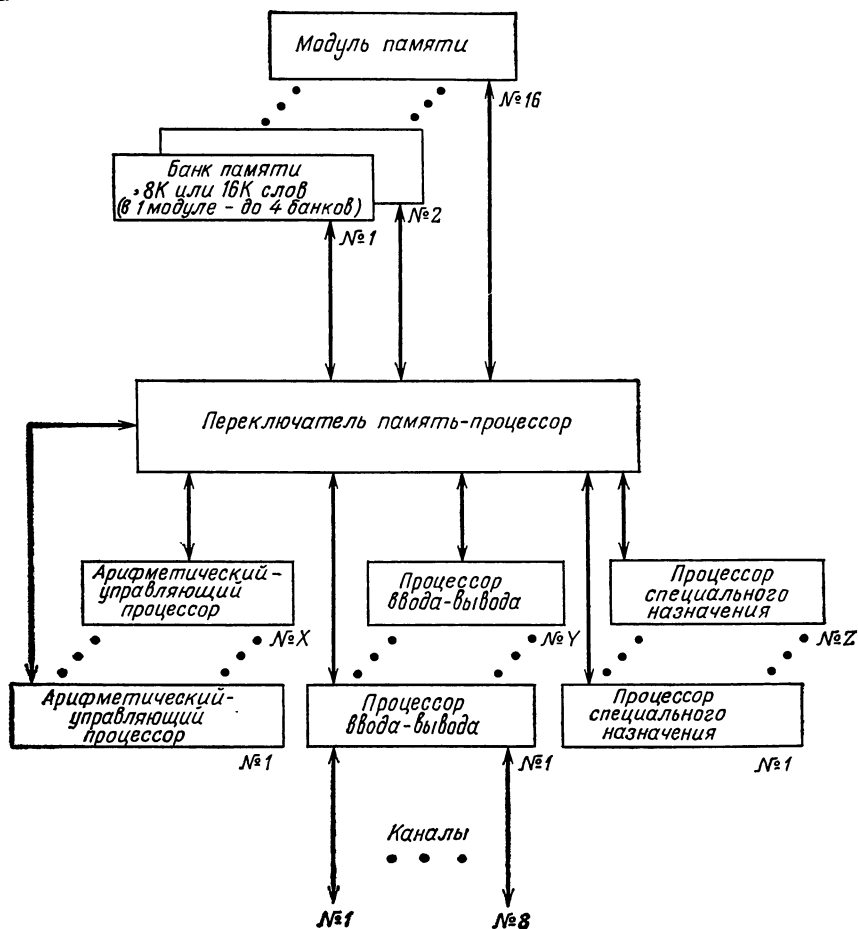


Рис. П.VIII.1. Конфигурация системы Н4400.

Система может содержать: от 1 до 7 арифметических — управляющих процессоров; от 1 до 7 процессоров ввода-вывода; до 6 процессоров специального назначения; всего до 8 процессоров; от 1 до 16 банков памяти (емкостью до 262 144 слова, или 1 048 576 байтов).

переключателя память-процессор можно осуществить блокировку вышедших из строя процессоров или запоминающих устройств; кроме того, осуществляется автоматическое переключение на новый исполнительный процессор или исполнительный модуль памяти, если ранее назначенные устройства вышли из строя.

Переключатель память-процессор управляет приоритетом доступа процессора к 16 входам в память (наивысшим приоритетом доступа к модулю памяти обладают входы 0 и 1, остальные входы обслуживаются по циклическому принципу). Аппаратные средства, содержащиеся в переключателе память-процессор, реализующие исполнительные функции, обеспечивают возможность функционирования в исполнительном режиме лишь одного процессора и обслуживают запросы на переход в исполнительное состояние или выход из него; направляют все системные прерывания в процессор, определенный как «руководитель прерываний»; формируют индикаторы состояния системы.

В условиях максимальной конфигурации системы ввода-вывода предусмотрено до 83 разрешенных прерываний (в том числе 56 канальных), подразделяющихся из 5 классов. Эти классы, в порядке их приоритета, включают:

1. Прерывания из-за сбоев аппаратных средств.
2. Локальные программные прерывания (межпроцессорная связь).
3. Внешние прерывания (от часов и внешних сигналов).
4. Прерывания по вводу-выводу.
5. Локальные непрограммные прерывания (внутренние программные ошибки).

Первые четыре класса прерываний обрабатываются переключателем память-процессор до их передачи на принимающий процессор. Переключатель обеспечивает буферизацию всех прерываний и устанавливает структуру приоритета для прерываний, направляемых в арифметический-управляющий процессор — «руководитель прерываний». Все локальные непрограммные прерывания обрабатываются в том арифметическом-управляющем процессоре, где они возникают, причем переключатель память-процессор не информируется об их возникновении. В противоположность первым четырем классам прерываний прерывания этого вида (не считая прерываний из-за арифметических ошибок) могут не блокироваться.

Прототип системы H4400 функционирует в центре обработки данных фирмы Hughes Aircraft с конца 1970 г. Этот прототип включает: два арифметических-управляющих процессора, два процессора ввода-вывода, память емкостью 65К слов, три накопителя на магнитных лентах, устройство чтения перфокарт, строчно-печатающее устройство, накопитель на магнитном диске, устройство чтения перфоленты и ленточный перфоратор, а также пишущую машинку на пульте оператора. В состав программного обеспечения входят операционная система, компилятор JOVIAL (J-6), мета-ассемблер, две отдельные системы отладки и полный набор общих программ-утилитов.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Арифметический-управляющий процессор системы Н4400 включает 35 регистров (16 регистров общего назначения, 16 регистров возврата прерываний, два регистра слова состояния программы и регистр установки метки прерывания). Каждый из этих регистров имеет длину 32 разряда и является доступным для программы. Один арифметический-управляющий процессор может выполнять до 600 000 команд в секунду, используя средства предварительного просмотра команд, которые позволяют совмещать выборку следующего 32-разрядного слова команды с выполнением предыдущей команды. Управление арифметическим-управляющим процессором обеспечивается при помощи постоянной памяти (память, допускающая только считывание) на больших интегральных схемах, содержащей микропрограммы объемом 2048 битов для каждого процессора, входящего в мультипроцессорную конфигурацию. Используя эту память в качестве «модульного варианта», можно в отдельном арифметическом-управляющем процессоре составить до 24 макрокоманд, реализованных аппаратными средствами. Это свойство дает разработчику системы возможность создания и реализации своего собственного набора команд, помимо стандартного набора из 108 команд, имеющих длину полуслова и полного слова. Другими «модульными вариантами» использования постоянной памяти являются добавление 18 команд для выполнения операций над шестнадцатеричными числами с плавающей запятой с нормальной и двойной точностью и добавление 9 команд, обрабатывающих любую непрерывную группу разрядов в пределах 32-разрядного слова.

Обеспечивается возможность адресации 524 287 полуслов при помощи 19-разрядных исполнительных адресов, генерируемых для различных методов адресации: прямого, многоуровневого косвенного, индексированного или метода база-плюс-смещение. Дополнительно предусматривается прямая адресация 8-разрядных байтов в пределах слова в целях загрузки, хранения и выполнения побайтовых команд, когда в операции принимает участие самый младший байт одного из 16 мультирегистров.

Каждый арифметический-управляющий процессор снабжен 64-разрядным словом состояния программы, функции которого приблизительно аналогичны функциям такого же слова в системе 360 фирмы IBM. Это слово содержит следующие коды: код условий арифметических операций; код условий запрещения арифметических операций вследствие ошибок; код состояний процессора; код номера процессора, определяющего входное соединение между переключателем память-процессор и арифметическим-управляющим процессором; ключ защиты памяти и счетчик команд программы. Однако в отличие от системы 360 каждый арифметический-управляющий

процессор снабжен 16 регистрами возврата прерывания (по 32 разряда каждый), которые организованы в пять групп по три регистра, плюс один резервный. Эти пять групп регистров соответствуют пяти названным выше классам прерываний. Таким образом, когда арифметический-управляющий процессор распознает прерывание, его текущее слово состояния программы записывается в два регистра возврата прерывания, соответствующих определенному классу прерываний. По сути дела, это «старое слово состояния программы», которое в системе 360 фирмы IBM должно храниться в резервируемом двойном слове в нижней части основной памяти. Другими словами, процедура прерывания здесь такая же, как в системе 360: «новое слово состояния программы» поступает из заданной ячейки памяти и заменяет «старое слово состояния программы».

СВОДКА ХАРАКТЕРИСТИК ПРОЦЕССОРА

Микропрограммная постоянная память позволяет использовать до 27 дополнительных команд, реализованных аппаратно (макрокоманд).

Семь из 16 регистров общего назначения могут использоваться как регистры базового адреса, а все 16 в целом — как индексные регистры.

Специальная команда REPEAT обеспечивает автоматическое повторение до семи последовательных команд (максимальное число повторений 524 287).

Возможна адресация 512К полуслов основной памяти (256К слов — максимум, реализованный в настоящее время) при помощи прямой, многоуровневой косвенной, индексированной адресации или адресации типа база-плюс-смещение.

Чередование памяти и использование средств предварительного просмотра и поиска команд обеспечивают максимальное быстродействие в одном процессоре 600 000 команд/с.

В одном из «модульных вариантов» использования постоянной памяти девять команд обеспечивают обработку строки, составленной из частей целого слова.

ОСНОВНАЯ ПАМЯТЬ

Основная память системы H4400 использует ферритовые сердечники с широким температурным диапазоном и организована по типу запоминающего устройства с разрушением информации при считывании; ее цикл работы равен 1,4 мкс, время обращения — 0,48 мкс. Длина слова составляет 36 разрядов (32 информационных разряда и 4 разряда контроля четности). 8К или 16К слов составляют банк памяти, а один модуль может содержать до 4 банков. Конфигурация системы может включать до 16 банков памяти, или 256К слов. Производительность системы максимально увеличена за счет обеспече-

ния доступа любого арифметического-управляющего процессора или процессора ввода-вывода к двум или большему числу модулей памяти в режиме совпадения под управлением переключателя памяти-процессор.

Модуль памяти содержит также средства внутреннего тестирования памяти. Этот механизм просматривает в цикле все слова памяти и сравнивает их содержимое с набором значений, заданных на пульте технического обслуживания системы. С этого пульта оператор может также занести в память или вывести из нее любое слово.

Отдельный модуль памяти при разных вариантах его использования обладает следующими характеристиками:

Наличие 8 или 16К слов в одном модуле с разрядом контроля четности для каждого 8-разрядного байта в 32-разрядном слове. Наличие устройства защиты секретности памяти, которое содержит 5-разрядный ключ секретности, причем 4 разряда представляют собой код ключа, который должен быть согласован с ключом защиты памяти для обеспечения доступа к блокам по 1024 слова, а один разряд используется для указания, какие операции допустимы при доступе к блоку, — только считывание или считывание и запись.

Всеми обращениями к памяти управляют два регистра-переключателя системы в переключателе памяти-процессор. Эти два регистра содержат шестнадцать 4-разрядных полей, каждое из которых соответствует конкретному входу в память. Каждое поле содержит логическое обозначение физического модуля памяти. Когда процессор (либо арифметический-управляющий, либо ввода-вывода) запрашивает доступ к памяти, четыре старших цифры адреса сравниваются с содержимым всех 4-разрядных полей в двух регистрах-переключателях системы. Затем данный запрос на доступ направляется во все модули памяти, имеющие совпадающие логические обозначения. Если условия секретности для всех этих модулей не идентичны или если содержимое памяти не соответствует запросу на считывание, то результаты производимой операции предсказать невозможно. Если вход в память не имеет закреплённого за ним модуля памяти, то его поле в двух регистрах-переключателях системы полностью игнорируется при запросе на доступ. Если вход с согласованным адресом банка не будет обнаружен, то произойдет прерывание по условию «несуществующего адреса».

ПРОЦЕССОР ВВОДА-ВЫВОДА

Процессор ввода-вывода обеспечивает канал связи между ЭВМ и периферийными устройствами, которые работают автономно и независимо от арифметического-управляющего процессора. В функции процессора ввода-вывода входит формирование последователь-

ностей команд и данных, защита данных в памяти, а также проверка на четность во всех интерфейсах. Процессор ввода-вывода может иметь от одного до восьми каналов, каждый из которых, независимо от типа, может приспосабливать свой внутренний стандартный интерфейс к требуемому системному интерфейсу ввода-вывода. Возможно использование нескольких различных типов каналов и их сочетаний без изменения внутрипанельного монтажа. Шесть типов канальных интерфейсов (см. ниже) являются взаимозаменяемыми в пределах модуля процессора ввода-вывода. Все каналы работают автономно и могут выполнять несколько операций ввода-вывода в режиме совпадения. Наличие фиксированного приоритета в переключателе память-процессор обеспечивает разрешение конфликтных ситуаций, возникающих в основной памяти. Каждый процессор ввода-вывода может осуществлять прямую адресацию максимум к 16 банкам памяти (262 144 слова). Способность процессора ввода-вывода выполнять совмещенные операции в памяти системы обеспечивает максимальную скорость передачи данных 1,4 млн. слов/с (при рабочем цикле памяти 1,4 мкс). Скорость передачи данных по каналам ввода-вывода для каждого стандартного типа канала представлена в таблице П.VIII.1. Ниже приведено описание типов

Таблица П.VIII.1

Скорость обмена данными по каналам ввода-вывода при цикле работы памяти 1,4 мкс

	Слов/с	Байтов/с	Битов/с
Суммарная скорость для одного процессора ввода-вывода	$1,4 \times 10^6$	$5,6 \times 10^6$	$4,5 \times 10^7$
Скорость для одного канала:			
поразрядной последовательной передачи	38×10^3	$1,4 \times 10^5$	$1,1 \times 10^6$
побайтовой последовательной передачи	250×10^3	$1,0 \times 10^6$	$8,0 \times 10^6$
пословной передачи	350×10^3	$1,4 \times 10^6$	$1,1 \times 10^7$
внешнего мультиплексорного ЭВМ — ЭВМ	250×10^3	$1,0 \times 10^6$	$8,0 \times 10^6$
ЭВМ — ЭВМ	350×10^3	$1,4 \times 10^6$	$1,1 \times 10^7$
поблочной передачи	250×10^3	$1,0 \times 10^6$	$8,0 \times 10^6$

каналов. Каждый стандартный тип канала, за исключением канала с поразрядной последовательной передачей, устанавливает связь с периферийными устройствами через 8-разрядный байтовый интерфейс.

Приоритет канала определяется типом запроса и позицией входа (физическим расположением канала в модуле). Наивысшим приоритетом обладают запросы на передачу данных, следующим приори-

тетом — запросы на прерывание, самым низким приоритетом — запросы на выборку команды. Среди запросов одного типа наивысшим приоритетом обладает запрос на нулевой вход, а самым низким — запрос на вход 7. Поскольку во всех операциях запросы на новые команды обладают наиболее низким приоритетом, структура приоритета служит своего рода схемой регулирования нагрузки, причем новые операции над данными не начинаются до тех пор, пока не будет устранена перегрузка в системе передачи данных.

ВСПОМОГАТЕЛЬНАЯ ПАМЯТЬ

Прототип системы H4400, используемый в центре обработки данных фирмы Hughes Aircraft, включает запоминающее устройство на магнитном диске с максимальной емкостью 58 млн. битов. Диск имеет 10 поверхностей по 203 дорожки на каждой поверхности. Среднее время доступа составляет 85 мс, максимальное — 135 мс.

ХАРАКТЕРИСТИКИ ОПЕРАЦИОННОЙ СИСТЕМЫ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Управление системой H4400 основано на операционной системе управления программными и аппаратными средствами (SHOC), которая определяется фирмой Hughes как универсальная программная исполнительная система для работы в реальном масштабе времени, предназначенная для мультипроцессорных операций. Объем резидентной части операционной системы, расположенной в основной памяти, составляет 12К слов.

Для организации мультипрограммирования используется обобщенный подход, при котором операционная система рассматривает множество выполняемых программ как сеть взаимосвязанных управляющих точек, называемых «узлами». Узел определяется либо как пульт управления системой (внешний узел), либо как основной блок прикладного программного обеспечения (внутренний узел, или «кластер»). Все узлы обладают следующими общими свойствами:

Каждый узел имеет «точку управления связью». Для внутренних узлов точка управления связью представляет собой специальное задание из кластера, которое играет двоякую роль: обрабатывает входные и выходные сообщения и планирует работу узла при помощи алгоритма и системы приоритетов, выбранных пользователем. Для внешних узлов точка управления связью представляет собой аппаратное средство, обладающее соответствующей способностью ввода-вывода информации, как, например, автоматическая пишущая машинка или терминал (с функциональной клавиатурой и дисплеем). Точка управления связью внешнего узла должна обязательно учитываться и распознаваться в под-

системе управления вводом-выводом операционной системы как допустимый тип периферийного устройства.

Каждый узел может объявить о своем исключительном праве на определенные ресурсы. Это обычно сопровождается указанием на аппаратное средство (если оно не съемное) или на определенные тома памяти (пакеты дисков, бобины магнитной ленты и т. д.). Области оперативной памяти также могут быть объявлены «собственностью» внутренних узлов.

Каждый узел может сам определить уровень своей секретности в сети по отношению к передаче данных. Определение любого конкретного узла в операционной системе включает имя узла и, в качестве необязательного параметра, список других узлов, с которыми допустимо установление связей («список разрешений»). Могут быть также указаны и дополнительные возможности, например совместное использование некоторых ресурсов. Узлы могут быть объединены в «группы», в которых разумно сочетаются требования секретности и совместного использования ресурсов.

Помимо указанных выше свойств, концепция «сети взаимосвязанных узлов» обеспечивает выполнение следующих функций управления заданиями, которые реализуются операционной системой:

1. Шаги задания по указанию пользователя могут выполняться параллельно несколькими процессорами.

2. Могут формироваться новые задания в виде дополнительных узлов сети по требованию любого узла (задания или внешнего устройства).

3. Задания, выполняемые в режиме совпадения, могут установить внутреннюю взаимосвязь при помощи точек управления связью.

4. Шаги задания данного узла могут синхронизировать начало своего выполнения с окончанием выполнения другого шага задания в этом же узле, с наличием меток и показаний счетчиков, установленных по указанию пользователя, с получением ответа на ранее переданное сообщение, с завершением операции ввода-вывода и с окончанием временного интервала или истинного времени, величины которых задаются операторами управления ссылкой на событие.

Для обеспечения этих возможностей в операционной системе предусмотрены два метода организации динамического запроса и управления обслуживанием. Первый использует системный вход операционной системы, осуществляющий маршрутизацию и передачу сообщений между узлами. Текст сообщения состоит из любой последовательности символов с максимальной длиной 240 символов. Каждый узел имеет закрепленный за ним идентификатор, который определяется передающим узлом как место назначения для сообщения. Если местом назначения является внешний узел, то маршрут сообщения задается с учетом печати или воспроизведения на данном узле (дисплей, строчно печатающее устройство и т. д.). Если же

местом назначения является внутренний узел, то сообщение помещается в очередь на входе данного узла и активизируется специальный механизм принятия сообщения при помощи точки управления данного узла. Существует возможность обхода очереди на входе для сообщений, обладающих более высоким приоритетом (например, для ввода с внешнего узла), и размещения данного сообщения в специальном буфере, выделенном точкой управления узла. Помимо этого, любой узел может объявить «потoki связи» резидентными на принадлежащих ему устройствах. Это обеспечивает возможность динамического распределения входных потоков для узлов, связанных с обработкой потоков заданий и (или) пакетов данных, поступающих с местных или удаленных терминалов.

Второй метод организации динамического обслуживания запросов системой программного обеспечения основан на комплексе стандартных программ SHOC, в которые можно войти из прикладных программ для получения требуемого обслуживания. Вход в эти стандартные программы обычно осуществляется при помощи механизма вызова монитора (команда MON), причем они выполняются как непосредственное расширение прикладных программ, которые их вызывают. Эти системные сервисные программы можно вызывать путем выдачи макрокоманды с соответствующими аргументами, определяющими характер требуемого обслуживания (например, объем области основной памяти), а в некоторых случаях — путем ввода сообщений в узел, содержащий ядро SHOC.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Система SHOC обеспечивает ввод-вывод при помощи централизованной системы управления вводом-выводом. В зависимости от типов устройств, в которых могут быть размещены данные, возможны несколько видов организации файла. Система управления вводом-выводом полностью ориентирована на файл (в противоположность многим системам, ориентированным на устройства).

В системе предусмотрено семь макрокоманд:

FILE — объявить файл.

FREL — освободить файл.

OPEN — открыть файл (получить доступ).

CLOSE — закрыть файл (уступить доступ).

FIN — ввести в файл.

FOUT — вывести из файла.

FPOS — указать позицию в файле.

Эти операции предназначены для создания минимального логического уровня ввода-вывода, необходимого для обеспечения независимости процедур ввода-вывода от конкретных устройств и централизованного управления вводом-выводом со стороны операционной системы SHOC.

ПРИЛОЖЕНИЕ IX

МУЛЬТИПРОЦЕССОРНАЯ СИСТЕМА 360, МОДЕЛЬ 65, ФИРМЫ IBM

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Фирма IBM определяет мультипроцессорную систему как «вычислительную систему, использующую два или больше взаимосвязанных устройств обработки данных для одновременного выполнения нескольких программ», и реализует эти возможности только на одной модели серии 360 — мультипроцессоре 65 МР. (Модель 67 предназначена главным образом для систем с разделением времени и обычно управляется операционной системой специального назначения, например системой TSS/360, хотя она может использоваться как модель 65 со стандартными средствами программного обеспечения путем блокировки механизма преобразования адреса виртуальной памяти.) Мультипроцессорная обработка данных в системе модели 65 предлагается в качестве одного из вариантов системы, называемого мультисистемным; этот вариант обладает следующими характеристиками:

Используется версия операционной системы OS/MVT, получившая название M65MP.

Одновременно могут выполняться две задачи; для предотвращения одновременного доступа обоих центральных процессоров к специальным супервизорным данным применяется программный метод, называемый блокировкой.

Любой центральный процессор имеет доступ к большинству периферийных устройств в результате использования двухканальных переключателей; доступ из процессоров к устройству, физически и логически подключенному к другому центральному процессору и не имеющему двухканального переключателя, зависит от связи между центральными процессорами. (Если устройство не располагает двухканальным переключателем, то невозможен доступ к нему от центрального процессора, с которым он непосредственно не связан.)

Реконфигурация системы может осуществляться оператором ЭВМ, причем помимо периферийных устройств можно исключать и заменять центральные процессоры, каналы и элементы памяти, не нарушая текущую обработку задания.

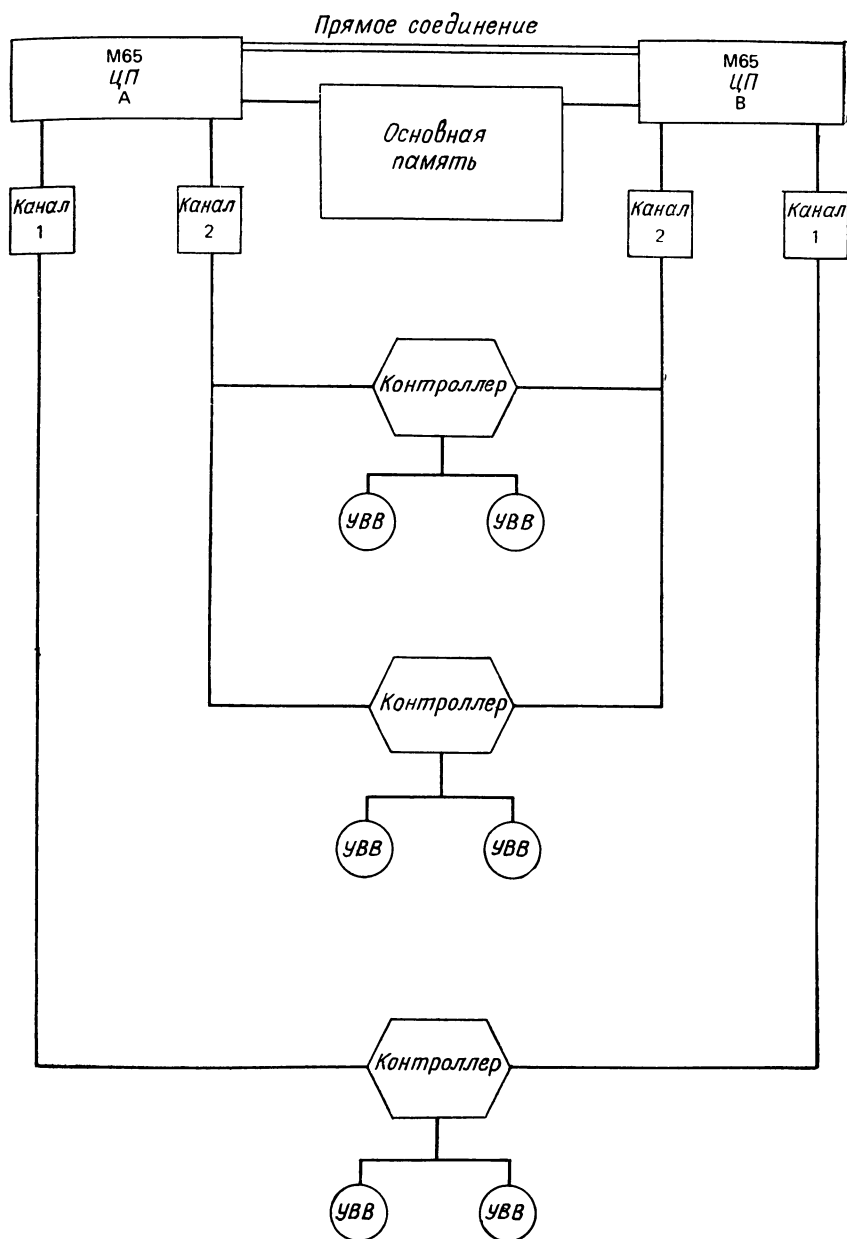


Рис. П.IX.1. Мультипроцессор модели 65 МР.
УВВ — устройство ввода-вывода.

Для организации двухпроцессорной системы используется многошинная-многовходовая конфигурация, которая обеспечивает взаимосвязь, необходимую для совместного использования элементов системы, как показано на рис. П.IX.1. Здесь представлена конфигурация типичного мультипроцессора. Центральный процессор имеет один вход для обеспечения доступа к его ресурсам, тогда как запоминающие устройства имеют несколько входов для связи с центральными процессорами и капалями.

Подход, разработанный фирмой IBM, допускает работу мультипроцессора модели 65 как единой вычислительной системы с двумя центральными процессорами, совместно использующими все запоминающие устройства под управлением одной операционной системы, или же как двух отдельных вычислительных систем, каждая со своими собственными центральными процессорами, памятью и периферийными устройствами, находящихся под управлением двух отдельных операционных систем. Между этими крайними случаями находится такая конфигурация, в которой устройства и память могут резервироваться для специальных условий использования их любым центральным процессором под управлением одной операционной системы. Основным недостатком рассматриваемой мультипроцессорной конфигурации заключается в том, что она не допускает применения массового запоминающего устройства на ферритовых сердечниках типа 2361 с временем цикла 8 мкс, емкостью до 8 388 608 байтов, имеющего прямую адресацию. Однако центральная память процессора обладает возможностью расширения до 2 097 152 байтов с полным циклом памяти около 750 нс путем добавления блоков по 256K байтов.

Основное преимущество, которое выявилось при эксплуатации модели 65 в режиме мультипроцессорной обработки, заключается в возможности применения стандартной аппаратуры и средств программного обеспечения, прошедших длительные испытания, с минимальными изменениями. Все объектные программы, которые обрабатываются при помощи стандартной операционной системы OS/MVT, можно обрабатывать и при помощи версии M65MP без внесения каких-либо изменений в команды, операторы управления заданиями и данные.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Каждый процессорный блок модели 2065 включает стандартные регистры системы IBM 360 и выполняет функции, указанные ниже:

1. Шестнадцать 32-разрядных регистров общего назначения, программно адресуемых как индексные регистры, накапливающие регистры или регистры базового адреса.

2. Четыре 64-разрядных регистра с плавающей запятой для выполнения арифметических операций с нормальной и двойной точностью.

3. 64-разрядное слово состояния программы, которое адресуется программами супервизорного состояния и используется для управления состоянием системы по отношению к программе, выполняемой в текущий момент.

4. Выполнение двух- или трехадресных команд пяти форматов с тремя длинами поля из универсального набора команд, содержащего 87 стандартных команд, 44 команды с плавающей запятой и 8 команд десятичной арифметики.

5. Система приоритетного прерывания, обеспечивающая распознавание пяти классов прерываний.

Предусмотрены следующие пять форматов команд: регистр-регистр, регистр-память с индексацией адреса, регистр-память, память-непосредственный операнд и память-память. Все обращения к памяти относятся к типу «база-плюс-смещение», где поля смещения представляют собой 12-разрядные положительные целые числа, добавляемые к 24 младшим разрядам регистра общего назначения, выбранного в качестве базового; в результате обеспечивается простой способ логической адресации до 16 777 216 байтов основной памяти.

Хотя 8-разрядный байт является основной единицей хранения данных, в системе имеется восемь различных форматов данных, так что каждая команда оперирует с одним и только одним форматом. На эти типы данных налагаются ограничения, связанные с выравниванием границ памяти. Адрес байта для поля данных должен быть кратен длине поля в байтах. Например, если процессор обращается к полному слову — операнду с фиксированной запятой, адрес которого не кратен четырем (граница полного слова), то происходит прерывание по условию неправильной адресации.

Процессор работает в одном из двух режимов: в состоянии задачи, когда выполнение всех команд ввода-вывода и группы управляющих команд является недопустимым, или в состоянии супервизора, когда все команды являются выполнимыми. Любое состояние процессора, независимо от того, находится ли он в состоянии обработки программы или ожидания, отражается словом состояния программы, которое представляет собой 64-разрядное управляющее слово. Маска системы (8 разрядов слова состояния программы) используется для запрещения прерываний от каналов ввода-вывода, таймера, ключа прерывания на пульте оператора или по сигналу периферийного устройства. 4-разрядный ключ защиты памяти используется для защиты блоков ферритовой памяти емкостью 2048 слов от неразрешенного считывания, считывания — записи или записи. Значения ключей хранятся в ячейках неадресуемой памяти, связанных с каждым блоком. Имеются две привилегированные команды: «установить ключ памяти» и «прочитать ключ памяти», кото-

рые используются супервизором системы для внесения изменений или проверки кода защиты, присвоенного блоку.

Поскольку каждый процессор должен реагировать автономно на автоматическую установку или обмена слова состояния программы с фиксированными ячейками памяти, необходимо было предусмотреть две специально выделенные области памяти, каждая из которых адресуется в диапазоне от 0 до 4095. Мультипроцессорные средства модели 65 обеспечивают прямое перемещение адреса, что достигается добавлением 12-разрядного префикса к адресу базы памяти, который всегда имеет нули в 12 старших разрядах, так как адреса 0—4095 могут формироваться без задания адреса базы или индекса. Таким образом, один процессор может обращаться к абсолютным адресам в нижней части ферритовой памяти, где находятся такие элементы, как векторы прерывания (различные слова состояния программы), слова состояния канала, показания таймера интервалов и слова управления вводом-выводом, в то время как другой процессор обращается к выделенному для него блоку емкостью 4К в верхней части памяти.

Для создания мультипроцессорной конфигурации модели 65 используется процессор модели 2065 в варианте мультипроцессора. Этот процессор должен иметь средства прямого управления для обеспечения непосредственной связи центральный процессор — центральный процессор, причем в состав одного из процессоров должен входить блок управления конфигурацией. Модель мультипроцессора соответствует схеме распределения потоков информации, представленной на рис. П.IX.2. Необходимо отметить, что она обладает возможностями доступа к центральной памяти при увеличении ее объема. На схеме показаны четыре основные логические части процессорного блока 2065:

1. Арифметико-логическое устройство, содержащее 60-разрядный параллельный двоичный, или 8-разрядный десятично-двоичный сумматор с временем цикла логической структуры 200 нс.

2. Внутренняя память, включающая рабочие регистры процессора, регистры общего назначения и регистры с плавающей запятой, имеющие время цикла 200 нс.

3. Постоянное запоминающее устройство, используемое для управления потоком данных и выполнением команд.

4. Устройство управления памятью, работающее независимо от процессора, обслуживающее его запросы на доступ к памяти и разрешающее конфликтные ситуации при одновременном доступе к памяти процессора и каналов передачи данных.

Следует отметить, что процессор 2065, показанный на рис. П.IX.2, осуществляет выборку команд 8-байтовыми порциями с временем цикла 200 нс. Кроме того, буфер команд обеспечивает быструю подготовку команд и перекрытие времени выборки и исполнения большинства команд.

С пульта управления можно задавать один из трех режимов: мультисистемный, обычный режим модели 65 и раздельный режим. В мультисистемном режиме обеспечивается непосредственная связь между центральными процессорами через основной интерфейс пря-

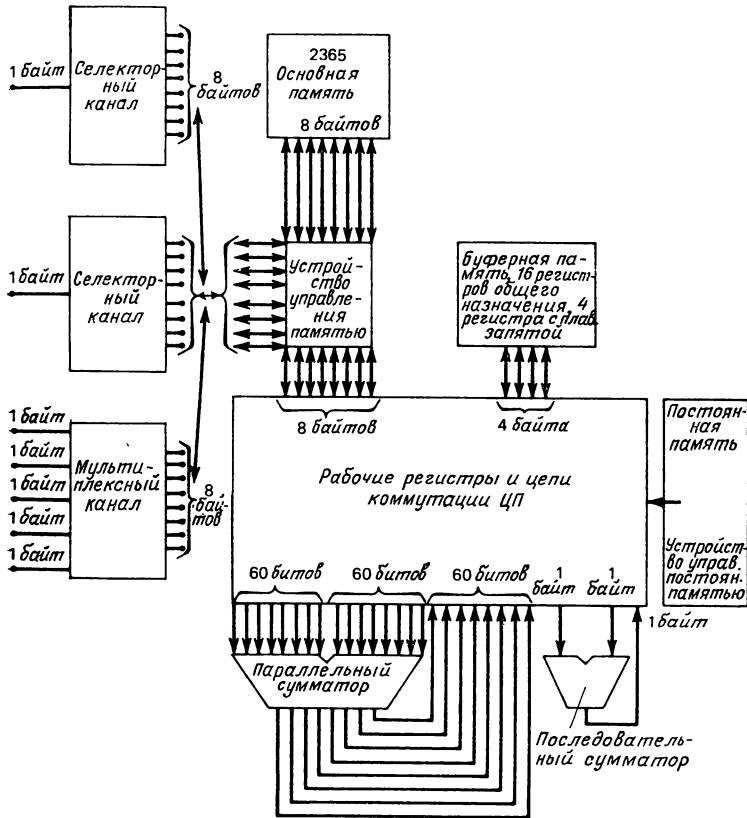


Рис. П. IX. 2. Схема распределения потоков информации в системе модели 65 МР.

мого управления для осуществления обычной мультипроцессорной обработки под управлением единой исполнительной программы. Режим модели 65 используется в тех случаях, когда интерфейс прямого управления должен функционировать точно так же, как в обычных моделях 65 системы 360. При работе в раздельном режиме мультисистемная прямая связь между центральными процессорами блокируется, так что каждый из них рассматривается как отдельная система, имеющая свою собственную управляющую программу, блоки основной памяти и устройства ввода-вывода. Раздельная система

могла бы продолжить использование операционной системы мультипроцессора при условии выделения одного центрального процессора из системы.

ОСНОВНАЯ ПАМЯТЬ

При работе модели 65 как мультипроцессорной системы можно построить конфигурацию, содержащую от двух до восьми блоков памяти процессора модели 2365-13, каждый из которых обеспечивает хранение данных объемом 265К байтов, так что в максимальной конфигурации память будет иметь емкость 2 089 152 байта. Каждый модуль памяти характеризуется временем цикла 750 нс при параллельном доступе к данным объемом 8 байтов. Функции хранения могут выполняться на побайтной основе (8 разрядов плюс разряд контроля четности), причем за один цикл работы в память может быть занесено любое число комбинаций последовательных байтов (включающих до восьми байтов).

Все модули памяти содержат по две автономные секции, каждая из которых имеет свой собственный адрес и буферные регистры. Каждая секция обеспечивает параллельный доступ к восьми байтам и в целом имеет емкость 131 072 байта, организованных в 16 384 двойных слов. Одна из секций содержит двойные слова, пронумерованные четными числами, другая секция — двойные слова, пронумерованные нечетными числами; этим достигается двустороннее чередование адресов.

Процессор мультипроцессорной модели обеспечивает для всей системы совместное использование памяти и устройств ввода-вывода, а также возможность адресации с помощью регистров с плавающей запятой. Каждое совместно используемое запоминающее устройство, как правило, содержит устройство управления буфером для организации интерфейса с блоком основной памяти. Соответствующая схема приоритетов определяет, какое устройство управления буфером может получить доступ к блоку памяти, и гарантирует, чтобы ни одно устройство не выполнило два последовательных обращения к данному блоку памяти, если другое устройство управления находится в состоянии ожидания доступа.

МОДУЛЬ УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Канал в системе IBM/360 представляет собой модуль управления вводом-выводом, обеспечивающий пути передачи данных и прямое управление контроллерами ввода-вывода и устройствами, подсоединенными к этим контроллерам. Имеется два типа каналов: селекторный и мультиплексный. Селекторные каналы предназначены для быстродействующих устройств, таких, как запоминающие устройства на магнитной ленте, на барабане или на диске. Эти каналы позволя-

ют осуществлять в каждый момент времени передачу информации лишь к одному устройству (или от него). Мультиплексные каналы обычно служат для устройств с малым быстродействием, таких, как автоматические пишущие машинки, устройства чтения перфокарт и печатающие устройства. Эти каналы допускают одновременную передачу данных к нескольким устройствам такого типа (или от них). Поскольку сигналы к этим устройствам передаются с чередованием, в функцию мультиплексора входит распределение сигналов по соответствующим устройствам, например отделение символов, предназначенных для карточного перфоратора, от символов, предназначенных для строчно печатающего устройства.

С каждым процессором 2065 может быть соединено не больше семи каналов в любой из двух комбинаций: один мультиплексный канал 2870 и шесть селекторных каналов 2860 модели 3 в двух конфигурациях или же два мультиплексных канала 2870 и пять селекторных каналов 2860 моделей 2 и 3. Каждый селекторный канал имеет возможность адресоваться к 256 устройствам ввода-вывода, по одному в каждый момент времени, с использованием перекрытия передаваемых групп данных. Основной мультиплексный канал может адресоваться к 16 устройствам ввода-вывода. Однако при использовании процессоров мультипроцессорной модели второй мультиплексный канал 2870 в каждом процессоре не может иметь селекторные подканалы. Каждый селекторный подканал может приводить в действие одно устройство ввода-вывода в режиме совпадения с основным мультиплексным каналом. Этот подканал позволяет подсоединить восемь контроллеров для устройств со скоростью передачи данных, превышающей 180 000 битов/с. При любом числе подсоединенных контроллеров максимальная общая скорость передачи данных для мультиплексного канала лежит в диапазоне от 110 до 670К битов в зависимости от числа установленных селекторных подканалов. Каждый действующий селекторный подканал снижает максимальную скорость передачи данных по основному мультиплексному каналу 110К битов, причем первые три селекторных подканала работают со скоростью 180К битов каждый, а четвертый подканал обладает максимальной скоростью передачи 100К битов, обеспечивая общую скорость передачи 670К битов.

В системе может быть использован адаптер канал-канал, обеспечивающий путь передачи данных между двумя каналами. Эти каналы могут относиться либо к одной и той же системе, либо к различным системам. Можно установить взаимосвязь двух центральных процессоров для совместного использования устройств ввода-вывода при помощи двухканального переключателя контроллеров и адаптера канал-канал для селекторных каналов 2860. Общее число совместно используемых устройств ввода-вывода для мультипроцессорной конфигурации не может превышать общего числа устройств, подсоединяемых к базовой модели 65.

ВСПОМОГАТЕЛЬНАЯ ПАМЯТЬ

Массовая память базовой модели 65 системы 360 включает ферритовые запоминающие устройства большого объема, магнитные барабаны и магнитные диски. Однако в мультипроцессоре модели 65 нельзя использовать ферритовую память большого объема.

В системе имеется два запоминающих устройства на барабанах со средней задержкой вращения 8,6 мс. Запоминающее устройство на барабане 2301 обеспечивает доступ примерно к 4,1 млн. байтов. К одному устройству управления памятью 2820 можно подсоединить четыре запоминающих устройства на барабане общей емкостью около 16,4 млн. байтов. Помимо этого, устройство управления 2820 может коммутироваться с двумя каналами при помощи двухканального переключателя. Скорость передачи данных к (от) центральному процессору составляет 1 200 000 байтов/с. Имеется барабан 2303 меньшего размера и с меньшим быстродействием, обеспечивающий прямой доступ к данным объемом 3,91 млн. байтов и передачу со скоростью 312 500 байтов/с. Двумя запоминающими устройствами на барабане можно управлять при помощи устройств управления памятью модели 2840 или 2841, обеспечивающих доступ к данным объемом 7,82 млн. байтов.

Запоминающее устройство на магнитных дисках представлено либо в виде пакета с постоянными (несъемными) дисками, либо в виде пакета съемных дисков. К устройствам первого типа относится запоминающее устройство на дисках 2302, модели 3 и 4, включающее соответственно один или два модуля по 112,79 млн. байтов каждый. К одному устройству управления памятью 2841 можно подсоединить до четырех запоминающих устройств на дисках 2302, что обеспечивает примерно емкость 902 млн. байтов памяти с прямым доступом, причем можно использовать до восьми различных механизмов обращения к памяти. Запоминающие устройства на пакете съемных дисков представлены устройствами моделей 2311 и 2314. Модель 2311 обеспечивает прямой доступ к памяти объемом 7,25 млн. байтов в одном пакете дисков. К устройству управления памятью 2841 можно подсоединить восемь дисководов, что дает в итоге емкость памяти 58 млн. байтов при среднем времени выборки 75 мс. Устройство 2314, модель 1, включает восемь автономных дисководов и устройство управления. В качестве резервного предусмотрен девятый дисковод. Другая конфигурация устройства 2314 (серия А) допускает наличие различного числа дисководов (до 9).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Мультипроцессорная работа модели 65 осуществляется под управлением операционной системы, предназначенной для системы 360, в варианте MVT. Этот вариант операционной системы обеспечивает

мультипрограммирование с переменным числом задач и выполнением до 15 заданий в режиме совпадения. Термин «задача» применяется фирмой IBM для обозначения самостоятельной части программы, которая рассматривается как основная единица работы для супервизора.

Задания помещаются в отдельные очереди на входе системы в зависимости от класса задания, причем место задания в очереди определяется его приоритетом. Одновременно в активном состоянии может находиться до 15 заданий.

Планирование запуска программ осуществляется под управлением MVT. Программа посылает запрос супервизору на подсоединение следующей задачи. После этого супервизор вырабатывает новую задачу и планирует ее выполнение. Эта новая задача является подзадачей исходной задачи. Для данной подзадачи может быть указан иной приоритет, чем для исходной задачи. Все задачи, относящиеся к одному заданию, могут выполняться асинхронно.

Определенные виды ресурсов системы, такие, например, как основная память и средства управления центральным процессором, назначаются только задачам, причем назначение осуществляется с учетом их приоритета. Система управляет распределением ресурсов, следит за осуществлением всех назначений, а также освобождает ресурсы после завершения задачи.

При работе системы в режиме мультипрограммирования несколько задач одновременно могут претендовать на получение одних и тех же ресурсов. В этом случае запросы устанавливаются в очередь и, когда ресурсы освобождаются, их предоставляют в распоряжение «старшего» в очереди. Для того чтобы управлять назначением ресурсов, супервизор формирует очереди из неудовлетворенных запросов на ресурсы и таблицы, идентифицирующие доступные ресурсы.

Для каждого шага задания или для задания в целом может быть указан объем требуемой основной памяти. Хотя для задания или шага задания назначается фиксированный объем памяти, под управлением системы MVT возможно дополнительное назначение памяти при помощи механизма «развертывания и свертывания». Этот механизм позволяет заданию временно динамически расширяться за границы первоначально выделенной области памяти. В случае отсутствия свободной памяти другое задание переписывается на диск (свертывается), а его область памяти используется первым заданием в течение всего необходимого времени.

Супервизор осуществляет динамическое управление памятью, распределяет ее области и позволяет прикладной программе посылать запрос на динамическое перераспределение областей памяти. Области памяти могут передаваться или совместно использоваться задачами под управлением MVT. Система предусматривает создание системы подпулов (блоков основной памяти емкостью 2К, отведен-

ных для конкретной задачи). Подпулы становятся доступными задаче, когда она переходит в состояние подсоединения. Подпулы могут быть освобождены после завершения задачи или оставлены за ней, если задача создала подзадачу. Основная память не может совместно использоваться последовательными шагами одного и того же задания.

Устройства ввода-вывода распределяются в начале выполнения задания, причем выполнение заданий не планируется до тех пор, пока не будут доступны все устройства. Распределение устройств может осуществляться при помощи идентификатора (ID) физического устройства, типа устройства (например, устройство чтения перфокарт, устройство записи на магнитную ленту) или категории устройства (например, устройства с последовательным и прямым доступом).

Система обеспечивает средства управления доступом для трех типов программ: программы, не используемые повторно, программы, используемые последовательно, и программы с повторной входимостью. Программы с повторной входимостью можно использовать одновременно в нескольких задачах. Программы, используемые последовательно, нельзя применять таким образом; запросы к ним устанавливаются в очередь, пока данная программа не станет доступной; когда она освобождается, управление программой передается первому запросу в очереди. Программы, не используемые повторно, загружаются с диска при каждом запросе на доступ к ним.

Машинное время распределяется на основе соперничества, т. е. право использования центрального процессора получает программа, обладающая наивысшим приоритетом. Программа освобождает центральный процессор в том случае, когда она ждет завершения какого-либо события, например операции ввода-вывода. В этот момент задача, обладающая более низким приоритетом, получает право использования центрального процессора. Когда задача с более высоким приоритетом готова возобновить обработку, задача с более низким приоритетом временно отстраняется и центральный процессор передается задаче с более высоким приоритетом.

В системе предусмотрена возможность работы в режиме разделения времени. Всем задачам, обладающим определенным приоритетом, центральный процессор может выделить интервал времени, величина которого задана предварительно. Группа задач, обрабатываемых в режиме разделения времени, соперничает за получение ресурсов центрального процессора, и разделение времени между ними происходит только до тех пор, пока группа находится на обслуживании центрального процессора.

Выполнение задачи может быть временно приостановлено, пока не произойдет какое-либо обусловленное событие или совокупность событий. При этом супервизор даст указание программе о том, что данное событие произошло, и начнет планирование ее выполнения.

Таковыми событиями могут быть определенные условия ввода-вывода, прерывания по таймеру и внешние прерывания. Выполнение задачи также может быть временно приостановлено, пока порождаемая ею подзадача не достигнет определенного этапа выполнения. Координация работы двух центральных процессоров осуществляется с использованием функции блокировки в целях предотвращения одновременного доступа обоих центральных процессоров к важнейшим системным данным, а также метода, называемого в практике фирмы IBM «похлопыванием по плечу», когда для установления связи между двумя центральными процессорами применяется команда прямой записи. В мультипроцессорном режиме работы, когда в одном центральном процессоре прерывания блокируются, а в другом не блокируются, необходимо «запирать» частично обработанные таблицы и очереди от их использования или внесения в них каких-либо изменений со стороны центрального процессора, работу которого нельзя прервать. Для обеспечения такого запирания предусмотрен специальный переключатель, называемый байтом запирания, который проверяется и включается всякий раз, когда блокируются прерывания в системе, а затем выключается, если прерывания снова разрешены. Если байт запирания включен, то задержанные программы могут выполняться только в том центральном процессоре, который выполнил команды включения байта запирания.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Если в системе одновременно появляются два или больше запросов на использование канала или контроллера, то один из них помещается в очередь на ввод-вывод. Очереди на ввод-вывод обычно обслуживаются в соответствии с диспетчерским приоритетом задачи, запрашивающей ввод-вывод, однако в ряде случаев супервизор может решать задачу оптимальной загрузки аппаратуры при выборе задачи из очереди. Например, запросы на доступ к диску могут обслуживаться таким образом, чтобы минимизировалось время поиска информации на диске.

Возможны следующие типы организации файла: последовательная, индексно-последовательная, с подразделением, с прямым доступом и телепроцессорная.

Для обработки данных используется два общих метода: с очередями и базовый. Метод, использующий очереди, предназначен для работы с отдельными записями и может применяться лишь при поиске записей в последовательном наборе. Записи заносятся в основную память до поступления фактического запроса на них, чем устраняется ненужное ожидание программных операций ввода-вывода. Базовый метод предназначен для работы с блоками записей. Блок заносится в основную память не раньше, чем поступит запрос на него.

В системе предусмотрено три метода выделения буферов: простая буферизация, обменная буферизация и буферизация с цепочками сегментов. Кроме того, применяется метод динамической буферизации (для доступа с очередями), когда буфер берется из пула системных буферов и присваивается входной записи. Этот буфер автоматически возвращается в пул после обработки входной записи. Супервизор осуществляет автоматическое объединение в блоки и распаковку блоков записей, что составляет часть его деятельности по буферизации. Доступ к данным может быть осуществлен либо в рабочей области, независимой от входного или выходного буфера, либо при помощи указателя ячейки, в которой хранится запись в пределах области буфера.

Стандартные телепроцессорные программы выполняют следующие функции: опрос терминалов, совместно использующих одну и ту же линию связи, анализ заголовков сообщений с целью определения маршрута следования входных и выходных сообщений, постановка в очередь и контроль последовательных номеров входящих сообщений, преобразование внешних кодов передачи данных во внутренние коды обработки, а также контроль ошибок передачи данных.

ДИАГНОСТИЧЕСКИЙ КОНТРОЛЬ ОШИБОК

Программа машинного контроля ошибок анализирует сбои с целью определения уровня допустимого восстановления. Если восстановление невозможно на функциональном уровне (в результате устранения неисправности и(или) повторного запуска прерванной команды), то предпринимается попытка на системном уровне связать эту неисправность с конкретной задачей, чтобы обеспечить селективное завершение соответствующего задания.

Возможна также попытка восстановления на уровне системной повторной инициации с использованием системных очередей, очередей заданий и очередей данных, автоматически поддерживаемых с помощью средств повторного запуска системы.

При обнаружении сбоя в канале ввода-вывода управление передается программе канального контроля ошибок. Анализ, осуществляемый с помощью этой программы, помогает программам восстановления исправить ошибки на входе. Программа канального контроля ошибок также генерирует специальную запись, указывающую условия сбоя в канале.

Пользователь имеет возможность создать свои собственные подпрограммы для контроля и обработки ошибок в своих программах. Он может задавать типы прерываний, которые будут обрабатываться его собственными подпрограммами, а также типы прерываний, которые должен обрабатывать супервизор. Супервизор обычно реаги-

рует на неисправимые программные ошибки, такие, например, как неверное выполнение привилегированных команд и нарушение защиты памяти.

При обнаружении неисправностей, которые могут быть устранены, управление передается обрабатываемой программе с указанием возможного источника ошибки. Если условия появления ошибки таковы, что дальнейшая обработка задачи приведет к разрушению накопленных данных или вызовет ухудшение характеристик системы, то управление передается программе останова по сбою.

ПРИЛОЖЕНИЕ X

МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ 370, МОДЕЛИ 158 И 168, ФИРМЫ IBM

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Фирма IBM определяет два типа «мультипроцессоров»: слабо связанные системы с раздельной основной памятью и с отдельными копиями операционной системы и жестко связанные системы с общей основной памятью и одной копией операционной системы. Системы последнего типа соответствуют определению мультипроцессоров, принятому в этой книге. Основные конфигурации таких систем показаны на рис. П.Х.1 и П.Х.2.

Мультипроцессорный режим работы жестко связанных систем моделей 158MP и 168MP предлагается фирмой IBM как один из вариантов работы системы 370. Такой режим получил название мультисистемного; он характеризуется следующими особенностями:

Используется основная версия операционной системы для ЭВМ системы 370, называемая операционной системой с виртуальной памятью (OS/VS2).

Каждый пользователь (или задание) системы имеет в своем распоряжении виртуальную адресуемую память емкостью 16 млн. байтов для размещения своих собственных и системных программ, общих данных и рабочих областей.

Система использует наборы «замков» для каждого из 13 типов многократно используемых ресурсов. Вследствие этого захват ресурсов одним центральным процессором не исключает возможности использования других ресурсов другим центральным процессором. (В операционной системе MVT для модели 65MP системы 360 возможен только один «замок».)

Технические средства, необходимые для организации взаимодействия двух моделей 158 или двух моделей 168, обеспечиваются блоком мультисистемной связи. Этот блок имеет в своем составе панель управления конфигурацией, при помощи которой можно вручную изменять конфигурацию системы.

Два процессора в системе 158MP имеют общую основную память емкостью от 1 до 8 млн. байтов, причем каждый из них имеет быстросействующую буферную память емкостью 8К байтов с доступом к информации объемом 8 байтов за 230 нс.

Два процессора в системе 168МР имеют общую основную память емкостью от 2 до 16 млн. байтов, причем каждый из них имеет буферную память емкостью 8К (с возможностью расширения до 16К байтов) с доступом к информации объемом 8 байтов за 80 нс.

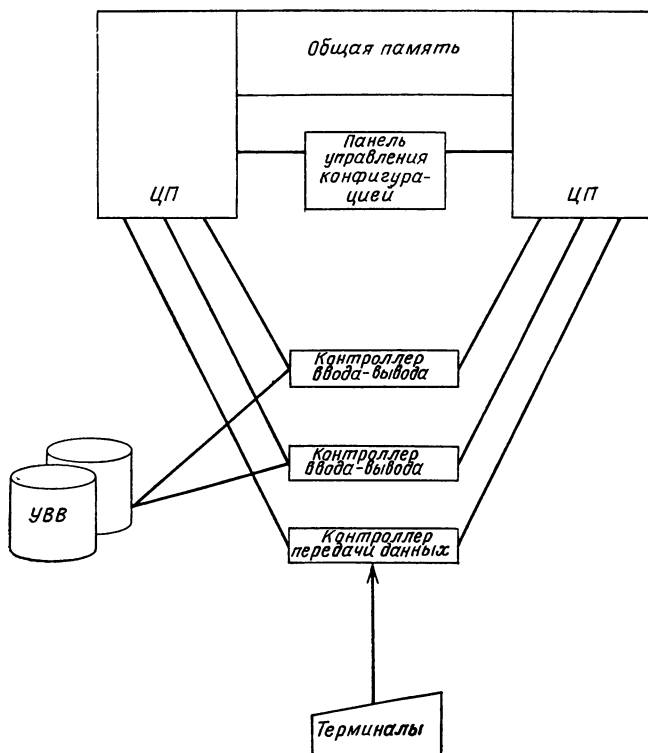
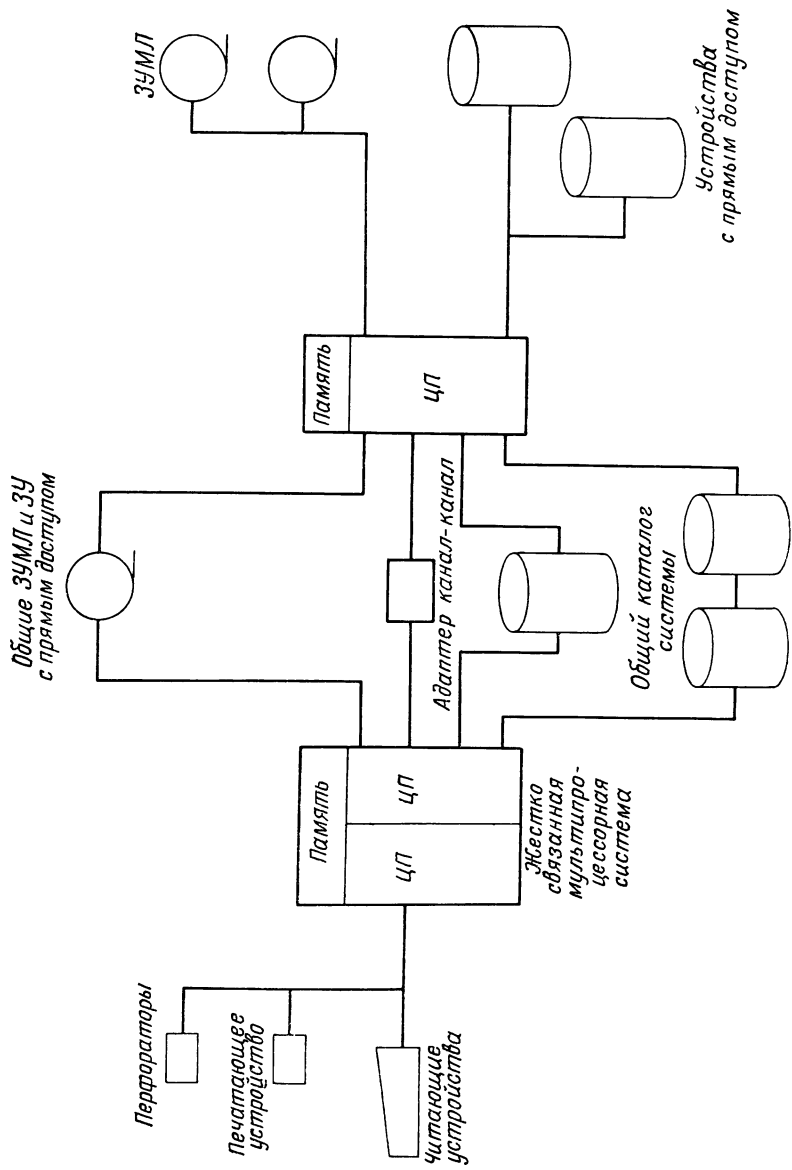


Рис. П.Х.1. Конфигурация жестко связанной мультимикропроцессорной системы фирмы IBM.

УВВ — устройство ввода-вывода.

Когда центральный процессор при работе в мультимикропроцессорной конфигурации выходит из строя, он генерирует специальный сигнал, прежде чем перейти в состояние длительного ожидания или останова. Средства восстановления в операционной системе OS/VS2 другого центрального процессора предпринимают действия по передаче на исправный центральный процессор работы, которую выполнял отказавший процессор.



Общий банк файлов

Рис. П.Х.2. Конфигурация многомашиной системы фирмы ИВМ. Жестко связанный мультипроцессор в слабо связанной системе.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОРЫ

Процессоры системы 370 совместимы («снизу вверх») с процессорами системы 360. В системе 370 предусмотрены следующие новые средства:

Средства динамической перекодировки адреса. Эти технические средства центрального процессора обеспечивают организацию виртуальной памяти в операционной системе. Динамическая перекодировка адреса позволяет использовать до 16 777 216 байтов виртуальной памяти.

Средства индикации времени. Эти средства включают часы астрономического времени, компаратор времени, таймер центрального процессора, а также таймер интервалов, который имеется и в системе 360.

Блок-мультиплексный канал. В таком канале можно выполнять в режиме совпадения несколько канальных программ, обслуживающих быстродействующие периферийные устройства.

Средства контроля ошибок и восстановления. Это средства обнаружения и исправления ошибок основной памяти, повторения команд центральным процессором и повторения операций ввода-вывода с целью уменьшения числа сбоя в системе.

ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ МУЛЬТИПРОЦЕССОРА СИСТЕМЫ 370

В дополнение к специальным командам, необходимым для выполнения таких функций, как, например, динамическая перекодировка адреса, в систему 370 включены также три класса команд, которые уже применялись в системе 360 для организации мультипроцессорной обработки:

Префиксация. Каждый центральный процессор использует нижний блок памяти (с адресами от 0 до 4096) для хранения управляющих данных и ключей состояния системы. Префиксация дает возможность логически перенести этот блок для каждого центрального процессора в различные физические блоки в основной памяти.

Синхронизация часов астрономического времени позволяет синхронизировать отсчет времени в двух центральных процессорах мультипроцессорной системы.

Сигнализация и ответы центрального процессора. Предусмотрена возможность связи между процессорами посредством команды «сигнал процессора». Эта команда обеспечивает передачу и прием сигнала, декодирование соответствующих кодов приказов и выдачу ответов на сигналы центрального процессора. Для организации связи между центральными процессорами в мультипроцес-

сорной системе используются 12 приказов, таких, например, как «Внешний вызов», «Экстренный сигнал», «Пуск», «Стоп», «Начальная загрузка программ» и «Сброс центрального процессора».

РАЗДЕЛЯЕМАЯ ОСНОВНАЯ ПАМЯТЬ

Модель 158MP (мультипроцессорная конфигурация) состоит из двух процессоров, каждый из которых имеет доступ к памяти одного и того же объема. Емкость общей памяти может составлять 1, 2, 3, 4, 6 и 8 млн. байтов. Для модели 168MP емкость общей памяти можно изменять от 2 до 16 млн. байтов путем наращивания блоками по 1 млн. байтов. Процессоры могут работать в мультипроцессорном режиме или как самостоятельные однопроцессорные системы, каждый со своей собственной частью памяти. В последнем случае область памяти для каждого процессора задается оператором с панели управления конфигурацией системы, причем вся память может быть целиком назначена одному процессору или же может быть разделена между двумя процессорами.

МОДУЛЬ УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Каналы в системе 370 делятся на три типа: селекторный, мультиплексный (как в системе 360) и блок-мультиплексный.

При передаче по блок-мультиплексному каналу его контроллер отключается от периферийного устройства на время выполнения операций, не связанных с передачей данных, и становится доступным для операций ввода-вывода другого устройства. Таким образом, операции передачи данных чередуются во времени. Блок-мультиплексный канал может также работать в режиме селекторного канала, обеспечивая совместимость с селекторным каналом системы 360.

Система 158MP может содержать до 10 блок-мультиплексных каналов и до 4 байт-мультиплексных каналов (максимальное число всех каналов не должно превышать 12). Общая скорость передачи в системе 158MP составляет 7,5 млн. байтов/с.

Система 168MP может содержать до 22 блок-мультиплексных каналов, а всего 24 канала, включая байт-мультиплексные. Общая скорость передачи данных в системе 168MP достигает 28 млн. байтов/с.

ВСПОМОГАТЕЛЬНАЯ ПАМЯТЬ

Запоминающая установка на магнитных дисках 3330 предоставляет пользователю высокоскоростную память большой емкости с прямым доступом. В установке используются пакеты дисков емкостью по 100 млн. байтов, а максимальная емкость всей установки равна 800 млн. байтов. Установка 3330 может быть подсоединена к

блок-мультиплексному каналу через контроллер 3830. Модули запоминающего устройства типа 3330 могут также подсоединяться к каналу через систему Управления интегрированной памятью. К каждой системе можно подсоединить до четырех установок 3330 (с общим числом дисководов 32). Установка 3330 с контроллером 3830 может использоваться совместно восемью центральными процессорами в конфигурации слабо связанной системы.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Мультипроцессорная работа системы 370 осуществляется под управлением операционной системы OS/VS2. Эта управляющая программа обеспечивает организацию виртуальной памяти, работу системы в режиме разделения времени и мультипроцессорную обработку. OS/VS2 позволяет создавать как слабо связанные, так и жестко связанные мультипроцессорные системы.

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Язык управления заданиями (JCL) операционной системы OS/VS2 в основном совпадает с языком OS/MVT, что обеспечивает совместимость этих операционных систем. Многие из действующих систем 360 использовали программные средства HASP или ASP для организации системного ввода и буферизации. В операционной системе OS/VS2 имеется подсистема ввода заданий, которая по существу сходна с системами HASP и ASP. Пользователь может выбрать любые стандартные средства операционной системы. Средства системного ввода и буферизации обеспечивают выполнение следующих функций:

Считывание и ввод в систему заданий.

Планирование выполнения заданий.

Подготовка данных для обрабатываемых заданий.

Поддержка средств управления системой.

Обработка всей выходной информации для заданий, выполняемых в пакетном режиме и режиме разделения времени.

Регулировщик системных ресурсов операционной системы OS/VS2 обеспечивает средства загрузки системы в соответствии с установленной производительностью ее функциональных устройств и управляет использованием системных ресурсов. Производительность системы и ее частей можно достаточно полно выразить при помощи совокупности коэффициентов, называемых коэффициентами интенсивности обслуживания; они характеризуют скорость предоставления пользователю ресурсов центрального процессора, системы ввода-вывода и запоминающих устройств при определенном уровне

загрузки системы. Для этих коэффициентов используются следующие единицы измерения:

Работа центрального процессора: одна единица соответствует выполнению 10 000 команд.

Работа системы ввода-вывода: количество событий ввода-вывода.

Загрузка реальной памяти: один занятый фрагмент для некоторого числа единиц работы центрального процессора.

Интенсивность обслуживания для всей системы будет определяться как линейная комбинация указанных выше факторов, умноженных на некоторые заданные заранее числовые коэффициенты. Задавая уровни загрузки различных функциональных устройств, можно определить, как соотносится качество обслуживания различных групп пользователей при разных уровнях загрузки системы. Система следит за изменениями интенсивности обслуживания пользователей и средним уровнем загрузки системы и осуществляет регулировку этих показателей.

Режим разделения времени, который в системе 360 представляет собой один из дополнительных режимов (TSO) операционной системы OS/MVT, в операционной системе OS/VS2 обеспечивается стандартными функциями управляющих программ. Все возможности, предусмотренные в TSO, полностью содержатся в системе разделения времени OS/VS2, за исключением командного языка, который остается необязательным.

Управляющая программа назначает собственную область виртуальной памяти, объемом 16 777 216 байтов для следующих пользователей и системных элементов:

Каждое задание в обрабатываемом пакете, запланированном инициатором.

Каждое задание, обрабатываемое в режиме разделения времени.

Главный планировщик.

Система ввода заданий.

Система виртуального телекоммуникационного доступа.

Система управления дополнительной памятью.

Несмотря на то что каждому заданию пользователя выделяется собственная область виртуальной памяти, задание не может управлять адресацией всей этой области. Каждая область разделена на системную подобласть, подобласть пользователя и подобласть общего назначения. Страницы памяти, используемые в виртуальной системе, имеют емкость по 4К байтов. Реальная память также делится на страницы емкостью по 4К байтов. Система управления виртуальной памятью может использовать при обработке заданий сегменты, объединяющие до 16 страниц (64К байтов). Таким образом, максимальная виртуальная память, назначенная пользователю, состоит из 256 сегментов.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Все операции ввода-вывода в системе 370 делятся в супервизоре ввода-вывода на два класса. Основная часть супервизора ввода-вывода управляет устройствами, контроллерами и каналами ввода-вывода. Супервизор обеспечивает также выполнение таких функций, как перекодировка адресов из виртуальных в абсолютные, фиксация страниц, контроль правомерности обращения, а также организует очереди запросов (отдельно для каждого типа запросов).

Динамическое распределение вспомогательной памяти, которое раньше выполнялось только в режиме разделения времени, в данном случае может осуществляться по запросам как высокоприоритетных, так и низкоприоритетных заданий. Наборы данных могут быть закрыты и не распределяться до окончания выполнения задания в целом или шага задания.

ДИАГНОСТИЧЕСКИЙ КОНТРОЛЬ ОШИБОК

Когда в одном из центральных процессоров происходит сбой и он не может продолжать нормально функционировать, включается система восстановления другого процессора. Происходит сброс системы ввода-вывода на отказавшем центральном процессоре и предпринимаются попытки повторно осуществлять операции ввода-вывода при помощи устройств, симметрично подсоединенных к каналам работающего центрального процессора. Если один из трактов канала отказал, то такая попытка будет повторена с другим трактом.

Новые команды и программное обеспечение позволяют динамически изменять конфигурацию сети периферийных устройств, отключая от системы источники ошибок, без перезапуска системы или ненормального окончания обработки текущего задания.

ПРИЛОЖЕНИЕ XI

ЭВМ ВОЕННОГО НАЗНАЧЕНИЯ МОДЕЛИ 215 ФИРМЫ RCA

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система RCA, модель 215, представляет собой мультипроцессор общего назначения с запоминаемой программой. Эта ЭВМ функционально совместима с ЭВМ серии SPECTRA 70 и в отличие от последней включает десять дополнительных команд для управления мультипроцессором и восстановления его работы после сбоев. Модель 215 была разработана с учетом специфических военных требований 70-х годов и поэтому отвечает соответствующим военным стандартам.

Базовая конфигурация модели 215 предусматривает наличие от 2 до 4 центральных процессоров, от 2 до 4 блоков ввода-вывода и от 1 до 8 модулей основной памяти. Блок распределения сигналов (БРС) осуществляет взаимосвязь процессоров и модулей памяти; он состоит из отдельных частей, каждая из которых снабжается энергией от соответствующего активного блока (каждый блок системы имеет свой собственный источник питания). Любое активное устройство, включая принадлежащую ему часть БРС, может быть автоматически выделено из системы и отключено, в то время как оставшиеся части системы будут продолжать нормально функционировать.

Каждый блок ЭВМ является автономным, включая указанную часть БРС, через которую осуществляется взаимосвязь этого устройства с системой. Система сконструирована таким образом, что к ней могут добавляться различные блоки без модификации других блоков; блоки являются взаимозаменяемыми; отказ какого-либо блока не влияет на работу других блоков системы.

Как видно из схемы на рис. П.ХI.1, БРС — это сердце системы, так как все связи и обмены информацией осуществляются и управляются этим блоком. БРС является по существу сетью логических схем операции исключающего ИЛИ, взаимосвязанных между собой, как показано на рис. П.ХI.2, и обеспечивающих, например, одновременный доступ двух блоков ввода-вывода и двух центральных процессоров к любым четырем из восьми модулей основной памяти. В случае одновременного обращения двух устройств ввода-вывода к

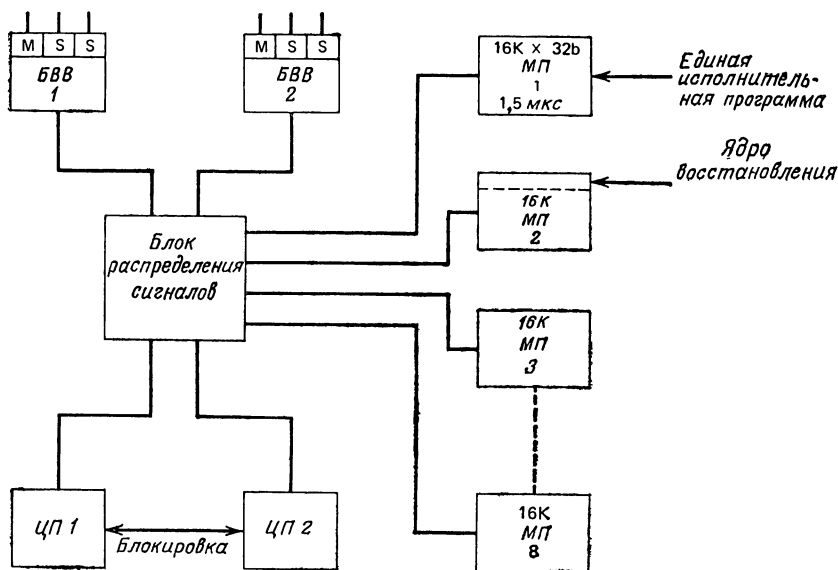


Рис. П.XI.1. Базовая конфигурация мультипроцессорной системы 215 фирмы RCA. М — мультиплексный канал, S — селекторный канал.
БВВ — блок ввода-вывода, МП — модуль основной памяти.

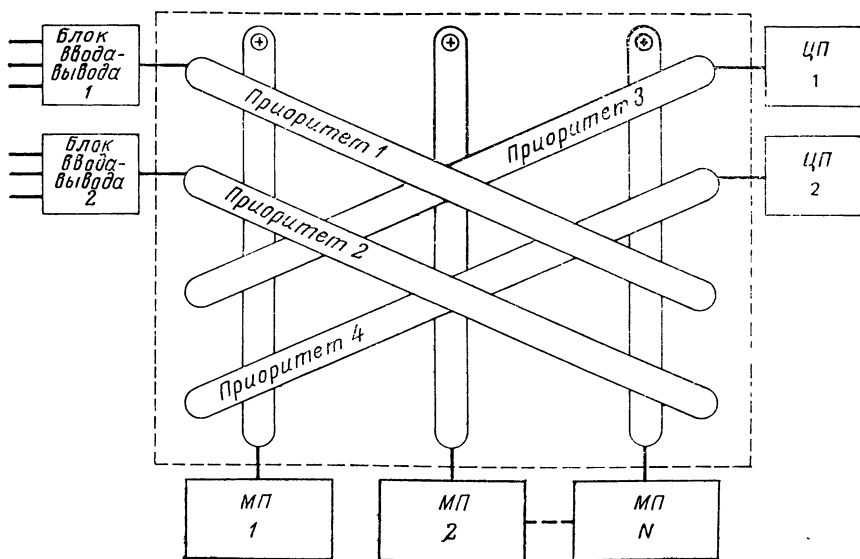


Рис. П.XI.2. Схема блока распределения сигналов с системой приоритетов доступа к основной памяти.
МП — модуль основной памяти.

какому-либо одному модулю основной памяти предпочтение будет отдано блоку ввода-вывода 1, обладающему более высоким приоритетом, чем блок ввода-вывода 2 (рис. П.ХI.2). Аналогично, если два центральных процессора обращаются к одному модулю основной памяти в одно и то же время, то центральный процессор 1 имеет более высокий приоритет, чем центральный процессор 2. В случае конфликта между блоком ввода-вывода и центральным процессором первый будет иметь более высокий приоритет, чем второй. Схемы БРС (главным образом устройства управления линиями и приемники) разбиты на отдельные блоки и снабжаются энергией от соответствующих активных устройств. Каждый из четырех процессоров (два процессора ввода-вывода и два центральных процессора) могут иметь одновременный доступ к любому из четырех модулей основной памяти. Такая организация позволяет системе обеспечивать суммарное быстродействие блоков ввода-вывода до 4 млн. байтов/с и суммарное быстродействие двух центральных процессоров, равное удвоенной величине 410К операций/с. Таким образом, система может быть расширена от минимальной конфигурации, содержащей один центральный процессор, один модуль основной памяти емкостью 16К слов и один блок ввода-вывода, до максимальной конфигурации с четырьмя центральными процессорами, восемью модулями основной памяти емкостью по 32К слов и четырьмя блоками ввода-вывода.

Все команды, управляющие слова, сигналы подтверждения, сигналы прерывания и ошибок, передаваемые между центральными процессорами, блоками ввода-вывода и модулями основной памяти, проходят через БРС. Этот блок состоит из 8 «отдельных блоков основной памяти», 4 «отдельных блоков процессора» и одного «отдельного блока управления эксплуатацией и восстановлением». Каждый модуль основной памяти связан с отдельным блоком основной памяти БРС и в свою очередь каждый блок ввода-вывода и центральный процессор связаны с соответствующими отдельными блоками процессора в БРС. Все отдельные блоки основной памяти в БРС идентичны, за исключением их физического адреса. Физический адрес модуля основной памяти определяется отдельным блоком в БРС, с которым связан этот модуль, и соответствует 3 старшим разрядам 19-разрядного адреса, определяющего местоположение байта в основной памяти. Все отдельные блоки процессора в БРС идентичны с точки зрения передачи сигналов между блоками ввода-вывода или центральным процессором и модулем основной памяти, за исключением их приоритетов. Отдельные блоки ввода-вывода идентичны между собой, за исключением маршрутов передачи сигналов прерывания и команд. Отдельные блоки центрального процессора также идентичны, за исключением их приоритета при управлении устройствами ввода-вывода.

Уникальной особенностью системы 215 является наличие ядра восстановления операционной системы; это устраняет проблему сбоя в модуле основной памяти, содержащем исполнительные программы. (Заметим, что на рис. П.ХІ.1 модуль основной памяти 1 является резидентным для единой исполнительный программы, управляющей всеми ресурсами системы в целом. В данном случае отсутствует взаимодействие между центральными процессорами по принципу «ведущий — ведомый». Каждый из них независимо обрабатывает программы самого высокого приоритета.) Ядро восстановления состоит из двух частей: одна часть представляет собой программу предварительного ввода, которая вызывает из внешнего запоминающего устройства другую копию единой исполнительный программы; другая часть содержит копию текущего состояния всех программ, которые обрабатывались в системе в момент появления сбоя памяти. Такая таблица состояния содержит информацию, позволяющую системе повторно приступить к выполнению операций с того места, где произошел сбой. Как один из вариантов, система может быть снабжена часами истекшего времени, обеспечивающими прерывание при завершении определенных временных интервалов. В системе 215 предусмотрены широкие возможности распознавания ошибок; она может обнаруживать: неисправности главного источника питания, сбой источников питания любого центрального процессора, блока ввода-вывода или модуля основной памяти, ошибки в командах, выявленные контролем четности, и сбой при обработке данных и управлении аппаратными средствами. Каждый центральный процессор может принимать сигналы, указывающие на сбой в других элементах системы, включая неисправности главного источника питания.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Центральный процессор системы 215 состоит из арифметического устройства и устройства управления, которые могут адресоваться к основной памяти, выполнять команды, инициировать операции ввода-вывода, выполнять машинные проверки и управлять вычислительными операциями. Регистры общего назначения, регистры с плавающей запятой, разряды состояния-управления и регистры передачи информации состояния-управления выполнены в виде быстродействующей буферной памяти блок-нотного типа емкостью 64 36-разрядных слов с временем цикла 300 нс. Центральный процессор может обращаться к основной памяти с временем цикла 1,5 мкс, что соответствует быстродействию 410 000 операций/с. Форматы данных, обрабатываемых системой, имеют 16 или 32 разряда для чисел с фиксированной запятой и 32 или 64 разряда для чисел с плавающей запятой. При каждом обращении к памяти центральный процессор

выбирает 36-разрядное слово, содержащее четыре 8-разрядных байта и 4 разряда контроля четности.

Система 215 имеет стандартный набор из 153 команд. Этот набор включает операции над числами с фиксированной и плавающей запятой, операции условного перехода, логические операции, операции десятичной арифметики, привилегированные операции, операции мультипроцессорной обработки и восстановления после ошибок.

Каждый центральный процессор является вполне независимым устройством, имеет отдельный источник питания и может реализовать исполнительные команды, получать доступ к основной памяти и управлять устройствами ввода-вывода независимо от действия другого центрального процессора. Сбой какого-либо центрального процессора или выполнение операций по обнаружению и устранению возникших в нем неисправностей не влияют на работу остальной части системы.

В системе 215 предусмотрено 32 уровня прерываний. Приоритет первого уровня назначается прерываниям по сбою источников питания и машинному контролю (условия машинного прерывания). Второй уровень связан с сигналами прямого управления, передаваемыми от одного центрального процессора к другому. Приоритеты от 9-го до 24-го уровня используются для обработки прерываний по вводу-выводу. Приоритет 25-го уровня зарезервирован для прерываний по часам истекшего времени и приоритет 26-го уровня — для прерываний, осуществляемых вручную с пульта оператора. Приоритет 27-го уровня предназначен для набора из 8 программных прерываний, приоритеты от 28-го до 31-го уровня выделены для индивидуально маскируемых программных прерываний, а приоритет 32-го уровня оставлен для контроля выполнения программ в центральном процессоре в режиме тестирования.

ОСНОВНАЯ ПАМЯТЬ

Основная память системы 215 представляет собой запоминающее устройство на ферритовых сердечниках емкостью 16 384 36-разрядных слова (четыре 8-разрядных байта и четыре разряда контроля четности). Каждый модуль памяти имеет свой собственный набор устройств управления, буферных регистров и адресных регистров. Кроме того, каждый модуль памяти имеет собственный источник питания. Объем памяти можно увеличивать наращиванием модулей по 16К или 32К слов; в максимальной конфигурации основная память может состоять из 8 модулей. Время цикла работы памяти равно 1,5 мкс. Каждый модуль имеет средства защиты памяти при записи и при считывании-записи, которые представляют собой ключи защиты, реализованные в блоке памяти емкостью 2048 байтов; ключ защиты памяти состоит из 5 разрядов (1 разряд определяет тип защиты, 4 разряда — ключ). Кроме того, с каждым центральным процессором и блоком ввода-вывода связаны ключи защиты, реали-

зуемые при помощи аппаратных средств. Каждый модуль памяти функционирует независимо от сбоев в другом модуле памяти или каких-либо мероприятий по устранению сбоев в любом другом блоке системы.

МОДУЛЬ УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Передача данных от внешних источников в систему 215 осуществляется через блок ввода-вывода. Обычно передача данных иницируется центральным процессором, и поток данных из основной памяти поступает в блок ввода-вывода или из этого блока в определенный модуль основной памяти только через блок распределения сигналов. Имеется два типа каналов, связанных с блоками ввода-вывода, — мультиплексные и селекторные каналы. Каждый блок ввода-вывода имеет быстродействующее буферное запоминающее устройство блок-нотного типа для хранения управляющей информации канала, а каналные регистры используются для приема передаваемых данных и для передачи управляющей информации и информации о состоянии канала в центральный процессор. Скорость передачи в мультиплексном канале равна 400 000 байтов/с, а в селекторном канале 800 000 байтов/с. Система 215 может обслуживать через каждый блок ввода-вывода один мультиплексный канал на 64—128 устройств ввода-вывода и два селекторных канала на 256 устройств ввода-вывода с общей скоростью 4 млн. байтов/с.

МАССОВАЯ ИЛИ ВСПОМОГАТЕЛЬНАЯ ПАМЯТЬ

Стандартное оборудование, предлагаемое фирмой RCA, состоит из следующих устройств:

Запоминающее устройство на магнитных дисках, включающее до 8 модулей, каждый из которых имеет емкость 7,25 млн. байтов на 1 сменный пакет дисков. Данные могут передаваться между процессором и пакетом дисков со скоростью 156 000 байтов/с. Среднее время поиска равно 75 мс, максимальное время — 135 мс. Запоминающее устройство на магнитном барабане емкостью 4,12 млн. или 8,25 млн. байтов. Среднее время доступа равно 17 мс, скорость передачи данных составляет примерно 325 000 байтов/с. Запоминающее устройство на магнитных дисках емкостью 0,5 млн. или 1,0 млн. байтов. Среднее время доступа равно 20 мс, скорость передачи данных составляет примерно 280 000 байтов/с.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Средства программного обеспечения системы RCA 215 в основном совпадают с операционной системой SPECTRA 70 TDOS. Эта система предназначена для выполнения в режиме совпадения до 6 программ,

разбитых на фрагменты различных размеров. Можно запланировать выполнение программ либо под управлением монитора, либо под управлением оператора с пульта системы. Любая программа может выполняться как часть цепочки программ, которые обрабатываются последовательно. Операции в этом режиме завершаются либо вызовом новой программы из цепочки обрабатываемых программ, либо при помощи операторов управления, включенных в поток заданий.

Области основной памяти назначаются для каждого задания в соответствии с требованиями, предъявляемыми заданием, и имеющимся свободным объемом памяти во время инициации.

Распределение памяти регулируется при помощи таблицы свободной памяти. Эта таблица определяет те области памяти, которые в текущее время еще не заняты. Конкретные устройства ввода-вывода могут назначаться для заданий при помощи команд управления, включенных в поток заданий, при инициации программы или с пульта управления оператора. Программа может динамически назначать устройства в процессе своего выполнения.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Стандартные программы ввода-вывода иницируют все операции ввода-вывода на всех устройствах, за исключением автоматической пишущей машинки на пульте системы. Составление очередей и последующее иницирование запросов на ввод-вывод осуществляются по принципу «первый пришел — первый обслужен» при условии, что устройства связи с удаленными пользователями не участвуют в работе. Если же система поддерживает связь с удаленными объектами, то операции ввода-вывода по каналам связи получают наивысший приоритет.

Система имеет контроллер файла, который позволяет пользователю выполнять операции записи или чтения на логическом уровне, без непосредственного взаимодействия с физическими устройствами. Этот логический уровень обеспечивает буферизацию в памяти, осуществляет подготовку логических записей (объединение в блоки и распаковка блоков) и разблокирование файлов для обработки в различных приложениях. Процессор управления файлами обеспечивает обработку файлов, записанных последовательно, например, на магнитной ленте, а также файлов, хранящихся в последовательном или произвольном порядке в запоминающих устройствах с прямым доступом.

Программа управления передачей данных от удаленных пользователей допускает выполнение в режиме совпадения одной программы передачи данных и пяти других программ, не связанных с передачей. Эта программа управляет всеми операциями телеобработки данных.

В контроллер файла входят также подпрограммы проверки и (или) записи стандартных меток файла на магнитной ленте.

ПРИЛОЖЕНИЕ XII

ОРТОГОНАЛЬНЫЕ ЭВМ OMEN-60 ФИРМЫ SANDERS ASSOCIATES

ВВЕДЕНИЕ

ЭВМ OMEN-60 фирмы SANDERS представляют собой семейство совместимых («снизу вверх») мини-ЭВМ с ортогональными матричными процессорами. Они предназначены для обработки в реальном масштабе времени данных, поступающих с высокой скоростью. По сравнению с другими системами в процессорах системы OMEN-60 особое внимание уделено быстрой связи между периферийными устройствами и ортогональной памятью большой емкости с двумя режимами доступа: обычным режимом пословного доступа и режимом вертикального доступа (одновременный доступ к разрядному срезу 64 слов).

В каждой ЭВМ OMEN-60 имеется два центральных процессора, которые обращаются к одному и тому же запоминающему устройству. Эту систему нельзя рассматривать как набор отдельных элементов со сложной организацией взаимосвязей. Один из центральных процессоров является обычным серийным процессором PDP-11 фирмы Digital Equipment, другой представляет собой вертикальное арифметическое устройство. Вертикальное арифметическое устройство системы OMEN-61 и -62 предназначено в основном для ассоциативных операций и операций над базами данных. Вертикальные арифметические устройства старших моделей серии 60 представляют собой идеальное средство для арифметических операций и весьма эффективны при обработке чисел с плавающей запятой и целых чисел.

Для программирования ЭВМ OMEN используются языки высокого уровня (например, расширенный язык Фортран-IV); они функционируют под управлением дисковой операционной системы. Поскольку система OMEN-60 включает в качестве основного элемента ЭВМ PDP-11, ее набор компиляторов OMEN и операционная система базируются на программном обеспечении фирмы Digital Equipment.

ОРГАНИЗАЦИЯ СИСТЕМЫ

Основой семейства ортогональных ЭВМ OMEN является специально разработанная система памяти — ортогональная память, доступ к которой может производиться в двух режимах. Как показано на рис. П.ХII.1, такая память предоставляет процессору PDP-11

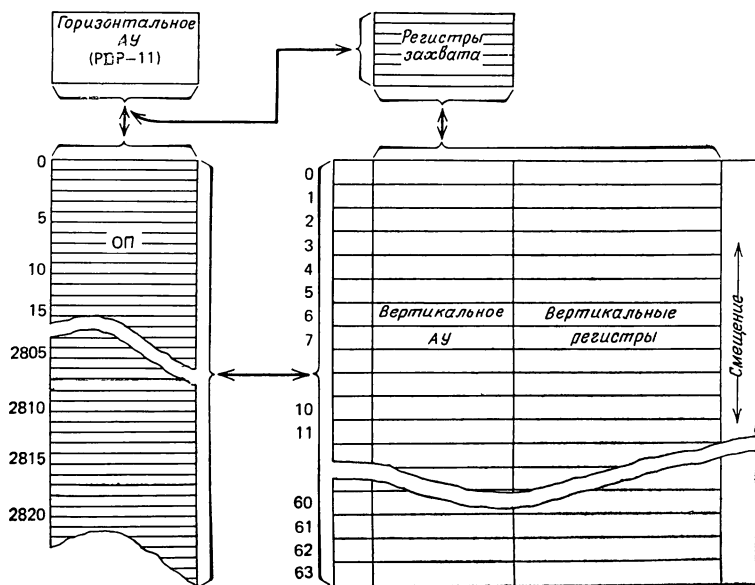


Рис. П.ХII.1. Схема организации системы OMEN-60.
ОП — ортогональная память, АУ — арифметическое устройство.

(«горизонтальное арифметическое устройство») в ответ на его запрос все разряды одного слова. Сочетание горизонтального арифметического устройства и ортогональной памяти представляет собой обычную машину фон Неймана. Потенциальные возможности системы OMEN обусловлены наличием второго режима доступа. В ответ на запрос вертикального арифметического устройства память предоставляет ему набор всех n -х разрядов 64 слов. Таким образом, последовательность вертикальных регистров может быть загружена для параллельной обработки 64 слов. Выборка разрядного среза слов используется в вертикальном арифметическом устройстве, представляющем собой набор параллельных элементарных процессоров. Эти логические элементы идентичны и выполняют в каждый момент времени одни и те же операции (за небольшими исключениями, которые задаются масками).

Следует особо отметить тот факт, что, как видно из рис. П.ХИ.2, несмотря на физическое разграничение, например, между словами 63 и 64 или словами 127 и 128, логическая граница между ними с точки зрения процессора PDP-11 отсутствует. Такая память представляет собой память обычного типа. Аналогично вертикальное

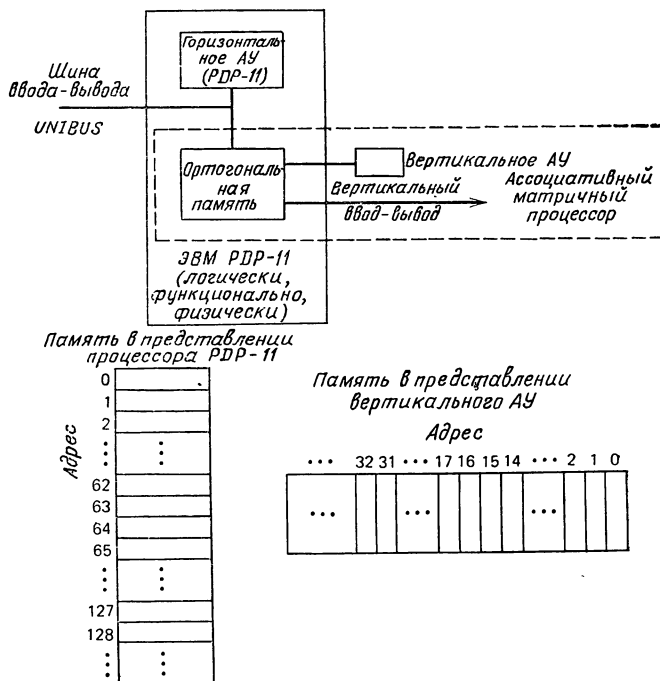


Рис. П.ХИ.2. Память системы OMEN-60.
АУ — арифметическое устройство.

арифметическое устройство произвольно адресуется к памяти и не чувствует границу между сечениями разрядов, например, с номерами 15 и 16 даже в тех случаях, когда между ними существует физическая граница. Вертикальное арифметическое устройство может также адресоваться к каждому байту, так что можно обращаться к блоку слов как к группе байтов, начинающейся с байта 101, столь же быстро, как к вектору слов или байтов, начинающемуся с байта 256. Если из блок-схемы исключить процессор PDP-11, то оставшаяся часть в штриховом прямоугольнике на рис. П.ХИ.2 будет обычным ассоциативным процессором или, как его иногда называют, ассоциативным матричным процессором.

Быстродействие системы OMEN-64 определяется частично быстродействием 64 процессоров, работающих параллельно, а также тем, что предусмотрен обширный набор машинных команд, и поэтому вызывается из памяти и декодируется гораздо меньшее число команд, чем в других системах. Многие из алгоритмов линейной алгебры и анализа сигналов требуют здесь для реализации не больше одной команды. Широкие возможности практических приложений системы OMEN-64 обусловлены, в частности, реализацией арифметических операций над числами с плавающей запятой в вертикальном арифметическом устройстве и унификацией системы, не предусматривающей отдельного устройства управления. Например, параллельный процессор с отдельным устройством управления не способен так легко контролировать взаимозависимость данных при выполнении команд, как ортогональная ЭВМ.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОРЫ

ГОРИЗОНТАЛЬНОЕ АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО

Горизонтальное арифметическое устройство представляет собой ЭВМ PDP-11 фирмы Digital Equipment. Это стандартный серийный процессор, оперирующий с 16-разрядными словами. Он снабжен 8 регистрами общего назначения и имеет набор из 400 команд. Все стандартные средства программного обеспечения PDP-11 пригодны для горизонтального арифметического устройства.

ВЕРТИКАЛЬНОЕ АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО

Вертикальное арифметическое устройство состоит из 64 идентичных элементарных процессоров, работающих в режиме пошаговой блокировки. Их сложность зависит от номера модели: в младших моделях используются одноразрядные сумматоры и логические схемы с 8-разрядным быстродействующим буферным регистром, в старших моделях — 16-разрядные сумматоры и логические схемы с 8 быстродействующими 16-разрядными буферными регистрами.

На рис. П.ХII.3 представлены некоторые подробности структуры системы OMEN-60. Каждый логический уровень в вертикальном арифметическом устройстве имеет логику смещений, которая включает следующие операции:

Инверсия. Эта операция перемещает данные «вверх дном»: уровень 0 меняется местом с уровнем 63, уровень 1 — с уровнем 62 и т. д.

Полное перемешивание. Эта операция аналогична делению колоды карт на две части с последующим чередованием карт из двух половин колоды: уровень 0 остается на месте, уровень 32 оказывается на месте уровня 1, уровень 1 — на месте уровня 2, уровень 33 — на месте уровня 3 и т. д.

Цилиндрическое смещение. Эта операция позволяет исключать или передавать по кругу вертикальные сдвиги.

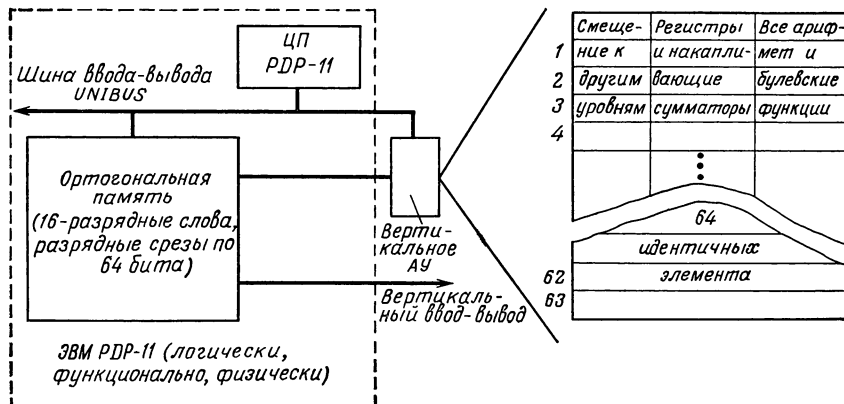


Рис. П.ХИ.3. Архитектура системы OMEN-60.

АУ — арифметическое устройство.

На каждом уровне вертикального арифметического устройства находится несколько регистров (емкостью 133 бита в системе OMEN-64): 8 16-разрядных регистров и 5 регистров масок. В этих регистрах выполняются все арифметические и булевские операции. При выполнении операции поразрядного сложения перенос устанавливается в нулевое начальное состояние, а затем на каждом временном цикле новые биты переноса заменяют старые в регистре переноса. Три входа в арифметико-логическое устройство, генерирующее сумму,—это два операнда и регистр переноса. Необходимо отметить, что во время операции сложения или булевой операции, которая выполняется тем же способом (но без участия регистра переноса), обработка начинается с младшего разряда операндов и проходит через все поле данных к более высоким разрядам. На каждом последовательном временном цикле работы машины выбирается следующий разрядный срез слов (от самого младшего разряда), и над ним выполняется операция; таким образом, если производится выборка из памяти, то заранее известно, что вторая выборка будет производиться по следующему разрядному срезу следующего самого

младшего разряда; в третий раз выборка будет производиться по третьему самому младшему разряду и т. д. Следовательно, чередование памяти дает здесь большие преимущества, так как при этом не возникает проблем, связанных с попыткой доступа к одной и той же физической памяти дважды за время одного рабочего цикла. Этот процесс аналогичен процессу выборки векторного числа из обычной памяти с чередованием, где также известно, что последовательные элементы вектора должны размещаться в физически раздельных запоминающих устройствах. Адресация памяти в ЭВМ серии OMEN фактически является побайтовой как для вертикального арифметического устройства, так и для процессора PDP-11. По тем же причинам, по которым процессор PDP-11 не чувствует границу между словами 63 и 64, доступ, связанный с выборкой или запоминанием блока целых чисел или символов, не обязательно должен производиться в границах определенного физического блока памяти. Другая особенность системы OMEN-64 — это средства, позволяющие организовать распределенный доступ к памяти. Они дают возможность каждому логическому уровню в вертикальном арифметическом устройстве определять, какие данные он будет обрабатывать. Единственное ограничение этого типа доступа связано с тем, что каждый уровень логики в вертикальном арифметическом устройстве должен присваивать свой номер слову, которое перемещается из физического блока.

Разные модели семейства OMEN-60 отличаются друг от друга своими арифметическими устройствами. Системы OMEN-61 и -62 имеют одноразрядную логику, позволяющую эффективно использовать эти модели для обработки данных, не требующей операций умножения и деления (например, для сортировки, распознавания образов и поиска). Системы OMEN-63 и -64 имеют сумматоры, обрабатывающие полное слово, и арифметические устройства для операций над числами с плавающей запятой, реализованные аппаратно; благодаря этому они удобны для решения задач линейной алгебры и других сложных арифметических вычислений. Модель 61 отличается от модели 62 только числом регистров на каждом уровне вертикального арифметического устройства.

ОРТОГОНАЛЬНАЯ ПАМЯТЬ

Ортогональная память сконструирована на базе электронных МОП-схем. В ней используется метод чередования как при вертикальном, так и при обычном (горизонтальном) доступе, причем оба типа доступа являются рандомизированными. Скорость передачи при горизонтальном доступе равна 2 М слов/с, а при вертикальном доступе — 45 М слов/с. Допускается емкость памяти от 8К до 128К 16-разрядных слов для моделей OMEN-61 и 62 и от 16К до 128К слов для моделей OMEN-63 и -64.

ВВОД-ВЫВОД

Характеристики системы ввода-вывода и периферийного оборудования в системе OMEN-60 аналогичны характеристикам этих устройств для ЭВМ PDP-11, так как основной канал ввода-вывода OMEN совпадает с каналом ввода-вывода ЭВМ PDP-11.

Устройства массовой памяти могут быть представлены в нескольких формах. Обычные запоминающие устройства на магнитных дис-

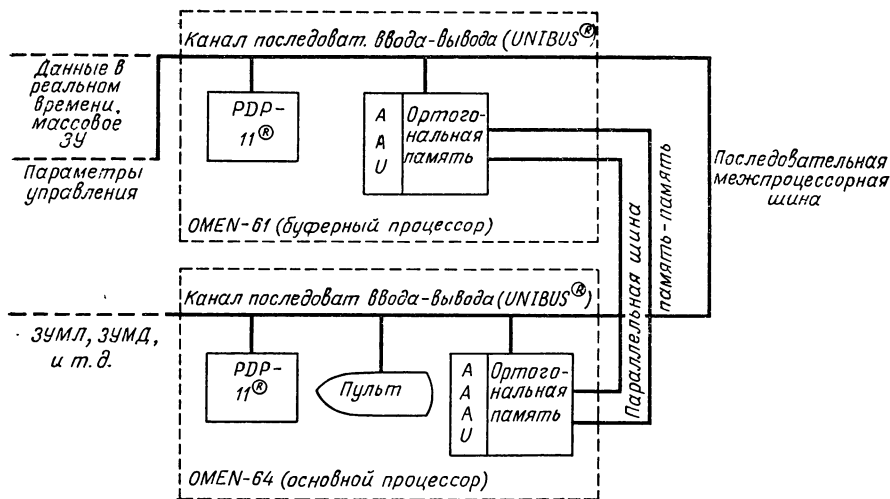


Рис. П.ХП.4. Типичное применение двухпроцессорной системы. (С пульта оператора осуществляется управление процессорами OMEN-61 и OMEN-64.)

как или лентах могут быть подсоединены к шине UNIBUS и позволяют при помощи операционной системы OMEN создавать необходимые системные файлы. Для тех случаев применения, когда скорость передачи данных дискового запоминающего устройства оказывается критической, в системе OMEN имеется дополнительная возможность подключить вертикальный канал ввода-вывода со скоростью передачи 8 000 000 разрядных срезов/с (что равно 32 000 000 слов/с) к специальным дискам с комбинированными головками или фиксированным пакетам дисков. В приложениях с такими высокими скоростями передачи данных обычно применяется конфигурация системы, в которой ЭВМ OMEN-61 используется в качестве буферного процессора для преобразования данных в требуемый формат, т. е. как буферное устройство между системой OMEN-64 и быстродействующим каналом ввода-вывода.

На рис. П.XII.4 показана типичная двухпроцессорная конфигурация, где небольшой процессор, в данном случае ЭВМ OMEN-61, функционирует в качестве буферной памяти и устройства для преобразования формата данных между быстродействующим каналом передачи данных в реальном времени и основным процессором. В тех случаях, когда данные поступают из каналов с циклическим опросом, ЭВМ OMEN-61 обеспечивает упорядочение чередующихся данных, а также выполнение других операций преобразования формата (например, операции размещения разрядов в обратном порядке); благодаря этому основной процессор OMEN-64 остается свободным для выполнения тех операций, которые осуществляются на нем наиболее эффективно, а именно операций обработки числовых данных в таких алгоритмах, как алгоритмы преобразования Фурье, формирования радиолуча, аппроксимации сигналов, обнаружения устойчивых сигналов, анализ записей сигналов и т. д. ЭВМ OMEN-61 может также использоваться для управления освободившимися блоками системы, если в этом появляется необходимость; такие условия могут иметь место в больших системах, которые функционируют полностью под цифровым управлением. Параллельная шина память-память имеет быстродействие, превышающее больше чем в 10 раз быстродействие последовательной межпроцессорной шины, поэтому на передачу информации затрачивается лишь очень небольшое число рабочих циклов. Это происходит потому, что ширина шины память-память равна 64 разрядам, а частота синхронизации в четыре раза выше, чем частота синхронизации у шины UNIBUS.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

ИСПОЛНИТЕЛЬНЫЕ ПРОГРАММЫ

В системе OMEN-60 используется либо операционная система ЭВМ PDP-11 на диске DOS, либо операционная система ЭВМ PDP-11 для работы в реальном времени RTOS.

ЯЗЫКИ

В системе используются главным образом языки высокого уровня. Исходные операторы таких языков, как расширенный Фортран, Бейсик, АПЛ, реализованы аппаратно в системе OMEN-64.

Программные средства для системы OMEN были упрощены за счет использования средств программного обеспечения ЭВМ PDP-11 и набора команд, реализованных при помощи микропрограммирования. Пользуясь языками высокого уровня, например языком Фортран, программист может записать программу обращения матрицы размерности 8×8 в следующем виде:

```

DIMENSION  ARRAY(8, 8),  VARRAY(8, 8)
      |
      |
      | (Операторы, определяющие элементы ARRAY)
      |
      |
VARRAY=RMTINV (ARRAY)
      |
      |
END

```

Интерпретатор языка Фортран для ЭВМ PDP-11 распознает RMTINV как операцию вертикального арифметического устройства и транслирует ее в машинную команду для этого устройства. Команда содержит операционный код для «обращения матрицы», начальный адрес и размер области ARRAY. Это командное слово затем посылается через шину UNIBUS в вертикальное арифметическое устройство, где соответствующая микропрограмма устанавливает последовательность операций, необходимых для обращения матрицы. Дополнение к обычному интерпретатору языка Фортран ЭВМ PDP-11 состоит в том, что добавлены наименования векторных функций языка Фортран в таблицу библиотечных функций и введен список машинных кодов, которые генерируются при обращении к векторным функциям. Аналогичные изменения в интерпретаторе языка Бейсик позволяют использовать его для выполнения параллельных операций.

Хотя языки Бейсик и Фортран обычно широко применяются для программирования научных задач, их использование для выполнения операций линейной алгебры связано с определенными трудностями. Компиляторы системы OMEN активизируют вертикальное арифметическое устройство при помощи специальных функций, однако программа, записанная для системы OMEN, обязательно должна содержать набор подпрограмм FUNCTION, которые добавляются для того, чтобы можно было прогнать эту программу в системе с последовательной обработкой. Это несколько громоздкий подход, но он легко реализуется средствами программного обеспечения. Более совершенные системы программирования могут использовать языки, ориентированные на матричные операции, такие, как язык АПЛ. Генератор машинных кодов в интерпретаторе языка АПЛ системы OMEN очень прост, поскольку каждый исходный оператор языка АПЛ преобразуется в одну машинную команду системы OMEN.

Потенциальные возможности системы OMEN, как и любой другой вычислительной системы, во многом зависят от языка ассемблера. Высокая степень микропрограммирования обеспечивает «высокий уровень» языка ассемблера системы OMEN. Например, при помощи записи

VII;; 2=ARRAY;

программист указывает, что матрица размерности 8×8 , размещенная в области памяти, начинающейся от местоположения ARRAY, должна быть транспонирована и затем инвертирована, а полученный результат должен быть занесен в вертикальный регистр 2. Все исходные операторы, полученные за счет расширения машинных команд, имеются в языке ассемблера, поэтому при использовании микропрограммирования гибкость системы не уменьшается. Пользователь может задать собственную последовательность машинных команд, используя для этой цели микропрограммирование на основе имеющихся исходных операторов. Ассемблер системы OMEN обладает возможностями макроязыка и позволяет создавать перемещаемые модули.

Регистры многократного воспроизведения (или «регистры захвата») имеют большое значение для операций линейной алгебры. Это хорошо видно из рис. П.ХII.5, где схематически показана операция «вектор \times матрица». В данном случае вектор (V_1, V_2, \dots, V_8) вызывается в регистр захвата и воспроизводится в нем несколько раз. При этом компонента V_1 повторяется 8 раз, V_2 повторяется также 8 раз и т. д. Поэтому, если матрица A находится в вертикальном арифметическом устройстве, а вектор (V_1, V_2, \dots, V_8) должен быть многократно воспроизведен, то для этого потребуется всего лишь одна операция многократного воспроизведения. Показанное на рис. П.ХII.5 поэлементное умножение представляет собой необходимый этап операции умножения вектора на матрицу.

Для получения окончательного результата необходимо выполнить суммирование попарных произведений элементов матрицы и вектора. Это можно осуществить следующим образом: разделить полученный массив пополам, сдвинуть вверх нижнюю половину и сложить обе половины; затем повторить эту операцию еще дважды. Вектор, полученный в результате умножения вектора на матрицу, будет находиться в первых восьми регистрах вертикального арифметического устройства и, следовательно, может быть непосредственно занесен в память системы. При умножении вектора (матрицы) на постоянную величину первый регистр захвата используется для передачи этой константы на все уровни вертикального арифметического устройства. Регистры захвата могут также применяться для выполнения операций умножения вектора на матрицу, если вектор и матрица формируются в памяти при помощи языка ПЛ-I. Это осуществляется так же, как было показано на примере языка Фортран, но в языке ПЛ-I вектор будет воспроизведен 8 раз в такой последовательности: $V_1 \rightarrow$ уровни 0, 8, 16 и т. д.; $V_2 \rightarrow$ уровни 1, 9, 17 и т. д.

Параллельные операции сравнения генерируют содержание регистров масок, которые определяют порядковые отношения для определенных операндов — векторов, матриц или констант. Эти ре-

зультаты часто используются в некоторых типах булевских операций для генерации маски в регистре маски памяти или регистре маски арифметического устройства. Первый из этих регистров управляет потоком данных в память. Если в определенном поле этого регистра находится 1 бит, то операция запоминания выполняется, в против-

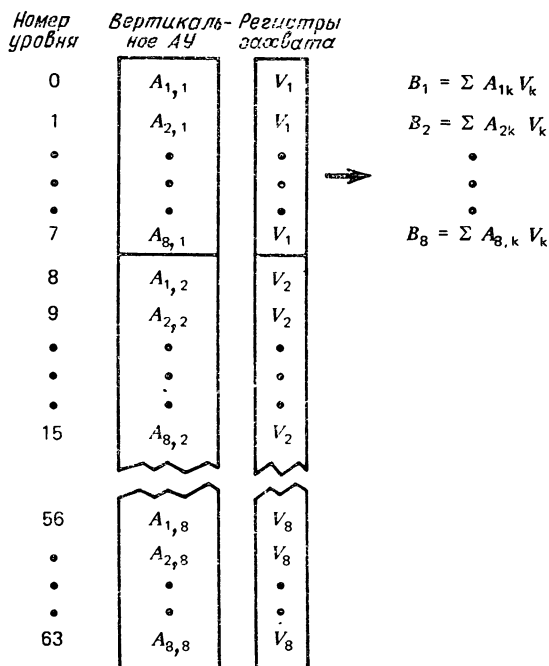


Рис. II.XII.5. Операция умножения вектора на матрицу.

ном случае содержимое ячеек памяти, к которым сделано обращение, не меняется. Регистр маски арифметического устройства действует точно так же по отношению к вертикальному арифметическому устройству, эффективно выполняя роль маски запрета операций для каждого уровня устройства (каждого элементарного процессора). Предусмотрены также специальные фиктивные маски, применение которых значительно сокращает число операций перемещения масок по сравнению с тем числом, которое потребовалось бы, если бы существовало всего по одному экземпляру масок каждого типа.

ПРИЛОЖЕНИЕ XIII

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ASC ФИРМЫ TEXAS INSTRUMENTS

ОРГАНИЗАЦИЯ СИСТЕМЫ

Вычислительная система ASC (Advanced Scientific Computer) является мощным мультипроцессором общего назначения. Она обеспечивает мультипрограммную обработку потоков заданий в режимах местной и дистанционной пакетной обработки и диалоговом режиме с удаленного терминала. Модульная структура и принципы проектирования системы обеспечивают возможность ее наращивания для удовлетворения требований пользователей относительно расширения функций и увеличения производительности системы. Система программного обеспечения помогает реализовать эти возможности расширения системы при высокой программной совместимости «снизу вверх».

Система ASC обладает рядом важных особенностей в отношении ее аппаратного и программного обеспечения. К наиболее важным характеристикам аппаратного обеспечения относятся:

Высокое быстродействие, магистральный принцип обработки информации в центральном процессоре, наличие команд для выполнения операций над векторными и скалярными величинами. Периферийная процессорная система с 8 независимыми виртуальными процессорами.

Устройство управления памятью со скоростью передачи информации 640 млн. слов/с, страничная организация памяти без временных потерь и защита памяти.

Быстродействующая центральная память с чередованием на полупроводниковых активных элементах с полосой рабочих частот, соответствующей скорости передачи 400 млн. слов/с.

Возможность расширения центральной памяти путем включения в систему вспомогательной памяти на полупроводниковых активных элементах емкостью несколько миллионов слов.

Множественные быстродействующие каналы передачи данных. Запоминающие устройства на высокоскоростных магнитных дисках большого размера с фиксированными головками, имеющие емкость 25 млн. слев.

Наиболее важными частями системы программного обеспечения являются:

Операционная система, функционирующая исключительно на базе периферийного процессора.

Компилятор языка Фортран, вырабатывающий оптимальный объектный код для операций центрального процессора над векторными и скалярными величинами.

Универсальная мультипроцессорная операционная система табличного типа, построенная в виде иерархии управляющих элементов и системных задач, выполняемых исключительно в виртуальных процессорах.

Мощная система управления файлами с древовидной структурой и средствами, обеспечивающими строгую секретность данных.

Схема конфигурации системы, представленная на рис. П.ХII.1, иллюстрирует роль устройства управления памятью как управляющего «перекрестного» коммутатора между памятью и всеми процессорами и контроллерами, которые сопрягаются между собой через входы устройства управления памятью. Высокоскоростная передача данных осуществляется при помощи множественных входов устройства управления памятью. Каждый вход на любой стороне устройства может обеспечивать скорость передачи данных до 80 млн. 32-разрядных слов в секунду. При включении в систему дополнительной центральной памяти используется добавочный вход, как показано на рис. П.ХII.1. Выходы устройства управления памятью могут работать в асинхронном режиме и полностью соответствуют требованиям многоскоростной передачи данных.

Устройство управления памятью сконструировано таким образом, чтобы всегда имелась возможность использовать память с более высоким быстродействием и подключать по требованиям пользователей большее число процессоров, применяя более высокие скорости передачи данных. Восемь модулей памяти с чередованием, каждый из которых имеет свой собственный вход в устройстве управления памятью, могут обеспечивать общую скорость передачи данных до 400 М слов/с, которая все же значительно ниже, чем могло бы допустить устройство управления. Скорость, которая необходима для обеспечения эффективной работы центрального процессора с 4 арифметическими устройствами («4-магистральный процессор») при выполнении операций над векторными величинами, вдвое ниже чем 400 М слов/с.

Устройство управления памятью имеет также необходимые аппаратные средства для управления доступом к памяти. Защита памяти выполняется при помощи регистров страничной организации памяти и граничных регистров. Регистры страничной организации исключают взаимное влияние областей памяти разных программ и позволяют использовать несмежные страницы памяти без каких-либо

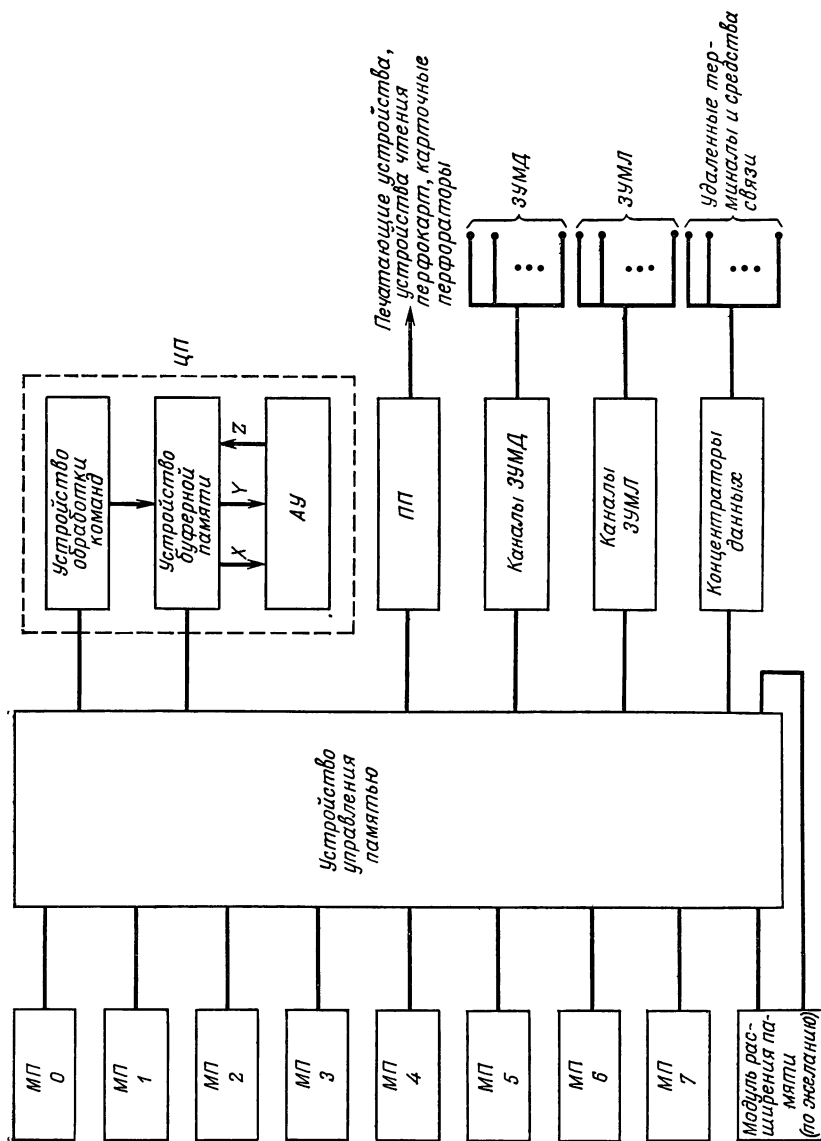


Рис. П.ХИИ.1. Конфигурация системы ASC с одной магистралью.

МП — модуль памяти, ПП — периферийный процессор, АУ — арифметическое устройство.

временных потерь. Граничные регистры разрешают или запрещают отдельной программе различные типы доступа (например, только считывание) к странично организованной памяти.

Центральный процессор состоит из устройства магистральной обработки команд, управляющего одним, двумя или четырьмя устройствами буферной памяти, каждое из которых соединено с маги-

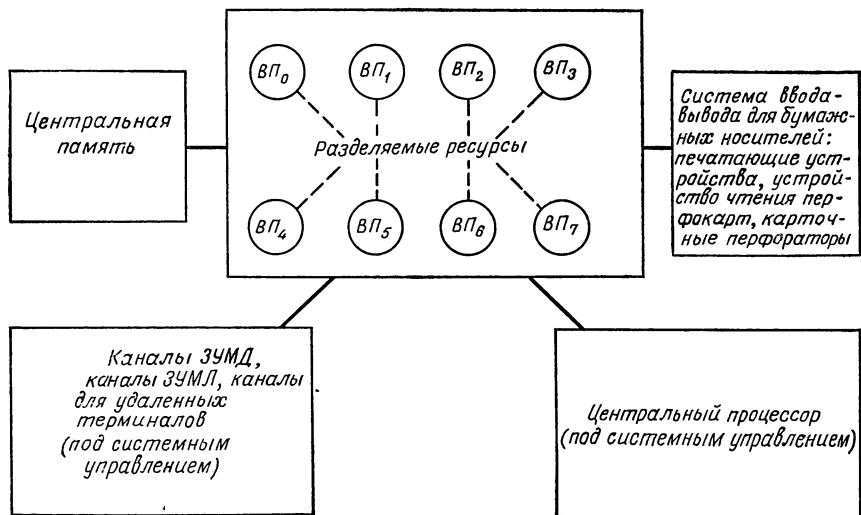


Рис. П.XIII.2. Логическая организация периферийного процессора.
ВП — виртуальный процессор.

стральным арифметическим устройством. Для обеспечения непрерывного потока команд из памяти системы в устройстве магистральной обработки предусмотрено 2 буфера по 8 слов. Один буфер содержит текущий набор команд, а другой в это время получает из памяти следующий набор из 8 команд. Вся обработка команд, за исключением той ее части, которая выполняется арифметическим устройством, осуществляется четырехуровневым устройством магистральной обработки. Каждое буферное устройство памяти содержит 3 пары буферных регистров по 8 слов: по одной паре для каждого из двух входных операндов и одна пара для выходного операнда. Эти буферы служат для организации доступа в память с перекрытием и чередованием для каждого из векторных операндов. Каждое магистральное арифметическое устройство состоит из восьмиуровневой арифметической «магистральной», связанной с соответствующим устройством буферной памяти.

Логическая организация периферийного процессора с восемью виртуальными процессорами представлена на рис. П.XIII.2. Основ-

ной функцией периферийного процессора является выполнение всех программ операционной системы, которая обеспечивает средства управления для всей системы ASC, ее периферийных устройств, памяти с прямым доступом и центрального процессора.

Каждый виртуальный процессор функционирует как самостоятельная ЭВМ со своими собственными командами и устройствами управления. Все виртуальные процессоры имеют общее арифметическое устройство и устройство постоянной памяти, используемое для часто повторяющихся последовательностей команд операционной системы. Буфер с соответствующим управлением используется несколькими виртуальными процессорами для организации доступа с перекрытием в память системы. Для передачи управляющей информации и информации о состоянии системы между периферийным процессором и другими процессорами или устройствами управления используется файл из 64 32-разрядных регистров связи в периферийном процессоре. Этот файл доступен для всех виртуальных процессоров и устройств, требующих управляющую информацию, например для контроллера дискового канала.

Вспомогательная память системы представлена главным образом запоминающим устройством на магнитных дисках. Контроллеры дискового канала имеют конфигурацию, отвечающую требованиям системы; они предназначены для управления памятью на магнитных дисках.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

В этом разделе приведены характеристики отдельных подсистем, составляющих конфигурацию системы ASC. В основу их обсуждения положена блок-схема аппаратных средств на рис. П.XIII.3. Данная конфигурация считается наиболее типичной для вычислительного оборудования, предназначенного для научных расчетов, которые требуют больших вычислительных мощностей.

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР

Центральный процессор, представленный на рис. П.XIII.3, является четырехмагистральным процессором, однако, если объем вычислительных работ недостаточен для загрузки всех четырех параллельных арифметических устройств, то можно использовать конфигурации, состоящие из одно- или двухмагистральных центральных процессоров. Базовая структура центрального процессора (рис. П.XIII.4) включает три основных элемента: устройство обработки команд (для неарифметических этапов обработки потока команд в центральном процессоре), устройство буферной памяти (для связи операнда с центральной памятью) и арифметическое устройство (выполняющее определенные арифметические или логи-

ческие операции). На рис. П.XIII.4 представлены схемы двухмагистрального и четырехмагистрального процессоров, каждого с соответствующим числом пар «устройство буферной памяти — арифметическое устройство». Следует отметить, что необходим один вход в память для устройства обработки команд и, кроме того, один вход для каждой магистрали центрального процессора (пары «устройство буферной памяти — арифметическое устройство»).

В конструкции центрального процессора использован принцип магистрали или трубопровода, позволяющий вести последовательную обработку с перекрытием нескольких независимых команд. В центральном процессоре применена магистраль команд для декодирования и обработки стандартных команд для выполнения операций над скалярными величинами. Помимо этого, возможности магистральной обработки в каждом арифметическом устройстве эффективно используются при выполнении векторных команд, а также некоторых последовательностей скалярных команд. Основной цикл центрального процессора равен 60 нс; поэтому, если принцип магистрали используется достаточно эффективно, то результаты арифметических операций могут появляться на выходе каждого арифметического устройства с интервалом 60 нс.

Центральный процессор обладает возможностями как векторного, так и скалярного процессора. Характеристики центрального процессора как устройства для обработки скалярных величин обусловлены наличием расширенного набора команд загрузки и хранения; предусмотрены команды с форматом полуслова, полного слова и двойного слова. Имеются программные средства загрузки и хранения файла регистров, а также загрузки эффективных адресов. Арифметические операции над скалярными величинами включают различные операции сложения, вычитания, умножения и деления чисел с фиксированной запятой, имеющих формат полуслова (16 разрядов) и полного слова (32 разряда), и чисел с плавающей запятой, имеющих формат полного слова и двойного слова (64 разряда). Логические команды для скалярных величин включают операции И, ИЛИ и исключающее ИЛИ. Допускаются операции смещения (в том числе кругового) для арифметических и логических операндов. Различные команды сравнения и сочетания этих команд с логическими командами предусмотрены для форматов полуслова, полного слова и двойного слова. Используются также различные сочетания команд для тестовых проверок и команд условного перехода. При помощи одиночной команды могут быть выполнены операции сцепления и переадресации арифметических регистров. Сопряжение подпрограмм производится при помощи команд условного перехода и загрузки. Предусмотрена возможность преобразования формата для нормальных и двойных слов, а также команды нормализации.

Характеристики центрального процессора как устройства для обработки векторных величин определяются функциями команд VECTL

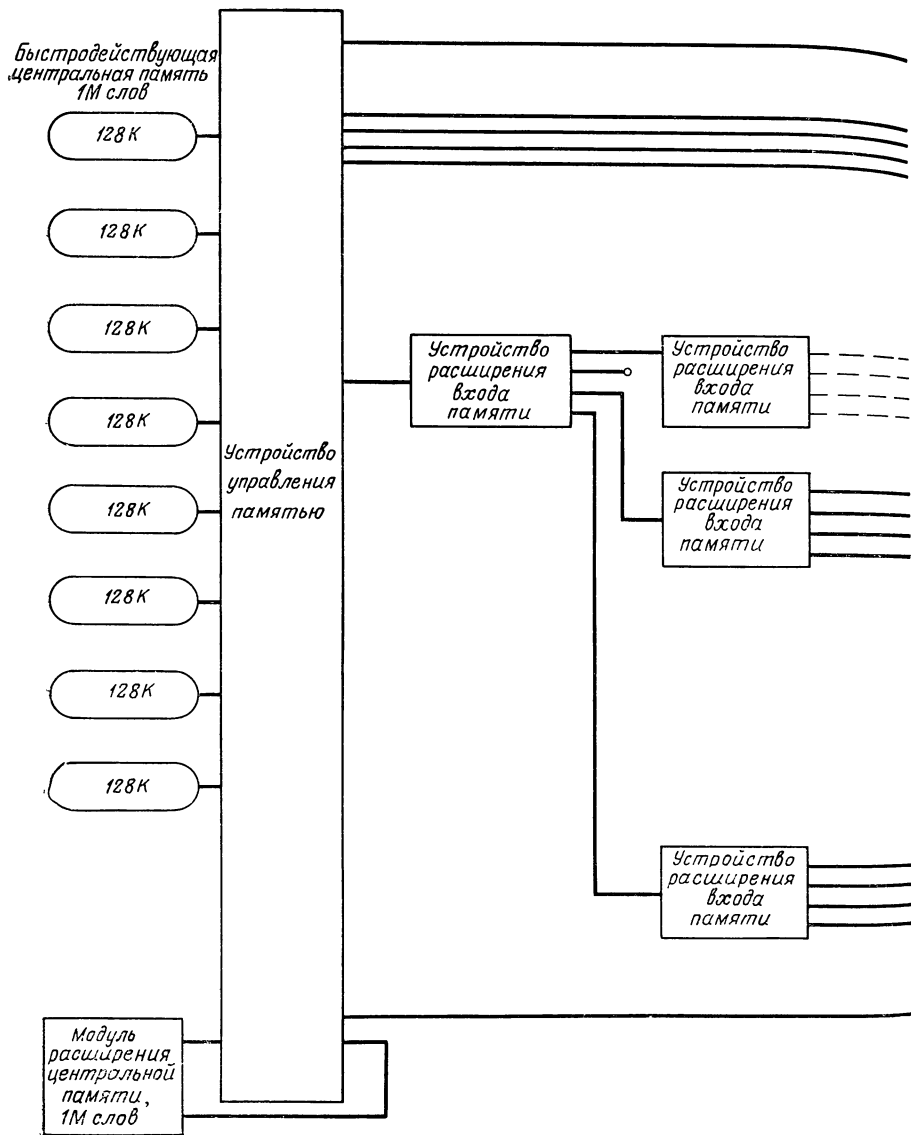
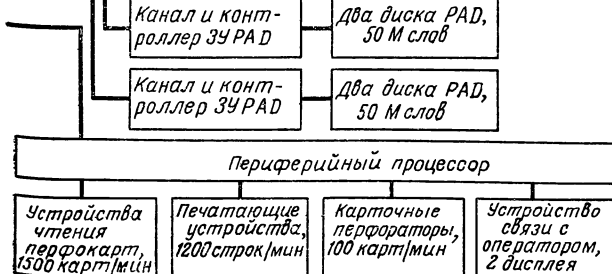
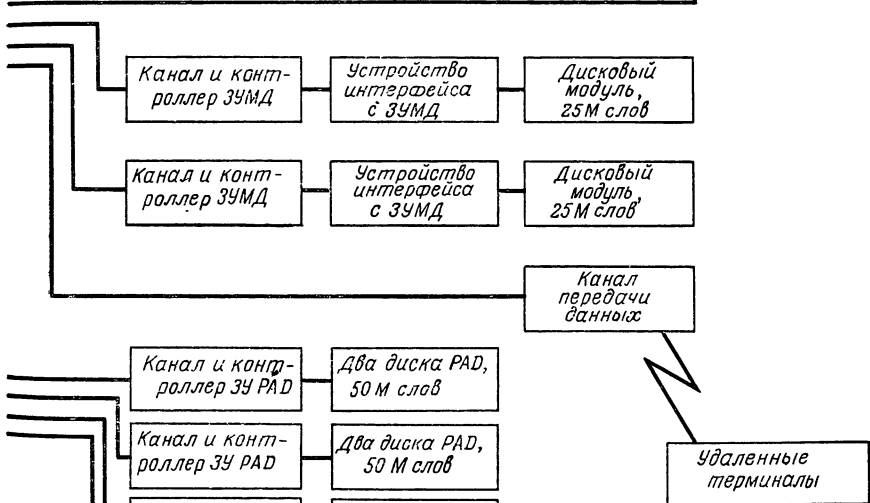
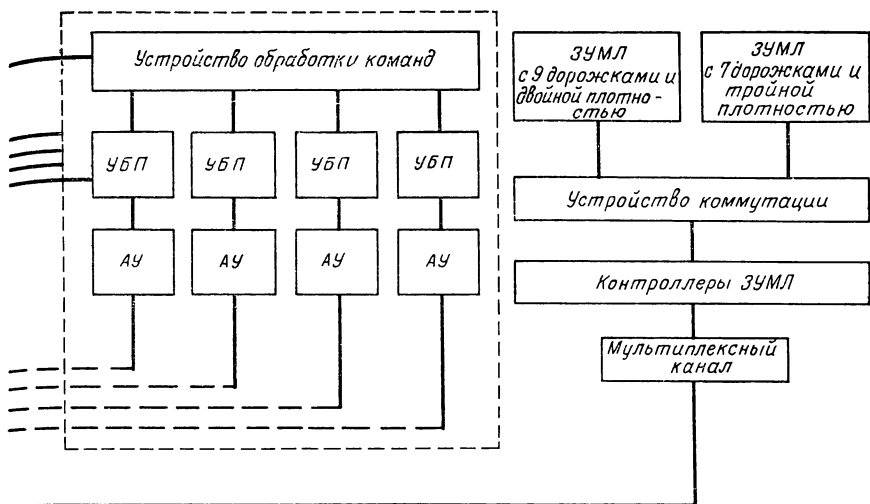


Рис. П.ХИИ.3. Одна из возможных конфигураций системы ASC.
УВП — устройство буферной памяти, АУ — арифметическое устройство.



(вектор после загрузки файла параметров вектора) и VEST (предполагается, что файл параметров уже загружен). Набор векторных команд включает такие арифметические операции, как сложение, вычитание, умножение, деление, скалярное умножение вектора на матрицу и другие операции над числами с фиксированной и плавающей запятой. Предусмотрены векторные команды для выполнения смещения, логических операций, сравнения, преобразования

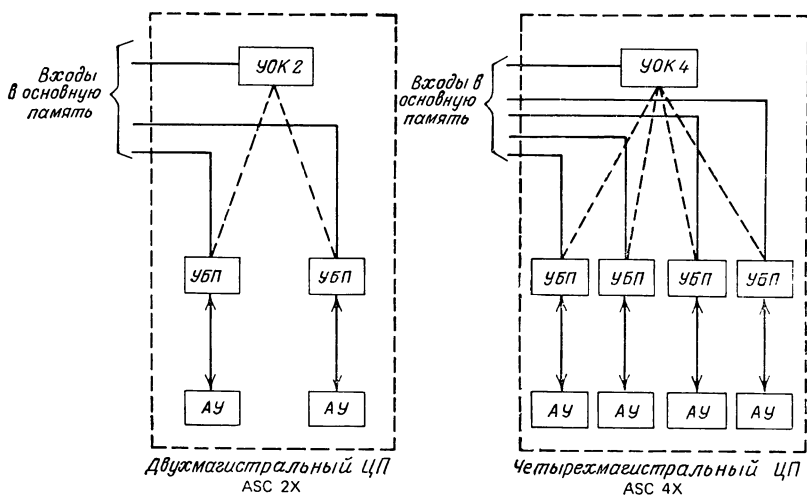


Рис. П.ХИИ.4. Базовая структура центрального процессора.

УОК — устройство обработки команд, УБП — устройство буферной памяти, АУ — арифметическое устройство.

формата, нормализации и специальных операций типа сортировки, поиска, определения экстремумов, выбора, замены.

Одной из основных характеристик векторных команд является *трехмерная* адресация внутри одной команды. Это эквивалентно введению в обычную ЭВМ специального узла, выполняющего три цикла индикации. Для хранения параметров, связанных с выполнением векторных команд, используется группа адресуемых регистров по 8 слов каждый (файл параметров вектора).

В центральном процессоре имеется 48 программно-адресуемых регистров. Эта группа 32-разрядных регистров состоит из 16 регистров базового адреса, 16 арифметических регистров, 8 индексных регистров и 8 регистров параметров вектора.

Функции центрального процессора могут быть распределены во времени между программами при помощи механизма автоматической коммутации контекста. Операционная система может произвести прерывание в любое время, даже во время выполнения векторной ко-

манды, и заставить центральный процессор автоматически переключиться с выполнения текущей программы на выполнение следующей программы. Благодаря этому система может работать как в режиме мультипрограммирования, так и в режиме реального времени. Автоматическая коммутация контекста может также вызываться программами центрального процессора, если их выполнение не может

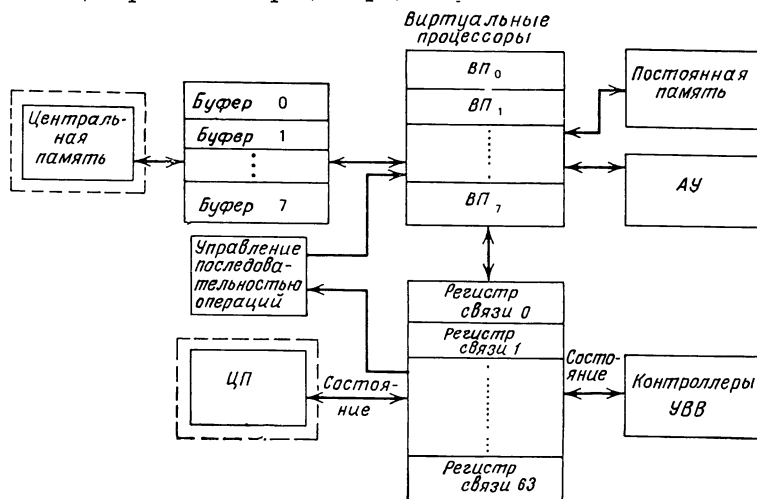


Рис. П.ХІІІ.5. Периферийный процессор.

АУ — арифметическое устройство, УВВ — устройство ввода-вывода.

быть продолжено до завершения определенных операций обслуживания, реализуемых операционной системой в периферийном процессоре.

Одной из основных аппаратных характеристик центрального процессора является возможность предварительного просмотра памяти и выбора операнда до того момента, когда он понадобится в расчетах. Двойная буферизация с использованием множественных буферов по 8 слов для каждого магистрального арифметического устройства обеспечивает сглаживание потока данных к (от) каждого арифметического устройства. Магистральное арифметическое устройство достигает наивысшей скорости обработки потока данных при выполнении операций над векторными величинами; обычно при этом результаты выдаются с интервалом 60 нс.

ПЕРИФЕРИЙНЫЙ ПРОЦЕССОР

Периферийный процессор представляет собой мультипроцессор, предназначенный для управления заданиями и данными в системе ASC. Реализация идеи периферийного процессора способствует повышению эффективности системы ASC. На рис. П.ХІІІ.5 пред-

ставлена схема логической организации периферийного процессора.

Периферийный процессор является набором из восьми отдельных процессоров, называемых виртуальными процессорами. Каждый виртуальный процессор имеет свой программный счетчик, а также арифметический, индексный, базовый и командный регистры. Восемь виртуальных процессоров совместно используют постоянную память, арифметическое устройство, устройство обработки команд и буфер центральной памяти. Эти устройства распределяются во времени между виртуальными процессорами, причем используются 16 циклов по 85 нс. Если применяется распределение с равными последовательными временными интервалами, то каждый из 8 виртуальных процессоров получает 2 цикла по 85 нс за 1,4 мкс. Обычной команде периферийного процессора требуется 2 цикла по 85 нс для завершения операции. Распределение предоставляемых временных интервалов можно варьировать в зависимости от конкретных требований к обработке информации.

Постоянная память в периферийном процессоре используется для хранения и выполнения коротких стандартных подпрограмм, часто применяемых виртуальными процессорами, например подпрограммы вызова удаленных терминалов. Постоянная память имеет емкость 4К 32-разрядных слов и состоит из элементов, сохраняющих информацию при выключении питания; время цикла памяти меньше 85 нс.

Так как периферийный процессор выполняет в основном функции управления, а не расчеты по математическим алгоритмам, набор его команд ориентирован на выполнение управляющих операций и не содержит операций умножения, деления или операций над числами с плавающей запятой. Формат команд идентичен формату команд центрального процессора — для каждой команды используется 32-разрядное слово. Команды обеспечивают выполнение операций с информационными полями длиной в 1 разряд, 1 байт (8 разрядов), полуслово (16 разрядов) и полное слово (32 разряда).

Каждый виртуальный процессор имеет прямой доступ ко всей центральной памяти для выполнения программ и хранения данных. Поэтому единственная копия программы с повторной входимостью может одновременно обрабатываться несколькими виртуальными процессорами.

Файл регистров связи содержит 64 32-разрядных регистра, которые программно адресуются виртуальными процессорами. Файл используется как главное запоминающее устройство для хранения управляющей информации, необходимой для координации всех частей системы ASC. Синхронизация процессов связи между всеми процессорами (центральным процессором, виртуальными процессорами, контроллерами каналов и периферийных устройств) осуществляется путем интерпретации битов состояния, поступающих от всех устройств в файл регистров связи.

ЦЕНТРАЛЬНАЯ ПАМЯТЬ

Центральная память системы ASC состоит из устройства управления памятью, одинаковых по размерам модулей быстродействующей центральной памяти и модулей расширения центральной памяти.

Устройство управления памятью организовано в виде двустороннего канала шириной 256 разрядов (8 слов), обеспечивающего сеть параллельных трактов доступа между 8 независимыми входами процессоров и 9 шинами памяти, причем каждый вход процессора имеет доступ ко всем модулям памяти. Девять шин памяти организованы таким образом, что могут обеспечивать восьмикратное чередование первых восьми шин с девятой шиной, используемой для доступа к модулю расширения центральной памяти. Устройство управления памятью осуществляет управление доступом к центральной памяти через восемь процессорных входов. Длина адресного слова равна 24 разрядам; при этом адресное пространство центральной памяти имеет объем 16 млн. слов. Для увеличения числа процессорных входов используется устройство расширения входа в память.

Устройство управления памятью предназначено для осуществления асинхронных операций, независимых от задержек в кабеле, частоты синхронизации процессора, а также от времени цикла доступа к памяти. Эти свойства устройства управления памятью позволяют легко вводить в систему усовершенствованные элементы аппаратуры памяти или процессора по мере их появления. Устройство управления памятью может обеспечить максимальную скорость передачи данных через один вход, равную 80 млн. слов/с, что в целом соответствует пропускной способности устройства 640 М слов/с. Эта пропускная способность значительно превышает достигнутые на сегодняшний день скорости передачи данных в устройствах центральной памяти и процессора.

Быстродействующая центральная память построена на активных элементах и имеет время цикла 160 нс. Передача данных осуществляется блоками по 256 разрядов (8 слов) в коде Хемминга, что позволяет автоматически обнаруживать и исправлять сбой в одном разряде каждого блока и обнаруживать, но не исправлять, ошибки в двух разрядах. Быстродействующая центральная память обычно делится на восемь равных по емкости модулей и допускает восьмикратное чередование. Однако управление декодированием адреса памяти осуществляется системой коммутации устройства управления памятью, в которой кратность чередования определяется операционной обстановкой и может быть меньше восьми.

В системе предусмотрена возможность расширения центральной памяти, при котором к высокоскоростной центральной памяти добавляются недорогие среднескоростные запоминающие устройства большой емкости. Эти устройства сконструированы на активных

элементах и имеют время цикла 1 мкс. Доступ к модулю расширения памяти осуществляется блоками по 8 слов. Для передачи данных также используется код Хемминга, при помощи которого обнаруживаются ошибки и автоматически исправляются ошибки в одном разряде. Модуль расширения памяти является частью памяти с прямой адресацией и к нему, следовательно, может адресоваться любой процессор или контроллер канала для получения команд или операндов. Можно также применить поблочную передачу информации между быстродействующей центральной памятью и модулем расширения памяти. Это вполне допустимо, так как модуль расширения центральной памяти имеет обычную шину памяти и вход для доступа в устройство управления памятью. При блочной передаче, инициируемой периферийным процессором, задается два адреса памяти и количество слов, назначенных для передачи. Модуль расширения осуществляет передачу данных автономно со скоростью 40 М слов/с и сообщает периферийному процессору об окончании передачи.

Емкость центральной памяти (16 млн. слов) определяется длиной адресного слова, равной 24 разрядам. Соотношение емкостей быстродействующей центральной памяти и ее модуля расширения можно изменять, чтобы наилучшим образом удовлетворить требованиям к системе в отношении ее общей производительности и эффективности решения конкретных задач. Быстродействующая центральная память объемом в 1 млн. слов вместе с несколькими миллионами слов модуля расширения отвечают требованиям к вычислительным системам во многих современных областях их применения.

Управление центральной памятью и доступом к ней со стороны различных устройств (процессоров и контроллеров) осуществляется при помощи средств двух типов: регистров переадресации страниц и регистров защиты. Каждая программа пользователя имеет свою собственную карту адресации страниц. Адреса страниц, не затребованные программой, отображаются на нулевую страницу физической памяти, которая недоступна для центрального процессора. Когда программа загружается в память, то чаще всего она размещается на несмежных страницах. Во время выполнения программы адреса страниц, генерируемые программой, преобразуются в физические адреса страниц памяти при помощи регистров переадресации; эта операция осуществляется без потери полезного времени процессора. Поскольку обращение к нулевой странице запрещено и соответствующий процессор уведомляется об этом, при помощи регистров переадресации можно организовать защиту областей памяти, с которыми работает пользователь, от программ других пользователей. На рис. П.ХIII.6 представлена схема страничной организации памяти. Возможность разбиения памяти на страницы и распределения ее между несколькими программами с предоставлением программе несмежных областей повышает эффективность ее использо-

вания. Размер страницы зависит от емкости центральной памяти и совокупности проблем, связанных с организацией вычислительного процесса при обработке конкретных заданий. В системе ASC может быть задано четыре различных размера страницы емкостью от 4K до 256K слов. Программа может использовать любой из имеющихся размеров страницы.

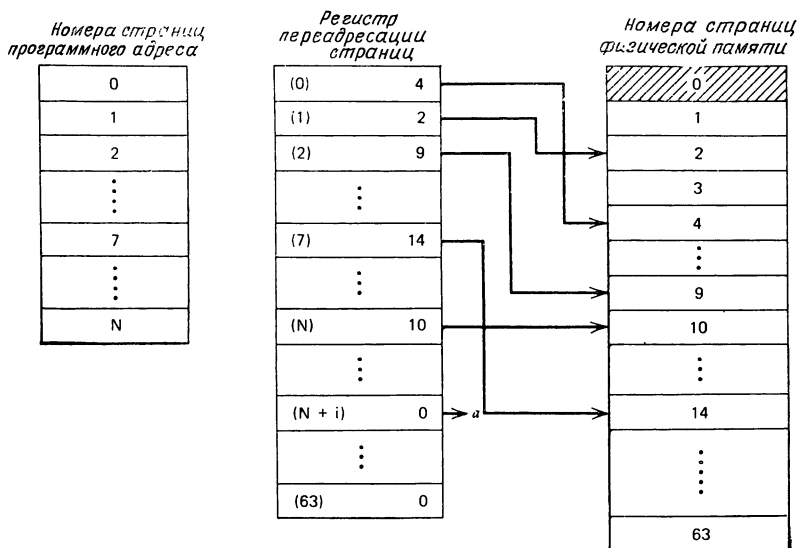


Рис. П.XIII.6. Страничная организация распределения памяти.

a — запрос на доступ к странице 0 в физической памяти отсылается операционной системе для вынесения решения (неверный запрос, ошибка и т. д.).

Регистры защиты предназначены для защиты рабочих областей памяти каждого пользователя. Эти регистры состоят из трех пар регистров границ, определяющих самый верхний и самый нижний адреса области доступа для считывания, записи или выполнения программы. Программное обеспечение системы позволяет использовать пять видов защиты с помощью регистров границ:

Только выполнение программы.

Только считывание.

Выполнение программы и считывание (операции записи запрещаются).

Считывание и запись (выполнение программ запрещается).

Считывание, запись и выполнение программ.

На любую попытку обратиться к памяти вне границ области, выделенной для определенного состояния управления системы, процессор получает отказ и извещается об этом отказе.

В больших системах ASC число обращений к памяти в данный момент со стороны процессоров и устройств управления может превысить число отдельных входов в память. В этих случаях для обеспечения дополнительных входов используются устройства расширения входа памяти; они могут также обслуживать устройства, для которых не требуются предусмотренные в системе входы в память с высокой пропускной способностью. Каждое устройство расширения входа памяти обеспечивает от 1 до 4 дополнительных входов с уменьшением пропускной способности не больше чем на 10% (это означает уменьшение скорости с 80 млн. примерно до 72 млн. 32-разрядных слов в секунду на одно устройство расширения). Устройства расширения входа памяти могут быть объединены в древовидную структуру. Приоритеты доступа на одном входе в память могут быть фиксированными или могут устанавливаться на основе динамического распределения. Конкретный способ выбирается при помощи коммутационной карты в устройстве расширения входа памяти.

ЗАПОМИНАЮЩИЕ УСТРОЙСТВА НА МАГНИТНЫХ ДИСКАХ

Дисковые запоминающие устройства являются главным типом вспомогательной памяти системы ASC. Применяются дисковые системы с фиксированными головками (по одной головке на тракт) и дисковые системы с подвижными головками.

Основная дисковая установка содержит по крайней мере один дисковый модуль с фиксированными головками, поскольку эффективная скорость передачи данных для этих дисков очень высока. Дисковый модуль с фиксированными головками имеет емкость 25 М 32-разрядных слов, скорость передачи 490К 32-разрядных слов в секунду и эффективное время доступа около 5 мс. Обычно для обеспечения равномерной нагрузки конфигурация должна включать от 2 до 4 модулей дискового запоминающего устройства с фиксированными головками.

Дополнительная дисковая система состоит из нескольких дисковых модулей с подвижными головками (PAD). С помощью модулей PAD можно формировать запоминающие устройства различной емкости и с различной скоростью передачи данных. Дисковые системы с подвижными головками имеют меньшую скорость передачи данных, чем дисковые системы с фиксированными головками при произвольной выборке данных.

Дисковая система с фиксированными головками является высокопроизводительным устройством, эффективность которого повышается благодаря наличию в операционной системе специального алгоритма сокращения времени доступа (SATF) при обмене данными. Такое сочетание программных и аппаратных средств обеспечивает очень высокую эффективную скорость передачи данных. Обычно система ASC содержит от двух до четырех дисковых модулей с

фиксированными головками и примерно такое же число дисковых модулей с подвижными головками.

Системой запоминающих устройств на магнитных лентах управляют мультиплексные каналы и контроллеры, обеспечивающие соответствие конфигурации и характеристики системы требованиям обработки данных. Обычно разрешается одновременный доступ к любым четырем лентопротяжным устройствам системы. В системе могут быть использованы семи- или девятидорожечные магнитные ленты со стандартной плотностью записи. Скорость передачи данных можно выбирать в широком диапазоне до 180К байтов/с.

Системой связи с удаленными терминалами управляет контроллер-концентратор данных, включенный в конфигурацию для обслуживания соответствующей сети терминалов. Поскольку данный контроллер-концентратор является программируемой системой (ЭВМ TI 980), он позволяет согласовать с системой ASC большинство имеющихся типов терминалов. В настоящее время система сопрягается с терминалами для пакетного ввода заданий с возможностью работы в диалоговом режиме (печатающее устройство, устройство чтения перфокарт и карточный перфоратор, устройство записи на магнитную ленту, пульта с дисплеями, графопостроитель и функциональная клавиатура).

Стандартные устройства чтения перфокарт и карточные перфораторы, строчно печатающие устройства и пульт оператора (с двумя дисплеями) могут быть подключены к периферийному процессору, который в этом случае функционирует как устройство управления и канал для средств ввода-вывода с бумажными носителями.

КАНАЛЫ

В системе ASC используется несколько типов устройств управления каналами, которые предназначены для обслуживания различных запоминающих устройств или терминалов, подключенных к системе. Аппаратура канала обеспечивает интерфейс и управление потоком информации между входом устройства управления памятью и контроллерами для различных запоминающих устройств или терминалов. Каждый канал связан при передаче данных только с контроллером и после первоначальной инициации продолжает работать независимо от центрального или периферийного процессоров. В наиболее типичных конфигурациях системы ASC каналы передачи данных подключаются к устройству расширения входа памяти.

Для обслуживания устройств, требующих высокой скорости передачи данных, таких, как диски, используется специальный быстродействующий канал. Этот канал может также выполнять операции чтения и записи данных в несмежных областях памяти.

Мультиплексный канал обеспечивает интерфейс для обслуживания среднескоростных устройств. В этом канале осуществляется

временное разделение передачи данных между памятью и одним — четырьмя контроллерами. Мультиплексор состоит из пяти буферов емкостью по 8 32-разрядных слов, четырех буферов подканала (по одному на каждый контроллер) и буфера канала. Буфер канала связан с устройством управления памятью, а четыре буфера подканала — с контроллерами.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение системы ASC состоит из обширной операционной системы и нескольких языковых процессоров: Фортрана, Языка спецификации заданий, Редактора связей и Метаасемблера.

Язык ASC — Фортран представляет собой расширенный стандартный язык ANSI Фортран, включающий многочисленные добавления, необходимые для операций над векторными величинами. Эти добавления соответствуют характеру данного языка и дают пользователю большие возможности программирования в системе обозначений, употребляемой математиками при решении различных математических задач. Специально разработанный объектный код эффективно использует возможности магистральной обработки и аппаратно реализованные команды для обработки векторных величин. Для выполнения скалярных операций была создана система кодирования, позволяющая добиться оптимального перекрытия выполнения команд при обработке данных в устройстве магистральной обработки команд и в арифметическом устройстве. Компилятор не только выполняет диагностические проверки исходных программ, посылая в случае необходимости соответствующие сообщения об ошибках, но и выдает информационные сообщения о невозможности оптимизации объектной программы вследствие особенностей структуры исходной программы.

Операционная система общего назначения ASC имеет структуру с высокой степенью модульности, включающую управляющие элементы и системные задачи, которые целиком выполняются виртуальными процессорами, входящими в периферийный процессор. Поэтому большие вычислительные возможности центрального процессора полностью остаются в распоряжении пользователя и не участвуют в выполнении функций управления, которые обычно связаны с работой операционной системы.

Модульность операционной системы была достигнута путем выделения основных элементарных работ операционной системы (небольших по объему). Множество таких элементарных работ (системных задач) затем объединяется в различные наборы для реализации требований к обработке информации в системе. Все эти модули имеют стандартную структуру и общий интерфейс. В результате получена система программного обеспечения, способная к непрерывному

совершенствованию и расширению: совершенствование возможно благодаря тому, что изменение временных или пространственных характеристик модулей происходит без воздействия на другие элементы системы, а расширение возможно за счет наличия строго обусловленного унифицированного интерфейса, который позволяет легко подключать новые модули для выполнения новых функций.

Системные задачи представляют собой программные модули с повторной входимостью в машинных командах, построенные таким образом, что их можно выполнять в любом виртуальном процессоре за очень короткое время, в среднем за 1 мс. Такие временные характеристики устраняют необходимость частых прерываний и соответствующих дополнительных операций по сохранению и восстановлению состояния и условий обработки. Связь между системными задачами в процессе их выполнения осуществляется управляющим элементом операционной системы. Планирование выполнения всех задач производится в соответствии с простым приоритетом; поэтому те результаты, которые в обычных системах достигаются за счет формирования прерываний, в рассматриваемой системе получаются путем выполнения каждой элементарной функции без прерывания с частотой, определяемой ее относительным приоритетом по сравнению с другими активными функциями.

Задачи объединяются в логические группы, названные системными командами. Многие задачи входят в состав нескольких системных команд, а объектный модуль одной задачи может одновременно и асинхронно выполняться как результат двух или большего числа активаций одной и той же команды или нескольких различных команд.

Команды являются такими элементами программ пользователя и (или) системных программ, которые требуют выполнения определенных функций. В качестве примера можно указать такие функции: 1) открытие файла; 2) считывание логической записи с диска; 3) загрузка программы. Большинство команд является элементами нескольких таких функций. Все подобные функции определяются в виде последовательности команд в системной таблице, хранящейся в резидентной части центральной памяти. Эта таблица интерпретируется одним из управляющих элементов операционной системы, благодаря чему обеспечивается высокая централизация управления системой ASC. Путем изменения этих таблиц можно модифицировать существующие в системе функции и добавлять новые функции. Дополнительное программирование необходимо только в тех случаях, когда требуются новые «строительные блоки» в форме команд и задач. Такие дополнения легко осуществляются за счет модульности операционной системы и строгой стандартизации интерфейсов.

Язык спецификации заданий позволяет пользователю программировать управление выполнением задания и вспомогательные функции операционной системы при помощи операторов языка, ориенти-

рованного по структуре и формату на пользователя. Программное обеспечение ASC предусматривает наличие развитой системы управления файлами. При каталогизации файлов используется древовидная структура, причем файлы могут находиться во всех узлах дерева, а не только на концах ветвей. Таким образом, на каждом пути движения по дереву может находиться большое число различных файлов, что позволяет каждому пользователю создавать для себя различные структуры базы данных.

Тщательно разработанная система обеспечения секретности файлов позволяет их владельцам ограничить доступ к файлам, расположенным в каком-либо отдельном узле, или же ко всем файлам, расположенным на ветвях дерева, следующих за данным узлом. Поэтому пользователи системы, имеющие доступ к каталогам файлов, могут быть ограничены или рамками отдельного файла, или же определенными путями в каталогизированной структуре.

Метаассемблер включает широкий спектр макрокоманд; он формирует совокупности машинных команд для центрального и периферийного процессоров. Этот тип ассемблера позволяет специфицировать характеристики языков других машин и поэтому может быть всегда определен для любой другой ЭВМ. Кроме того, в Метаассемблере достаточно легко выполняются специальные требования к расширению функций языка ассемблера для центрального и периферийного процессоров.

Программа Редактор связей обеспечивает выполнение стандартных функций, необходимых для создания модулей загрузки из отдельных объектных модулей. Кроме того, этот процессор позволяет осуществлять загрузку в память временных связей и обеспечивает автоматическое выполнение программ оверлейной структуры.

ПРИЛОЖЕНИЕ XIV

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА UNIVAC 1108 ФИРМЫ SPERRY RAND CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Вычислительная система UNIVAC 1108 является универсальной системой с однопроцессорной или мультипроцессорной конфигурацией, модульная структура которой позволяет осуществлять выбор вычислительных средств, необходимых для удовлетворения требований по быстродействию и производительности в широком диапазоне применений — от обычных вычислительных устройств предприятий и организаций до мощных информационно-вычислительных систем коллективного пользования. Централизованное управление системой возложено на операционную систему, получившую название «Ехес-8». Система «Ехес-8» предназначена для обеспечения различных режимов работы (пакетной обработки, разделения времени, работы в реальном времени, мультипрограммирования), причем использование какого-либо одного из этих режимов не связано с дополнительными издержками из-за наличия средств, обеспечивающих работу системы в других режимах. Устройства, которые не нужны для требуемых режимов, могут быть исключены в процессе построения системы.

В минимальной мультипроцессорной конфигурации эта вычислительная система включает два центральных процессора, два банка основной памяти (каждый по 65К 36-разрядных слов) и один контроллер ввода-вывода. Способность к расширению, обусловленная модульной структурой системы, позволяет добавить к минимальной конфигурации еще один центральный процессор, два куба памяти по 65К слов и один контроллер ввода-вывода. Взаимосвязь основных устройств в такой вычислительной системе показана на рис. П.XIV.1.

Система 1108 позволяет одновременно решать на нескольких процессорах большое число задач. Каждый функциональный блок, входящий в состав системы, имеет несколько каналов доступа, а приоритетная логика разрешает любые возможные конфликты. Отказы каких-либо отдельных элементов не вызывают прекращения непрерывной работы системы, а сами элементы могут логически заменяться действующими в процессе работы системы.

Каждый процессор системы 1108 способен осуществлять все функции, требуемые для выполнения любых команд, включая команды арифметических операций, процедур ввода-вывода и управления обработкой. В каждый процессор включен набор собственных

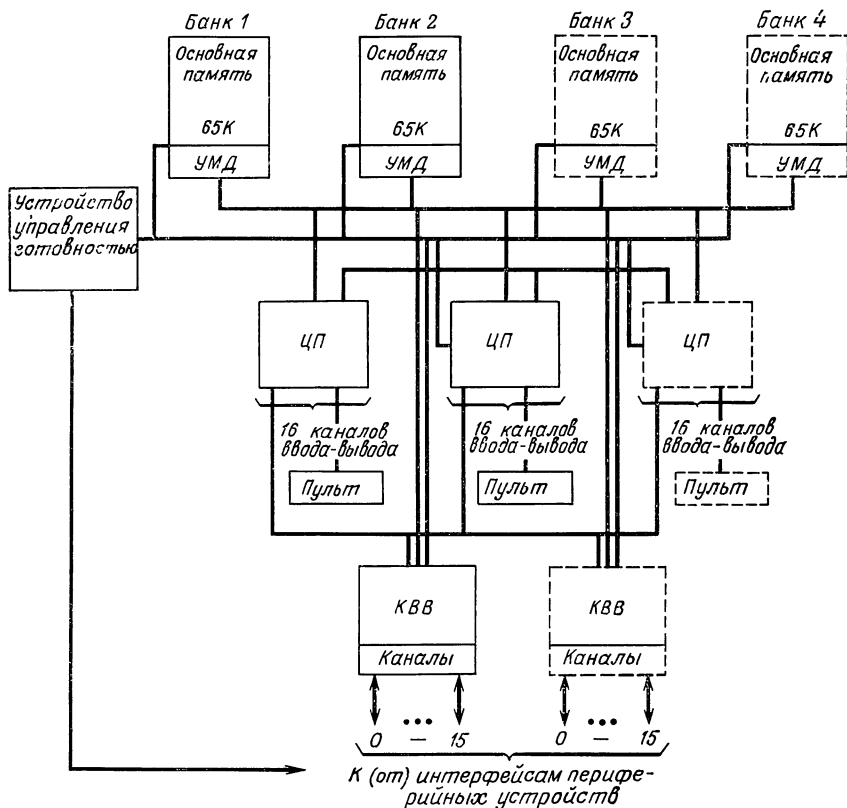


Рис. П.XIV.1.Мультипроцессорная система UNIVAC 1108.

УМД — устройство мультимодуля к памяти, КВВ — контроллер ввода-вывода.

управляющих регистров, выполняющих функции накопителей, индексных регистров, регистров управления вводом-выводом и специальных регистров.

Контроллер ввода-вывода является независимым процессором и используется в системе 1108 для расширения ее возможностей по организации ввода-вывода. Контроллер ввода-вывода содержит до шестнадцати двусторонних высокоскоростных 36-разрядных каналов ввода-вывода. Он обеспечивает независимый доступ к основной памяти системы и возможность формирования цепочек данных; кроме

того, он имеет свою собственную быстродействующую индексную память для управления буфером. Часть аппаратных средств контроллера ввода-вывода включает три группы регистров:

Шестнадцать указательных регистров, определяющих место расположения в основной памяти функционального слова, которое управляет операциями контроллера ввода-вывода.

Шестьдесят четыре внутренние специфицируемых индексных регистра, управляющих внешними функциями и доступом к данным (регистры ISI).

Сто девяносто два внешне специфицируемых индексных регистра, управляющих связью и доступом к данным, с возможным добавлением 256 дополнительных регистров (регистры ESI).

Регистры ISI используются для каналов ввода-вывода, подключенных к периферийным подсистемам, таким, как запоминающее устройство на магнитной ленте, магнитные барабаны, печатающие устройства, карточные перфораторы.

Регистры ESI используются для передачи и приема данных из подсистем связи. При работе системы в соответствующем режиме поток данных в канале связи от каждого терминала управляется при помощи его собственного слова управления доступом к данным, записанного в индексной памяти.

Основная память системы 1108 может быть расширена подключением блоков по 65K слов до емкости 262 144 36-разрядных слов. Время цикла чтение-запись основной памяти составляет 750 нс, а эффективное время обращения к памяти, состоящей из 4 логических блоков, за счет перекрытия операций при выборке команд и данных может быть снижено до 375 нс. Кроме того, используя двукратное чередование адресов (четные, нечетные ячейки) в двух модулях памяти внутри каждого банка, можно значительно уменьшить вероятность конфликтных обращений.

Вспомогательная память в системе 1108 состоит главным образом из запоминающих устройств на магнитном барабане (до восьми), которые подсоединяются к системе через один или два контроллера.

Характерным для мультипроцессорной системы является разделение во времени всеми процессорами всей основной памяти и всех подсистем ввода-вывода. Доступ к основной памяти осуществляется при помощи устройств мультидоступа к модулям памяти, а доступ к подсистемам ввода-вывода — с помощью разделяемого периферийного интерфейса.

Подсистемы ввода-вывода в системе 1108 могут быть одно- и двухканальными, причем в одноканальной подсистеме в каждый момент времени выполняется только одна операция ввода-вывода, а в двухканальной подсистеме имеется возможность выполнять одновременно две операции, относящиеся к различным периферийным устройствам.

В системе имеется программный 18-разрядный таймер интервалов. Его временной цикл равен 200 мкс, так что через каждые 200 мкс из содержимого таймера вычитается 1. Когда на счетчике окажется нуль, в системе происходит прерывание.

Важным элементом системы 1108 является устройство управления готовностью. В соответствии с требованиями к готовности системы должна быть предусмотрена возможность расчленения системы для выполнения специальных работ и технического обслуживания оборудования. Устройство управления готовностью выполняет следующие функции:

Расчленяет мультипроцессор на ряд независимых систем.

Отключает устройства для технического обслуживания или ремонта без нарушения функционирования системы.

Обеспечивает защиту основной памяти при отказе электропитания центральных процессоров или контроллеров ввода-вывода. Автоматически вызывает средства восстановления системы после отказа.

Устройство управления готовностью обеспечивает разделение аппаратных средств системы на ряд специфических конфигураций при помощи соответствующих подключений и блокировок интерфейсов между различными устройствами. Таким способом можно образовать до трех логически независимых конфигураций, работающих в режиме совпадения под управлением одной операционной системы. Эти конфигурации могут быть определены заранее для каждого конкретного случая применения. В то же время устройство управления готовностью может отключить некоторые блоки для ремонта или технического обслуживания. Устройство управления готовностью является независимым устройством с собственным питанием; оно логически расположено между периферийными подсистемами, центральными процессорами, контроллерами ввода-вывода и основной памятью. Оно может сопрягаться с тремя центральными процессорами через каналы ввода-вывода каждого процессора, двумя контроллерами ввода-вывода, четырьмя банками основной памяти и шестью периферийными подсистемами с мультидоступом. К системе могут быть добавлены еще до 24 периферийных подсистем (группами по шесть подсистем). Устройство управления готовностью включает панель управления (расположенную физически на пульте оператора), на которой отображается текущее состояние системы (ее расчленение на отдельные конфигурации), а также указывается, какие из устройств отключены. Панель позволяет вручную управлять конфигурацией системы, включая в систему или отключая отдельные устройства. Последовательность операций автоматического восстановления после отказов обеспечивается при помощи системного таймера восстановления в устройстве управления готовностью. Период времени на этом таймере может быть установлен в диапа-

зоне от одной до пятнадцати секунд. Если операционная система не возвращает таймер в исходное положение в течение заданного временного интервала, устройство управления готовностью решает, что в системе произошло существенное нарушение работы, и инициирует последовательность операций автоматического восстановления. Процессор может затем запросить устройство управления готовностью, какие устройства находятся в активном состоянии и могут быть использованы для обработки информации.

В системе 1108 предусмотрен отдельный пульт оператора для каждого центрального процессора. На пульте имеется клавиатура, дисплей, автоматическая пишущая машинка и часы. Часы на пульте указывают суточное время в часах, минутах и сотых долях минуты. Через каждые 600 мс они посылают в центральный процессор значение суточного времени и через каждые 6 с посылают сигнал прерывания. Эти часы могут быть вручную отключены от пульта оператора, и в мультипроцессорной системе значение суточного времени можно задавать или от внешнего источника, или программно.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Главной составной частью мультипроцессорной системы модели UNIVAC 1108 является центральный процессор. Каждый центральный процессор в мультипроцессорной конфигурации выполняет как арифметические, так и логические операции и может управлять работой до 16 каналов ввода-вывода. Центральный процессор обеспечивает возможность адресации частей слова по 6, 9, 12 и 18 разрядов, а также адресации полного слова (36 разрядов) или двойного слова (72 разряда). Перекрытие при выполнении команд позволяет достичь быстродействия около 610 000 команд/с, включая операции над числами с плавающей запятой.

Каждый центральный процессор логически подразделяется на шесть взаимодействующих секций:

Регистры управления — 128 программно-адресуемых регистров, используемых для арифметических операций, индексации и управления буфером ввода-вывода.

Арифметическая секция.

Секция управления.

Секция ввода-вывода, управляющего мультиплексной передачей данных между основной памятью и 16 каналами ввода-вывода. Эта секция включает приоритетную схему прерываний и пути к периферийным подсистемам для сигналов и данных.

Секция индексации, содержащая параллельные индексные сумматоры и схемы тестирования.

Секция управления классами памяти, которая получает адреса последнего операнда от индексного сумматора и устанавливает адрес и маршрут данных к одному из восьми модулей памяти.

Набор команд системы включает следующие типы операций: операции передачи данных, арифметические операции над числами с фиксированной запятой (36-разрядными при нормальной точности и 72-разрядными при двойной точности), арифметические операции над числами с плавающей запятой (с нормальной точностью в диапазоне от 10^{38} до 10^{-37} с 8 значащими цифрами, с двойной точностью в диапазоне от 10^{307} до 10^{-308} с 18 значащими цифрами), арифметические операции над отдельными частями слова, операции над индексными регистрами, логические операции, операции сдвига, операции поиска, операции условной и безусловной передачи управления, команды подготовки следующей команды и операции ввода-вывода.

Система прерываний обеспечивает выполнение операций реального времени, мультипрограммирование и разделение времени. Специальные ячейки прерываний размещаются в начальных адресах основной памяти для каждого условия прерывания. Эти ячейки используются для установления связи с программами, которые захватывают и вводят подпрограммы обработки прерывания в операционную систему. Синхронизация операций ввода-вывода и реакция системы на события, происходящие в реальном времени, выполняются при помощи некоторых из этих прерываний. Другие прерывания используются в системе контроля ошибок в центральном процессоре.

Каждый центральный процессор имеет общую секцию управления вводом-выводом, которая управляет процессом передачи данных между основной памятью и периферийными подсистемами. Данные передаются по одному из 16 двусторонних каналов ввода-вывода (каждый канал имеет 72 линии передачи данных — 36 для ввода и 36 для вывода), и хотя большинство периферийных подсистем являются одновременно входными и выходными, поток данных в канале имеет лишь одно направление для каждой конкретной команды ввода-вывода. Ввод-вывод может выполняться в любом из двух режимов: режим внешней спецификации индексов (ESI) для мультиплексных устройств передачи данных, передающих по 36, 18 или 9 разрядов, и режим внутренней спецификации индексов (ISI) для других типов периферийного оборудования. Все каналы ввода-вывода, кроме канала 15 (пульт оператора), могут работать в любом режиме в зависимости от положения переключателя режимов, связанного с каждым каналом.

При использовании режима внутренней спецификации индексов обеспечивается параллельная передача 36-разрядных слов к (от) периферийным подсистемам в соответствии с содержимым входных (выходных) регистров управления доступом, связанных с данными

каналом в процессоре. Функциональные слова, переданные подсистеме, определяют операции, которые должна выполнить подсистема. Содержание слова и адрес буфера, помещенные процессором в регистр управления доступом при инициации операции ввода-вывода, связаны только с характеристиками канала, а не с устройствами, подключенными к подсистеме. Если переключатель режимов для данного канала установлен в положение режима внешней спецификации индексов, то регистры управления доступом становятся регистром ESI, связанным со списками слов управления доступом, которые расположены в области основной памяти, назначенной для устройств, подключенных к данному каналу. Когда устройство передает данные, оно посылает процессору адрес своего собственного слова управления доступом. Это управляющее слово передается из основной памяти к подсистеме каждый раз, когда данные передаются под управлением подсистемы. После того как слово управления доступом модифицируется подсистемой для индикации окончания передачи данных, оно возвращается в основную память и остается там до тех пор, пока не поступят новые данные от того же источника. При каждой передаче данных к (от) основной памяти требуется три обращения к памяти: два для слова управления доступом и одно для данных, которые могут иметь формат 18-разрядного полуслова или 9-разрядного байта. Построение цепочек буферов данных для операций распределенного чтения групповой записи невозможно осуществить в режиме ISI. Приведенное выше описание двух режимов работы процессора при вводе-выводе можно подытожить следующим замечанием: в режиме ISI процессор внутренне связывает единственный регистр управления буфером с определенным каналом, в то время как в режиме ESI один из многих регистров управления буфером связывается внешне с одним из многих устройств, подключенных к каналу.

Операции ввода-вывода организуются в определенную последовательность системой управления при помощи специальной системы приоритетов, и в каждый момент времени с центральным процессором может быть связан только один канал, хотя все 16 каналов ввода-вывода могут быть одновременно готовы для передачи данных.

ОСНОВНАЯ ПАМЯТЬ

Базовый модуль памяти системы 1108 построен на ферритовых сердечниках и имеет емкость 32К слов. Каждое слово памяти состоит из 36 разрядов и двух дополнительных разрядов контроля четности — по одному разряду на каждое полуслово. Четность проверяется при выполнении операций чтения или записи в каждом случае обращения к памяти. Когда обнаруживается ошибка в четности, банк памяти посылает прерывание по нарушению четности в процессор. Процессор обрабатывает код прерывания и переписывает

вает ошибочное слово в специальную область памяти с целью индикации последующих неправильных результатов и облегчения изоляции ошибочных данных.

Минимальная конфигурация основной памяти для мультипроцессорной системы 1108 включает 2 банка по 65К слов (каждый банк состоит из двух модулей по 32К слов). Эта конфигурация может быть расширена добавлением банков по 65К слов до емкости 262К слов. Основная память имеет время полного цикла обращения 750 нс и обеспечивает доступ до трех процессоров и двух контроллеров ввода-вывода к каждому модулю памяти объемом 32К слов.

Эти характеристики доступа реализуются при помощи устройства мультидоступа к модулю. Устройство мультидоступа может создать пять упорядоченных по приоритетам путей внутри банка к каждому модулю памяти по 32К слов. Если при доступе возникают конфликтные ситуации, устройство предоставляет доступ к памяти запросу, имеющему наибольший приоритет, затем следующему (с меньшим приоритетом) и т. д. Таким образом, связь между процессором и единичным модулем памяти может быть асинхронной. Так как задержки при вводе-выводе могут привести к нежелательным потерям циклов обращения к магнитному барабану, к повторному чтению или повторной записи на магнитную ленту, или к полной потере данных при вводе их в реальном времени, контроллеры ввода-вывода обычно имеют в устройстве мультидоступа наиболее высокий приоритет, а следующий приоритет назначается центральным процессорам.

Аппаратная защита памяти осуществляется по границе каждого блока емкостью 512 слов. Управляющим элементом в этом случае является регистр границы памяти. Он загружается операционной системой с целью задания допустимых границ рабочих областей для каждой программы, находящейся в обработке. Перед каждым обращением к основной памяти процессор осуществляет контроль границ как в области команд, так и в области данных. Если адрес оказывается в области, выходящей за допустимые границы, то генерируется прерывание по условию защиты, управление передается операционной системе, и она выполняет соответствующие действия по контролю и восстановлению. Системой 1108 могут быть установлены два различных режима управления. Один из них — «привилегированный режим» — запрещает привилегированным программам выполнение операций записи за пределами выделенной им области. (К таким программам относятся программы, обеспечивающие работу в реальном времени или управляющие подпрограммы операционной системы.) Другой режим — это режим задач пользователя, в котором осуществляется защита при чтении, записи и переходах в памяти.

Относительная адресация и динамическое перемещение программ в основной памяти осуществляется с помощью базовых регистров. Относительная адресация предусматривает возможность ди-

намического перемещения области памяти, назначенной для одной программы, с целью обеспечения непрерывной рабочей области памяти для другой программы; благодаря этому программы имеют возможность динамически запрашивать дополнительную память в соответствии с их потребностями.

Предусмотрено два специальных метода для обращения к основной памяти, повышающих производительность системы и уменьшающих вероятность возникновения конфликтных ситуаций при доступе. Первый из них называется перекрытием, он дает центральному процессору возможность осуществлять одновременно выборку операнда текущей команды и код следующей команды. Второй называется чередованием; он дает возможность двум процессорам (двум центральным процессорам, двум контроллерам ввода-вывода или одному центральному процессору и одному контроллеру ввода-вывода) получать доступ к паре модулей памяти при минимальном числе конфликтных ситуаций.

В процессе работы системы можно исключить из конфигурации основной памяти один банк емкостью 65К слов для технического обслуживания, причем система в целом будет продолжать функционировать.

КОНТРОЛЛЕР ВВОДА-ВЫВОДА

Контроллер ввода-вывода (рис. П.XIV.2) является независимым устройством, которое управляет работой до 16 периферийных подсистем и находится в свою очередь под управлением трех различных процессоров. Функционально контроллер ввода-вывода сходен с секцией ввода-вывода центрального процессора, которая обсуждалась выше. Сразу после поступления к нему запроса на ввод-вывод контроллер берет на себя полное управление операциями, передавая данные между основной памятью и периферийными устройствами без дальнейшего участия центрального процессора, который инициировал запрос.

Оба режима ISI и ESI обеспечиваются контроллером ввода-вывода, однако имеются некоторые особенности их реализации по сравнению с работой секции ввода-вывода центрального процессора, описанной выше. Как показано на рис. П.XIV.2, контроллер ввода-вывода имеет индексную память емкостью 256 слов (с возможностью расширения до 512 слов), которая содержит слова управления доступом в обоих указанных режимах передачи данных. Эти слова управления аналогичны словам управления доступом ESI, содержащимся в главной памяти и используемым секцией ввода-вывода процессора. Их применение позволяет обойтись одним обращением к основной памяти при каждой передаче данных в режиме ESI, в отличие от трех обращений, которые необходимы при передаче данных в режиме ESI процессором. Помимо этого, в режиме

ESI возможны операции сцепления данных, которые отсутствуют при работе процессора в режиме ESI. В режиме ESI можно эффективно использовать мультиплексные подканалы в каналах, работающих в этом режиме, поэтому подсистемы связи, подключенные к

По два модуля основной памяти емкостью 65 536 слов (с чередованием)

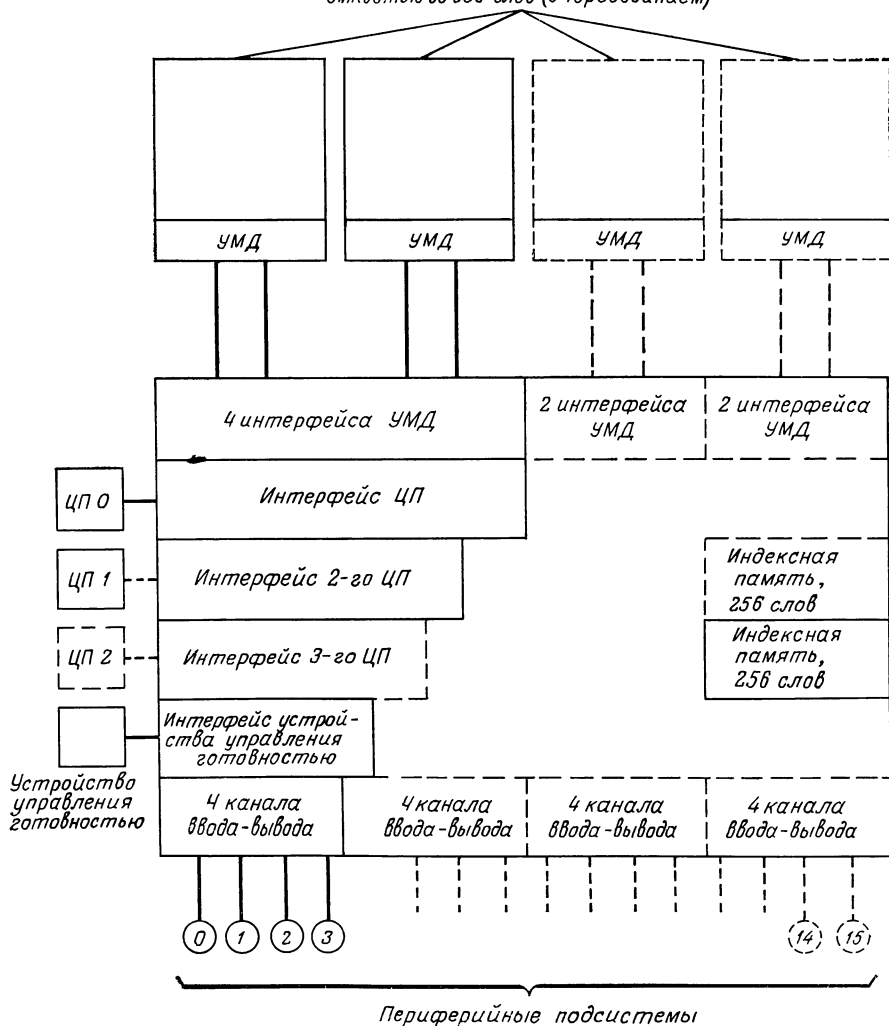


Рис. П.XIV.2. Контроллер ввода-вывода системы UNIVAC 1108.

Сплошными линиями изображена базовая часть системы, пунктирными линиями — дополнительные блоки, расширяющие систему.
УМД — устройство мультимодульного доступа к памяти.

модулям контроллеров удаленных терминалов, могут обслуживаться через один канал ввода-вывода без нарушения программной последовательности операций процессора.

Мультипроцессорная система 1108 включает один или два контроллера ввода-вывода, каждый из которых управляет работой своей группы периферийных подсистем и обеспечивает прямой путь между основной памятью и 4, 8, 12 или 16 быстродействующими двусторонними каналами данных. Каждый канал имеет скорость передачи 1 333 000 слов/с.

Передача данных между периферийными устройствами и основной памятью производится независимо от центрального процессора, благодаря чему процессорам остается больше времени для обработки информации. Прямая передача данных от одного периферийного устройства к другому невозможна, и вся передаваемая информация в этом случае должна быть буферизована в основной памяти системы.

Здесь необходимо обратиться к описаниям системы UNIVAC 1110 и UNIVAC AN/UYK-7, которые приведены в следующих приложениях, чтобы объяснить аналогичные методы буферизации в системах ввода-вывода этих мультипроцессоров. В частности, система 1110 обеспечивает режимы передачи данных ESI и ISI только через специальное устройство доступа к системе ввода-вывода, а не через центральный процессор, как система 1108. Однако это устройство доступа системы 1110 обслуживает до 24 каналов и обеспечивает создание цепочек буферов данных в обоих режимах ввода-вывода, в то время как контроллер ввода-вывода системы 1108 обслуживает только 16 каналов и может организовать цепочки буферов только в режиме ISI. В системе AN/UYK-7 ввод-вывод также осуществляется без помощи процессоров (в противоположность системе 1108), однако в ней предусмотрены нормальный буферный режим и режим ESI, аналогичные режимам ISI и ESI в системе 1108, соответственно. Кроме того, контроллер ввода-вывода системы AN/UYK-7 обеспечивает режим внешней спецификации адреса (ESA) и режим межмашинного обмена, которые дают дополнительные средства управления передачей данных, отсутствующие в системах 1108 и 1110.

ВСПОМОГАТЕЛЬНАЯ, ИЛИ МАССОВАЯ ПАМЯТЬ

С системой 1108 могут использоваться различные запоминающие устройства большой емкости. Вспомогательная память состоит главным образом из быстродействующих магистральных барабанов большой емкости с временем доступа от 4,3 до 17 мс и емкостью от 262К до 22 000К 36-разрядных слов. Характеристики этих запоминающих устройств приведены ниже.

Емкость памяти	262 144 слова по 36 информационных разрядов и 1 разряду контроля четности, или 1 572 864 алфавитно-цифровых символов на один барабан
Среднее время доступа	4,3 мс
Скорость вращения барабана	7200 об/мин
Количество головок чтения-записи	432, по одной на каждую дорожку
Скорость передачи символов	1 440 000; 720 000; 360 000; 180 000; 90 000
Скорость передачи слов	240 000; 120 000; 60 000; 30 000; 15 000
Количество требуемых каналов ввода-вывода	Один на подсистему
Количество барабанов в подсистеме	От 3 до 9 (до 14 155 776 символов)
Емкость памяти	2 097 152 слов по 36 информационных разрядов и 1 разряду контроля четности, или 12 582 912 алфавитно-цифровых символов на один барабан
Среднее время доступа	17 мс
Скорость вращения барабана	1800 об/мин
Количество головок чтения-записи	1782 (33 блока по 54 головки на блок)
Скорость передачи символов	1 440 000; 720 000; 360 000; 180 000; 90 000
Скорость передачи слов	240 000; 120 000; 60 000; 30 000; 15 000
Количество требуемых каналов ввода-вывода	Один или два на подсистему
Количество барабанов в подсистеме (максимальное)	8 (общая емкость 100 663 296 символов)
Емкость памяти	220 020 096 слов, или 132 120 576 алфавитно-цифровых символов на одно устройство
Среднее время доступа	92 мс
Плотность записи	1000 битов/дюйм
Количество дорожек на 1 дюйм	106
Скорость вращения барабана	870 об/мин
Количество подвижных головок чтения-записи	64
Скорость передачи символов	153 540 символов/с
Скорость передачи слов	25 590 слов/с
Количество фиксированных головок чтения-записи	24
Среднее время доступа (с фиксированными головками)	35 мс
Емкость памяти, обслуживаемой одним набором фиксированных головок	258 048 символов
Защита операций записи	Имеется
Каналы ввода-вывода	Один или два на подсистему
Количество барабанов в подсистеме	8 (1 056 964 608 символов на подсистему)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Диспетчеризация и распределение ресурсов в системе 1108 осуществляются операционной системой Ехес-8. Эта система управляет порядком поступления заданий на обработку и непосредственным выполнением программ в системе 1108. Она предназначена для управления одновременной обработкой большого числа программ и исключает какое-либо их взаимное влияние.

В супервизоре системы Ехес-8 имеется три уровня планирования: предварительное планирование, динамическое распределение областей памяти и диспетчеризация ресурсов центрального процессора. Программы (в системе 1108 этот термин обозначает также полное задание), поступающие в систему, сортируются и записываются в виде файла, который используется супервизором для планирования и обработки заданий. Предварительное планирование каждого задания зависит от его приоритета и требований к оборудованию. Динамический распределитель берет задание, установленное в очередь предварительным планировщиком, и выделяет ему область памяти в соответствии с его требованиями. Когда в основной памяти используется режим разделения времени, распределитель инициирует операцию «вытеснения памяти». Эта операция осуществляет обмен информацией между запоминающим устройством на барабане и основной памятью, она используется только для обеспечения приемлемого времени ответа на запрос от удаленного терминала. Уплотнение памяти осуществляется операционной системой в тех случаях, когда в памяти отсутствует достаточно большая непрерывная свободная область, необходимая для обработки задачи.

Подпрограмма «диспетчер» представляет третий уровень планирования. Она определяет, какая из задач, находящихся в основной памяти, должна получить машинное время центрального процессора, и осуществляет функцию передачи управления от одной задачи к другой. Обычно ресурсы времени центрального процессора предоставляются какой-либо программе пакетной обработки, и она использует их до тех пор, пока не оказывается заблокированной в результате какого-либо события в системе или передачи центрального процессора программе (программам) с более высоким приоритетом.

Состав и структура технических средств системы указываются во время генерации системы; операционная система назначает эти средства как необходимые и доступные для выполнения задания. Это осуществляется путем формирования таблицы текущего состояния ресурсов, в которой указаны средства, доступные для назначения, и средства, используемые в данный момент различными заданиями.

Связь оператора с системой осуществляется с пульта, оборудованного клавиатурой и дисплеем. Операционная система передает

на дисплей различную информацию, например данные о текущей загрузке системы, или дает ответы на запросы оператора, связанные со структурой системы ввода-вывода и блокировкой ее элементов. Оператор может затребовать и другую информацию, например, данные о состоянии невыполненных работ.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Операциями ввода-вывода управляет центральная подпрограмма ввода-вывода, которая принимает и устанавливает в очередь запросы и прерывания и управляет устройствами ввода-вывода, когда это необходимо. Обращение к системе управления вводом-выводом в конечном итоге передается подпрограмме, управляющей конкретным устройством, которое назначено для выполнения операций ввода-вывода. Эта подпрограмма в свою очередь рассматривает запрос и устанавливает его в очередь для данной подсистемы ввода-вывода. Когда подсистема освобождается, входная запись извлекается из очереди подсистемы и подпрограмма ввода-вывода запускается с соответствующей точки. Очередь не формируется, если подсистема сразу оказалась свободной.

Запросы на ввод-вывод подразделяются операционной системой на три категории в зависимости от процесса, который посылает запрос. Категориям запросов приписываются соответствующие приоритеты, и все запросы одной категории будут выполняться до конца, прежде чем может быть обслужен какой-либо запрос следующей категории, имеющей более низкий приоритет. Приоритеты категорий запросов ранжируются в следующем порядке: запросы процессов реального времени, запросы операционной системы и запросы пакетной обработки. Всякий раз, когда это необходимо, в рамках каждой категории используется система предварительного просмотра команд в памяти, в результате чего среднее время обработки запросов на ввод-вывод может быть уменьшено.

Система ввода-вывода обеспечивает интерфейс между периферийными устройствами системы 1108 (все стандартные терминалы, устройства чтения перфокарт, карточные перфораторы, печатающие устройства, устройства чтения перфолент, ленточные перфораторы) и программой пользователя. Данные, поступающие к (от) этим устройствам, буферизуются в массовой памяти для обеспечения эффективной связи центрального процессора с асинхронными и сравнительно медленными периферийными устройствами.

Для устройств записи на магнитную ленту, массовой памяти и устройств связи система обеспечивает буферизацию данных путем выделения отдельных буферов или буферного пула. Пул может содержать любое требуемое число буферов. Работа с отдельным буфером наиболее эффективна, так как она не требует значительного

вмешательства со стороны супервизора; кроме того, в этом случае расход основной памяти на один буфер будет несколько меньше, чем при создании буферного пула, поскольку в последнем случае требуются дополнительные области памяти для хранения информации управления пулом. Создаются отдельные буферные пулы для файлов данных и для устройств связи. Каждому каталогизированному файлу, назначенному для определенного задания, должен быть выделен буферный пул. Буферный пул может быть предназначен для одного отдельного файла или для многих файлов. Размер каждого буфера должен быть равен максимальному размеру блока, который задается для файла, плюс три дополнительных слова для обеспечения управления. Буферный пул для устройств управления связью может быть создан в любой части основной памяти, которую пользователь может выбрать и выделить в качестве области ввода-вывода.

Подпрограммы управления наборами данных предназначены для обработки широкого класса форматов файлов. На форматы, принятые в системе 1108, налагаются незначительные ограничения. Файлы могут обрабатываться на уровне элементов, записей или блоков, причем физические характеристики устройств ввода-вывода не учитываются. Доступ к файлу может быть произвольным или последовательным. Каждый файл описывается форматом файла и определением элементов данных. Таблица управления элементами файлов дает возможность пользователю осуществлять по запросу доступ к любому элементу записи.

Предусмотрена подсистема телеобработки, которая находится в резидентной части операционной системы и используется для управления всеми процессами связи с большим числом независимых терминалов. Программа управления связью обеспечивает два режима работы системы телеобработки. Первый относится к управлению буфером: программа управления связью управляет передачей и приемом сообщений в отдельные буферы, независимо от содержания каждого буфера. Второй предполагает наличие формата, заданного системой для устройств, способных участвовать в передаче и подтверждать получение сообщения.

Работа системы в режиме запрос-ответ инициируется и управляется при помощи языка управления обработкой. Команды вводятся через удаленный пульт пользователя в диалоговом режиме. Предусмотрены также следующие возможности: 1) связь удаленного терминала с центральной системой при помощи набора телефонного номера (в дополнение к связи по выделенным каналам и связи с удаленными пультами); 2) ввод информации с перфоленты, предусматривающий предварительную печать программ, для обеспечения высокой эффективности передачи; 3) связь пользователя с вычислительным центром, другими пультами и с самой операционной системой.

Входные устройства сопряжения, работающие в запросном режиме, обычно допускают ввод данных от удаленного терминала. Таким образом, частота вводов в систему определяется работой оператора удаленного терминала. Команды, поступающие от терминала, буферизируются в основной памяти так же, как и входной поток информации с перфокарточных устройств, находящихся в вычислительном центре.

Система управления файлами управляет формированием и использованием всех файлов программ и данных. Эта система формирует и непрерывно корректирует главный справочник файлов, каталогизированных в системе, и таблицы состояния различных областей массовой памяти. Для каждого файла, известного системе (за исключением временных файлов), в главном справочнике файлов выделяется область, содержащая его идентификаторы и характеристики.

Секретность файла обеспечивается при помощи двух ключей, которые должны быть специфицированы в определенном операторе обращения при организации доступа к каталогизированному файлу. Первоначально ключи задаются в операторе обращения, который вызывает каталогизацию файла. Главный справочник содержит счетчик числа доступов к файлу и время последнего доступа. Эти поля в справочнике наряду с системным регистрационным журналом доступны пользователю и предназначены для контроля использования файла и обнаружения случаев нарушения его секретности.

ДИАГНОСТИЧЕСКАЯ ОБРАБОТКА ОШИБОК

Операционная система Ехес-8 содержит стандартную процедуру восстановления для каждого типа отказа системы ввода-вывода. Если в системе ввода-вывода возникает состояние ошибки и пользователь не блокирует ее восстановление, то система Ехес-8 иницирует стандартную процедуру восстановления. В случае, если автоматическое восстановление оказалось невозможным, сообщение об ошибке выводится на экран пульта управления системой, и оператор может ввести специальные управляющие команды для восстановления системы.

К машинным ошибкам относятся: ошибки четности при управлении вводом-выводом данных, ошибки четности при выполнении операций чтения-записи в основной памяти и нарушение электропитания. Если обнаружено нарушение четности при управлении вводом-выводом, то операционная система определяет, какой является ошибка — случайной или постоянной. Если ошибка случайная, то операции ввода-вывода иницируются повторно и управление возвращается к адресу прерывания. Если же ошибка не является случайной, а оборудование другого канала не может принять на себя функции отказавшего канала, то выполнение программы будет

прекращено, а сам канал объявлен неисправным. Если случайная ошибка возникла в каких-либо регистрах, не используемых стандартной программой восстановления, то предпринимается попытка автоматического восстановления.

Обработка ошибок четности при вводе-выводе данных и ошибок четности при чтении-записи в основной памяти осуществляется аналогично. Если возникает случайная ошибка, то программа пользователя получает управление в точке повторного запуска при условии, что такая точка предусмотрена. Если точка повторного запуска не была предусмотрена, то выполнение программы прекращается. Если ошибка возникла внутри операционной системы и повторный запуск невозможен, то система в целом останавливается. Если ошибка не является случайной, то выполнение прерванной программы окончательно прекращается и связанный с ней блок памяти объявляется неисправным. Если искаженный программный модуль операционной системы особенно важен, то система останавливается.

Когда происходят сбои в электропитании, система инициирует следующую последовательность операций: 1) после прерывания сохраняются адрес прерывания и область памяти, связанная с управляющими функциями, а каналы, выполняющие операции ввода-вывода, отмечаются специальными метками; 2) если ЭВМ запускается повторно без полного сброса, то метки повторного запуска операций ввода-вывода снимаются, управляющая память повторно загружается и происходит возвращение к адресу прерывания; 3) если ЭВМ запускается повторно после полного сброса, то отмеченные операции ввода-вывода снова устанавливаются в очередь в списке запросов на ввод-вывод, управляющая память повторно загружается и управление передается прерванной программе.

Условия, возникающие в процессе выполнения программы и приводящие к прерыванию или псевдопрерыванию в ЭВМ, называются программной контингенцией. Пользователь может сам заранее определить, каким образом он будет обрабатывать такие прерывания, если стандартные средства, предусмотренные в системе, ему по какой-либо причине не подходят. Известны следующие типы программных контингенций: применение неразрешенных кодов операций, неисправности основной памяти, использование кодов привилегированных команд в прикладных программах, переполнение в операциях над числами с плавающей запятой или выход результата за разрядную сетку в области младших разрядов, переполнение при делении, прерывания от схем контроля и блокировки. Прерывания последнего типа возникают тогда, когда программа пользователя пытается получить доступ к системным данным, которые в текущий момент заблокированы другой обрабатываемой программой.

ПРИЛОЖЕНИЕ XV

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА UNIVAC 1110 ФИРМЫ SPERRY RAND CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Система 1110 состоит из трех основных групп элементов: процессоров, памяти и периферийных подсистем. Каждый элемент функционально независим и имеет следующие характеристики:

Два или больше трактов доступа.

Разрешение конфликтных ситуаций при доступе на основе приоритетной логики.

Сохранение непрерывной работоспособности системы при выходе из строя любого отдельного элемента системы.

Возможность логического отключения отдельных элементов для их технического обслуживания без остановки работы всей системы.

Базовая конфигурация этой вычислительной системы включает три функционально и физически независимых устройства: два командно-арифметических устройства и одно устройство доступа к системе ввода-вывода. Организация процессора предназначена для работы в мультипрограммном и мультипроцессорном режимах. Базовый процессор может быть расширен за счет добавления отдельных командно-арифметических устройств (до 4) и устройств доступа к системе ввода-вывода (также до 4). Базовая конфигурация системы показана на рис. П.XV.1.

В системе 1110 все управляющие и арифметические функции выполняются с помощью двух командно-арифметических устройств. Каждое из них является многозадачным устройством со стековой организацией команд, способным управлять выполнением до четырех команд в различных стадиях обработки. Каждое командно-арифметическое устройство может устанавливать интерфейс с четырьмя блоками основной памяти по трактам доступа к командам и трактам доступа к операндам. Двойные тракты передачи данных связывают каждое устройство с расширенной памятью через интерфейсы мультидоступа, максимальное число которых может достигать восьми. Тракты передачи данных к основной и расширенной памяти имеют возможность работать с перекрытием и чередованием.

Каждое командно-арифметическое устройство может вести обработку со скоростью 1,8 млн. команд/с и имеет время выполнения одной базовой команды 300 нс. Устройство обеспечивает также обработку символов при помощи команд, ориентированных на работу с байтами.

Устройство доступа к системе ввода-вывода управляет всеми процессами обмена данными между периферийными устройствами и основной или расширенной памятью, причем обмены инициируются этим устройством и осуществляются под программным управлением. Устройство доступа к системе ввода-вывода включает два тракта передачи данных, работающих в режиме совпадения, — один для основной и один для расширенной памяти. Устройство состоит из двух секций: секции управления, которая включает всю логику, связанную с передачей функций, данных и слов состояния между основной или расширенной памятью и подсистемами, и секции, содержащей от 8 до 24 каналов ввода-вывода.

Системный пульт обеспечивает средства для связи с операционной системой и состоит из следующих основных элементов:

Дисплей с клавиатурой и специальным печатающим устройством. Система связи, работающая в реальном времени, которая обеспечивает возможность организации диагностических проверок с удаленных пунктов по телефонным каналам.

Индикатор сбоев, осуществляющий визуальную индикацию состояния ошибки в основных элементах системы.

Устройство расчленения системы, которое включается в систему по желанию, позволяет осуществлять техническое обслуживание устройств системы, временно исключенных из конфигурации, дает оператору возможность логического деления системы на две или три независимые подсистемы и инициирует последовательность действий по восстановлению работоспособности в случае сбоя. При помощи устройства расчленения системы оператор получает возможность:

расчленить систему на две или три меньшие системы;
изолировать устройства и вывести их из системы без нарушения работы оставшейся части системы;
выполнять функции системного монитора, осуществляя контроль состояния основных элементов;
осуществлять первоначальную загрузку системы при вводе ее в действие;
запускать процедуры автоматического восстановления системы в тех случаях, когда прерывания не предусмотрены.

Под управлением программного обеспечения устройство расчленения представляет устройствам доступа к системе ввода-вывода информацию о состоянии ресурсов системы.

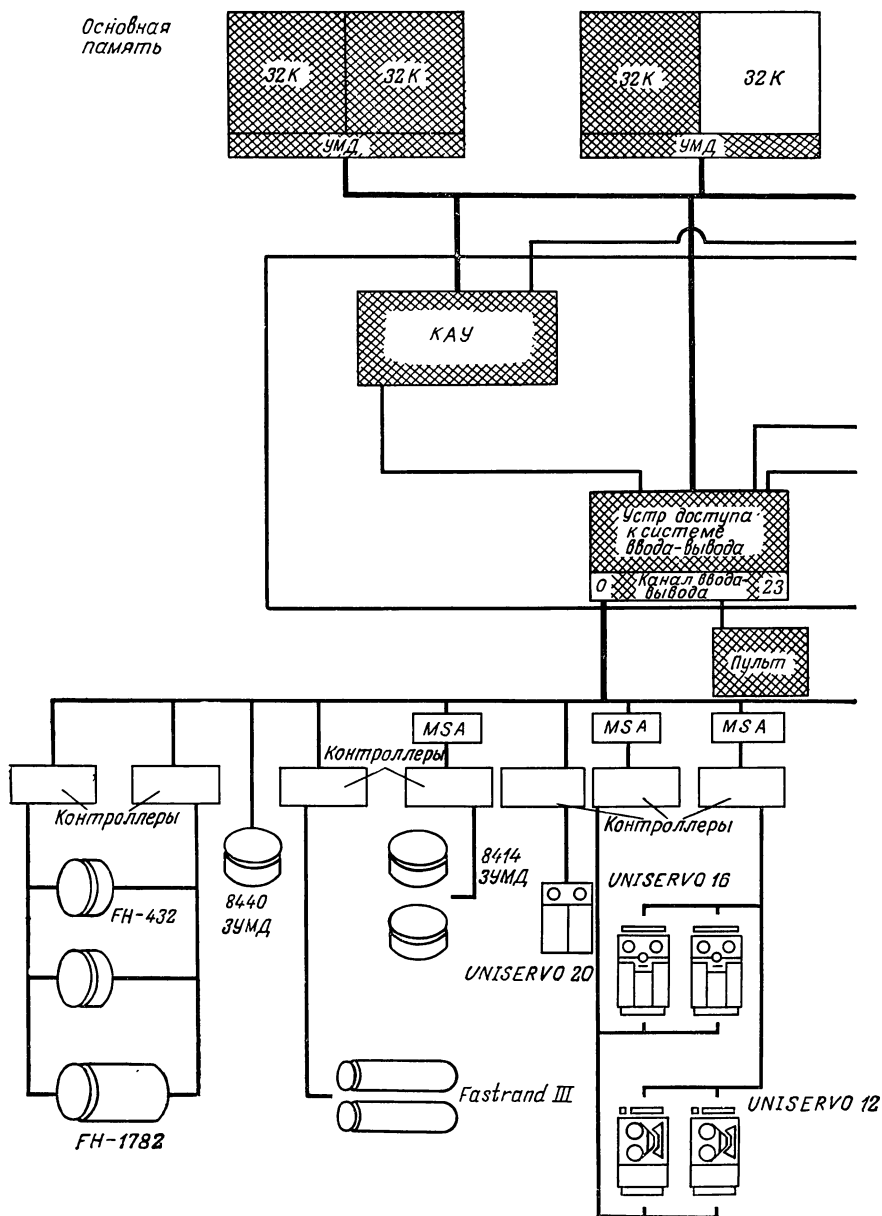
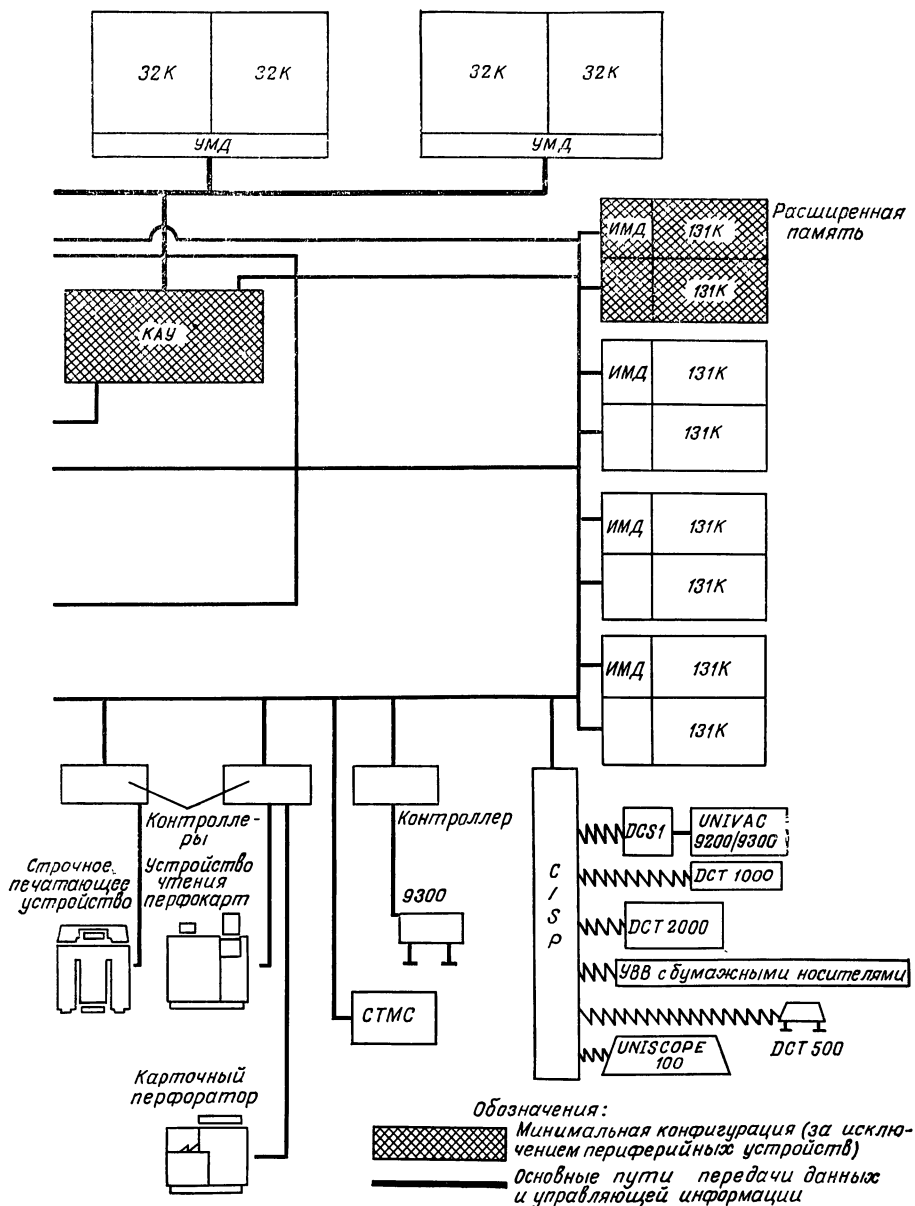


Рис. П.XV.1. Конфигурация процессора и УМД — устройство мультимодульного доступа к памяти, ИМД — интерфейс



основной памяти системы UNIVAC 1110.
 мультидоступа, КАУ — командно-арифметическое устройство.

В системе 1110 основная память состоит из быстродействующих элементов памяти на тонких пленках с прямой адресацией и без разрушения информации при считывании. Базовый блок памяти емкостью 32К 36-разрядных слов состоит из четырех модулей по 8К слов и может быть расширен добавлением блоков по 32К слов до максимальной емкости 264К слов. Минимальная емкость памяти, требуемая для работы системы 1110, составляет 96К слов.

Второй уровень прямо адресуемой памяти в системе 1110 представляет собой систему расширенной памяти. Минимальная конфигурация расширенной памяти состоит из двух блоков, каждый по 131К 36-разрядных слов. Расширенная память может быть увеличена до 1048К 36-разрядных слов добавлением блоков по 131К слов. Расширенная память подключается к системе через интерфейс мультидоступа, который обслуживает до десяти каналов для каждого блока памяти.

В системе предусмотрен программируемый 18-разрядный таймер интервалов. Время его рабочего цикла равно 200 мкс; каждый раз по истечении этого периода содержимое счетчика таймера уменьшается на единицу, и, когда оно достигает нуля, в системе генерируется прерывание.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

Командно-арифметическое устройство представляет собой многозадачный процессор со стековой организацией команд. Базовая конфигурация этого устройства показана на рис. П.XV.2.

Командно-арифметическое устройство может управлять одним или двумя устройствами доступа к системе ввода-вывода и состоит из пяти взаимодействующих секций:

Секция формирования адреса. В этой секции определяются относительные и абсолютные адреса операндов и команд. В течение цикла генерации адреса операнда проверяются конфликтные ситуации при доступе к памяти, формируется базовый адрес, контролируются ограничения программы и осуществляется приращение индекса.

Стек регистров общего назначения. Стек регистров общего назначения состоит из 112 программно адресуемых регистров управления на интегральных схемах, каждый емкостью 36 битов. Стек включает индексные регистры (18 регистров), накопители, счетчики повторений, регистр маски, регистр часов реального времени и регистр областей памяти, в которых размещается регистр состояния процессора. Стек подразделяется на четную и нечетную области для организации одновременного обращения к памяти

при выполнении команд двойной точности с 72-разрядным форматом.

Секция условного перехода. Эта секция минимизирует время обработки информации в командно-арифметическом устройстве, когда встречаются команды условного перехода. Секция про-

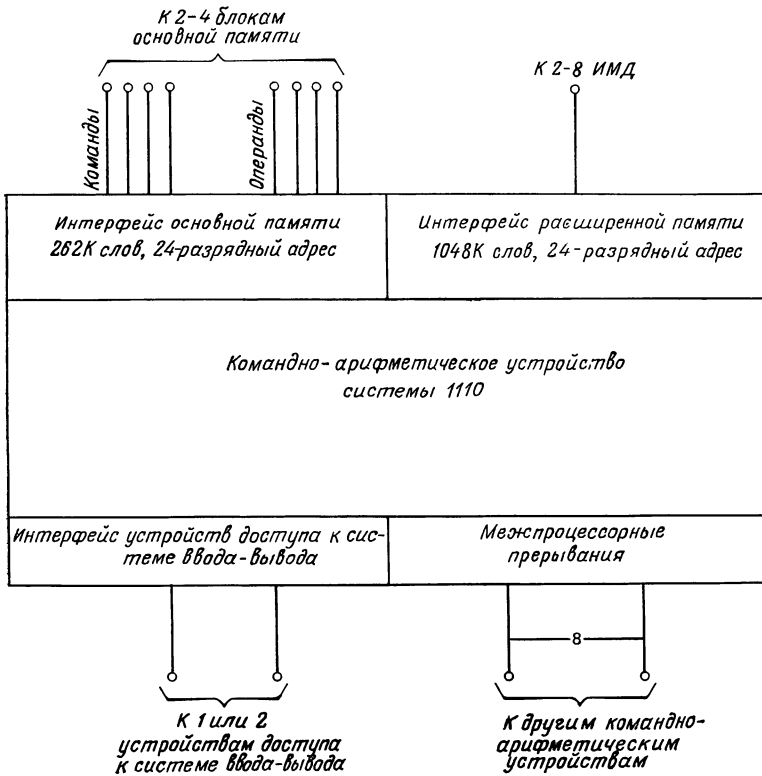


Рис. П.XV.2. Конфигурация командно-арифметического устройства.

веряет выполнение условия перехода и передает управление арифметической секции.

Секция выполнения операций в памяти. Эта секция является частью секции управления командно-арифметического устройства, она управляет выполнением операций записи в память: Секция выполняет операции, необходимые для передачи в память полных слов и их частей.

Арифметическая секция. В этой секции производятся такие операции над данными, как сложение, вычитание, умножение, деление, сдвиг, -

Быстродействие командно-арифметического устройства в системе 1110 составляет 1,8 млн. команд/с, а эффективное время исполнения базовой команды равно 300 нс. Набор команд включает команды передачи частей слов или расщепленных слов, команды арифметических операций над расщепленными словами, команды сдвига, команды операций с двойной точностью над числами с фиксированной и плавающей запятой, команды загрузки и хранения, команды повторного поиска, команды подготовки следующей команды, управляющие команды операционной системы, команды перехода, логические команды, команды ввода-вывода, команды операций над байтами, команды преобразования из байтовой в битовую форму и команды операций десятичной арифметики.

ЦЕНТРАЛЬНАЯ ПАМЯТЬ

Система 1110 включает двухуровневую иерархическую исполнительную память с прямой адресацией. Внутри системы эта память трактуется как совокупность элементов системы, аналогичных периферийным устройствам. Первый уровень, или основная память, имеет время считывания 320 нс и время записи 520 нс. Основная память состоит из модулей на тонких пленках (без потери информации при считывании) и встроенных устройств мультидоступа. Основная память обеспечивает одновременный доступ к модулям по 8К слов и контроль по четности адресов и данных. Доступ к основной памяти осуществляется через устройства мультидоступа; его могут одновременно получить до 4 командно-арифметических устройств и до 4 устройств доступа к системе ввода-вывода. Основная память может быть расширена добавлением блоков по 32К слов до максимальной емкости 262К слов. Обеспечивается возможность доступа к памяти с чередованием (за счет четной и нечетной адресации двух смежных модулей по 8К слов) и разрешения конфликтных ситуаций, связанных с доступом в пределах модулей емкостью 8К слов.

Каждый блок основной памяти (32К слов) имеет свой собственный источник питания, устройство мультидоступа и панель для технического обслуживания. Каждое слово состоит из 36 информационных разрядов, двух разрядов контроля четности и двух резервных разрядов. Блок основной памяти емкостью 32К слов способен одновременно обслуживать 4 запроса, по одному на каждый модуль по 8К слов. Расширенная до 65К слов основная память может обслуживать до 8 одновременно пришедших запросов.

Устройство мультидоступа физически расположено в том же шкафу, что и блок основной памяти, а функционально размещено между блоком основной памяти, командно-арифметическими устройствами и устройствами доступа к системе ввода-вывода. Устройство мультидоступа обеспечивает 8 (возможно расширение до 16) упорядочен-

ных по приоритетам трактов к каждому модулю основной памяти. Если возникает соперничество среди устройств, запрашивающих доступ, то устройство мультидоступа разрешает доступ к основной памяти процессору, имеющему наибольший приоритет. Задержки при передаче информации для наиболее предпочтительных операций ввода-вывода могут быть минимизированы путем подключения устройства доступа к системе ввода-вывода к трактам устройства мультидоступа с более высоким приоритетом.

Второй уровень памяти с прямой адресацией называется расширенной памятью. Эта память имеет модули по 131К слов и может быть расширена добавлением блоков по 131К слов до максимальной емкости 1048К слов. Такая память имеет время цикла чтение-записи 1,5 мкс и предусматривает проверку четности в адресах и данных. Доступ к расширенной памяти осуществляется через интерфейсы мультидоступа. Эти интерфейсы выполняют те же функции, что и рассмотренные выше устройства мультидоступа к основной памяти и могут сопрягаться с четырьмя командно-арифметическими устройствами и четырьмя устройствами доступа к системе ввода-вывода.

Память обладает следующими основными характеристиками:

- независимый доступ к модулям;
- непрерывная структура адресации.

Для предотвращения случайного обращения программы к области памяти, находящейся вне разрешенного для нее адресного пространства, система 1110 имеет аппаратную защиту памяти. Эта функция осуществляется регистром границ памяти, который загружается операционной системой для каждой программы, обрабатываемой в данный момент. Регистр определяет как границы команд, так и границы наборов данных.

МОДУЛЬ УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Устройство доступа к системе ввода-вывода полностью управляет всеми операциями ввода-вывода. Базовая секция устройства имеет 8 каналов и может быть расширена до 24 каналов. Общая скорость передачи данных, которую может обеспечить устройство, составляет 4 млн. 36-разрядных слов/с (24 млн. символов/с). Любой канал может работать в режиме внутренней спецификации индекса (ISI) или в режиме внешней спецификации индекса (ESI). При работе в режимах ESI и ISI обеспечивается образование цепочек данных с целью реализации в устройстве доступа к системе ввода-вывода операций распределенного чтения — групповой записи. В режиме ESI осуществляется формирование таблиц, необходимых для организации внешних прерываний или прерываний от монитора при помощи табличных указателей для каждого канала.

Работа системы ввода-вывода в режиме ESI совместно с оборудованием передачи данных позволяет организовать связь удаленных мультиплексированных устройств с основной памятью системы через единственный канал ввода-вывода на основе автономного управления без нарушения работы основной программы. Каждое такое удаленное устройство устанавливает связь со своей собственной областью в основной памяти. Любой канал ввода-вывода может быть переключен на работу в режиме ESI при помощи переключателя или коммутационной карты. Кроме того, при помощи таких коммутационных карт канал в режиме ESI может быть подготовлен для работы с элементами информации длиной в полуслово (18 разрядов) или четверть слова (9 разрядов). Так как канал ввода-вывода в режиме ESI может быть использован многими устройствами, поток данных должен регулироваться словом управления доступом, которое определяется однозначно для каждого устройства, работающего в данный момент, а не для канала, как в режиме ISI. Эти слова управления доступом размещаются в памяти с относительными адресами, назначенными для соответствующих устройств. Когда устройство начинает передавать данные, оно сообщает адрес своего слова управления доступом, и поэтому здесь нет необходимости в сложных программах-мониторах для управления потоком данных.

ВСПОМОГАТЕЛЬНАЯ, ИЛИ МАССОВАЯ ПАМЯТЬ

Стандартный вариант конфигурации массовых запоминающих устройств для системы 1110 в основном такой же, как для системы 1108; он описан в соответствующем разделе предыдущей главы.

В качестве дополнительного варианта система 1110 может включать две дисковые подсистемы. Одна подсистема имеет емкость памяти до 39,44 млн. 36-разрядных слов, скорость передачи данных 69 444 слов/с и среднее время доступа 60 мс. Каждый пакет дисков в этой подсистеме имеет емкость 5 млн. 36-разрядных слов.

Другая дисковая подсистема имеет емкость памяти до 152 млн. 36-разрядных слов, скорость передачи данных 138 888 слов/с и среднее время доступа 35 мс.

ХАРАКТЕРИСТИКИ ОПЕРАЦИОННОЙ СИСТЕМЫ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Основным элементом операционной системы UNIVAC 1110 является супервизор, который управляет определением последовательности, инициацией и выполнением всех программ в системе. Он предназначен для управления обработкой большого числа программ без какого-либо взаимодействия между ними. Супервизор содержит

три уровня планирования: предварительное планирование, динамическое распределение пространства памяти и диспетчеризацию ресурсов командно-арифметического устройства. Задания, поступающие на вход системы UNIVAC 1110, сортируются и записываются в информационных файлах, а затем эти файлы используются супервизором для планирования и управления заданиями. Операторы управления для каждого задания извлекаются и просматриваются интерпретатором управляющих команд супервизора с целью выбора последовательности заданий для подготовки их в процессе предварительного планирования. Такое планирование каждого задания зависит главным образом от двух факторов: приоритета задания и его требований к оборудованию.

Динамический распределитель рассматривает задания, подготовленные предварительным планировщиком, и распределяет пространство памяти в соответствии с потребностями каждой задачи задания. Можно считать, что каждое задание состоит из задач (под задачей подразумевается некоторая элементарная процедура по обработке информации в процессоре системы или выполнение некоторой программы пользователя). Все задачи определенного задания обрабатываются последовательно, но не обязательно одна за другой. Если имеются несколько заданий, то задачи отдельных заданий могут выполняться в режиме чередования.

Когда система работает в режиме разделения времени, динамический распределитель инициирует операции «вытеснения» памяти. При этом одна программа в основной памяти временно заменяется другой, а сама переписывается на магнитный барабан. Такие действия предпринимаются только для сокращения времени отклика на запросы с удаленных терминалов, работающих в запросном режиме.

Подпрограмма диспетчеризации ресурсов процессора осуществляет третий уровень планирования. Она рассматривает различные задачи, находящиеся в памяти, и решает, нужно ли переключить одно из командно-арифметических устройств с обработки одного задания на обработку другого. В нормальных условиях командно-арифметическое устройство может использоваться программой пакетной обработки до тех пор, пока она не будет заблокирована в результате возникновения какого-либо системного события или пока какая-либо программа с более высоким приоритетом не будет готова к обработке.

Состав и размещение оборудования задаются системе при ее генерации. После этого операционная система распределяет это оборудование по мере его необходимости и доступности. Операционная система формирует и ведет таблицы текущего состояния, которые содержат информацию о том, какое оборудование доступно для использования и какое уже занято различными заданиями, выполняемыми в данное время.

Связь оператора с системой осуществляется через пульт оператора, оборудованный дисплеем с клавиатурой. Операционная система выдает на экран информацию о текущей загрузке системы, ответы на запросы оператора о состоянии системы ввода-вывода и блокировках отдельных устройств. Помимо этого, оператор может запросить информацию о состоянии невыполненных заданий.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Управление вводом-выводом осуществляется точно так же, как в системе 1108 при использовании операционной системы Ехес-8.

ДИАГНОСТИЧЕСКАЯ ОБРАБОТКА ОШИБОК

Диагностическая обработка ошибок осуществляется точно так же, как в системе 1108.

ПРИЛОЖЕНИЕ XVI

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА UNIVAC AN/UUK-7 ФИРМЫ SPERRY RAND CORPORATION

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

ЭВМ AN/UUK-7 представляет собой мультипроцессорную систему военного назначения. Система предназначена для работы в реальном времени с ориентацией на пакетную обработку и характеризуется высокой надежностью, легкостью обслуживания, а также возможностью быстро устранять и заменять отказавшие модули. Система состоит из функциональных автономных модулей, которые размещаются в одном или нескольких шкафах AN/UUK-7. Конструкция шкафов и модулей позволяет быстро осуществлять замену съемных модулей.

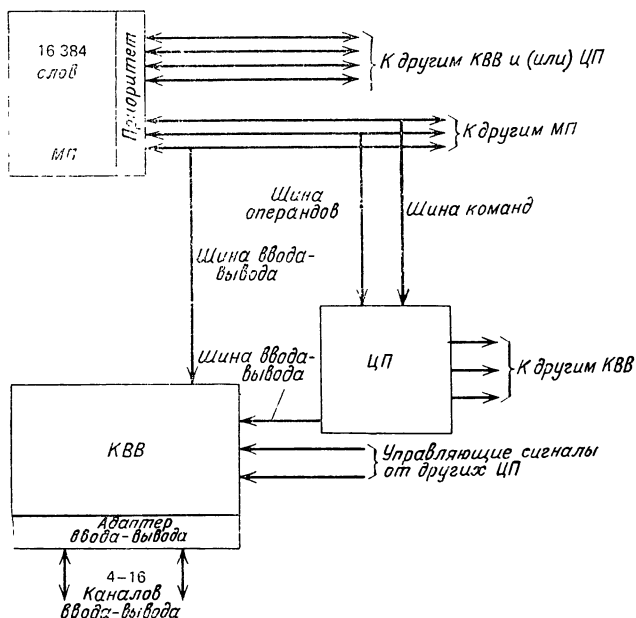


Рис. П.XVI.1. Конфигурация системы AN/UUK-7.
МГ — модуль памяти, КВВ — контроллер ввода-вывода.

Основные модули системы AN/UUK-7 — это центральные процессоры (ЦП), контроллеры ввода-вывода, адаптеры ввода-вывода, блоки памяти, а также блоки электропитания. Функциональная взаимосвязь этих модулей показана на рис. П.XVI.1. Минимальная конфигурация системы содержит один ЦП, один контроллер ввода-вывода, один четырехканальный адаптер ввода-вывода, три модуля памяти емкостью 16К слов каждый и один блок питания (размещающиеся в одном шкафу). Возможно расширение минимальной конфигурации до трех ЦП, четырех контроллеров ввода-вывода, четырех адаптеров ввода-вывода и ферритовой памяти емкостью 256К. Каждый модуль памяти имеет восемь трактов доступа, которые используются совместно следующим образом: два тракта на каждый ЦП и один тракт на каждый контроллер ввода-вывода. Таким образом, совместный доступ к общей памяти может осуществляться максимум тремя ЦП и двумя контроллерами ввода-вывода или двумя ЦП и четырьмя контроллерами ввода-вывода. В мультипроцессорной конфигурации все ЦП имеют одинаковый статус и могут функционировать либо в режиме задачи, либо в режиме прерывания. Прерывания посылаются во все процессоры, включенные в конфигурацию, за исключением прерываний от монитора, которые посылаются только в процессоры, допускающие прерывания от монитора в канале прерывания (управляемом процессором).

ФУНКЦИОНАЛЬНЫЕ БЛОКИ

ПРОЦЕССОР

ЦП включает все управляющие, арифметические и временные схемы, требуемые для обработки алфавитно-цифровых данных и для выполнения управляющих функций. Каждый центральный процессор может адресоваться к памяти объемом 256К слов через посредство двух шин: одна для команд и одна для операндов. При наличии двух или большего числа модулей памяти такая конфигурация позволяет каждому ЦП осуществлять выполнение команды и выборку следующей команды с перекрытием во времени независимо от конфликтных ситуаций, возникающих при обращении к памяти. В двух- или трехпроцессорной системе каждый ЦП может быть подсоединен ко всем модулям памяти и контроллерам ввода-вывода.

Центральные процессоры функционируют в двух различных режимах, или состояниях: в режиме прерывания процессор выполняет управляющие функции, а в режиме задачи процессор обрабатывает прикладные, или рабочие программы. В целях облегчения переключения режимов каждый процессор содержит два набора из семи индексных регистров и восьми регистров базового адреса, а также два набора из восьми адресуемых накапливающих сумматоров. Наличие этих средств исключает необходимость хранения и восстано-

ления базовой информации на регистрах при прекращении и восстановлении режима задачи. В режиме прерывания используется другой заданный набор управляющих регистров.

Предусмотрены аппаратные средства для выполнения операций над числами с фиксированной и с плавающей запятой с использованием-8, 16-, 32- или 64-разрядных операндов. Набор команд включает 130 основных команд, имеющих формат полного слова и полуслова, с прямой или косвенной адресацией и адресацией строк символов переменной длины, а также набор из 18 привилегированных команд. Кроме того, каждый процессор снабжен управляющим таймером, работающим со скоростью 1024 отсчетов/с.

Каждый ЦП включает блок памяти емкостью 512 слов на магнитных сердечниках со считыванием без разрушения информации (NDRO), в котором хранятся программы анализа аппаратных прерываний, начальной загрузки и автоматического восстановления, а также диагностическая программа. В случае необходимости в этой памяти можно хранить и другие программы, однако их следует отбрасывать и вносить в память при изготовлении системы. Память NDRO отделена от общей адресуемой основной памяти, причем тип используемой памяти (NDRO или основная) зависит от рабочего состояния системы, а также от положения переключателя на пульте оператора. Прерывания по сбоям в аппаратных средствах вызывают в ЦП обращение к программе анализа аппаратных прерываний для реализации диагностической проверки и перехода к процедуре восстановления или останова. В зависимости от причин, вызвавших прерывание, программа из памяти NDRO может передавать управление другой программе или перейти к обусловленному останovu.

В качестве управляющей памяти для каждого ЦП служат 82 регистра на интегральных схемах с произвольным доступом. Время выборки для каждого стека в этих регистрах различно и зависит от временных требований к управляющей и арифметической секциям ЦП. Управляющая память ЦП содержит следующие элементы:

Наименование элемента	Режим задачи	Режим прерывания
Арифметический накапливающий регистр	8	8
Индексный регистр	7	7
Регистр базового адреса	8	8
Регистр точки останова	—	1
Регистр активного состояния	—	1
Монитор центрального процессора	—	1
Слово указателя памяти	—	8
Слово начального условия	—	8
Регистр защиты памяти	—	8
Регистр идентификации сегмента	—	8
Неприсваиваемый регистр	—	1

Один из разрядов в регистре активного состояния определяет, в каком режиме (прерывания или задачи) будут использоваться арифметические и индексные регистры. Хотя предусмотрено наличие двух наборов регистров базового адреса, оба они адресуются главным образом в режиме прерывания. Регистр точки останова в соответствии с заполнением управляющих разрядов позволяет провести сравнение содержимого этого регистра с адресом и (или) операндом каждой выполняемой команды. При их совпадении происходит прерывание в точке останова в программе или общий останов машины (когда управление осуществляется оператором с пульта системы). Регистр активного состояния управляет состоянием различных операций в процессоре и осуществляет индикацию состояния. Содержимое регистра идентифицирует ЦП и определяет класс происшедшего прерывания, маски блокировки прерываний с более низким приоритетом, включение-выключение блокировки памяти, состояние индикатора режимов задача — прерывание, а также индикаторов базового, индексного и арифметического регистров. Слова указателя памяти и слова начального условия используются процессором при выполнении прерываний.

Каждый процессор снабжен 20-разрядным регистром адреса программы, в котором 3 разряда указывают один из восьми базовых регистров той группы, с которой работает ЭВМ, а 16 младших разрядов содержат относительный адрес следующей команды. При определении адреса каждая команда дополнительно указывает адрес смещения, который может быть задан в одном из семи индексных регистров для формирования относительного адреса. Окончательный адрес эффективного *операнда* формируется путем прибавления выбранного базового адреса к этому относительному адресу. Окончательный адрес *команды* формируется путем прибавления содержимого базового регистра, определяемого тремя разрядами регистра адреса программы, к относительному адресу, содержащемуся в 16 младших разрядах этого регистра.

При помощи средств блокировки памяти можно предотвратить доступ рабочих программ к определенным сегментам основной памяти. Любой процессор в режиме прерывания может осуществить блокировку выполнения операций записи и (или) чтения в определенных областях какого-либо модуля памяти для своих собственных управляющих операционных программ. Блокировка прекращается, когда процессор возвращается к режиму до прерывания, и вся область памяти становится доступной для управляющей программы. Выполнение функций блокировки памяти регулируется тремя группами соответствующих регистров управляющей памяти. Для любого блока памяти емкостью до 65К слов существуют следующие средства управления: базовый регистр, содержащий начальный адрес; связанный с ним регистр защиты памяти, который определяет функцию блокировки и размер блока смещения; регистр идентификации сег-

мента, содержащий относительный адрес в основной памяти, из которого передаются данные о блокировке. Средства защиты памяти по отношению к сегменту, заданному базовым регистром и соответствующим регистром защиты, управляют следующими операциями в режиме задачи:

1. В пределах защищенной области

запрещение или разрешение чтения операндов;
запрещение или разрешение хранения операндов;
запрещение или разрешение выполнения команд;
запрещение или разрешение косвенной адресации.

2. За пределами защищенной области

запрещение любых ссылок на операнды;
запрещение выполнения команд.

3. Запрещение или разрешение использования набора индексных или базовых регистров в режиме прерывания для косвенной адресации.

Для того чтобы обеспечить всестороннюю защиту памяти, работа каждого процессора должна быть запрограммирована в соответствии с текущим состоянием системы, поскольку заблокированная область одного процессора может свободно использоваться другим процессором. Кроме того, активный контроллер ввода-вывода может установить связь с заблокированной областью какого-либо процессора.

Вводом и выводом информации полностью управляет модуль контроллера ввода-вывода, адресуемый центральным процессором. Цепочки команд ввода-вывода хранятся в модуле памяти, доступном для контроллера при выполнении команд. ЦП выполняет привилегированную команду инициации ввода-вывода, которая идентифицирует контроллер (один из четырех возможных) и адрес первой команды в последовательности программ ввода-вывода. Адресуемый контроллер получает абсолютный адрес на шине операндов памяти-процессор, которая подсоединена также к контроллеру. Последующие действия управляются программой, находящейся в ведении контроллера.

Система UYK-7 обрабатывает данные, поступающие из нескольких источников в реальном времени с учетом происходящих событий или в соответствии с планом. Прерывания могут возникнуть в каком-либо удаленном периферийном устройстве или в самой ЭВМ. Так как одновременно могут происходить несколько прерываний, процессор снабжен схемой приоритетов и методами принятия решений, которые обеспечивают возможность выбора соответствующей подпрограммы, необходимой для решения самой неотложной задачи. При передаче управления выбранной программе все другие прерыв-

вания либо образуют очередь в соответствии с их приоритетами, либо вообще игнорируются. Прерывания в системе УУК-7 обрабатываются при помощи специальной исполнительской программы, когда ЦП находится в режиме прерывания. Можно выделить четыре класса прерываний в порядке возрастания приоритета:

Класс I — прерывания из-за аппаратных сбоев и ошибок, а также прерывания при недопустимом отклонении напряжения питания, которые никогда не блокируются.

Класс II — прерывания из-за сбоев и ошибок в программе.

Класс III — прерывания из-за сбоев в программах ввода-вывода; прерывания, инициируемые мониторами при управлении обменом входных-выходных данных; прерывания по часам монитора контроллера ввода-вывода.

Класс IV — прерывания при инициируемом программой переходе системы в режим прерывания, требующий обслуживания со стороны операционной системы.

При выполнении прерывания любого класса переход в режим прерывания блокирует все другие прерывания данного класса, а также прерывания тех классов, которые имеют более низкий приоритет. Коды прерывания генерируются как контроллером ввода-вывода, так и процессором. В ЦП формируются кодовые слова для прерываний, связанных с работой процессора, а в контроллере ввода-вывода — кодовые слова для прерываний, обусловленных контроллером ввода-вывода. Контроллер ввода-вывода генерирует 8-разрядный код прерывания для каждого запроса на прерывание; процессор, обрабатывающий прерывание, назначает дополнительный 2-разрядный код, идентифицирующий контроллер ввода-вывода, который посылает прерывание.

Каждый раз, когда контроллер ввода-вывода генерирует прерывание класса I или III по условию неправильно сформированных команд, от часов монитора и от ЦП, он посылает запрос на прерывание во все процессоры системы. Запросы на прерывание от монитора (класс III) посылаются в ЦП, которые разрешают осуществить это прерывание в канале прерывания (управляемом программой процессора). В процессор, который первым реагирует на запрос о прерывании, передается код прерывания, имеющий наивысший приоритет в пределах данного класса, если он в данный момент не заблокирован. Так как здесь могло быть и любое другое прерывание, помимо того, которое генерирует запрос, схемы контроллера ввода-вывода могут поддерживать очередь прерываний, ожидающих своевременного ответа со стороны процессора. Процессор, отвечающий на запрос контроллера ввода-вывода о прерывании, уже обрабатываемый другим процессором, получит сигнал NO OPERATION (не обрабатывать). После этого он продолжает выполнение своих операций без изменения режима.

При выполнении прерывания процессор запоминает в регистре слова указателя памяти содержимое регистра адреса программы, регистра активного состояния и код режима прерывания. Для отражения новых условий в регистр текущего активного состояния будут внесены соответствующие изменения. После завершения процесса вызова прерывания адрес программы будет перемещен из слова начального условия, связанного с классом прерывания в управляющей памяти, в адрес подпрограммы прерывания. Содержимое ячейки слова начального условия в управляющей памяти первоначально устанавливается операционной системой. Реализация программы прерывания заканчивается выполнением команд возврата прерывания. Управление возвращается прерванному режиму путем восстановления в управляющей памяти содержимого регистров активного состояния и адреса программы, которое имело место в момент прерывания.

Прерывания, происходящие синхронно с выполнением операций в ЦП, не задерживаются в состоянии ожидания, если они заблокированы. Это относится ко всем прерываниям класса II, за исключением межпроцессорных прерываний и прерываний по часам монитора ЦП, а также к прерываниям класса I, за исключением прерываний из-за останова ЭВМ и возобновления работы контроллера ввода-вывода. Все другие виды прерываний, возникающие асинхронно с выполнением операций ЦП, находятся в состоянии ожидания, и процессор может обнаружить их в процессе просмотра и анализа приоритетов последовательности команд. Прерывание, которое не было заблокировано, может приостановить работающую программу, даже если программа сама обрабатывает прерывание. Следовательно, прерывания могут переходить каскадами от класса к классу или в пределах одного класса, в результате очистки соответствующего указателя блокировки в регистре активного состояния.

С точки зрения обеспечения надежности и восстановления и повторного запуска системы особый интерес представляет прерывание при отклонении напряжения питания от заданного уровня. Источник питания содержит средства защиты памяти от передачи данных, которые могут оказаться искаженными при прерываниях из-за нарушений первичного электропитания. Однако, если периферийные устройства попытаются передать данные в память в период ее защиты при сбоях в питании, эти данные могут быть утеряны в интерфейсе ввода. Выход системы из состояния полного повреждения питания или временного отклонения напряжения от заданных значений может осуществляться без вмешательства оператора. Прерывание генерируется, когда напряжение питания падает ниже допустимого уровня. При этом ЭВМ может продолжить функционировать в течение 250 мкс на запасах энергии, накопленных в аккумуляторах блока питания. Подпрограмма прерывания при повреждении питания содержит команду перехода, задаваемую вручную, для пе-

редачи управления в программу повторного пуска после запоминания данных, которые могут быть потеряны из-за сбоя в питании. Рабочие характеристики этой специфической команды перехода не допускают ее выполнения при отклонении напряжения питания ниже определенного уровня. В зависимости от масштабов и степени повреждения источника питания возможны две последовательности действий:

1. Если напряжение питания нормализуется раньше достижения опасного уровня, то выполняется команда перехода и осуществляется повторный запуск.

2. Если напряжение питания снижается до опасного уровня, то источник питания генерирует сигнал MASTER CLEAR (полное гашение), реализуемый системой автоматически. Возобновление нормальной подачи электроэнергии после очистки ЭВМ приведет к автоматическому запуску из последнего адреса памяти NDRO.

СВОДКА ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОЦЕССОРА

Наибольший интерес представляют следующие характеристики процессора:

Постоянная память емкостью 512 слов с сохранением информации при выключении электропитания для программ начальной загрузки, программы анализа сбоев и восстановления, а также программы диагностики аппаратных средств.

82 регистра с произвольной выборкой на интегральных схемах для управляющей памяти ЦП, в том числе два набора из семи индексных и восьми базовых регистров и два набора из восьми адресуемых арифметических накапливающих регистров.

Возможность адресации до $256K$ слов ($K=1024_{10}$) основной памяти при помощи прямой, косвенной, индексной адресации или адресации «база-плюс-смещение».

Выполнение команд с перекрытием, когда используют два или больше модулей памяти.

ОСНОВНАЯ ПАМЯТЬ

Основная память системы AN/UYK-7 включает модули (банки) памяти на магнитных сердечниках с разрушением информации при считывании и с произвольной выборкой; время полного цикла памяти равно 1,5 мкс. Каждый модуль памяти содержит четыре блока емкостью 4K, преобразователь электропитания, адресный транслятор, а также временную и управляющую схемы. Все блоки памяти взаимозаменяемы и обладают емкостью 16K 32-разрядных слов.

Взаимодействие между ЦП, памятью и контроллером ввода-вывода обеспечивается шинной системой связи, содержащей три канала

соответственно для передачи команд, передачи операндов и для осуществления ввода-вывода. Каждый ЦП сопрягается с блоками памяти при помощи шины передачи команд и шины передачи операндов, а также с контроллером ввода-вывода через шину передачи операндов данных. Каждый контроллер ввода-вывода связан с блоками памяти через шину ввода-вывода. Для установления связи с другими модулями в каждом модуле емкостью 16К слов предусмотрено восемь трактов сопряжения (одна шина и один вход на тракт), обеспечивающих доступ к памяти. Шина памяти служит в качестве двустороннего тракта передачи данных и тракта передачи адресов между инициатором запроса и памятью. В шину памяти включены линии управления связью, предназначенные для запрашивающего устройства и для памяти. На каждый канал приходится по 32 линии передачи данных и 20 линий управления связью; при этом только 9 из 20 линий управления используются для любого блока памяти. К каждому блоку памяти может адресоваться любая совокупность процессоров и контроллеров ввода-вывода при соблюдении одного ограничения: общее число комбинаций, составленных из двух доступов на каждый процессор и одного доступа на контроллер ввода-вывода, не должно превышать восьми доступов к любому одному блоку памяти. В любой момент могут поступать несколько запросов к модулю памяти; они выполняются в соответствии с приоритетом, определяемым схемами приоритета модулей памяти. Порядок приоритета устанавливается при монтаже сопрягающих шин системы.

Наличие в модулях памяти логических схем готовности-возобновления операций позволяет осуществлять асинхронное взаимодействие с процессорами и контроллерами ввода-вывода. Каждая шина, соединяющая вход памяти с контроллером ввода-вывода или ЦП, передает запрос на обслуживание и соответствующий операнд или адрес команды и запрашиваемые операнды и команды. Для специального обращения к памяти ЦП или контроллер ввода-вывода выдают на сопрягающую шину сигнал запроса и адрес ячейки памяти. Модуль памяти, указанный пользователем, находясь в состоянии готовности, отвечает на запрос (чтение, запись или чтение-запись), и выполняет соответствующие функции. Операнды и команды передаются по той же шине, что и соответствующие адреса и запросы. Последовательность и направление потока данных по всем шинам, подсоединенным к модулю памяти, регулируются при помощи временной схемы, логической схемы готовности-возобновления, а также схемы приоритета. Если модуль не находится в состоянии готовности по какому-либо входу, то любой запрос на этот вход хранится в схеме приоритета, ожидая своей очереди. Такой асинхронный принцип работы позволяет осуществлять в любой момент столько обращений к операциям считывания, записи или считывания-записи информации, сколько модулей памяти имеется в системе.

Все модули памяти со смежной адресацией содержат адреса 00000—77777₈. Для выбора слова в пределах адресуемого модуля адресные трансляторы преобразуют 14-разрядную часть 18-разрядного адреса, заданного инициатором запроса. Каждый модуль отвечает на запросы в тех случаях, когда значение сигнала в канале выборки соответствует номеру, присвоенному этому модулю в системе. Систему памяти можно по желанию модифицировать для применения адресации с чередованием памяти в парных модулях (по 32К слов). При такой конфигурации адреса с четными номерами в диапазоне 00000—77776₈ выбираются и преобразуются четным модулем пары, а адреса с нечетными номерами в диапазоне 00001—77777₈ выбираются и преобразуются нечетным модулем пары.

КОНТРОЛЛЕР ВВОДА-ВЫВОДА

Контроллер ввода-вывода системы AN/UYP-7 включает управляющие и временные схемы, необходимые для передачи данных ввода-вывода, внешних команд и внешних прерываний между доступными модулями памяти и периферийными устройствами по 4, 8, 12 или 16 полным дуплексным каналам, а также для обновления содержимого регистра часов реального времени и регистра часов монитора. Каждый контроллер ввода-вывода может выполнять передачу данных с прямым доступом к (от) максимальному числу 16 модулей памяти под управлением одного, двух и трех ЦП. В каждом контроллере ввода-вывода имеется управляющая память на интегральных схемах для хранения слов управления буфером, указателей адресов команд, полей управления функциями контроллера, а также показаний часов. Управляющая память содержит 64 слова (по 56 разрядов каждое). Одно слово предназначено для каждого канала и для каждой из следующих операций: ввод данных, вывод данных, выполнение внешней функции и внешнего прерывания.

Функциями контроллера ввода-вывода управляет цепочка команд (программы ввода-вывода), инициируемая одним или несколькими ЦП. Программы ввода-вывода определяют буферную область, номера каналов, используемые мониторы и типы передачи, а также функции, связанные с заданием форматов информации (слов или байтов). Цепочки формируются с учетом соответствующих требований из 15 команд, входящих в набор команд контроллера ввода-вывода. Последовательность составления цепочки команд, а также передачи информации (данные, коды прерывания, команды на периферийные устройства) регулируется при помощи заданных значений разрядов определенных полей управляющих слов в управляющей памяти контроллера ввода-вывода. Эти слова выбираются и загружаются в соответствии с командами, выполняемыми контроллером ввода-вывода. Слова управления буфером определяют порядок выполнения всех операций передачи информации по активному каналу и по за-

вершении этого процесса указывают адрес следующей команды в цепочке.

Каждый канал, определяемый сменной монтажной схемой, может осуществлять связь в одном из следующих режимов: обычный буфер, внешняя спецификация индекса (ESI), внешняя спецификация адреса (ESA) и межмашинный интерфейс. В обычном режиме адаптер канала ввода-вывода передает или принимает данные, содержащие 32-разрядные слова, и выполняет операции над данными, заданные программой. Передача, прием и обработка слов данных осуществляются с использованием обычных управляющих сигналов. В режиме ESI используются 16 младших разрядов данных в каналах: передачи входных данных для определения адреса основной памяти, содержащего слово управления буфером. Разряды 0—17 слова управления буфером содержат начальный адрес данных, предназначенных для передачи, а разряды 18—28 содержат конечный адрес буфера. В режиме ESA периферийные устройства обеспечиваются средствами спецификации ячеек абсолютной памяти для хранения или поиска данных на пословной основе. Для выдачи ответа на запрос с внешней спецификацией адреса требуется активный канал. Межмашинный режим работы адаптера канала ввода-вывода заключается в обеспечении связи между адаптерами ввода-вывода. Передающая ЭВМ в этом случае не ожидает сигнала подтверждения от принимающей ЭВМ; вместо этого передающая ЭВМ посылает одновременно сигнал готовности и сами данные на выходные каналы. Когда передающая ЭВМ получает сигнал возобновления, с выходных каналов снимаются сигналы готовности и передаваемые данные, и в эти каналы вводятся дополнительные данные или же межмашинные операции завершаются. Если принимающая машина не получит данные и не пришлет обратный сигнал в пределах заданного периода времени (регулируемого путем выбора межмашинного периода простоя), то возникает прерывание класса I. Модуль выбора режима, содержащийся в системе контроллера ввода-вывода, включает логическую схему для определения межмашинного периода простоя, тип интерфейса для каждого канала (обычный буфер, ESI, ESA или межмашинный интерфейс) и регулирует приоритет каждого канала. Выбор периода простоя и типа интерфейса для каждого канала следует производить до изготовления модуля выбора режима.

Запрашиваемые операции выполняются контроллером ввода-вывода в соответствии с приоритетами для двух групп запросов: буферизуемые запросы (получаемые по каналам ввода-вывода) и небуферизуемые запросы (инициируемые процессором или средствами управления контроллера ввода-вывода, такими, как часы реального времени и цепочки команд, связанных с работой каналов). Если запросы обеих групп приходят одновременно, то контроллер ввода-вывода чередует выполнение между двумя группами. После выполне-

ния запроса осуществляется упорядочение последовательности приоритетов запросов.

16-разрядный регистр часов монитора в контроллере ввода-вывода можно привести в действие путем загрузки в него положительной величины. При работе регистра эта величина уменьшается в соответствии с показаниями часов (внутренних или внешних). Когда она достигает 0, происходит прерывание класса III по часам монитора контроллера для последующей организации прерывания ЦП, связанного с данным контроллером. Нуль, загруженный в регистр часов монитора, дает прерывание класса III в системе ЦП — контроллер ввода-вывода. Любая отрицательная величина в этом регистре приводит к блокировке работы часов монитора. 32-разрядный регистр часов реального времени осуществляет счет со скоростью 1024 отсчетов/с при использовании внутреннего генератора импульсов и со скоростью до 100 кГц, когда используется внешний генератор импульсов.

Модуль адаптера интерфейса, связанный с каждым контроллером ввода-вывода, включает схему для 4, 8, 12 или 16 каналов ввода-вывода. Он содержит выходные регистры, устройства управления линиями связи, входные усилители, а также средства синхронизации подтверждений. В соответствии с номерами и характеристиками каналы объединены в группы по четыре канала ввода и четыре канала вывода. Входные линии передачи данных и линии управления оканчиваются входными усилителями, а выходные линии управления приводятся в действие одним из устройств управления линиями связи. Четыре канала вывода каждой группы совместно используют 32 устройства управления линиями, но только один из этих каналов принимает соответствующий сигнал управления в ходе передачи. В качестве одного из вариантов передачи для любой группы из четырех каналов можно выбрать последовательный интерфейс для обычного и межмашинного режимов передачи данных. Сигналы управления линиями связи, используемые в параллельных каналах, заменяются управляющими кодовыми комбинациями, которые предшествуют передаче слов по последовательным каналам. Все передаваемые биты управления генерируются аппаратными средствами последовательного интерфейса.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Операционная система, разработанная для вычислительной системы AN/UUK-7, называется COMMON CONTROL PROGRAM и состоит из четырех функциональных элементов для работы в реальном времени и одного элемента для других режимов. Эти пять элементов программного обеспечения — общий управляющий модуль, общий периферийный модуль, модуль отладки, общий системный

модуль и пакет программ-утилитов — управляют всей мультипроцессорной обработкой и мультипрограммированием.

К основным функциям общего управляющего модуля относятся инициация задач, планирование, инициация ввода-вывода, обработка сообщений и обработка прерываний. Все модули операционной системы, за исключением общего управляющего модуля и программы восстановления, могут функционировать, когда ЭВМ находится в режиме задачи. Общий управляющий модуль обеспечивает выполнение операций в реальном времени путем планирования ресурсов центрального процессора (центральных процессоров). Помимо этого, общий управляющий модуль обеспечивает возможность установления взаимосвязи между пользователями, а также выполнение функций, которые были выделены в качестве привилегированных для операционной системы при разработке ЦП системы AN/UYK-7. Общий управляющий модуль снабжает прикладные программы различными средствами для формирования реакций на внешние события и выполнения некоторых функций, связанных с временными ограничениями и изменениями во времени, а также функций, необходимых для работы не в реальном времени. Этим универсальным планированием работ в системе управляет прикладная программа.

Общий управляющий модуль осуществляет обработку запросов на планирование ресурсов, используя заданный набор правил, называемый алгоритмом планирования. Схемы данных и индексная структура общего управляющего модуля обеспечивают возможность планирования до 256 отдельных программных модулей. Программный модуль может иметь один из следующих типов входов: приоритетный, периодический, фоновый, вход типа сообщения, вход прерывания по вводу-выводу, вход по требованию. Из 256 модулей 63 модуля могут одновременно обрабатываться с приоритетного входа. Разработчик системы должен указать допустимую величину ожидания обработки для каждого типа входа. Эта величина простая рассматривается как характеристика помехоустойчивости системы, позволяющая общему управляющему модулю восстановить управление процессором в аварийных условиях. Общий управляющий модуль не разрешает процессору войти в какой-либо модуль через приоритетный, периодический, фоновый входы или вход типа сообщения в том случае, если другой процессор уже работает с данным модулем, войдя в него через один из этих входов. Однако входы прерывания по вводу-выводу и входы по требованию не имеют защиты от повторного входа.

Каждый модуль снабжается именем, номером приоритета и определенной величиной периодического временного интервала. Это делается под руководством разработчика системы. Самый высокий уровень планирования предусмотрен для приоритетных входов. Номер приоритета модуля указывает, какой модуль должен получить предпочтение в том случае, если планируется выполнение одновре-

менно нескольких модулей с приоритетным входом. Каждый модуль должен обеспечить вход сообщений от процессора, чтобы иметь возможность принять по крайней мере управляющие сообщения. Периодический временной интервал, установленный для каждого модуля, указывает желаемое время между последовательными использованиями периодического входа. Для всех процессоров с фоновым входом предусматривается автоматическое планирование разделения времени с величиной кванта времени, выбранного разработчиком системы. Вход по требованию предназначен главным образом для выполнения тех функций, для которых необходимо точное согласование по времени, так как выполнение их с помощью входов прерывания может привести к значительному расходу ресурсов. При запросе на обслуживание по требованию необходимо задать величину показаний часов реального времени, и в дальнейшем этот вход инициируется в результате прерывания по показаниям часов.

Все модули должны регистрировать свои требования на ввод-вывод, передаваемые общему управляющему модулю, включая номер (номера) контроллера ввода-вывода, номер (номера) канала и режимы работы (только ввод, только вывод или ввод и вывод). Кроме того, модуль, осуществляющий операции вывода, должен указать действия для обработки прерываний от монитора входных данных и от монитора внешних прерываний. В связи с работой системы в реальном времени вход прерывания по вводу-выводу должен быть ограничен инициацией функций, которые не могут быть задержаны до того, пока модуль не получит приоритетный вход. Вход прерывания по вводу-выводу должен позволять двум или большему числу ЦП одновременно управлять операциями, связанными с работой соответствующего процессора ввода-вывода.

Общий управляющий модуль обеспечивает программные средства, необходимые для установления связи между элементами прикладной программы с помощью сообщений. Кроме того, этот модуль обеспечивает защиту памяти, централизованную обработку прерываний, а также выполнение привилегированных команд, таких, как инициация функций ввода-вывода. Средства защиты программ реализуются при помощи таких свойств ЭВМ, как блокировка памяти и использование привилегированных режимов работы.

Общий периферийный модуль обеспечивает установление универсального интерфейса между прикладной программой (программами), оператором и стандартными периферийными устройствами: устройством чтения-записи магнитной ленты, устройством чтения-записи бумажной перфоленты, автоматической пишущей машинкой, символьным дисплеем. Общий периферийный модуль осуществляет запись данных и их поиск, а также позволяет оператору управлять и контролировать работу программного обеспечения. Эти функции может запрашивать прикладная программа при помощи межмодульного сообщения или оператор при помощи клавиатуры пульта управ-

ления или устройства чтения бумажной перфоленты. Кроме того, этот модуль выполняет функции по преобразованию и изменению формата данных, управлению устройствами записи на магнитную ленту, управлению дисплеем, а также осуществляет общесистемный контроль ошибок и восстановление.

Общий системный модуль обеспечивает централизованную обработку программных ошибок, сохранение их в памяти системы и назначает общей программе функциональные средства для обработки данных в общем системном модуле. Общая программа включает средства для выполнения функций обновления общих системных данных; она использует межмодульные сообщения для ввода новых данных или корректировки ранее накопленных общих системных данных, а также для передачи управления после завершения корректировки.

Модуль отладки обеспечивает средства работы в реальном времени, используемые при отладке программ, их тестировании и объединении, а также для общего обслуживания выполняемых программ. Модуль отладки обеспечивает доступ в режиме on-line к памяти или арифметическим регистрам для событий, происходящих в данный момент времени, и других событий, обрабатываемых в реальном масштабе времени. Помимо этого, модуль отладки может вызывать информацию из непосредственно подключенных к ЭВМ устройств или устройств, задающих рабочий цикл системы.

Пакет программ-утилитов обеспечивает выполнение функций, задаваемых оператором и необходимых для управления системой AN/UYK-7 и стандартными периферийными устройствами. Этот пакет обеспечивает возможность загрузки и распечатки памяти ЭВМ и выполнения программ не в реальном времени при их отладке вручную. Интерфейс между пакетом утилитов и общим управляющим модулем отсутствует.

ПРИЛОЖЕНИЕ XVII

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА SIGMA 9 ФИРМЫ XEROX DATA SYSTEMS

ВВЕДЕНИЕ В ОРГАНИЗАЦИЮ СИСТЕМЫ

Вычислительная система SIGMA 9 представляет собой быстродействующую цифровую вычислительную систему общего назначения. Основная система включает ЦП, подсистему основной памяти,

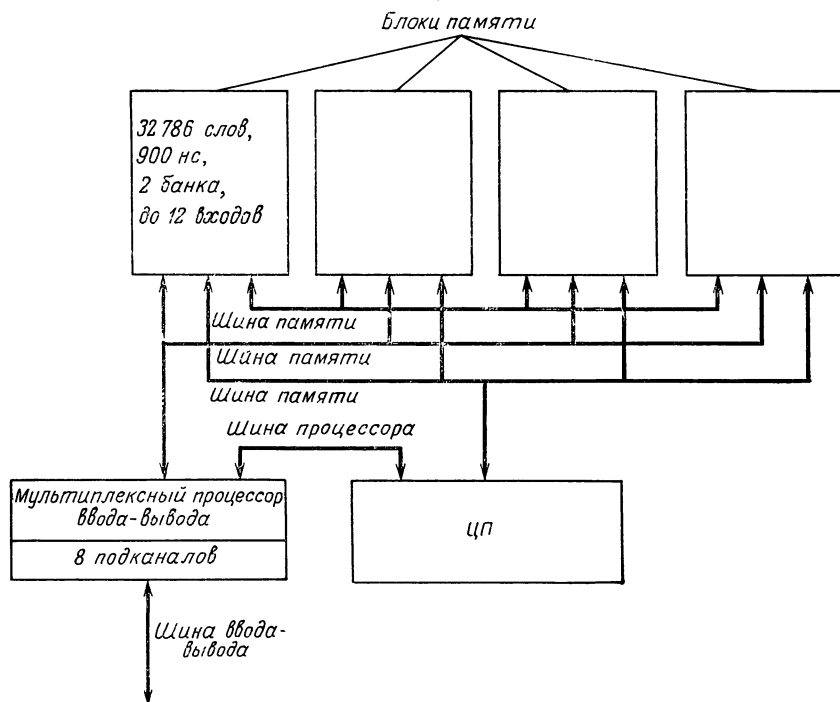


Рис. П.XVII.1. Базовая конфигурация системы SIGMA 9.

а также автономную подсистему ввода-вывода, причем каждый основной элемент системы функционирует асинхронно по отношению к остальным элементам (рис. П.XVII.1).

Система SIGMA 9 предназначена для работы в качестве мульти-процессорной системы с общей памятью. Вся система в целом может содержать до четырех ЦП и до 11 процессоров ввода-вывода (общее число процессоров обоих типов определяется максимальным числом входов в память, равным 12). Все процессоры системы SIGMA 9 имеют единообразную адресацию памяти.

Система распределения памяти обеспечивает динамическое размещение программы в различных несмежных сегментах памяти. При использовании этой системы любая программа может быть разбита на страницы, содержащие по 512 слов, и распределена по всей памяти на имеющихся свободных страницах.

Возможно расширение основной памяти системы от минимальной емкости 131К слов (по 32 информационных разряда и 1 разряду контроля четности) до максимальной емкости 524 288 слов; расширение осуществляется блоками по 32 786 слов. Время цикла основной памяти системы SIGMA 9 равно 900 нс.

Операции ввода-вывода в системе SIGMA 9 выполняются главным образом под управлением одного или нескольких процессоров ввода-вывода. Процессоры ввода-вывода требуют от ЦП только инициации соответствующих программ, после чего каждый из них действует автономно, без последующего вмешательства ЦП.

В стандартной конфигурации предусмотрено наличие двух часов реального времени и возможно по желанию добавление еще двух. Часы вырабатывают информацию об истекшем времени и астрономическом времени, причем каждые часы могут иметь собственный период отсчета и приоритет.

Система приоритетного прерывания в SIGMA 9 имеет иерархическую структуру. Предусмотрено до 224 уровней прерывания, причем каждый уровень имеет свой однозначный адрес в основной памяти и свой приоритет.

КОМПОНЕНТЫ СИСТЕМЫ SIGMA 9

ПРОЦЕССОР

ЦП системы SIGMA состоит из двух основных частей — быстродействующей памяти и арифметико-управляющего блока. ЦП выполняет арифметические и логические операции; формирует и контролирует выполнение последовательностей команд; осуществляет управление обменом информацией между основной памятью и другими частями системы. ЦП включает регистры общего назначения, которые можно использовать для различных целей: в качестве накапливающих, индексных или временных рабочих регистров. Общее число этих 32-разрядных регистров, объединенных в блоки по 16 штук, может достигать 64.

ЦП располагает тремя быстродействующими запоминающими устройствами на интегральных схемах для хранения схемы распределения памяти, кодов защиты доступа к памяти, соответствующих схеме распределения, а также кодов защиты при записи в память.

Система распределения памяти обеспечивает динамическое размещение программы в несмежных сегментах памяти. При работе этой системы любая программа может быть разбита на страницы, содержащие по 512 слов, и распределена по всей памяти на имеющихся в наличии свободных страницах. Таким образом, система преобразует виртуальные адреса, выбранные в соответствии с требованиями отдельной программы, в реальные адреса, назначенные в соответствии с требованиями системы управления памятью.

ЦП может функционировать в режиме ведущего или ведомого либо в режиме ведущего с защитой. Режим работы регулируется путем установки значений трех разрядов в управляющем слове. В режиме ведущего ЦП может выполнять все необходимые управляющие функции, а также осуществлять модификацию любой части системы. Единственное ограничение, налагаемое на операции ЦП в данном режиме, связано с блокировкой операций записи в память. В режиме ведомого ЦП выполняет обычные прикладные программы. При этом в программах используются коды защиты доступа к памяти и запрещаются все привилегированные операции, если действует система распределения памяти. К привилегированным относятся операции, связанные с вводом-выводом и внесением изменений в основное состояние управления ЭВМ. Режим работы ЦП, определяемый как режим ведущего с защитой, представляет собой модификацию обычного режима работы ведущего процессора с применением дополнительной защиты управляющих программ, выполняемых в этом режиме. Режим ведущего с защитой можно применять только в случае, когда ЦП работает в режиме ведущего и при этом функционирует система распределения памяти.

В мультипроцессорной конфигурации каждый ЦП имеет возможность получить исключительное право на управление системными ресурсами (областью памяти, периферийными устройствами и программными средствами) при помощи специальной команды, которая блокирует использование ресурсов до их освобождения данным процессором. Вследствие того что адресация всех процессоров к памяти идентична, необходимо, чтобы каждый ЦП имел свою отдельную область памяти. Эта собственная память используется для информации, которая является уникальной для каждого ЦП; к ней относятся, например, условия прерывания, схемы связи с устройствами ввода-вывода и т. п. Собственная память каждого ЦП содержит 1024 слова. Неисправные блоки можно изолировать от системы путем их селективной блокировки вручную через шины системы.

В мультипроцессорной конфигурации предусмотрена функция управления, обеспечивающая:

- 1) управление внешней шиной ввода-вывода для целей технического обслуживания системы и блоков специального назначения;
- 2) централизованное управление расчленением системы;
- 3) управление прерываниями межпроцессорного обмена, позволяющего одному процессору непосредственно сигнализировать другому о необходимости выполнения определенных операций.

Любой ЦП имеет возможность назначить операции ввода-вывода на любой процессор ввода-вывода; однако последовательностью операций окончания процесса ввода-вывода управляет только один из четырех ЦП. Это означает, что задача окончания операций ввода-вывода закрепляется за одним ЦП во избежание проблем, связанных с конфликтными ситуациями.

Набор команд системы SIGMA 9 содержит более 100 основных команд. К этим командам относятся: команды загрузки и записи в память с полями длиной в байт (8 разрядов), полуслово, слово и двойное слово; команды анализа — интерпретации; арифметические команды операций над числами с фиксированной запятой, в которых используются поля длиной в полуслово, слово или двойное слово; команды сравнения; команды преобразования; команды сдвига; логические команды; арифметические команды операций над числами с плавающей запятой; десятичные команды; команда обработки байтовой строки; команды вытеснения; команды ветвления; команды вызова; команды управления; команды ввода-вывода. Время выполнения команды колеблется в диапазоне от 730 нс (для команды загрузки) до 38 мкс (для десятичного умножения).

ОСНОВНАЯ ПАМЯТЬ

Минимальная оперативная память системы SIGMA 9 содержит 131 172 слова. Ее можно расширить до 524 288 слов, добавляя блоки по 32К слов. Память подразделяется на банки (по 16К слов), представляющие собой наименьшую часть памяти, к которой процессор может осуществлять автономный доступ. Два банка по 16К слов каждый, совместно использующие одну общую логическую схему входа, называются блоком памяти. Такая структура помогает организовать двукратное чередование адресов между двумя банками внутри одного блока и четырехкратное чередование адресов между четырьмя банками в двух блоках памяти. При двукратном чередовании адресов четные адреса присваиваются одному банку, а нечетные — другому; при четырехкратном чередовании каждый четвертый адрес присваивается соответствующему банку в двух смежных блоках памяти.

Память содержит 33-разрядные слова (четыре 8-разрядных байта и разряд контроля четности); возможна прямая адресация всей памяти как со стороны центральных процессоров, так и со стороны процессоров ввода-вывода. Обмен данными между памятью и про-

цессорами осуществляется асинхронно. В настоящее время память имеет время полного цикла 900 нс, обеспечивая возможность обработки в банке памяти (16К слов) более 35 млн. битов в 1 с. При асинхронных операциях в системе SIGMA 9 отдельные функциональные элементы работают в непрерывном режиме или в режиме «готовность — продолжение работы». Специальная конструкция памяти, функционирующей в асинхронном режиме, позволяет инициировать рабочий цикл памяти практически в любой момент времени и обеспечивает для каждого банка максимальную частоту работы свыше 1 110 000 Гц. Каждый блок основной памяти в системе имеет как минимум два входа (возможно увеличение их числа до 12), причем каждый вход подсоединяется к отдельной шине памяти. Поскольку модуль памяти обслуживает только одно запрашивающее устройство (ЦП или процессор ввода-вывода) во время одного цикла чтение-запись, один его вход должен обладать более высоким приоритетом по сравнению с другим на случай одновременного появления нескольких запросов; в противном случае запросы на доступ обрабатываются по принципу «первый пришел — первый обслужен». Все данные и адреса, передаваемые в любом направлении по шинам между блоками памяти и процессорами, проверяются на четность.

Существует два способа управления использованием основной памяти программой: при помощи системы распределения памяти и при помощи запираания памяти. При использовании первого способа ЦП выполняет тестирование, чтобы определить, имеются ли какие-либо ограничения на использование виртуального адреса в программах, выполняемых ЦП в режимах ведомого или ведущего с защитой. Существуют четыре типа защиты доступа:

Программа, выполняемая ЦП в режиме ведомого или ведущего с защитой, может осуществлять запись, считывание или доступ к командам на данной странице виртуальных адресов.

Программа, выполняемая ЦП в режиме ведомого или ведущего с защитой, не может осуществлять запись, но может осуществлять считывание или доступ к командам на данной странице виртуальных адресов.

Программа, выполняемая ЦП в режиме ведомого или ведущего с защитой, не может осуществлять запись или доступ к командам, но может осуществлять считывание данной страницы виртуальных адресов.

Программе, выполняемой ЦП в режиме ведомого или ведущего с защитой, запрещается любой доступ к данной странице виртуальных адресов.

Второй способ — запираание памяти — обеспечивается при помощи метода «замка и ключа». Для каждой страницы (512 слов) в области первых 128К слов реальных адресов памяти предусмотрен 2-разрядный замок защиты при записи. Замок защиты при записи можно

изменить лишь в результате выполнения привилегированной команды. Замки и ключи защиты памяти осуществляют управление доступом для операции записи в соответствии со следующими правилами:

Если значение замка памяти равно 00, то соответствующая страница памяти «разблокирована», а операция записи на эту страницу разрешается независимо от значения ключа памяти.

Если значение ключа памяти равно 00, то ключ представляет собой «отмычку», которой можно открыть любой замок памяти; при этом разрешается запись на любую страницу памяти независимо от значения замка.

Любое значение замка памяти, кроме 00, позволяет осуществлять операцию записи на эту страницу лишь в том случае, если значение ключа совпадает со значением замка памяти.

Таким образом, программа может осуществить запись на заданную страницу памяти, если значение замка памяти равно 00, или если значение ключа равно 00, или если значение ключа совпадает со значением замка памяти.

МОДУЛЬ УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

Операции ввода-вывода выполняются главным образом под управлением одного или нескольких процессоров ввода-вывода. Процессор ввода-вывода требует от ЦП лишь инициации и затем действует автономно.

Система SIGMA 9 может включать до 11 процессоров ввода-вывода. Это могут быть: мультиплексные процессоры ввода-вывода, предназначенные для использования с периферийными устройствами, имеющими стандартную скорость, в том числе среднескоростными запоминающими устройствами с прямым доступом RAD; процессоры ввода-вывода, работающие совместно с высокоскоростными запоминающими устройствами RAD; устройства специального назначения. Интерфейсы между ЦП, модулями основной памяти и устройствами ввода-вывода унифицированы настолько, что при добавлении в систему новых типов процессоров ввода-вывода не возникает необходимости в модификации интерфейсов.

Мультиплексный процессор ввода-вывода позволяет осуществлять управление передачей данных с разными скоростями (от низкой до высокой) по двум каналам, с использованием метода мультиплексирования во времени. Этот процессор может обслуживать до 32 подканалов (в базовой конфигурации используются восемь подканалов), он располагает собственной памятью, регистрами и арифметическими устройствами. В мультиплексном процессоре ввода-вывода по желанию можно применять режим «перемещения», который обеспечивает возможность пересылки больших блоков данных из одной области

памяти в другую без вмешательства ЦП, после того как он инициировал эту пересылку. Данные передаются со скоростью 500 000 слов/с.

Структура системы ввода-вывода предусматривает формирование цепочки команд (обеспечивающей выполнение операций многократной записи) и цепочки данных (обеспечивающей выполнение операций считывания распределенных данных и записи групповых данных) без вмешательства ЦП.

ВСПОМОГАТЕЛЬНАЯ, ИЛИ МАССОВАЯ ПАМЯТЬ

Массовая память системы SIGMA 9 включает файлы на устройствах с быстрым доступом, обеспечивающих скорость обмена от 750К до 25 млн. байтов/с при среднем времени доступа 17 мс, а также до 4 файлов на сменных дисках, каждый из которых обеспечивает скорость передачи данных до 312 500 байтов/с, емкость памяти от 24 млн. до 49 млн. байтов при среднем времени доступа 87,5 мс.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

УПРАВЛЕНИЕ ЗАДАНИЯМИ

Программное обеспечение системы SIGMA 9 включает операционную систему XOS фирмы Хегох и универсальную систему разделения времени UTS. Обе системы обеспечивают выполнение процессов в реальном времени наряду с другими видами обработки данных в режиме совпадения. Операционная система XOS предусматривает три основных класса заданий: параллельные задания, порождающие задания и последовательные задания. Параллельные задания представляют собой каталогизованные задания, каждое из которых содержит лишь один шаг задания; они инициируются оператором с пульта. Планировщик заданий операционной системы активирует задания в очереди параллельных заданий с учетом их приоритетов и требований на ресурсы. Задания, относящиеся к классу порождающих, обычно содержат несколько шагов задания. Для обеспечения обработки этих шагов с перекрытием класс порождающих заданий подразделяется на пять подклассов, каждый из которых обладает некоторым относительным приоритетом. В любой заданный момент пользователь может хранить в памяти системы одновременно пять порождающих заданий. Класс последовательных заданий содержит лишь один подкласс и одну очередь ожидающих заданий. Задания последовательного типа обычно представляют собой долгосрочные задания, выполняемые в режиме пакетной обработки, причем в памяти в каждый момент времени содержится лишь одно последовательное задание.

Для обеспечения мультипрограммирования операционная система XOS может управлять выполнением нескольких заданий, находящихся в памяти:

- любого числа параллельных заданий;
- до пяти порождающих заданий;
- одного последовательного задания.

Имеются общие элементы (симбионты), предназначенные для использования заданиями любого класса.

Планировщик заданий выбирает задания, которые должны активироваться с учетом класса задания, наличия ресурсов, требуемых для задания, а также относительного приоритета задания (для последовательных заданий). Наивысшим приоритетом обладает класс параллельных заданий, поэтому планировщик сначала проверяет очередь параллельных заданий, затем очередь порождающих заданий и, наконец, очередь последовательных заданий.

До начала выполнения любого задания планировщик проверяет наличие ресурсов, чтобы определить возможность его активации; если такая возможность отсутствует, то задание устанавливается в очередь ожидания. В ходе выполнения задания программа пользователя может осуществлять запрос — освобождение страниц памяти.

Оператор может послать с пульта запрос о состоянии задания, об инициации параллельного задания, а также о блокировке или восстановлении периферийного устройства, выпешшего из строя.

Система включает средства контроля четности для обнаружения ошибок в памяти и восстановления и средства автоматического восстановления и перезапуска системы.

УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Супервизор ввода-вывода обслуживает все запросы на операции ввода-вывода. Он иницирует ввод-вывод, устраняет ошибки, а также информирует систему о завершении конкретной процедуры ввода-вывода. Супервизор ввода-вывода оптимизирует поток данных от периферийных устройств, используя различные методы в зависимости от типа периферийного устройства. Например, в зависимости от положения диска с фиксированной головкой запросы на ввод-вывод выстраиваются в цепочку с учетом начального адреса сектора и количества передаваемых данных; при оптимизации движения держателей головок в дисковом устройстве с подвижными головками для каждого диска устанавливается две очереди ожидания, причем запросы в каждой очереди размещаются в соответствии с адресом цилиндра и порядком активации очередей.

Универсальная система разделения времени UTS обслуживает методом совпадения до 128 пользователей в режиме on-line, совмещая операции локальной и удаленной пакетной обработки и

работу в реальном времени. Система позволяет пользователю осуществлять инициацию и доступ к обслуживанию в режиме on-line при помощи различных систем связи, ориентированных на обработку символов.

Система обеспечивает создание, опрос, корректировку и стирание файлов различных форматов, задаваемых пользователем, на различных вспомогательных запоминающих устройствах. Возможны форматы файлов с последовательным, индексно-последовательным и прямым доступом, а также расчлененные файлы. В качестве запоминающих устройств используются устройства на магнитной ленте, устройства RAD, а также съемные диски. Для магнитной ленты и магнитного диска используются два вида стандартных меток. Пользователь специфицирует устройство, файл и операцию, и система выполняет необходимые действия в режиме совпадения, используя при передаче данных метод буферизации.

ДИАГНОСТИЧЕСКАЯ ОБРАБОТКА ОШИБОК

Для изоляции ошибок и сбоев, возникающих в аппаратуре и в программном обеспечении, система предоставляет в распоряжение пользователя такие средства, как контрольный таймер, определяющий правильность ответной реакции устройства в обусловленный промежуток времени; прерывание от системы контроля электропитания; обнаружение и исправление ошибок памяти с использованием контроля четности; журнал регистрации ошибок.

Система обеспечивает автоматическое восстановление или перезапуск системы, «фотографирование» поврежденной системы для последующих диагностических исследований и диагностические процедуры с периферийного оборудования. Предусмотрено также создание контрольных точек для перемещения и перезапуска программ.

ОГЛАВЛЕНИЕ

Предисловие к русскому изданию	5
Предисловие	8
Глава 1. Причины создания мультипроцессорных систем и систем параллельной обработки информации	13
Улучшение характеристик систем	13
Тенденции развития характеристик систем	14
Совпадение операций	15
Повышение надежности и готовности	16
Одномашинные вычислительные системы	18
Основная пятиблочная ЭВМ — машина фон Неймана	20
Прямой доступ к памяти	21
Канал ввода-вывода	22
Процессорный блок	23
Многомашинные вычислительные системы	23
Сателлитные ЭВМ	25
Несвязанные вычислительные системы с автономной периферийной ЭВМ	25
Характеристики оборудования	27
Последовательность операций	27
Результат	27
Связанные системы	28
Основы построения мультипроцессорных систем	32
Определение Мультипроцессорной системы	33
Развитие мультипроцессорных систем и их назначение	34
Историческая эволюция концепции мультипроцессора	35
Глава 2. Аппаратура	40
Основные требования	40
Типы организации систем	41
Системы с общей или разделенной во времени шиной	42
Системы с перекрестной коммутацией	47
Многошинные — многовходовые системы	51
Асимметричные и неоднородные системы	54
Системы с магистральной обработкой	56
Примеры систем с магистральной обработкой	59
Параллельные системы. Матричные и векторные процессоры	60
Системные организации, ориентированные на устойчивость к сбоям и отказам	64
Пути соединения функциональных блоков	69
Виртуальные процессоры	70
Множественные арифметические устройства	73

Основная память для мультипроцессорных систем	74
Доступ к памяти с перекрытием	74
Конфликты при обращении к памяти	76
Назначение физических и логических адресов памяти	79
Организация ввода-вывода и интерфейсы	81
Надежность и готовность аппаратного обеспечения систем	89
Устойчивость против отказов и терпимость к отказам	89
Реконфигурация системы	91
Аппаратное обеспечение мультипроцессорных систем	93
Рассмотрение организации мультипроцессорной системы с функциональной точки зрения	95
Глава 3. Операционные системы и другое программное обеспечение для мультипроцессорных систем	99
Введение	99
Организация мультипроцессорных операционных систем	100
Система типа «ведущий-ведомый»	101
Система с раздельным выполнением заданий в каждом процессоре	103
Система с симметричной, или однородной обработкой во всех процессорах	104
Основные требования к функциональным характеристикам	106
Распределение и управление ресурсами	107
Распределение и управление памятью	107
Планирование и диспетчеризация	109
Взаимосвязь процессоров	111
Ненормальное окончание обработки заданий	112
Балансировка нагрузки процессоров	112
Защита системных таблиц и наборов данных	113
Балансировка загрузки системы ввода-вывода	113
Реконфигурация	113
Клинич	114
Специальные проблемы программного обеспечения мультипроцессорных систем	116
Разделение памяти и организация доступа	116
Доступ к системным таблицам и наборам данных и их защита	118
Исправление ошибок	121
Распознавание параллельных участков в задании и использование режима параллельной обработки	123
Разработка и контроль качества программного обеспечения систем	126
Заключение	128
Глава 4. Настоящее и будущее мультипроцессорных систем	130
Сравнительные характеристики современных мультипроцессорных систем и систем с параллельной обработкой	130
Степень достижения мультипроцессорной системой целей, поставленных при ее планировании	131
Цели и преимущества	131
Готовность	131
Гибкость	139
Эффективность	139
Сравнительная оценка Витта	142
Недостатки мультипроцессорных систем	144
Некоторые дополнительные соображения относительно работы в режиме совпадения и параллельной обработки	145
Другие типы классификации	145
Степень параллельности работы	148
Будущее мультипроцессорных систем	148

Литература	150
Словарь терминов	153
Приложение I. Система PEPE управления перспективных разработок по противоракетной обороне Армии США	161
Приложение II. Мультипроцессорная система D825 фирмы BURROUGHS CORPORATION	175
Приложение III. Система обработки информации B6700 фирмы BURROUGHS CORPORATION	194
Приложение IV. Системы CDC 6500, CYBER-70, модели 72-2X, 73-2X и 74-2X, фирмы CONTROL DATA CORPORATION	218
Приложение V. Системы DEC 1055 и 1077 фирмы DIGITAL EQUIPMENT CORPORATION	233
Приложение VI. Вычислительная система STARAN фирмы GOOD-YEAR AEROSPACE SYSTEMS	239
Приложение VII. Система MULCICS 6180 и системы серии 6000 фирмы HONEYWELL	249
Приложение VIII. Вычислительная система H4400 фирмы HUGHES AIRCRAFT COMPANY	260
Приложение IX. Мультипроцессорная система 360, модель 65, фирмы IBM	270
Приложение X. Мультипроцессорные системы 370, модели 158 и 168, фирмы IBM	284
Приложение XI. ЭВМ военного назначения модели 215 фирмы RCA	292
Приложение XII. Ортогональные ЭВМ OMEN-60 фирмы SANDERS ASSOCIATES	299
Приложение XIII. Научно-исследовательская вычислительная система ASC фирмы TEXAS INSTRUMENTS	310
Приложение XIV. Вычислительная система UNIVAC 1108 фирмы SPERRY RAND CORPORATION	329
Приложение XV. Вычислительная система UNIVAC 1110 фирмы SPERRY RAND CORPORATION	346
Приложение XVI. Вычислительная система UNIVAC AN/UYK-7 фирмы SPERRY RAND CORPORATION	357
Приложение XVII. Вычислительная система SIGMA 9 фирмы XEROX DATA SYSTEMS	372

П/р Ф. Г. Эяслоу
МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ
И ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

Редактор Л. И. Титомир
Художник В. А. Чернецов
Художественный редактор В. К. Биссенгалиев
Технический редактор Т. А. Максимова
Корректор М. А. Смирнов

Сдано в набор 23/I 1976 г. Подписано к печати 24/VI 1976 г.
Бумага тип. № 1 $60 \times 90 \frac{1}{16} = 12$ бум. л. 24 печ. л.
Уч.-изд. л. 25,15. Изд. № 20/8303. Цена 1 р. 97 к. Зак. 136.

ИЗДАТЕЛЬСТВО «МИР»

Москва, 1-й Рижский пер., 2

Московская типография № 11 Союзполиграфпрома при
Государственном комитете Совета Министров СССР по де-
лам издательств, полиграфии и книжной торговли
Москва, 113105, Нагатинская, 1