

Государственный комитет РФ по высшему образованию

Санкт—Петербургский государственный электротехнический
университет имени В.И. Ульянова (Ленина)

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

Методические указания
по выполнению курсового проекта

Санкт—Петербург

1995

Электронные вычислительные машины: Методические указания по выполнению курсового проекта/ Сост.: А.И. Водяхо, А.Ф. Казак, А.С. Маркин; ГЭТУ. — С.—Пб., 1995. — 32 с.

Приводятся варианты заданий и даются рекомендации по выполнению курсового проекта по дисциплине "Электронные вычислительные машины". Определяются содержание разрабатываемой проектной документации и требования к ее оформлению.

Предназначены для студентов специальности 220100 и бакалавров по направлению 552800.

Утверждено
редакционно—издательским советом университета
в качестве методических указаний

Введение

Курсовой проект предназначен для практического освоения разделов курса "Электронные вычислительные машины", посвященных изучению архитектуры и принципов структурной организации и функционирования ЭВМ.

Эта цель наилучшим образом достигается самостоятельной проработкой студентом архитектурного, схемного и алгоритмического описаний ЭВМ.

Курсовой проект выполняется в течение одного семестра. Учебный план дисциплины предусматривает аудиторную групповую проработку основных разделов задания, проработку отдельных вопросов выполнения курсового проекта в процессе индивидуальных консультаций с преподавателем, самостоятельную работу студента и защиту в условиях, близких к защите дипломного проекта.

Качественное выполнение курсового проекта требует от студентов знаний отдельных разделов смежных дисциплин ("Узлы и устройства ЭВМ", "Периферийные устройства ЭВМ", "Основы построения ЭВМ") и систематической работы над проектом.

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1. Предмет проектирования

Курсовой проект для бакалавров посвящается разработке архитектуры однокристального RISC процессора. В зависимости от варианта проектируется либо процессор общего назначения, предназначенный для использования в качестве центрального процессора (ЦП) рабочей станции, ориентированной на работу в многопользовательском режиме, либо процессор для встроенных применений, который может использоваться, например в составе систем управления в реальном масштабе времени.

1.2. Общие требования к разрабатываемому процессору

Система команд должна удовлетворять следующим требованиям.

1. Операции обращения к памяти отделены от операций, связанных с обработкой данных.

2. Операции, связанные с преобразованием данных, выполняются по принципу регистр—регистр.

3. В общем случае аппаратно поддерживаются операции над данными, представленными в формате с фиксированной точкой (ФТ) и плавающей точкой (ПТ).

4. Система команд должна быть функционально полной.

Процессор должен иметь систему прерываний.

Должны быть предусмотрены средства отключения процессора от шины с переводом внешних выводов в третье состояние.

Для процессоров с традиционной (принстонской) архитектурой должны быть предусмотрены механизмы работы с виртуальной памятью, а также возможность работы в многозадачном режиме.

Процессоры с гарвардской архитектурой должны иметь встроенную кэш—память данных, емкость которой равна емкости внутренней памяти данных.

1.3. Исходные данные для курсового проектирования

Исходные данные для выполнения курсового проекта определяются как общими требованиями к разрабатываемому процессору, так и вариантом задания.

Варианты заданий приведены в табл. 1.1 — 1.4. Состав исходных данных определяется типом архитектуры. Для традиционной архитектуры варианты заданий в соответствии с табл. 1.1, 1.2 включают следующие исходные данные для проектирования:

- перечень аппаратно поддерживаемых типов данных;
- адресность операционных команд;
- способы адресации;

Традиционная архитектура

N ва- ри- ан- та	Формат данных				Адрес- ность	Способ адресации				Регистровая память		
	8	16	32	64		Н	О	П	К	Ко- ли- чес- тво	Тип	Раз- ряд- ность
1	+	+	+	-	1	+	+	+	-	8	У	16
2	-	+	+	+	2	-	+	+	+	8	У	32
3	+	-	+	+	3	+	+	+	-	8	ФО	16
4	+	+	-	+	1	+	+	+	-	8	ФО	32
5	+	+	+	-	2	+	+	-	+	16	У	32
6	-	+	+	+	3	+	+	-	+	16	У	32
7	+	-	+	+	1	+	+	+	-	16	ФО	16
8	+	+	-	+	2	+	+	+	-	16	ФО	32
9	+	+	+	-	3	-	+	+	+	32	У	16
10	-	+	+	+	1	-	+	+	+	32	У	32
11	+	-	+	+	2	+	+	+	-	32	ФО	16
12	+	+	-	+	3	+	+	+	-	32	ФО	32
13	+	+	+	-	1	-	+	+	+	64	У	16
14	-	+	+	+	2	-	+	+	+	64	У	32
15	+	-	+	+	3	+	+	+	-	64	ФО	16
16	+	+	-	+	1	+	+	-	-	64	ФО	32
17	+	+	+	-	2	+	+	+	-	64	ФО	16
18	-	+	+	+	3	+	+	+	-	64	ФО	32
19	+	-	+	+	1	+	+	-	+	64	У	32
20	+	+	-	+	2	+	+	-	+	64	У	16
21	+	+	+	-	3	+	+	+	-	32	ФО	32
22	-	+	+	+	1	+	+	+	-	32	ФО	16
23	+	-	+	+	2	+	+	-	+	32	У	32
24	+	+	-	+	3	+	+	-	+	32	У	16
25	+	+	+	-	1	+	+	+	-	16	ФО	32
26	-	+	+	+	2	+	+	+	-	16	ФО	16
27	+	+	-	+	3	+	+	-	+	16	У	32
28	+	+	-	+	1	+	+	-	+	16	У	16
29	-	+	+	+	2	+	+	+	-	8	ФО	32
30	-	+	+	+	3	+	+	+	-	8	ФО	16

Традиционная архитектура

N ва- ри- ан- та	Шина адрес- данные		Наличие сопро- цессора	Основная память		Ввод-вывод		Преры- вания	
	С	Р		Объем, Мб	ШД	И	П	Р	О
1	+	-	+	32	32	+	-	+	-
2	+	-	+	64	16	+	-	+	-
3	+	-	+	1024	32	+	-	+	-
4	+	-	+	1024	16	+	-	+	-
5	+	-	+	64	16	+	-	+	-
6	-	+	+	32	32	-	+	+	-
7	-	+	+	32	16	-	+	+	-
8	-	+	-	64	32	-	+	+	-
9	-	+	-	1024	16	-	+	+	-
10	-	+	-	1024	32	-	+	+	-
11	+	-	-	64	16	+	-	+	-
12	+	-	-	32	32	+	-	+	-
13	+	-	-	32	16	+	-	+	-
14	+	-	-	32	32	+	-	+	-
15	+	-	+	64	16	+	-	+	-
16	-	+	+	64	16	-	+	-	+
17	-	+	+	32	32	-	+	-	+
18	-	+	+	1024	16	-	+	-	+
19	-	+	+	1024	32	-	+	-	+
20	-	+	+	1024	16	-	+	-	+
21	+	-	+	64	32	+	-	-	+
22	+	-	-	64	16	+	-	-	+
23	+	-	-	32	32	+	-	-	+
24	+	-	-	32	16	+	-	-	+
25	+	-	-	64	32	+	-	-	+
26	-	+	-	32	16	-	+	-	+
27	-	+	-	1024	32	-	+	-	+
28	-	+	-	1024	16	-	+	-	+
29	-	+	+	32	32	-	+	-	+
30	-	+	+	64	16	-	+	-	+

Гарвардская архитектура

N ва- ри- ан- та	Форматы данных			Адрес- ность	Способы адресации				Регистровая память		
	8	16	32		Н	О	П	К	Коли- чество	Тип	Разряд- ность
1	+	+	-	1	+	+	+	-	8	У	8
2	-	+	+	2	-	+	+	+	8	У	16
3	+	+	-	3	+	+	+	-	8	ФО	8
4	-	+	+	1	+	+	+	-	8	ФО	16
5	+	+	-	2	+	+		+	16	У	32
6	-	+	+	3	+	+		+	16	У	32
7	+	+	-	1	+	+	+	-	16	ФО	8
8	-	+	+	2	+	+	+	-	16	ФО	16
9	+	+	-	3		+	+	+	32	У	8
10	-	+	+	1		+	+	+	32	У	16
11	+	+	-	2	+	+	+	-	32	ФО	8
12	-	+	+	3	+	+	+	-	32	ФО	16
13	+	+	-	1	-	+	+	+	64	У	8
14	-	+	+	2		+	+	+	64	У	16
15	+	+	-	3	+	+	+	-	64	ФО	8
16	-	+	+	1	+	+	-	-	64	ФО	16
17	+	+	-	2	+	+	+	-	64	ФО	8
18	-	+	+	3	+	+	+	-	64	ФО	16
19	+	+	-	1	+	+	-	+	64	У	8
20	-	+	+	2	+	+	-	+	64	У	16
21	+	+	-	3	+	+	+	-	32	ФО	8
22	-	+	+	1	+	+	+	-	32	ФО	16
23	+	+	-	2	+	+	-	+	32	У	8
24	-	+	+	3	+	+	-	+	32	У	16
25	+	+	-	1	+	+	+	-	16	ФО	8
26	-	+	+	2	+	+	+	-	16	ФО	16
27	+	+	-	3	+	+	-	+	16	У	8
28	-	+	+	1	+	+	-	+	16	У	16
29	+	+	-	2	+	+	+	-	8	ФО	8
30	-	+	+	3	+	+	+	-	8	ФО	16

Гарвардская архитектура

N ва- ри- ан- та	Шина адрес- данные		Память данных		Память команд		Ввод- вывод		Преры- вания	
	С	Р	Объем, Кб	ЩД	Объем, Кб	ШК	И	П	Р	О
1	+	-	1	16	1	16	-	+	+	-
2	+	-	2	16	2	32	-	+	+	-
3	+	-	4	16	4	64	-	+	+	-
4	+	-	4	32	4	16	-	+	+	-
5	+	-	2	16	2	16	-	+	+	-
6	+	-	1	16	1	32	-	+	+	-
7	+	-	2	16	2	16	-	+	+	-
8	-	+	1	32	1	32	+	-	+	-
9	-	+	4	16	8	64	+	-	+	-
10	-	+	4	16	4	16	+	-	+	-
11	-	+	1	16	2	16	+	-	+	-
12	-	+	2	32	4	32	+	-	+	-
13	-	+	2	16	2	16	+	-	+	-
14	-	+	4	16	8	32	+	-	+	-
15	+	-	1	16	1	64	-	+	+	-
16	+	-	1	32	2	16	-	+	-	+
17	+	-	2	16	4	16	-	+	-	+
18	+	-	4	16	8	32	-	+	-	+
19	+	-	4	16	4	16	-	+	-	+
20	+	-	1	32	1	32	-	+	-	+
21	+	-	2	16	2	64	-	+	-	+
22	-	+	2	16	4	16	+	-	-	+
23	-	+	1	16	2	16	+	-	-	+
24	-	+	4	32	8	32	+	-	-	+
25	-	+	4	16	4	16	+	-	-	+
26	-	+	2	16	2	32	+	-	-	+
27	-	+	1	16	2	64	+	-	-	+
28	-	+	1	32	1	16	+	-	-	+
29	-	+	4	16	8	16	-	+	-	+
30	-	+	2	16	4	32	-	+	-	+

- основные характеристики регистровой памяти (количество регистров, их тип и разрядность);
- тип локальной шины;
- наличие сопроцессора;
- характеристики основной памяти (объем и разрядность шины данных);
- способ организации ввода — вывода;
- основные характеристики системы прерываний.

Аппаратно могут поддерживаться 6 основных форматов:

- 8 разрядов с ФТ;
- 16 разрядов с ФТ;
- 32 разряда с ФТ;
- 64 разряда с ФТ;
- 32 разряда с ПТ;
- 64 разряда с ПТ.

Для тех вариантов, в которых предусмотрены 32- и 64-разрядные форматы, требуется реализовать их в форматах с фиксированной и с плавающей точками.

Понятие адресности относится к операционным командам, т. е. командам, выполняющим действия по обработке данных. Например, нет смысла использовать трехадресный формат для команд, реализующих операции перехода. Таким образом, заданная адресность для разрабатываемой системы команд является максимальной. Для одноадресной команды предполагается наличие аккумулятора, что позволяет в операционных командах один из адресов рассматривать как жестко заданный, не требующий наличия поля в команде (подразумеваемый адрес).

Всего предусмотрено использование 4 основных способов адресации памяти: непосредственная (Н), относительная (О); прямая (П) и косвенная (К). Основной является относительная адресация. Реализация механизмов непосредственной адресации для тех вариантов, в которых это не предусмотрено

заданием, необходима в том случае, когда нужно обеспечить функциональную полноту системы команд. Например, если предусматривается наличие индексных регистров, то следует предусмотреть механизмы их загрузки.

Для регистровой памяти задаются количество регистров, их тип, а также разрядность. Регистры могут быть либо универсальными (У), либо функционально ориентированными (ФО). В первом случае речь идет о регистрах общего назначения (РОНах), которые предназначены для хранения как адресов, так и чисел с ФТ. Если предусматривается наличие сопроцессора, то для хранения чисел с ПТ должны использоваться отдельные регистры с ПТ, которые физически размещаются внутри сопроцессора. Если сопроцессор отсутствует, то возможны различные варианты организации РОНов для данных с ПТ:

— использование одной и той же физической памяти для данных с ФТ и ПТ. В этом случае, как правило, данные с ПТ занимают несколько РОНов, используемых для хранения данных с ФТ. Вариант совмещения РОНов для хранения данных с ФТ и с ПТ ставит перед разработчиком архитектуры следующую задачу: может ли данное с ПТ размещаться, начиная с любого РОНа с ФТ, или же только определенным образом, например только с четных адресов. Во втором случае встает вопрос задания адресов для данных с ПТ: либо они совпадают с адресами для данных с ФТ, либо вводится своя сквозная нумерация (0, 1, 2, ...);

— использование различных модулей физической памяти для реализации РОНов для данных с ФТ и ПТ.

Функциональная ориентация РОНов подразумевает их разбиение на группы в зависимости от функционального назначения (например, регистры для хранения данных, индексов, базы, указателя стека и т.п.). Внутри групп возможно дополнительное разбиение (например, регистр базы

данных, регистр базы программного кода, регистр базы стека т.д.).

В процессоре может использоваться либо совмещенная шина адреса и данных (С), либо отдельные шины адреса и данных (Р).

В состав системы может входить сопроцессор, обеспечивающий выполнение операций с ПТ, который должен иметь собственные регистры с ПТ.

В задании определяется объем оперативной памяти, к которому необходимо обеспечить доступ в режиме прямоадресуемой памяти. При невозможности или затруднительности обеспечить такой режим допускается использование страничной организации памяти. Разрядность шины данных определяет разрядность памяти.

Ввод-вывод может быть либо изолированным (И), т.е. с использованием специальных команд ввода-вывода, либо по аналогии с ячейками памяти (П).

Изолированный ввод-вывод подразумевает использование специальных команд ввода-вывода. Идентификация обращения к регистрам внешних устройств осуществляется по коду операции.

Организация ввода-вывода по аналогии с обращением к ячейкам оперативной памяти (ОП) предполагает использование единого адресного пространства для ячеек памяти и портов ввода-вывода. В этом случае адресное пространство делится между ячейками ОП и регистрами внешних устройств. Данный подход позволяет по коду адреса определить, идет ли обращение к ячейке ОП или к регистру ВУ, не требует введения специальных команд ввода-вывода и дает возможность использовать различные способы адресации при обращении к регистрам ВУ.

Возможны три основных варианта физического подключения внешних устройств:

— внешние устройства подключаются к той же шине, что и память;

— память подключается к локальной шине, а внешние устройства — к некоторой внешней шине;

— для подключения памяти и внешних устройств используются отдельные интерфейсы.

Для всех вариантов требуется разработать систему прерываний, при этом требуется реализовать либо равные (P), либо относительные приоритеты. Дисциплина обслуживания прерываний с равными приоритетами подразумевает простейшую одноуровневую систему прерываний. В случае задания относительных приоритетов подразумевается ранжирование причин прерываний и реализация многоуровневой системы прерываний.

Отличительной особенностью гарвардской архитектуры является наличие отдельной памяти команд (см. табл. 1.4). Предполагается, что память команд размещается внутри кристалла.

Раздельная реализация памяти команд и данных упрощает организацию конвейера по выборке команд из памяти и по размещению их в очередь для дальнейшего исполнения. Данная организация позволяет избежать конфликтов при обращении к памяти, которые имеют место при использовании традиционной архитектуры, однако не снимает проблем, связанных с реализацией команд переходов.

Наличие виртуальной памяти для процессоров с традиционной архитектурой подразумевает решение следующих задач:

— разработку формата и способа хранения таблицы виртуальных и физических страниц;

— выбор способа преобразования виртуального адреса в физический и его аппаратную поддержку;

— выбор варианта удаления страницы из ОП и его алгоритмическую проработку;

— разработку алгоритма обмена страницами между ОП и внешней памятью.

Возможность работы ЭВМ в многозадачном режиме должна обеспечиваться проработкой структуры и форматов системных дескрипторов задач, а также аппаратной поддержкой, определяющей состояние текущей программы.

Наличие кэш-памяти данных для процессоров с гарвардской архитектурой подразумевает решение следующих задач:

— разработку основных алгоритмов работы кэш-памяти;

— проработку вопросов структурной организации кэш-памяти;

— определение разрядов статуса и управления накопителем кэш-памяти;

— разработку стратегии замены строк и ее алгоритмическую проработку.

1.4. Результаты проектирования

В процессе выполнения курсового проекта разрабатываются:

1) форматы команд и данных;

2) регистровая модель процессора;

3) структура выводов кристалла;

4) система команд, включая семантику их выполнения и установку флажков;

5) обобщенный алгоритм функционирования процессора, включая алгоритмы выполнения основных команд;

6) алгоритмы выполнения основных операций на шине (чтение, запись, ввод, вывод, захват шины, прерывание);

7) внутренняя организация (до структурного уровня), включая механизмы виртуальной памяти, механизмы защиты памяти и механизмы работы с кэш-памятью.

2. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА

2.1. Уточнение структуры системы

Разрабатываемый процессор предназначен для использования в качестве ЦП рабочей станции (варианты с традиционной архитектурой) либо в качестве встроенного процессора.

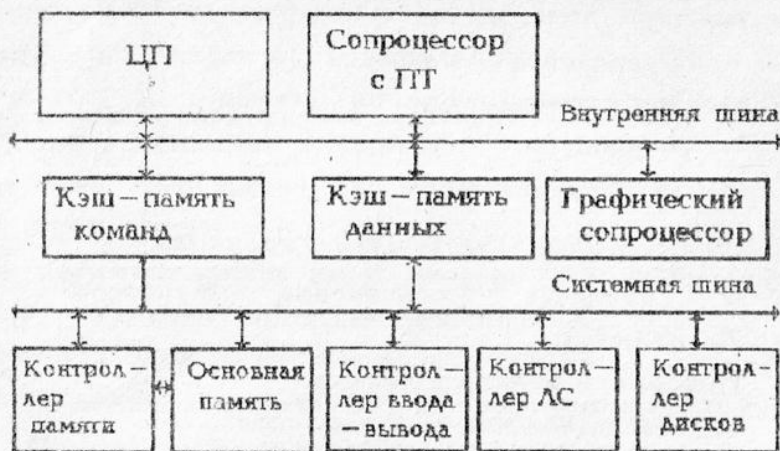


Рис. 2.1

Структура обобщенной рабочей станции показана на рис. 2.1 и содержит ЦП, сопроцессор с ПТ, кэш-память команд, кэш-память данных и графический сопроцессор, которые работают на внутреннюю (локальную шину). На системную (глобальную) шину работают основная (оперативная) память, контроллеры ввода-вывода, контроллер локальной сети (ЛС), а

также контроллер гибкого и жесткого дисков. Основная память выполняется на кристаллах динамической памяти большой емкости. Контроллер памяти обеспечивает, в частности процедуру регенерации памяти.

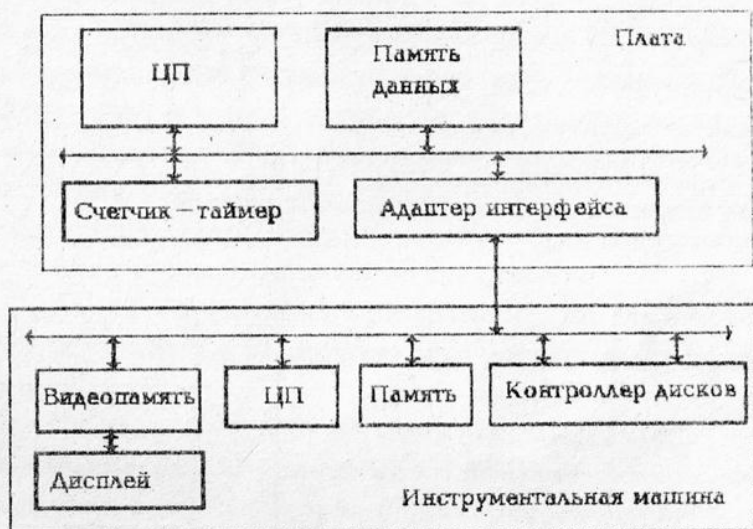


Рис. 2.2

Обобщенная структура, соответствующая гарвардской архитектуре, приведена на рис. 2.2. Она представляет собой одноплатную ЭВМ, предназначенную для встроенных применений. На плате размещаются: ЦП, память данных, счетчик-таймер, адаптер интерфейса. Кроме того, на плате могут размещаться вспомогательные схемы и интерфейсные схемы для подключения внешних устройств (на рис. 2.2 эти схемы не показаны).

Разрабатываемая плата не имеет собственной дисковой памяти, дисплея и клавиатур. Предполагается, что разработка и отладка программного обеспечения осуществляются на инструментальной машине с использованием кросс-систем программирования. В качестве инструментальной машины может

использоваться, например рабочая станция, имеющая в своем составе дисплей, клавиатуру, дисковую память и т. п.

Основной задачей данного этапа проектирования является уточнение структуры и разрядности внутренней шины.

2.2. Разработка архитектуры внешних выводов

Для вариантов, не требующих использования сопроцессора, разрабатывается архитектура внешних выводов только ЦП. Для вариантов, требующих сопроцессора, разрабатывается архитектура внешних выводов ЦП и сопроцессора.

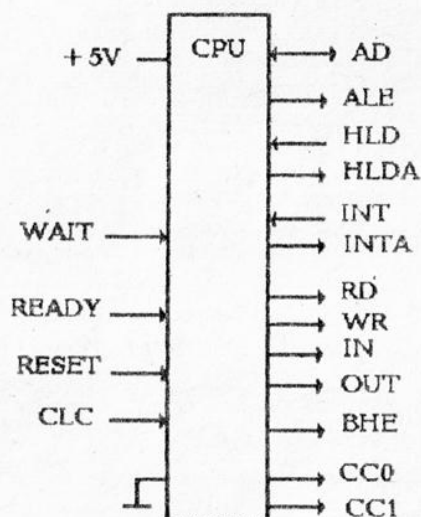


Рис 2.3

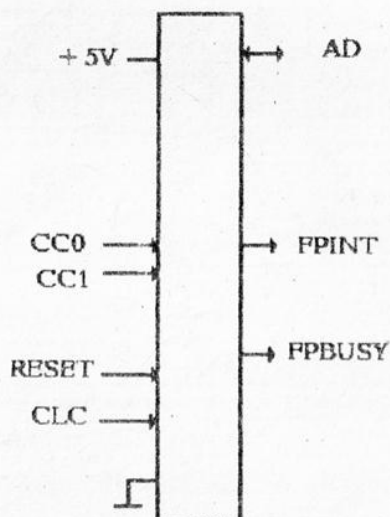


Рис. 2.4

При разработке архитектуры внешних выводов следует ориентироваться на использование стандартных корпусов, имеющих, соответственно, 40, 68, 132, 144, 168, 172 выводов. В случае, если остаются "лишние" выводы, они могут быть использованы для следующих целей:

- дублирования контактов "Земля" и "Питание";

- увеличения разрядности шины адреса;
- введения нескольких уровней прерывания.

Обобщенная архитектура внешних выводов кристалла ЦП показана на рис. 2.3, а сопроцессора — на рис. 2.4.

Процессор, изображенный на рис. 2.3, имеет совмещенную шину адреса и данных (AD). Сигнал **ALE** используется для фиксации адреса на внешнем регистре — защелке. Пара сигналов **HLD** и **HLDA** используется для реализации механизма захвата шины. Сигналы **INT** и **INTA** являются сигналами запроса и подтверждения прерывания. Если на корпусе имеется достаточное число свободных выводов, то целесообразно ввести несколько уровней запроса на прерывания.

Линии **RD** (Чтение), **WR** (Запись), **IN** (Ввод), **OUT** (Вывод), **BHE** (Разрешение записи старшего байта) задают выполняемую на шине операцию.

Сигналы **CC0**, **CC1** и **WAIT** используются для организации взаимодействия с внешним математическим сопроцессором. Линии **CC0** и **CC1** служат для синхронизации работы ЦП и сопроцессора и могут использоваться, например следующим образом:

CC0	CC1	
0	0	Нет операции
0	1	Очистка очереди
1	0	Запись команды в буфер
1	1	Выборка команды из буфера

На контакт **WAIT** поступает сигнал от сопроцессора об окончании вычислений.

Контакт **READY** (Готовность) служит для приема сигнала готовности от медленных внешних устройств. Назначение выводов питания, **RESET** (Сброс) и **CLC** (Синхронизация) очевидны.

На рис. 2.4 показан возможный вариант архитектуры внешних выводов сопроцессора.

В сопроцессоре (рис. 2.4) на вывод **FPBUSY** выдается единичный сигнал, указывающий на то, что сопроцессор занят. По линии **FPINT** выдается сигнал прерывания в случае возникновения ошибочной ситуации типа попытки деления на нуль. Назначение прочих выводов такое же, как и одноименных выводов ЦП.

2.3. Выбор форматов данных

На рис. 2.5 показаны основные используемые типы данных.

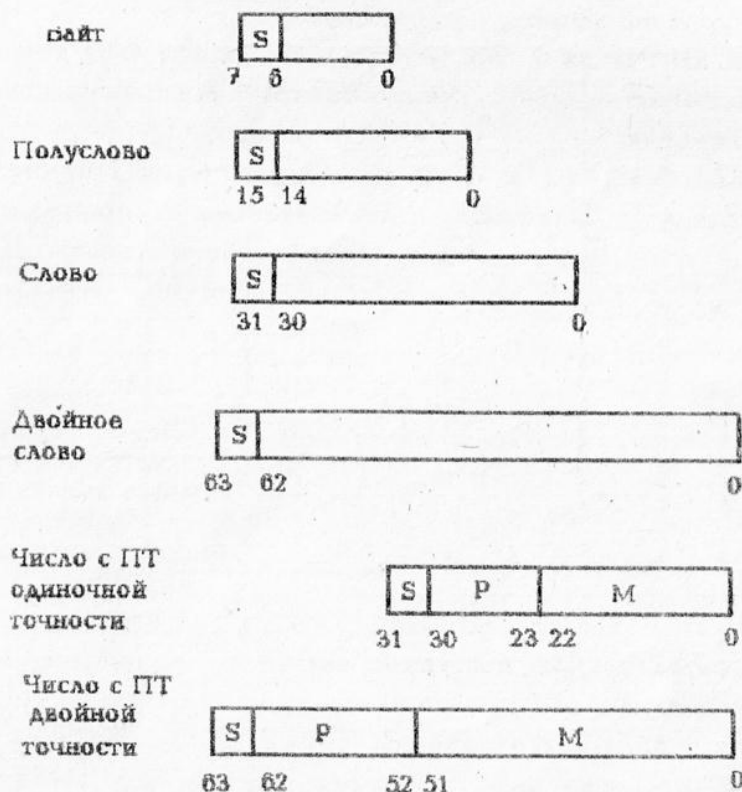


Рис. 2.5

Числа с ФТ представляют собой целые со знаком или без знака. Целые со знаком представляются в дополнительных кодах. Целые без знака используются также для представления адресов. Для представления чисел с ПТ рекомендуются следовать требованиям стандарта (ANSI/IEEE Standart 754). При этом следует дать описания представления особых ситуаций (нуль переполнение, NAN).

2.4. Определение структуры регистровой памяти

На этапе определения структуры регистровой памяти выбираются:

- число регистров различных типов и их разрядность;
- состав и структура регистра флажков.

Один из возможных вариантов организации регистровой памяти ЦП для вариантов с использованием внешнего сопроцессора показан на рис. 2.6.

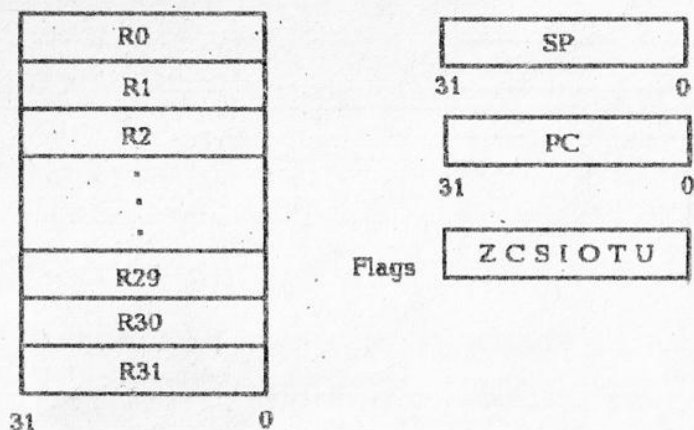


Рис. 2.6

В структуре, приведенной на рис. 2.6, регистровая память включает тридцать два 32-разрядных РОНа и 32-разрядный счетчик команд. Регистр флажков содержит 7 флажков, которые используются следующим образом:

- Z — признак нулевого результата;
- C — признак переноса из старшего разряда;
- S — знак результата;
- O — признак переполнения результата;
- I — разрешение прерывания;
- T — пошаговый режим;
- U — режим супервизор/пользователь.

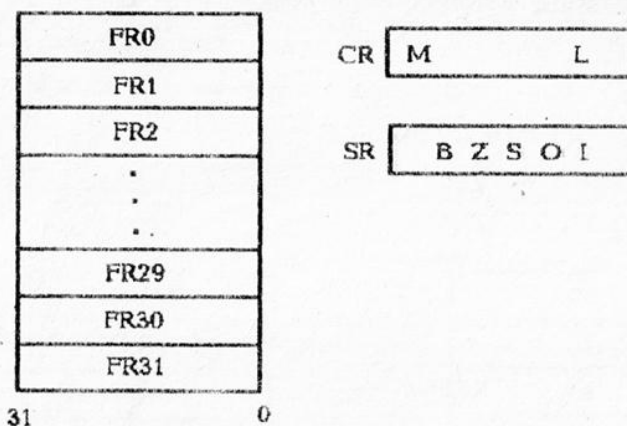


Рис. 2.7

Возможный вариант организации регистровой памяти сопроцессора показан на рис. 2.7.

2.5. Выбор форматов команд

При выборе форматов следует исходить из принципов RISC — обработки, в соответствии с которыми используется минимальное число форматов команд. Наиболее желательным является использование одного (32-разрядного) формата. Приемлемым можно также считать вариант, когда команды имеют разную длину [1].

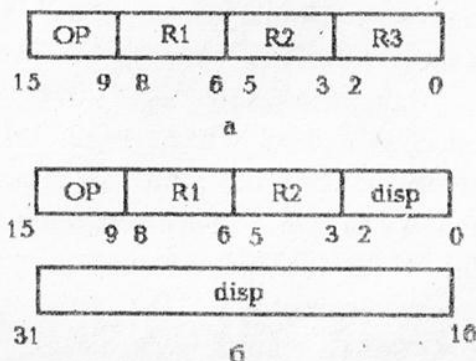


Рис. 2.8

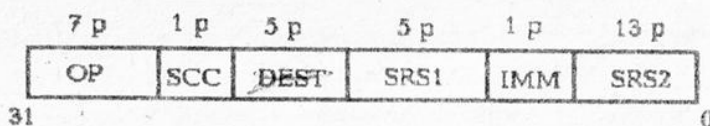


Рис. 2.9

На рис. 2.8 и 2.9 приведены возможные форматы команд. На рис. 2.8 показаны форматы команд, соответствующие второму случаю. Операционные команды (рис. 2.8, а) имеют длину 16 разрядов и содержат 4 поля: 7-разрядное поле кода операции (OP) и три 3-разрядных поля для задания регистров источников и приемника. Типичная двухместная операция типа сложения выполняется по схеме $R1 \leftarrow \langle R2 \rangle + \langle R3 \rangle$. Команды обращения к памяти, обращения к подпрограммам, переходов, а

также команды, использующие непосредственные операнды имеют длину 32 разряда (рис. 2.8,б).

Кроме полей **OP**, **R1** и **R2** имеется 19-разрядное поле смещения (**disp**). Адрес памяти определяется сложением содержимого **R2** и **disp**: $A = \langle R2 \rangle + disp$.

На рис. 2.9 показан пример 32-разрядного формата [2].

Команда на рис. 2.9 содержит 6 полей: 7-разрядное поле кода операции (**OP**), 1-разрядное поле **SCC**, разрешающее или запрещающее установку флажков. Поля **DEST**, **SRS1** и **SRS2** задают, соответственно, регистр-приемник и регистры-источники данных. Одноразрядное поле **IMM** определяет содержание 13-разрядного поля **SRS2**. Если **IMM** = 0, то содержимое указанного поля интерпретируется как номер регистра, в противном случае — рассматривается как 13-разрядная константа. В командах обращения к памяти оно используется в качестве смещения. При этом адрес памяти определяется как $A = \langle SRS1 \rangle + SRS2$.

2.6. Разработка системы команд

Результаты проектирования следует оформить в виде таблицы:

Система команд

N	Мне- мони- ка	Название	Содержание	Флажок			Код
				C	Z	S	
1	ADD	Сложение с ФТ	$R1 = \leftarrow \langle R2 \rangle + \langle R3 \rangle$	+	+	+	0000001
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	JMP	Безуслов- ный переход	$A = \leftarrow \langle R2 \rangle + disp$	-	-	-	1010101

Система команд должна быть функционально полной и включать, как минимум, следующие команды:

- 1) обращения к памяти по чтению и записи;
- 2) арифметические с ФТ (сложение, вычитание, умножение, деление);
- 3) арифметические с ПТ (сложение, вычитание, умножение, деление);
- 4) логические (поразрядное "И", "ИЛИ" и "Исключающее ИЛИ");
- 5) сдвигов на произвольное число тактов;
- 6) условных и безусловных переходов;
- 7) работы с подпрограммами;
- 8) загрузки в регистры непосредственных операндов;
- 9) ввода — вывода;
- 10) управления вычислительным процессом.

2.7. Внутренняя организация

При разработке структуры процессора следует исходить из следующих моментов.

В процессоре реализуется конвейерный принцип обработки информации. Используется либо двухступенчатый, либо трехступенчатый конвейер. Два возможных варианта структурной организации процессора показаны на рис. 2.10 и 2.11. Для двухступенчатого конвейера (рис. 2.10) выделяются фаза выборки команды и фаза выполнения команды. Параллельно и асинхронно могут работать два процессора: процессор памяти и исполнительный процессор. Процессор памяти обеспечивает работу с памятью команд и памятью данных, а также с кэш-памятью. На исполнительный процессор возлагаются функции, связанные с дешифрацией и выполнением команд.

Для трехступенчатого конвейера (рис. 2.11) выделяются фазы: выборки команды из памяти; дешифрации команды и выборки операндов; выполнения команды и записи результата.



Рис. 2.10



Рис. 2.11

При использовании трехступенчатого конвейера (рис. 2.11) в состав процессора могут входить, например адресный процессор, процессор шины, процессор команд и исполнительный процессор. Процессор шины управляет выполнением основных операций на шине. Адресный процессор выполняет операции адресной арифметики и операции, связанные с преобразованием виртуальных адресов в физические с использованием таблицы переадресации (TLB).

Процессор команд выполняет функции, связанные с дешифрацией команд, и извлечение операндов. Исполнительный процессор занят собственно выполнением операции.

Регистровая память реализуется на регистровых файлах, допускающих в одном такте выборку двух операндов и запись одного операнда. Данные при этом выбираются из регистровых файлов по переднему фронту импульса синхронизации и записываются по заднему фронту. Операционные устройства, как с ФТ, так и с ПТ, представляют собой комбинационную схему. Все операции с ФТ выполняются за 1 такт, а с ПТ — за фиксированное число тактов. Например, сложение с ПТ и умножение с ПТ выполняются за 2 такта, а деление с ПТ — за 5 тактов. Арифметические операции с ПТ выполняются либо в сопроцессоре, либо в отдельном арифметико-логическом устройстве (АЛУ).

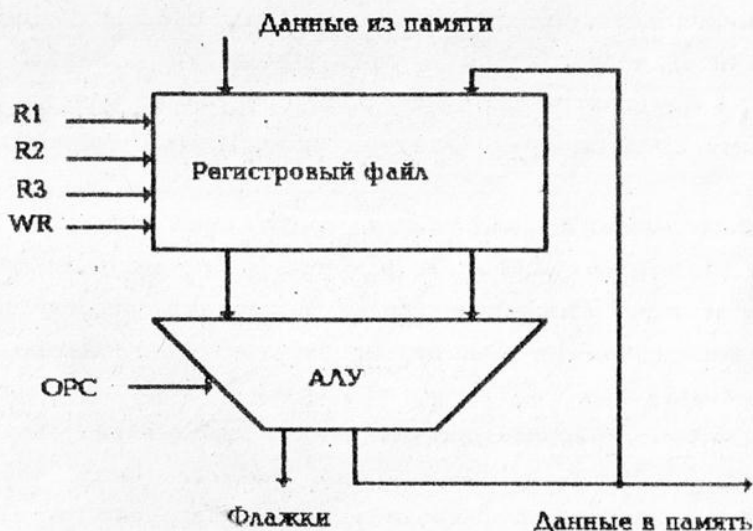


Рис. 2.12

Обобщенная структура исполнительной подсистемы показана на рис. 2.12. Основными элементами исполнительной подсистемы являются регистровый файл и АЛУ. По переднему фронту синхроимпульса из регистрового файла одновременно выбираются 2 операнда, соответствующих адресам **R2** и **R3**. В течение времени длительности тактового импульса сигналы распространяются через комбинационные схемы АЛУ. При этом выполняемая операция определяется подаваемым на вход АЛУ кодом операции (**ОП**). Если на управляющий вход регистрового файла подается сигнал разрешения записи **WR**, то по заднему фронту в регистровом файле по адресу **R1** запоминается результат выполнения операции.

В случае, если заданием предусмотрено поддержание механизма работы с виртуальной памятью, то в состав адресного процессора необходимо ввести регистры переадресации (**TLB**), в которых хранятся отдельные строки таблицы переадресации. Число **TLB** может находиться в пределах от 16 до 256. В системе команд необходимо предусмотреть команды работы с данными регистрами и позаботиться о том, чтобы эти команды были отнесены к привилегированным. Кроме собственно регистров **TLB** необходимо предусмотреть наличие необходимых управляющих регистров.

Типовая схема формирования физического адреса показана на рис. 2.13. Виртуальный адрес разбивается на два поля: поле **N** сегмента и поле смещения. Поле **N** сегмента задает номер виртуального сегмента, с которым работает пользователь. Это поле используется в качестве точки входа в таблицу переадресации, которая обычно имеет достаточно большой размер и хранится в оперативной памяти. Часть таблицы хранится в аппаратных регистрах (**TLB**). Загрузка отдельных частей таблицы осуществляется под управлением операционной системы.

Каждая строка таблицы переадресации имеет два поля, в которых указываются, соответственно, начальный адрес

физического сегмента и атрибуты. В качестве атрибутов могут выступать: признак наличия данного сегмента в оперативной памяти, длина сегмента, поле изменения и поле защиты.

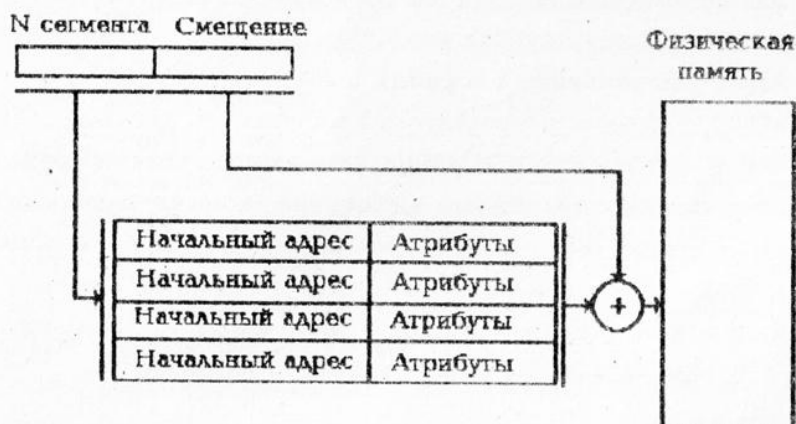


Рис. 2.13

В поле изменения фиксируется факт обращения к данному сегменту по записи. Поле защиты может содержать следующую информацию:

- разрешение чтения данных;
- разрешение записи;
- разрешение выполнения.

Более подробно данный вопрос рассмотрен в [3, 4].

2.8. Основные алгоритмы функционирования

В процессе выполнения курсового проекта должны быть разработаны основные алгоритмы функционирования процессора, а именно:

- обобщенный алгоритм функционирования системы;
- алгоритмы функционирования отдельных подсистем (процессоров);
- алгоритмы взаимодействия отдельных подсистем (процессоров).

Разработка обобщенного алгоритма предполагает выделение отдельных фаз выполнения. Например, возможно выделение следующих фаз выполнения команды:

- 1) **IF** — выборка команды из кэш — памяти команд;
- 2) **RD** — выборка данных из регистровой памяти, при этом одновременно дешифрируется код операции;
- 3) **ALU** — выполнение операции в АЛУ;
- 4) **MEM** — запись результата в кэш — память данных;
- 5) **WB** — запись в РОНЫ результата выполнения операции.

Использование подобного конвейера команд позволяет в ряде случаев заканчивать выполнение команды в каждом такте.

Для вариантов, предусматривающих разработку процессора с традиционной архитектурой, необходимо разработать основные механизмы работы с виртуальной памятью.

2.9. Алгоритмы выполнения отдельных операций

Необходимо подробно рассмотреть структуру и основные алгоритмы функционирования исполнительной подсистемы. Достаточно полно алгоритмы выполнения отдельных операций описаны в [1].

В командах ветвления используется механизм отложенного условного перехода [2]. Данный механизм предполагает, что при появлении команды перехода выполняется "по инерции" одна команда, которая располагается в программе непосредственно за командой перехода.

При реализации команд работы с подпрограммами возможно либо использование стека, расположенного в оперативной памяти, либо запоминание адреса возврата в одном из РОНов.

В системе команд необходимо предусмотреть команды программного прерывания, при выполнении которых следует запоминать наряду с адресом возврата состояние регистра флажков.

2.10. Разработка алгоритмов выполнения основных операций на шине

Требуется разработать алгоритмы выполнения основных операций на шине:

- чтение команд;
- чтение данных;
- запись данных;
- прерывание;
- ввод, вывод;
- захват шины.

При работе с кэш-памятью обмен данными между ней и основной памятью осуществляется в режиме передачи блоков данных фиксированной длины.

В пояснительной записке следует привести временные диаграммы работы шины.

3. ОФОРМЛЕНИЕ КУРСОВОГО ПРОЕКТА

Курсовой проект выполняется в виде пояснительной записки и графической части. Пояснительная записка должна содержать:

- титульный лист;
- оглавление с указанием страниц;
- задание на курсовой проект;
- структурную схему разрабатываемой системы;
- подробное описание архитектуры на регистровом уровне, включающее как пользовательские, так и системные регистры;
- описание форматов данных, команд и способов адресации;
- описание системы команд, включающее описание логики установки флажков;
- описание архитектуры внешних выводов кристалла ЦП и сопроцессора;

— описание временных диаграмм выполнения основных операций на шине: чтения, записи, ввода, вывода, прерывания, захвата, взаимодействия с сопроцессором;

— структурно-функциональную схему ЦП и сопроцессора и ее описание;

— обобщенный алгоритм функционирования системы, определяющий взаимодействие отдельных подсистем (процессоров);

— алгоритмы функционирования отдельных подсистем (процессоров);

— описание методов и средств управления памятью.

Ориентировочный объем пояснительной записки составляет 20–30 страниц формата 210x297 мм.

Графическая часть курсового проекта должна содержать:

— структурную схему разрабатываемой системы;

— структурно-функциональную схему ЦП и сопроцессора;

— временные диаграммы выполнения основных операций на шине;

— граф-схемы алгоритмов.

Оформление пояснительной записки и графической части курсового проекта должно быть выполнено в соответствии с требованиями ЕСКД и ЕСПД.

Библиографический список

1. Архитектура ЭВМ, ВС и сетей: Учеб. пособие / В.Н.Балакин, А.Ф.Казак, А.О.Тимофеев и др.; ЛЭТИ. — Л., 1988.
2. Компьютеры на СБИС: В 2 кн. / Т. Мотоока, Х. Хорикоси, М. Сакаутти и др. / Пер. с япон. — М.: Мир, 1986.
3. Григорьев В.Н. Микропроцессор i486. Архитектура и программирование. — М.: ГРАНАЛ, 1993.
4. Паппас К., Марри У. Микропроцессор 80386: Справ. — М.: Радио и связь, 1993.

Оглавление

Введение.....	3
1. Техническое задание.....	3
1.1. Предмет проектирования.....	3
1.2. Общие требования к разрабатываемому процессору.....	4
1.3. Исходные данные для курсового проектирования.....	4
1.4. Результаты проектирования.....	13
2. Последовательность выполнения курсового проекта...14	
2.1. Уточнение структуры системы.....	14
2.2. Разработка архитектуры внешних выводов.....	16
2.3. Выбор форматов данных	18
2.4. Определение структуры регистровой памяти.....	19
2.5. Выбор форматов команд	21
2.6. Разработка системы команд.....	22
2.7. Внутренняя организация.....	23
2.8. Основные алгоритмы функционирования.....	27
2.9. Алгоритмы выполнения отдельных операций.....	28
2.10. Разработка алгоритмов выполнения основных операций на шине.....	29
3. Оформление курсового проекта.....	29
Библиографический список.....	30

Редактор И.Б. Сенишева
Лицензия ЛР N 020617 от 10.08.92

Подписано в печать 22.05.95. Формат 60 x 84 1/16. Бумага тип. N 2.
Печать офсетная. Усл. печ. л. 1,86. Уч. — изд. л. 2,0.
Тираж 150 экз. Заказ 82
Редакционно — издательский отдел ГЭТУ им. В.И. Ульянова (Ленина)

Ротапринт МГП "Поликом"
197376, С. — Петербург, ул. Проф. Попова, 5