

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Национальный минерально-сырьевой университет «Горный»

Кафедра информационных систем и вычислительной
техники

ЭВМ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА

*Методические указания
по курсовому проектированию
для студентов бакалавриата
направления подготовки 09.03.01*

САНКТ-ПЕТЕРБУРГ
2015

ЭВМ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА: методические указания к выполнению курсового проекта / Национальный минерально-сырьевой университет «Горный». Сост.: *М.В. Копейкин, В.В. Спиридонов, Е.О. Шумова*. СПб, 2015, 47 с.

Методические указания разработаны в соответствии с Федеральным государственным образовательным стандартом высшего профессионального образования и рабочей программой дисциплины.

Дисциплина посвящена изучению вопросов функциональной и структурной организации ЭВМ и их периферийных устройств. В основных ее разделах изучаются принципы построения, функционирования и оценки характеристик ЭВМ и периферийных устройств, отдельных их компонент и блоков.

Предлагаемая в указаниях тематика курсового проектирования связана с разработкой специализированного процессора ЭВМ. Круг рассматриваемых при этом вопросов охватывает составление микропрограмм выполнения операций в арифметическо-логическом устройстве, разработку микропрограмм выполнения команд, блока управления и общей структуры процессора, составление функциональных и принципиальных схем отдельных устройств процессора, а также выполнение их поведенческого моделирования.

Методические указания предназначены для студентов направления 230100 – Информатика и вычислительная техника.

Табл.2. Ил.12. Библиогр.: 9 назв.

Научный редактор проф. *И.В. Иванова*

© Национальный минерально-сырьевой университет «Горный», 2015 г.

Введение

Курсовой проект по дисциплине "ЭВМ и периферийные устройства" является одной из основных самостоятельных работ. Задание на курсовой проект построено так, что для работы над ним требуется привлечение знаний и данного курса, и ранее изученных дисциплин. Общий порядок выполнения проекта соответствует порядку работы над выпускной квалификационной работой.

Целью данного курсового проекта является приобретение практических навыков в использовании полученных знаний при разработке структуры ЭВМ, ознакомление со средствами автоматизации проектирования ЭВМ, а также закрепление основных теоретических положений курса. В результате выполнения проекта студенты должны получить четкое представление о взаимодействии основных узлов и блоков ЭВМ в процессе обработки информации и научиться использовать аппарат, методы и средства проектирования ЭВМ. Этой цели наилучшим образом соответствует самостоятельное выполнение студентом разработки структуры, алгоритмического описания, схем и конструкции специализированного процессора ЭВМ.

Курсовой проект выполняется на втором курсе. Исходными данными для проектирования являются основные требования к функциональным и техническим характеристикам процессора, перечисленные в форме задания на курсовой проект (прил. 1). Варианты задания выбираются по шифру студента. Таблицы заданий приведены в прил. 2.

Общие указания

1. Содержание курсового проекта

В ходе курсового проектирования студент должен разработать:

- 1) структурную схему арифметико-логического устройства (АЛУ) и микропрограммы выполнения в нем заданных операций;
- 2) блок управления АЛУ или процессором;
- 3) структурную схему процессора и микропрограммы обработки команд;

4) принципиальную схему АЛУ или блока микропрограммного управления (в соответствии с вариантом задания) и конструктивную реализацию этой схемы и процессора в целом.

Курсовой проект оформляется в виде пояснительной записки и графической части.

В пояснительную записку должны входить:

1) титульный лист, на котором указываются название работы, фамилии студента и руководителя, номер группы и шифр студента;

2) задание на курсовой проект;

3) раздел 1 "Арифметико-логическое устройство", в котором должны быть представлены:

- описание вариантов структуры АЛУ, их оценки по заданному критерию и структурная схема выбранного варианта;

- краткое описание назначения и особенности организации узлов выбранного варианта структуры АЛУ;

- микропрограммы выполнения заданных арифметических и логических операций, представленных в виде граф-схем;

- список, реализуемых в АЛУ микроопераций;

4) раздел 2 "Процессор", который должен содержать:

- описание структуры, форматов и общего порядка выполнения команд;

- описание характеристик и назначения основных блоков структурной схемы процессора;

- микропрограммы выполнения команд процессором, представленные в виде граф-схем;

- список реализуемых в процессоре микроопераций;

- таблицу оценки времени выполнения команд в тактах;

5) раздел 3 "Блок управления" должен включать:

а) для вариантов управления на основе автомата Мили или

Мура:

- граф автомата управления АЛУ;

- функции переходов и выходов автомата;

- таблицу кодирования состояний;

- окончательный вид функций возбуждения элементов памяти автомата;

б) для вариантов с блоком микропрограммного управления:

- перечень всех микрокоманд, различающихся операционной частью;
- матрицу совместимости микроопераций АЛУ и матрицу совместимости микроопераций процессора по выборке и обработке команд и выборке операндов;
- список подмножеств совместимых микроопераций (полученный методом прямого включения);
- описание формата микрокоманд и коды микроопераций;
- таблицу размещения микрокоманд в микропрограммной памяти;
- общее описание структуры блока микропрограммного управления и назначения его узлов;

б) раздел 4 "Конструкция устройства" должен содержать общее описание конструкции процессора, ориентированной на модульный вариант его исполнения.

Примерный объем пояснительной записки 25...35 страниц формата 210 x 297 мм. Граф-схемы микропрограмм, рисунки, таблицы и другой пояснительный материал может выполняться в карандаше на миллиметровой бумаге.

Графическая часть курсового проекта должна содержать:

- 1) структурную схему процессора (с указанием на ней всех управляющих сигналов и информационных связей) формат 12 или 22;
- 2) функциональную схему блока управления АЛУ или процессора (в соответствии с требованиями задания);
- 3) функциональную схему одного разряда всех узлов АЛУ в их взаимосвязи между собой;
- 4) принципиальную схему АЛУ или блока управления (согласно варианту задания) в заданном элементном базисе;
- 5) конструкторский чертеж платы и общего вида процессора.

Допускается изображение схем на миллиметровой бумаге. Используемые в пояснительной записке и графической части проекта обозначения должны удовлетворять требованиям ЕСКД.

2. Основные этапы работы над курсовым проектом

При работе над курсовым проектом целесообразно придерживаться следующего порядка проработки отдельных вопросов:

- 1) определение структуры АЛУ процессора и разработка алгоритмов выполнения в нем заданных операций;
- 2) составление функциональной схемы разряда АЛУ;
- 3) разработка структуры процессора и алгоритмов выполнения команд процессора с соответствующими микропрограммами;
- 4) синтез управляющего автомата АЛУ или блока управления процессора;
- 5) разработка принципиальной схемы определенного заданием устройства процессора;
- 6) разработка вопросов конструктивной реализации процессора.

Законченный проект сдается на кафедру для проверки.

Проект, получивший положительную рецензию, после устранения отмеченных недостатков выносится на защиту.

Структурная организация проектируемого процессора

В курсовом проекте разрабатываются вопросы организации и функционирования центрального ядра ЭВМ, включающего процессор, рассматриваемый как совокупность АЛУ и устройства управления, и оперативную память (ОП). Организация ввода-вывода, обмена с внешними ЗУ и выполнения пультовых операций не прорабатываются.

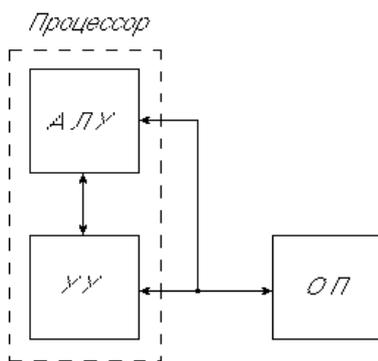


Рис. 1. Общая структура центральной части ЭВМ

В самом общем виде структурная схема проектируемой системы может быть представлена так, как показано на рис. 1. В ходе работы над проектом такое начальное представление раскрывается и детализируется, по-

полняется при необходимости требуемыми блоками. Особенности отдельных устройств данной структуры рассматриваются ниже.

1. Арифметико-логическое устройство

Структура АЛУ разрабатываемого в курсовом проекте процессора в основном определяется указанными в задании составом операций, организацией сумматора, особенностями представления чисел и способом реализации сдвига (в отдельном блоке или в любом блоке). Однако и при этих ограничениях существуют возможности построения различных структур АЛУ и составления разных микропрограмм выполнения операций.

Рекомендации по построению и выбору структурной организации АЛУ излагаются в разделе "Порядок выполнения основных этапов разработки процессора". Тем не менее, здесь целесообразно принять излагаемые ниже соглашения.

Особенности выполнения операций и связь АЛУ с другими блоками ЭВМ

При разработке структуры АЛУ и микропрограмм выполнения операций следует исходить из того, что в начальный момент перед выполнением операции операнды находятся на регистрах АЛУ, при этом числа с плавающей точкой представлены в нормализованном виде. Предполагается, что операнды, подлежащие обработке, передаются на регистры АЛУ устройством центрального управления. Результат операции передается из АЛУ в другие блоки также устройством управления.

Числа с фиксированной запятой в зависимости от варианта задания представляются либо в виде целых чисел со знаками, либо в виде дробных чисел (строго меньших по модулю, чем единица) со знаками. Различие в способах представления не сказывается на операциях сложения и вычитания. Однако для умножения и деления это различие проявляется в следующем.

При выполнении операции умножения произведение имеет в общем случае вдвое больше разрядов, чем сомножители, т. е. двойную длину. Поскольку в курсовом проекте не предусматривается обработка операндов двойной длины, то в случае представления

дробных чисел следует производить округление произведения до стандартного формата.

В случае умножения целых чисел округление принципиально недопустимо и необходимо следить за тем, чтобы количество значащих цифр произведения не превышало разрядности АЛУ. Выявить это можно с помощью анализа на нулевое содержимое старшей половины разрядов произведения двойной длины. И если произведение имеет больше значащих разрядов, чем сомножители, то следует зафиксировать переполнение.

При выполнении операции деления, наоборот, переполнение возможно в случае дробных чисел, если делитель по модулю меньше делимого. Для целых чисел такая ситуация, напротив, приводит к нормальному, отличному от нуля, результату. Количество цифр частного при делении целых чисел следует ограничить необходимым для представления целой части частного. Для уменьшения погрешности вычислений целесообразно определять еще одну (первую дробную) цифру частного и производить округление.

В случае деления дробных чисел количество разрядов частного естественно ограничить обычной разрядностью операндов, также определяя одну дополнительную цифру для округления.

Заданием предусматривается также некоторое различие в форме представления чисел с плавающей запятой, касающееся порядков.

В случае представления порядков в виде целых чисел со знаками все операции над ними выполняются, как над обычными двоичными числами с фиксированной запятой.

Если порядки представляются как характеристики (как "смещенный порядок" или целое число без знака), то вычисление их разностей и сумм имеет свои отличия.

При внимательном анализе можно заметить, что представление порядков в виде характеристик подобно представлению двоичных чисел со знаком в дополнительном коде. Разница состоит в том, что старший разряд равен инверсии знака. Тогда нетрудно вывести и общее правило обработки характеристик, сводящееся к следующему. Характеристики суммируются или вычитаются как целые без знака. Результатом этого является прямой (дополнительный) код суммы или разности, указанием на это является наличие (отсутствие)

переноса из старшего разряда. Для получения характеристики суммы (разности) характеристик теперь следует к результату добавить единицу старшего разряда (фактически представляющую собой смещение порядка).

В процессе выполнения операций сложения и вычитания чисел с плавающей запятой при выравнивании порядков и нормализации результата возможно изменение мантисс операндов. Для уменьшения погрешности результата операции в алгоритмы их выполнения следует включить округление.

При выполнении всех операций в АЛУ, кроме результатов, должны формироваться значения признаков (нулевой результат, переполнение, отрицательный результат и т.д.), позволяющих анализировать результат выполнения операций в АЛУ. Значения признаков фиксируются на регистре состояния АЛУ.

Особенности схемы АЛУ

Выполнение микроопераций в АЛУ может распределяться по-разному между блоками АЛУ. Микрооперации суммирования, например, выполняются в специальном блоке - сумматоре. Микрооперации сдвига и преобразования кодов, напротив, часто выполняются в нескольких регистрах АЛУ, что вызвано их относительной простотой и большой повторяемостью. Эти два варианта распределения микроопераций по блокам АЛУ называют структурами с общими и закрепленными микрооперациями соответственно [2].

В случае закрепленных микроопераций одинаковые по содержанию микрооперации могут выполняться в различных операционных блоках АЛУ. Например, микрооперация сдвига вправо на один разряд может выполняться в аккумуляторе и в одном из входных регистров АЛУ.

При построении структур с общими микрооперациями предполагается, что для каждой микрооперации в АЛУ существует отдельный и обычно только один блок. Для случая микроопераций сдвига таким блоком является сдвигатель, с помощью которого и производятся все микрооперации сдвига.

Этот блок, как правило, является комбинационной схемой, хотя возможно и использование схемы с памятью, регистра (но та-

кой вариант проигрывает по быстродействию). Сдвиг информации, хранимой в каком-либо регистре, осуществляется с помощью комбинационного сдвигателя посредством подачи информации с выходов регистра на вход сдвигателя и подачи в сдвигатель соответствующего управляющего сигнала, определяющего вид сдвига. При этом на выходе сдвигателя окажется сдвинутая требуемым образом входная информация, занесение которой в регистр-получатель должно осуществляться в данном же такте.

Прием информации может осуществляться либо непосредственно в регистр, содержимое которого сдвигается, либо через промежуточный регистр. В первом случае регистр, содержащий сдвигаемую информацию и одновременно принимающий сдвинутую, должен строиться на M-S-триггерах. Во втором случае на сдвиг требуется два такта.

Описанные способы изображены на рис. 2, а и б соответственно, где Р – регистр, СДВ – сдвигатель, а ПР – промежуточный регистр.

В задании предусматривается построение АЛУ с закрепленными или общими микрооперациями сдвига. Все остальные микрооперации могут реализовываться тем или иным способом, выбор которого осуществляется в соответствии с принятым критерием.

При организации передачи данных между блоками АЛУ необходимо учитывать принятый (парафазный или однофазный) способ передачи и используемые в регистрах типы триггеров. Так, при построении регистра на RS-триггерах однофазной передаче кода должна предшествовать подача сигнала установки в нуль триггеров регистра, принимающего информацию. Парафазная передача или организация регистра на D-триггерах не требуют предварительного сброса регистра.

В ходе построения схемы АЛУ следует производить оценку различных вариантов по заданному критерию (см. раздел "Порядок выполнения основных этапов разработки процессора").

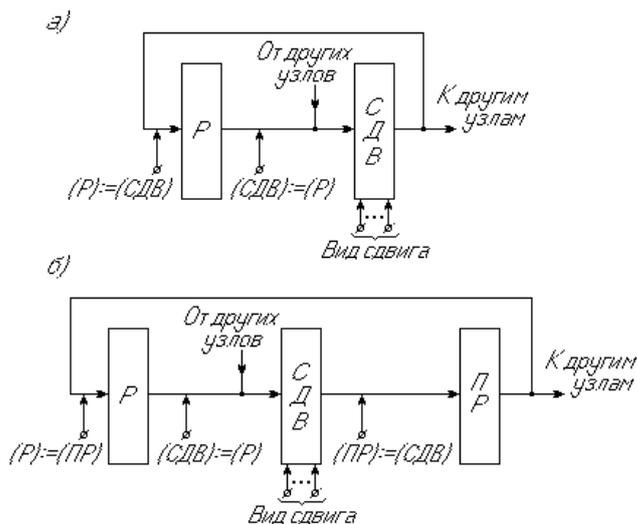


Рис.2. Сдвиг информации с помощью сдвигателя с одновременным возвратом сдвинутой информации (а) и с возвратом через промежуточный регистр (б)

2. Оперативная память

Независимо от способов адресации команд и данных оперативная память (ОП) рассматривается как устройство, выборка и запись информации в котором осуществляется словами по 2 или 4 байта в соответствии с вариантом задания. Предполагается, что ОП представляет собой модуль с необходимыми схемами хранения информации и управления.

На структурной схеме процессора оперативную память рекомендуется представлять, как показано на рис. 3. На рис. 3, а, кроме собственно блока оперативной памяти ОП, изображены регистр адреса РА и регистр данных РД с соответствующими шинами, а также схема управления памятью СхУ ОП. Такое представление, конечно не является обязательным, так как в ряде случаев (особенно для статических полупроводниковых ЗУ) регистры РА и РД могут отсутствовать вовсе или их роль могут выполнять другие регистры процессора. Вариант структуры с непосредственным подключением

оперативной памяти к шинам адреса и данных представлен на рис. 3, б. Аналогично часто самостоятельно не выделяется схема управления СхУ ОП.

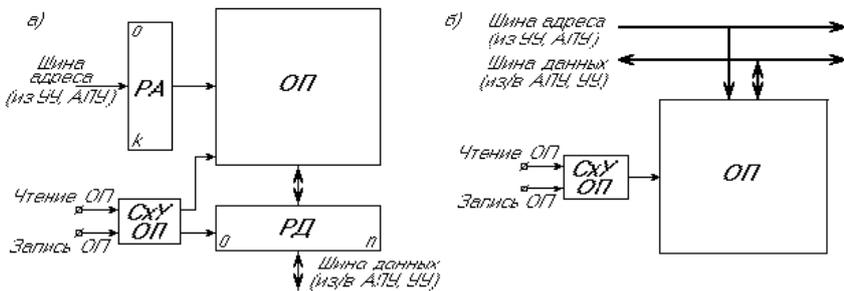
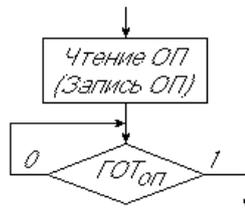


Рис.3. Варианты представления оперативной памяти на структурной схеме процессора: с буферными регистрами адреса и данных (а) и с непосредственным подключением оперативной памяти к шинам адреса и данных (б)

Количество разрядов регистра адреса определяется числом адресуемых слов ОП, а при необходимости адресации полуслов или байтов (например, для случаев команд не кратных длине слова) может быть увеличено на 1-2 разряда. Объем ОП в задании определен в байтах, поэтому количество слов в зависимости от разрядности ОП в 2 или 4 раза меньше.

Количество разрядов РД равно количеству разрядов слова памяти. Разряды контроля на четность для каждого байта хотя и могут предполагаться существующими, но в общую разрядность памяти не включаются и не рассматриваются.



Предполагается, что для выполнения операций записи или чтения слова необходимо однократно подать соответствующий сигнал в СхУ ОП: "Запись ОП" или "Чтение ОП". Момент окончания операции в ОП, инициированной этим сигналом, определяется осведомительным сигналом ГОТоп, вырабатываемым СхУ ОП. Нулевое значение этого сигнала указывает на то, что операция в ОП не закончена, а единичное – говорит о ее завершении. Такое средство, вообще говоря, имеется не во всех вариантах организации ОП, так как цикл обращения к памяти, как правило, фиксирован. Однако поскольку цикл ОП часто имеет большую длительность, чем такт процессора, то необходимо предусматривать синхронизацию работы процессора и ОП.

Рис. 4. Синхронизация циклов процессора в ОП с помощью "ждущих" вершин микропрограммы

В курсовом проекте это рекомендуется обеспечивать введением в граф-схемы микропрограмм "ждущих" вершин, помещаемых после обращения к памяти (рис. 4). В интервале времени между инициированием операции чтения или записи в ОП и моментом, когда сигнал ГОТоп примет единичное значение, состояние регистра данных (выхода ОП) считается неопределенным и использоваться не может. Следует отметить, что в ряде случаев, особенно в микропроцессорах, режим ожидания сигнала готовности от различных устройств, в том числе от ОП, реализуется аппаратно.

Оперативная память всегда реализует один из следующих режимов: "чтение", "запись", или "хранение". В режим "чтение" или "запись" ОП переходит при подаче соответствующих управляющих сигналов на СхУ ОП. Формы и длительности сигналов "чтение ОП" и "запись ОП" могут различаться для разных типов ОП (полупроводниковых, статических, динамических, ферритовых и др.). В рамках курсового проекта предлагается считать что при подаче соответствующего управляющего сигнала на вход СхУ ОП оперативная память переходит в режим "чтение" или "запись" (при этом на шине ГОТоп появляется нулевой сигнал).

По завершении всех действий (переходных процессов), связанных с инициированным режимом, СхУ ОП помещает единичный сигнал на выходе ГОТоп. К моменту появления этого сигнала слово памяти, заверщенное по режиму "чтение", помещено в РД (или на

выходе шины данных) или по режиму "запись" занесено в соответствующую ячейку памяти.

3. Устройство управления процессора

Устройство предназначено для реализации определенных заданием команд процессора, поэтому его состав и структура должны соответствовать указанным ограничениям и требованиям. К ним в первую очередь относятся способы адресации, адресность команд, характер связей между узлами (непосредственные или магистральные), размещение регистров общего назначения и др.

Способы адресации

В каждом варианте задания на курсовой проект предусматривается проработка трех вариантов адресации из восьми различных схем: прямой, индексной, двойной индексной, с автоиндексированием, относительно счетчика команд, косвенной, косвенной регистровой, непосредственной. Различные способы адресации отличаются порядком преобразования исходного (указанного в команде) адреса в исполнительный адрес (по которому при выполнении команды происходит обращение к ОП).

При *прямой адресации* исполнительный адрес совпадает с исходным и должен позволять адресовать операнд, расположенный в любой части памяти. Поэтому для команд обработки данных (имеющих формат слов) прямой адрес должен позволять адресовать любое слово памяти. В командах управления прямой адрес должен позволять адресовать команду при любом возможном ее расположении в ОП (см. с. 31). Обеспечивать прямую адресацию для всех типов команд не обязательно.

При *индексной адресации* исполнительный адрес образуется сложением некоторой величины - смещения, указываемой в поле исходного адреса, с содержанием регистра общего назначения (или специального индексного регистра), в котором записано базисное значение адреса, трактуемое в некоторых случаях как приращение адреса. Поле адреса при этом имеет вид, представленный на рис. 5, а, где *I* – номер регистра общего назначения (РОН), содержимое

которого складывается со смещением для получения исполнительного адреса, *СМ* – смещение.

РОН обычно представляют собой отдельную регистровую память на триггерных регистрах, являющуюся частью процессора. Количество РОН в большинстве случаев составляет от 4 до 32 регистров. Однако в ряде ЭВМ эти регистры размещаются в оперативной памяти, что оправдано при частых прерываниях, большом количестве РОН и быстродействующей ОП. В этих случаях расположение РОН в ОП обычно задается специальным регистром-указателем, имеющимся в процессоре. В рамках курсового проекта в вариантах задания с размещением РОН в ОП допускается считать их расположение в ОП фиксированным, например, в начальных словах памяти. Способ реализации РОН и их количество указаны в задании.

Двойной индексной адресацией называется такая модификация индексной адресации, при которой исполнительный адрес образуется посредством сложения смещения и содержимого двух РОН, хранящих базисное значение адреса и его приращение. В этом случае в адресном поле команды указываются смещение и номера двух РОН, как показано на рис 5, б, где *Б* – номер РОН, содержащего базисный адрес, *И* – номер РОН, содержащего приращение адреса (индекс), *СМ* – смещение. Такой способ адресации позволяет организовать работу с перемещаемыми в ОП программами и массивами данных.

Автоиндексированием называется такая модификация индексной адресации, при которой содержимое РОН после его сложения со смещением увеличивается. При этом увеличение производится таким образом, что в следующий раз при использовании данного РОН для индексирования будет получен адрес следующего слова (полуслова, байта) памяти, т.е. если ОП адресуется только пословно, то при автоиндексировании содержимое РОН наращивается на единицу. (Если ОП адресуется, например, побайтно, то при двухбайтовом операнде содержимое РОН увеличивается на 2). Разрядность РОН должна позволять указывать в них полноразрядный адрес па-



Рис. 5. Содержимое адресного поля команды при индексной (а) и двойной индексной (б) адресации

мяти. Если это не удастся выполнить по ограничениям задания (варианты с большой емкостью и малой разрядностью ОП), допускается задавать в базисном адресе только старшие, а в индексе – только младшие разряды адреса.

В вариантах с организацией РОН в виде отдельной регистровой памяти в процессоре рекомендуется снабдить их дешифратором номера РОН и двунаправленной шиной данных РОН. В этом случае предлагается изображать их на структурной схеме процессора так, как показано на рис 6, где $РОН_0...РОН_n$ – регистры общего назначения, $ДшРОН$ – дешифратор номера РОН, ШД РОН – шина данных регистров общего назначения, $ЧтРОН$ и $ЗпРОН$ – сигналы чтения содержимого РОН и записи информации в РОН соответственно.

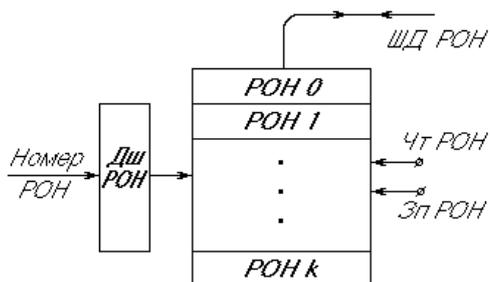


Рис. 6. Представление блока регистров общего назначения на структурной схеме (для вариантов с размещением РОН в процессоре)

Для извлечения информации из РОН следует подать на $ДшРОН$ номер нужного регистра и сигнал $ЧтРОН$, при этом на $ШД РОН$ будет выдано содержимое указанного регистра. Для записи информации в РОН нужно подать на $ДшРОН$ номер регистра, в который следует записать информацию, поместить записываемую информацию на

$ШД РОН$ и подать сигнал $ЗпРОН$, выполнив все эти действия, очевидно, в одном такте.

Под *косвенной адресацией* понимается такой способ, при котором исходный адрес указывает слово памяти, содержимое которого рассматривается как исполнительный адрес.

Косвенной регистровой адресацией называется такая модификация косвенной адресации, при которой исходный адрес является не адресом слова памяти, а номером РОН.

При *адресации относительного счетчика* команд исполнительный адрес получается посредством сложения исходного адреса с содержимым счетчика команд. Типичным использованием такой схемы адресации является применение ее в командах передачи

управления. Поэтому в вариантах задания, включающих этот способ адресации, достаточно ограничиться ее реализацией только в этих командах.

При *непосредственной адресации* в поле исходного адреса указывается значение операнда, вступающего в операцию. При этом учитывая, что разрядность адреса обычно меньше разрядности операнда, непосредственный операнд может иметь неполную длину (например, в один бит). Другим вариантом, позволяющим задать непосредственно полноразрядный операнд, является увеличение длины команды с непосредственной адресацией до необходимого числа разрядов. Следует отметить, что в двух- и трехадресных командах, как правило, непосредственно адресуются только один из операндов.

В задании не предусматривается требование реализации непосредственной адресации во всех командах обработки данных. Достаточно обеспечить ее, например, в одной из логических операций, введя в систему команд процессора дополнительную команду, задающую эту операцию с непосредственной адресацией.

Особенности системы команд процессора

В зависимости от варианта задания список команд, определенный заданием, может потребовать некоторых уточнений и введения дополнительных команд, связанных с загрузкой РОН, реализацией какого-либо способа адресации и пр. (см. с. 27)

При реализации команд управления порядком выполнения программы наибольшие затруднения может вызвать разработка команд безусловного перехода с возвратом (обращения к подпрограмме) и управления циклом. Для ознакомления с различными способами реализации этих команд рекомендуется изучать их выполнение по [1]. Если этой информации окажется недостаточно, то целесообразно ознакомиться с порядком выполнения аналогичных команд в некоторых моделях ЭВМ, пользуясь [3...6].

Порядок выполнения основных этапов разработки

1. Определение структуры АЛУ и разработка алгоритмов выполнения заданных операций

При работе над данным разделом должны быть рассмотрены следующие основные вопросы:

а) определение формата данных и способа их представления в процессоре;

б) анализ алгоритмов выполнения заданных операций и соответствующих структур АЛУ для их реализации с учетом ограничений задания;

в) разработка структурной схемы операционной части АЛУ и микропрограмм выполнения операций в соответствии с заданным критерием эффективности;

г) разработка функциональной схемы одного разряда АЛУ (с учетом ограничений заданного элементного базиса).

В результате должна быть получена полная структурная схема АЛУ со всеми внутренними связями между блоками АЛУ, а также установлены внешние связи с блоками, не входящими в состав операционной части, микропрограммы выполнения в АЛУ заданных операций и функциональная схема разряда АЛУ.

Определение формата данных и способа представления их в АЛУ

В различных вариантах задания требуется проработать вопросы преобразования данных следующих типов:

а) двоичных чисел с фиксированной запятой;

б) двоичных или шестнадцатеричных чисел с плавающей запятой;

в) десятичных чисел с фиксированной запятой;

г) логических величин.

В представлении каждого вида данных имеется ряд альтернатив (место и способ фиксации запятой, взаимное расположение полей и т.д.). Однако в связи с

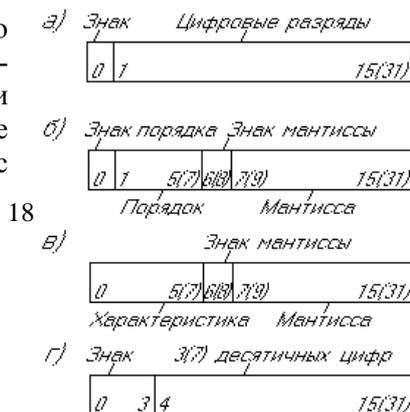


Рис. 7. Форматы данных

учебным характером проекта и заданными ограничениями рекомендуется использовать форматы данных, приведенные на рис. 7.

Формат двоичных чисел с фиксированной запятой показан на рис. 7, а. Согласно заданию, запятая фиксирована либо после знакового разряда, либо после младшего цифрового разряда. В первом случае числа будут дробными, не превосходящими по модулю единицу, во втором – целыми.

Различия в выполнении операции для этих случаев указаны ранее (см. с.7). Хранение чисел с фиксированной запятой в памяти и представление их в процессоре может осуществляться как в прямом, так и в дополнительном коде.

Для представления двоичных чисел с плавающей запятой рекомендуется формат, показанный на рис. 7, б и в. Следует отметить, что формат, представленный на рис. 7, в, имеет модификации, отличающиеся расположением знака мантиссы. (У чисел с плавающей запятой характеристика, в отличие от порядка, как известно [1], является целым числом без знака и равняется порядку, увеличенному на константу вида $100..0$, разрядность которой равна разрядности порядка со знаком. Так, например, порядку -14 (-01110) в шести разрядной сетке соответствует характеристика $010010 = -01110 + 100000$, а порядку $+14$ ($+01110$) – характеристика 101110 . Числа с плавающей запятой, как правило, хранятся в памяти в прямом коде.

При представлении двоичных чисел с фиксированной и плавающей запятыми во всех узлах процессора (кроме, может быть, сумматора АЛУ) позиции знаков занимают один разряд.

Рекомендуемый формат десятичных чисел приведен на рис. 7, г. Для унификации обработки всех десятичных разрядов чисел и исключения неполных тетрад под знак десятичного числа отводятся 4 двоичных разряда (одна тетрада). Десятичные числа также обычно хранят в памяти в прямом коде.

Для представления логических величин в ЭВМ используются наборы двоичных разрядов: байт, слово, последовательность байтов и т.д. В курсовом проекте для ограничения объема работ рекомендуется принять в качестве формата логических данных одно слово.

Анализ алгоритмов выполнения заданных операций и структур АЛУ для их реализации с учетом ограничений задания

При анализе алгоритмов выполнения операций и структур АЛУ целесообразно учесть следующие рекомендации общего характера.

В связи с тем, что в ряде заданий предусматривается обработка чисел в двоичной и десятичной системе счисления, сумматор АЛУ должен быть рассчитан на выполнение операций сложения и вычитания в обеих системах счисления.

Логические операции могут выполняться как на самом сумматоре АЛУ, так и на отдельной схеме. Первый вариант предпочтительнее при использовании сумматора накапливающего типа. Выполнение логических операций на отдельной схеме позволяет несколько упростить внутреннюю структуру сумматора, хотя увеличивает количество узлов АЛУ и связей между ними.

Для упрощения микропрограмм работы устройства управления при выборке операндов целесообразно независимо от типа выполняемой операции АЛУ считать, что в исходном состоянии операнды находятся всегда на одних и тех же регистрах (сумматоре). Результат выполнения операции по ее окончании также должен помещаться в одном и том же регистре (сумматоре).

При разработке структуры АЛУ, как и других систем, можно выделить две основные стадии:

- а) построение структуры из набора типовых блоков,
- б) оценка полученного варианта по некоторому критерию.

Первая стадия достаточно формализована, хотя интенсивно исследуется в настоящее время. Одним из перспективных путей, дающих определенные рекомендации по построению систем (обработки информации), является функционально-структурный подход [3]. Вторая стадия – оценка вариантов структур – разработана в большей мере, и в рамках данного курсового проекта рекомендуется применять методы оценки с помощью интегральных критериев, из-

ложенных, например, в [8]. Эта методика обязательна также в дипломном проектировании.

Анализ алгоритмов и учет ограничений задания с целью формирования различных вариантов структуры АЛУ в соответствии с рекомендациями методологии функционально-структурного подхода проводится в следующем порядке.

1. Исходным, заданным принимается общий алгоритм выполнения операции (операций), для реализации которой проектируется АЛУ. Конечно, имеется возможность выбора различных видов алгоритмов. В самом общем случае, это может быть даже не алгоритм в строгом смысле данного термина, предполагающего однозначную последовательность преобразований, а просто перечень действий, выполняемых для получения результата, с указанием их информационно-логической взаимосвязи. Однако в данном проекте указанный анализ не обязателен.

2. Для каждого шага (блока, оператора) S_i алгоритма рассматривается возможность реализации его типовыми узлами и микрооперациями АЛУ (сдвиг, передача, преобразование кода, суммирование, счет и т.п.). Если такая возможность имеется, то фиксируются все найденные варианты r_{ji} реализации рассматриваемого шага S_i . Если же шаг алгоритма предполагает выполнение относительно сложных преобразований, то его необходимо разбить на более мелкие действия, а затем найти варианты их реализации. Такое разбиение обычно предполагает разделение обрабатываемой информации на фрагменты и отдельное манипулирование с ними, выполнение требуемого преобразования в виде однотипной последовательности шагов и т.п.

3. Возможные варианты реализации r_{ji} каждого из шагов S_i алгоритма предварительно оцениваются в соответствии с заданным критерием эффективности (см. далее с.22). Если при такой оценке какой-либо из вариантов оказывается существенно хуже, то он далее не принимается во внимание. Если же имеются варианты реализации шага S_i алгоритма, оценки которых близки, то эти варианты подлежат дальнейшему рассмотрению.

4. Полученные варианты реализации r_{ki} и r_{lj} для каждой пары шагов S_i, S_j , алгоритма проверяются на возможность (или целесообразность) совместного использования. После этого различные вари-

анты структуры АЛУ строятся объединением узлов, необходимых для реализации хотя бы одного r_{ki} для каждого из шагов S_i алгоритма.

Выбор структурной схемы операционной части АЛУ в соответствии с заданным критерием эффективности и составление микропрограмм выполнения операций

При сравнении различных вариантов структурной реализации систем используют оценки их по ряду показателей (параметров), сводимых в некоторый обобщенный интегральный критерий. Одной из простых форм такого критерия является линейная форма

$$F(u) = \sum_{i=1}^{i=n} k_i * u_i,$$

где n – количество различных показателей u_i (параметров), по которым оценивается система; k_i – коэффициент при i -м показателе, согласующий шкалы измерений различных показателей, а также учитывающий относительную значимость отдельных показателей в общей оценке системы. Такая линейная форма – частный случай аддитивного критерия – широко распространена в силу достаточной простоты получения оценок. Однако следует помнить, что линейные аппроксимации обычно связаны с определенной потерей точности.

Известны различные методы и алгоритмы построения интегральных критериев [9]. В настоящем курсовом проекте в качестве критерия эффективности разрабатываемого АЛУ принят критерий

$$F(u) = \sum_{i=1}^{i=5} k_i * u_i, \quad (1)$$

где коэффициенты k_i выбираются из соответствующей таблицы задания, а показатели u_i , $i = (1 \div 5)$ имеют следующий смысл: u_1 – среднее время выполнения операции в АЛУ в условных единицах (см. далее); u_2 – количество элементов (вентилей И, ИЛИ, И-НЕ и др.) в схеме АЛУ, u_3 – количество узлов (регистров, счетчиков, сумматоров, схем сравнения, преобразователей и т.д.) в АЛУ, u_4 – количест-

во связей между узлами АЛУ (с учетом их разрядности), u_5 – количество управляющих сигналов (микроопераций АЛУ).

Эти показатели учитывают особенности структуры АЛУ или любого иного блока обработки информации, связывая их с особенностями интегральной технологии (см., например, [11, с.180]). (Есть, конечно, и другие факторы, определяющие технические характеристики блоков обработки информации в интегральном исполнении. К ним, помимо используемой технологии, в первую очередь относятся (топологические) особенности связей между узлами и временные соотношения в процессах передачи и хранения информации.)

При сравнении вариантов структуры АЛУ по критерию (1) более предпочтительным вариантом является, очевидно, тот, который обеспечивает наименьшее значение $F(u)$. Количество исследуемых вариантов должно быть не менее 3.

При определении значений $u_1 \dots u_5$ требуется учитывать следующее.

Показатель u_1 вычисляется как

$$u_1 = n_{\text{оп}} * \tau_{\text{АЛУ}},$$

где $n_{\text{оп}}$ – среднее количество тактов, затрачиваемое на выполнение операций в АЛУ, а $\tau_{\text{АЛУ}}$ – длительность такта АЛУ.

В свою очередь, среднее количество тактов

$$n_{\text{оп}} = \sum_{j=1}^{j=4} a_j * t_j,$$

где a_j – относительная частота выполнения операций j -типа, а t_j – среднее время операции j -типа в АЛУ в тактах. Верхний индекс суммирования равен 4, так как для упрощения и в связи с учебным характером проекта принимаются во внимание лишь четыре операции, определенные заданием для разработки микропрограмм. Тогда окончательно имеем

$$u_1 = \tau_{\text{АЛУ}} \sum_{j=1}^{j=4} a_j * t_j.$$

Значения величин a_j определяются заданием. Значения времен t_j (в тактах) определяются по (составляемым) микропрограммам их выполнения. Длительность такта АЛУ $\theta_{\text{АЛУ}}$ принимается числен-

но равной глубине схемы АЛУ, т. е. максимальному количеству элементов (И, ИЛИ, И-НЕ,), через которые проходит сигнал от входа к выходу АЛУ при выполнении самой длительной микрооперации (обычно это суммирование).

Определение значений показателей $u_2...u_5$ производится непосредственно по функциональной и структурной схемам АЛУ. Однако необходимость наличия составленной функциональной схемы для получения численного значения показателя u_2 требует разработки всех рассматриваемых вариантов до уровня функциональной схемы, что может быть излишне трудоемко. В этом случае целесообразно довести до уровня функциональной схемы один из вариантов и на его основе получать оценку u_2 для других вариантов, прорабатывая функциональные схемы лишь для отличающихся узлов этих вариантов. Определяя значение показателя u_4 в случае магистральной связи одного узла с несколькими, следует учитывать эту связь столько раз, сколько узлов подключено к источнику.

При выборе варианта структуры АЛУ фиксируются способы реализации шагов алгоритмов выполнения операций. Поэтому составление микропрограмм теперь сводится к подстановке вместо каждого блока S_i алгоритма (последовательности микроопераций, соответствующих выбранной его реализации r_{jib} , т. е. каждому оператору и логическому условию блок схемы алгоритма ставится в соответствие оператор (последовательность операторов) микропрограмм, состоящих из одной или нескольких микроопераций, или условная вершина.

Данный этап целесообразно выполнять, начиная с наиболее сложной (арифметической) операции из числа заданных. После рассмотрения алгоритма первой операции переходят к следующей. Перед этим для каждого из операционных блоков целесообразно построить таблицу, в которой должны быть отражены наборы микроопераций и логических условий, закрепленных за блоками или управляющих обменом информацией с другими блоками. Это позволяет в итоге отображения всех алгоритмов на структуру операционной части АЛУ получить определенный набор микроопераций и логических условий АЛУ.

После реализации всех операций необходимо изобразить микропрограммы операций, представив их в виде граф-схем. При

этом рекомендуется придерживаться обозначений микроопераций принятых в [1, с.119 и далее].

Разработка функциональной схемы одного разряда АЛУ

Составлением этой схемы завершается разработка операционной части АЛУ (в некоторых вариантах задания необходимо представить еще и принципиальную схему). При построении данной схемы исходным является перечень узлов АЛУ, выполняемых в них микроопераций и взаимосвязей между узлами, что определяется на предшествующих этапах.

Схема строится для любого разряда АЛУ (можно взять типовой средний разряд, представляющий любую цифру числа с фиксированной и плавающей запятыми). На схеме должны быть изображены по одному (одноименному) разряду всех регистров и по одному разряду всех узлов передачи и преобразования: сумматора, сдвигателя, мультиплексоров и т. д. Допускается представлять разряды регистров в виде триггеров, однако для оценки значений показателей u_1 и u_2 необходимо представлять, сколько элементов и в какой взаимосвязи входят в триггер. Допускается не показывать на схеме общие узлы АЛУ типа счетчика циклов, умножения, триггеров состояния, схем анализа. Однако необходимо представлять себе их устройство и связи для получения оценок $u_2 \dots u_4$. Каждый изображенный узел должен обеспечивать выполнение всех возложенных на него в соответствии с составленными микропрограммами микроопераций.

При построении функциональной схемы следует ориентироваться на тип, число входов, нагрузочные способности и другие особенности элементов, имеющих в серии, определенной заданием на курсовой проект. В этом случае переход к принципиальной схеме АЛУ (для тех вариантов задания, где это требуется) значительно упрощается.

При работе над данной частью проекта (как и при составлении принципиальной схемы) рекомендуется использовать программы систем автоматизированного проектирования.

2. Разработка структуры процессора, алгоритмов и микропрограмм выполнения команд

При работе над данным разделом проекта необходимо проработать следующие вопросы:

- а) определение списка команд;
- б) определение структуры и формата команд;
- в) разработка структурной схемы процессора и микропрограмм выполнения команд;
- г) оценка времени выполнения команд.

В результате должна быть получена полная структурная схема процессора, микропрограммы выполнения команд и оценки времени их выполнения.

Определение списка команд

Список команд в курсовом проекте определяется заданием. Однако в большинстве случаев, помимо заданного набора команд, для организации функционирования процессора требуется введение некоторых дополнительных команд. Так, задание не предполагает реализацию непосредственной адресации и адресации относительно счетчика команд во всех типах команд. Поэтому может оказаться целесообразным введение дополнительных команд, например логической операции, выполняемой с непосредственным операндом, или команды перехода с адресацией относительно счетчика команд. Аналогичное замечание относится к косвенной и косвенной регистровой адресациям при наличии в процессоре кроме них еще и индексной (двойной индексной адресации). То же относится к ситуации с автоиндексированием, т. е. общим требованием является реализация каждого из заданных типов адресации хотя бы в одной команде, а индексной или косвенной в каждой команде.

Определение структуры и формата команд

Структура команды определяется на основе адресности процессора, способов адресации и типов команд. В задании на курсовой проект частично оговорены элементы структуры команды. Явно не указанные требования реализуются студентом по собственному ус-

мотрению, т.е. в данной части проекта критерий эффективности для оценки различных вариантов можно не использовать. Структуру и формат команд первоначально целесообразно разрабатывать отдельно для каждой группы команд. При этом в качестве прототипа можно использовать системы команд известных ЭВМ, процессоров семейства Intel и др. [4, 5, 6].

Разрабатывая формат команды для согласования разрядности команд с разрядностью процессора необходимо получить команды, кратные по длине слову памяти или байту. При этом команда может занимать более одного слова памяти. В таких случаях желательно получить формат команды, кратный целому числу слов или (что менее удобно) полуслову памяти. Это позволит упростить процедуру выборки команды из памяти. В любом случае следует руководствоваться соображениями получения минимальной длины команды.

Структура всех разрабатываемых команд зависит от типа операции. Не требуется выбирать единственную структуру и формат для всех типов команд, но следует стремиться к унификации. (Здесь под форматом команды, в отличие от структуры, понимается распределение функциональных полей команды по позициям кода команды.)

При разработке формата команд для различных групп команд следует вначале перечислить все функциональные поля команд и определить их разрядность. Предполагается, что структура группы логических и арифметических команд одинакова.

Количество разрядов в поле кода операции выбирается достаточным для того, чтобы закодировать все команды составленного списка команд. С учетом возможного расширения списка команд по отношению к заданию для поля кода операции достаточно 4...5 двоичных разрядов. При кодировании операций целесообразно близкие по характеру операции обозначать кодами, имеющими одинаковые части, например сложение – 0100, вычитание – 0101, умножение – 0110, деление – 0111. Тогда код операции, первые два разряда которого равны 01, указывает арифметическую команду.

Количество разрядов в адресном поле команды зависит от адресности (количества адресов) команды, способов адресации, объема оперативной памяти.

При прямой адресации каждого байта памяти поле адреса должно иметь $\log_2(E_{OP})$ разрядов, где E_{OP} – емкость оперативной памяти в байтах. Если память адресуется словами, то разрядность адреса будет $\log_2(E_{OP}/l)$ разрядов, где l – количество байтов в слове памяти. Задание на курсовой проект предусматривает в качестве обязательной обработку слов. Однако в проекте могут быть реализованы операции и над байтами. Кроме того, в командах управления прямой адрес должен содержать столько разрядов, чтобы позволить адресовать команду, к которой осуществляется переход, при любом ее расположении в памяти.

Для индексной адресации в адресном поле каждого адреса должно быть выделено $\log_2(N_{РОН})$ разрядов для указания номера РОН, содержащего базисный адрес (или индекс), где $N_{РОН}$ – количество РОН. Поле смещения обычно имеет 7...12 разрядов, хотя можно отвести под него и большее число разрядов. Такой же вид имеет адрес и в случае автоиндексирования. При адресации с двойным индексированием в поле адреса должен указываться адрес еще одного РОН, т. е. нужно еще $\log_2(N_{РОН})$ разрядов.

В случае косвенной адресации разрядность адреса может быть равной разрядности прямого адреса или, при необходимости, меньшей (допуская косвенную адресацию только через часть ОП). Разрядность адреса в случае косвенной регистровой адресации составляет $\log_2(N_{РОН})$ разрядов. Поле непосредственного операнда можно выбрать в пределах от 8 разрядов (байта) до полного слова.

При адресации относительно счетчика команд поле адреса обычно совпадает с разрядностью смещения в индексной адресации.

Способ адресации может указываться явно (с помощью специальных признаков разрядов в адресной части команды) или неявно – с помощью кода операции, задающего, например, операцию с непосредственной адресацией или команду перехода с адресацией относительно счетчика команд. При наличии индексной (двойной индексной) и прямой адресации также можно использовать неявное указание способа адресации. Например, нулевое содержание поля РОН индексного (базисного) адреса будет указывать на прямую адресацию.

Признаки адресации, задаваемые явно, могут быть отдельными для каждого адреса команды или общими для всех адресов.

Расположение признаков в поле команды и их количество выбирается самостоятельно.

Некоторые возможные варианты форматов двухадресных команд приведены на рис 8, где КОП – код операции, П – признак способа адресации, Б – номер РОН, содержащего базисный адрес, И – номер РОН, содержащего индексный адрес, СМ – смещение.

Различны варианты расположения (общего и отдельных для каждого адреса) признаков адресации показаны на рис. 8, а, б, в. Вариант неявного задания способа адресации, при котором ненулевые значения полей И и Б определяют двойное индексирование, при нулевом поле Б – задана индексная адресация, а при $I = B = 0$ – прямая адресация, приведен на рис. 8, г. На рис. 8, д показан пример команды загрузки РОН. Эта команда может иметь разрядность меньше, чем разрядность арифметической команды, так как при одинаковой их разрядности часть поля второго адреса не будет использоваться.

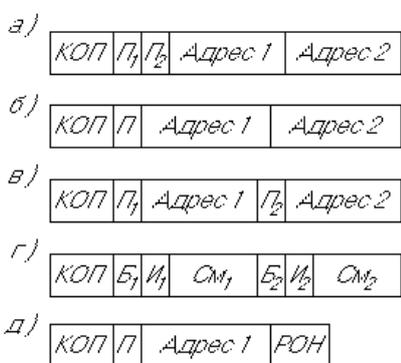


Рис. 8. Варианты форматов двухадресных команд

Одновременно с определением структуры и формата команд в курсовом проекте следует дать функциональное описание порядка выполнения команд.

Разработка структурной схемы процессора и микропрограмм выполнения команд

Исходными данными для выполнения этой части проекта являются описание команд процессора и ограничения задания на состав аппаратных средств. В результате должны быть построены операционная часть устройства управления процессора, микропрограммы выполнения его команд и полная структурная схема процессора.

Как и в разделе "АЛУ", выполнение этой части проекта сводится к последовательной детализации общего порядка выполнения

команд вплоть до уровня микроопераций и выбору необходимых для этого узлов процессора. Но в отличие от разработки АЛУ здесь допускается осуществлять выбор вариантов без оценки их по заданному критерию эффективности.

Первоначально общий порядок выполнения команд можно представить так, как показано на рис. 9, разделив команды на две большие группы: команды, определяющие какие-либо преобразова-

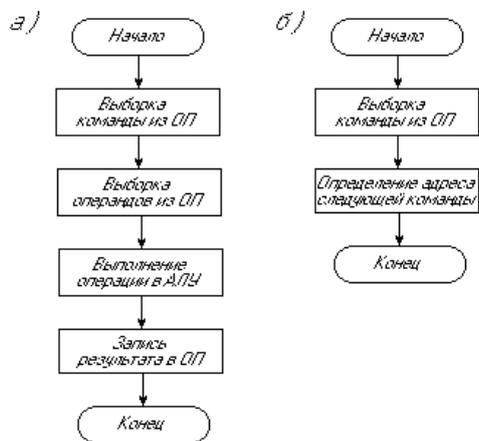


Рис. 9. Общий порядок выполнения операционных команд (а) и команд управления (б)

ния данных – операционные команды (рис. 9, а), и команды, определяющие порядок выполнения программы – команды (передачи) управления (рис. 9, б). Отметим, что в одноадресных операционных командах запись результата не является обязательной.

Различные этапы выполнения команд, соответствующие рис. 9, имеют следующие особенности.

А. Выборка команды. Для того, чтобы выполнить некоторую команду, ее необходимо выбрать из ОП и переслать в регистр команд. Разрядность регистра команд обычно должна позволять разместить команду полностью, хотя в ряде случаев это необязательно имеет место.

При естественном порядке следования команд программа занимает в ОП область последовательно расположенных ячеек. Если длина каждой команды равна слову памяти, то каждая команда размещается в одной ячейке памяти. При этом выборка команды производится за одно обращение к ОП и не вызывает затруднений.

В ряде вариантов длина команды может превосходить разрядность слова памяти. Если при этом длина команды кратна длине слова памяти (состоит из целого количества слов), то она размеща-

ется в соответствующем количестве ячеек ОП, а выборка ее производится за столько обращений к ОП, сколько слов занимает команда.

Возможны случаи, когда длина команды не кратна длине слова памяти. Тогда размещение каждой команды программы, начиная только с первого (старшего) разряда ячейки памяти, приводит к бесполезному расходу объема ОП, так как остаются неиспользуемыми часть ячеек памяти, в которых располагаются окончания команд. Во избежание этого приходится располагать команды по разному относительно границ ячеек памяти.

Например, если разрядность слова памяти составляет 4 байта, а длина команды - 6 байтов, то возможны два варианта размещения команды в ОП (рис. 10), от чего зависит и порядок выборки команды. Определить, какой вариант расположения команды имеет место, можно по младшим разрядам (разряду) адреса команды. Действительно, последние два разряда адреса байта в рассматриваемом примере для любой ячейки памяти равны 00 для самого левого байта, 01 - для второго байта, 10 - для третьего байта и 11 - для правого байта. В случае адресации полуслов будет иметь всегда четный (оканчивающийся нулем) адрес, а правое - нечетный.

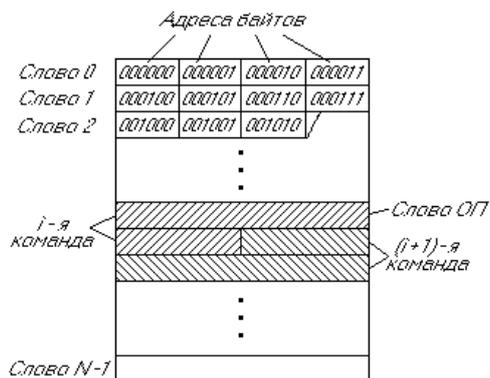


Рис. 10. Размещение команд программы в ОП при длине команды, равной 1,5 слова

жущим только последние 4 байта этой команды. Но следует проверить, не является ли выполняемая команда командой, осуществляю-

Учитывая, что при выборке последних двух байтов i -й команды из ОП выбираются первые два байта $(i+1)$ -й команды, размещенные в той же ячейке памяти, для повышения быстродействия процессора целесообразно сохранить эти два байта на некотором буферном регистре. Это позволит осуществить выборку $(i+1)$ -й команды за одно обращение к памяти за словом, содержащим только последние 4 байта этой команды.

щей передачу управления к другому участку программы, так как использование содержимого буферного регистра для формирования следующей команды в таком случае невозможно.

Б. Выборка операндов, выполнение операции и завершение команды Основное функциональное содержание этапа выборки операндов при выполнении операционных команд связано с формированием исполнительных адресов операндов.

В вариантах с индексной (двойной индексной) адресацией, автоиндексированием и адресацией относительно счетчика команд при наличии адресного сумматора целесообразно использовать сумматор комбинационного типа. В отсутствие адресного сумматора требуемые преобразования адресов осуществляются на сумматоре АЛУ.

При выполнении операционных команд после формирования исполнительного адреса операнд извлекается из ОП и посылается в регистр (сумматор) АЛУ, определяемый микропрограммой выполнения операции в АЛУ. Если размещение операндов для всех операций одинаково (см. с. 20), то эта посылка во всех случаях одна и та же.

Микропрограмма выполнения всех этих действий состоит в основном из микроопераций пересылок, в структуре процессора для ее реализации используются лишь те узлы, которые оговорены заданием (РОН, адресный сумматор) или уже введены в него при разработке АЛУ и этапа выборки команд (регистр команд, возможно буфер команд).

В командах (передачи) управления после соответствующего анализа условий перехода аналогичным образом и фактически по той же микропрограмме производится формирование исполнительного адреса, который указывает расположение в ОП следующей команды.

В микропрограмме выполнения команд процессором для команд арифметических и логических операций после выборки и засылки операндов в АЛУ собственно выполнение операции обозначается в виде одного блока. Это делается как для операций, микропрограмм выполнения которых составлены в разделе "АЛУ", так и для тех арифметических и логических операций, перечисленных в

списке команд задания на курсовой проект, микропрограммы выполнения которых составлять не требовалось.

Примерный общий вид микропрограммы выполнения команд процессором представлен на рис. 11 (Конечно, в пояснительной записке все блоки, кроме вышеназванных, должны быть раскрыты до уровня микроопераций). Причем на рис. 11 предполагается, что коды операций сложения, вычитания и умножения равны 00100, 00101 и 00110 соответственно, коды операций команд управления начинаются с единицы, а операционных команд - с нуля. Команды посылочных операций (чтение, пересылка, загрузка регистра, запись и др.) в самостоятельную группу не выделены, так как по общей схеме выполнения их можно уподобить операционным командам, в которых преобразования операндов (пересылаемых данных) отсутствуют. Обычно выполнение команд пересылки также сопровождаются формированием признаков результата, вырабатываемых в результате анализа посылаемого слова.

При завершении выполнения команды перед выборкой очередной команды обычно осуществляется проверка на отсутствие условий, препятствующих нормальному продолжению программы. К таким условиям относится наличие запросов на прерывание, задание на пульте режима командной обработки ("цикл") и др. Из этих условий одни могут требовать вмешательства программных средств операционной системы или аппаратной обработки, другие - прекращения выполнения команд процессором. Для учета таких ситуаций в конце микропрограммы выполнения команд рекомендуется поместить две условных вершины так, как показано на рис. 11: одну, анализирующую наличие особых ситуаций (вершина "ОС = 1"), и вторую, выделяющую те ситуации, когда возникли условия, требующие останова процессора (вершина "УО = 1"). Отметим, что команда останова процессора заданием не предусмотрена. Операционную вершину "Обработка особого случая", показанную на рис. 11, раскрыть на уровне микроопераций не требуется.

Таким образом, в результате выполнения этой части проекта будут получены полная структурная схема процессора и микропрограмма выполнения всех его команд. Образец выполнения одного из возможных вариантов структурной схемы процессора приведен в прил. 3.

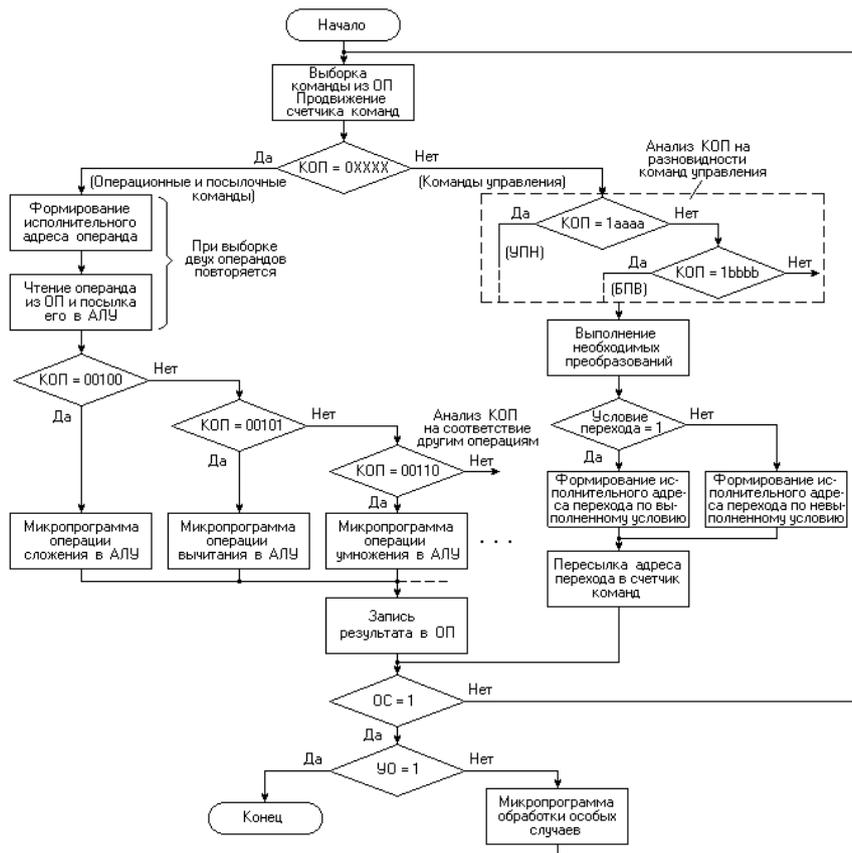


Рис. 11. Примерный укрупненный вид микропрограммы выполнения команд процессором

Оценка времени выполнения команд

Эта оценка дается в тактах и производится непосредственно по граф-схемам составленных микропрограмм подсчетом общего количества или определением среднего количества операторов (тактов) микропрограммы по известным правилам (см., например, [10]).

При наличии в микропрограммах разветвлений следует принимать вероятности выполнения каждой ветви разветвления одинаковыми, если нет иной информации.

Необходимо отдельно подсчитать количество обращений к памяти, производимых в процессе выполнения команд при различных способах адресации. Для операционных команд, микропрограммы выполнения которых в АЛУ составлять по заданию не требовалось, оценки времени не даются.

Результаты оценки следует представить в виде таблицы, содержащей наименования команд, способ адресации, среднее количество тактов, затрачиваемое на выполнение команды и среднее количество обращений к ОП в команде. В вариантах с блоком микропрограммного управления после выполнения следующего раздела оценку времени следует осуществить повторно.

3. Синтез управляющего автомата АЛУ или блока микропрограммного управления процессора

В зависимости от варианта задания в данном разделе проекта требуется разработать либо автомат управления АЛУ, либо устройство микропрограммного управления процессором.

Для вариантов, в которых необходимо синтезировать автомат управления АЛУ, синтез осуществляется только для тех операций, для которых составлены микропрограммы их выполнения в АЛУ.

Для этих операций строится объединенная граф-схема микропрограммы, по которой и синтезируется автомат. Синтез рекомендуется проводить каноническим методом, изложенным, например, в [2]. Для исключения гонок в автомате следует использовать двойную память. Необходимо ввести шину установки памяти автомата в исходное состояние.

В общем случае рекомендуется использовать в качестве элементов памяти D- или RS-триггеры и строить автомат по схеме автомата Мили. При этом требуется проанализировать, выполняется ли для всех переходов автомата условие независимости выходных сигналов, вызывающих переход, от выходных сигналов, вырабатываемых на этом переходе. В тех случаях, когда подобной независи-

мости нет, необходимо принять меры к устранению возможности некорректной работы автомата. В пояснительной записке следует привести таблицу переходов и сигналов, для которых наблюдается указанная зависимость.

Кодирование состояний автомата допускается выполнить произвольно.

Данный этап работы завершается составлением функциональной схемы автомата в элементном базисе, соответствующем заданной серии элементов. При построении автомата рекомендуется использовать программы синтеза управляющих автоматов для ЭВМ.

Для вариантов, в которых необходимо синтезировать блок микропрограммного управления микропроцессора, прежде всего, следует выписать все различные операторные вершины (без повторов) из составленных микропрограмм выполнения операций в АЛУ и микропрограмм выполнения команд процессором. Этот список и будет представлять собой перечень всех микрокоманд, отличающихся только операционной частью. Данный список вместе с ранее составленными списками микроопераций АЛУ и процессора позволяет построить известными методами [12] матрицу совместимости микроопераций. Ввиду большой размерности такой матрицы заданием предусматривается раздельное построение матриц совместимости микроопераций процессора. (При этом возможное в ряде вариантов использование микроопераций АЛУ для формирования исполнительных адресов не оказывается существенным).

После получения матриц совместимости методом прямого включения [2] строятся подмножества совместимых микроопераций. Допускается построение подмножеств совместимых микроопераций по функциональным признакам, содержательному смыслу микроопераций, однако количество подмножеств при этом может быть слишком большим.

Формат операционной части микрокоманд определяется количеством полученных подмножеств и числом микроопераций в каждом из них. Кодирование микроопераций в пределах подмножеств осуществляется произвольным образом. Формат адресной части микрокоманд (МК) может быть выбран простейшим: с одним полем номера условия и полноразрядным (без сегментации) адресом следующей МК так, как показано на рис.12, где ОЧ МК операционная

часть МК; КМО – код микрооперации; АЧ МК – адресная часть МК; Нусл – номер логического условия; АСМК – адрес следующей МК.

Количество разрядов в поле номера условия должно быть

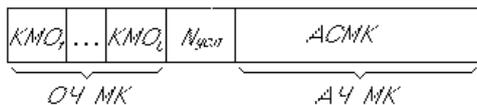


Рис. 12. Формат микрокоманды

достаточным для нумерации всех условий, используемых в составленных микропрограммах. Количество разрядов в адресе следующей МК определяется исходя из требований к емкости памяти микропрограмм, которая должна позволять разместить все составленные микропрограммы. Можно использовать более сложный формат МК, если он обеспечивает меньшее время выполнения микропрограммы. Примеры таких форматов приводятся в [2].

При размещении МК в памяти микропрограмм целесообразно начать с тех МК, которые следуют за условными вершинами, приписывая им адреса, отличающиеся значением младшего разряда в соответствии со значением условия, определяющего разветвление. Окончательное размещение микропрограмм производится в пояснительной записке в виде таблицы, отражающей содержимое ячеек памяти (постоянного ЗУ) микропрограмм.

В заключение данного раздела следует привести структурную схему блока микропрограммного управления, описать назначение его узлов и дать временную диаграмму их взаимодействия.

4. Разработка принципиальной схемы и конструкции устройства

В различных вариантах задания предусматривается разработка принципиальной схемы одного из следующих блоков: устройства микропрограммного управления процессора, арифметико-логического устройства или автомата управления АЛУ. Принципиальные схемы составляются на основе разработанных в предыдущих частях проекта соответствующих схем с учетом особенностей заданного элементного базиса.

Для вариантов, связанных с составлением принципиальной схемы блока микропрограммного управления, указаны две серии микросхем, одна из которых (К556) – это микросхемы постоянного

ЗУ, а другая (К133 или К531) используется для построения остальных узлов блока, за исключением задающего генератора, который представлять в схеме не обязательно.

Две серии микросхем К555 и К556 указаны и для одного из вариантов, в котором строится принципиальная схема автомата управления АЛУ. В этом варианте комбинационная часть автомата должна быть построена на ПЛИМ К556РТ1 (РТ2), а память – на микросхемах серии К555.

При построении принципиальной схемы АЛУ необходимо составить ее для того разряда, для которого была построена функциональная схема, и для всех общих узлов операционной части АЛУ, не вошедших в схему этого разряда (счетчиков, схем анализа и др.). Все остальные разряды АЛУ на принципиальной схеме раскрывать не следует, считая их одинаковыми с тем разрядом, схема которого составлена (для уменьшения объема работы).

На этапе разработки конструкции должна быть спроектирована плата, на которой размещаются элементы составленной принципиальной схемы (в случае принципиальной схемы АЛУ это может быть часть схемы на 20...40 микросхем).

В проекте должен быть представлен сборочный чертеж платы, а также чертеж общего вида процессора, конструктивное оформление которого должно быть реализовано в виде модуля органов управления, монтируемого в стойку и подключаемого к остальной части системы через соединители (разъемы), либо в виде ТЭЗа. Конструктив должен позволять размещать в себе все платы процессора, количество которых легко определить, исходя из заданной элементной базы.

На данном этапе выполнения проекта целесообразно широко использовать пакеты программ САПР логического и конструктивно-технологического проектирования дискретных устройств.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основной:

1. *Цилькер, Б.Я.* Организация ЭВМ и систем: учебник для вузов / Б.Я. Цилькер, С.А. Орлов. – СПб.: Питер, 2011 – 688 с..

2. *Таненбаум, Э.* Архитектура компьютера / Э. Таненбаум. – СПб.: Питер, 2007. – 844 с.

Дополнительный:

3. *Гук, М.* Аппаратные средства IBM PC: энциклопедия / М. Гук. – 3-е изд. – СПб.: Питер, 2008. - 1072 с.

4. *Бройдо, В.Л.* Архитектура ЭВМ и систем: учебник для вузов / В.Л. Бройдо, О.П. Ильина. – СПб.: Питер, 2009. – 720 с.

5. *Копейкин, М.В.* Организация ЭВМ и систем: (память ЭВМ): учеб. пособие / М.В. Копейкин, В.В. Спиридонов, Е.О. Шумова. – СПб.: Изд-во СЗТУ, 2004. - 153 с.

6. *Копейкин, М.В.* Управление ЭВМ: учеб. пособие / М.В. Копейкин, В.Я. Пашкин, В.В. Спиридонов. - Л.: СЗПИ, 1988. - 84 с.

7. *Организация ЭВМ и систем:* учеб.-метод. комплекс / сост.: М.В. Копейкин, В.В. Спиридонов, Е.О. Шумова. – СПб.: Изд-во СЗТУ, 2009. - 187 с.

8. *Спиридонов, В.В.* Проектирование структур АЛУ: учеб. пособие / В.В. Спиридонов. – СПб.: СЗПИ, 1992. - 84 с.

9. *Хамахер К.* Организация ЭВМ. 5-е изд / *К. Хамахер, Э. Вранешиг, С. Заки.* Организация ЭВМ. 5-е изд. – СПб.: Питер, 2003. - 848 с.

Образец оформления задания

ЗАДАНИЕ
на курсовой проект по дисциплине
“ЭВМ и периферийные устройства”

студенту _____
(фамилия, имя, отчество) (шифр)

Данные для проектирования

1. Организация сумматора АЛУ
2. Микропрограммы АЛУ
3. Реализация микрооперации сдвига
4. Разрядность данных АЛУ и ОП
5. Представление чисел с фиксированной запятой
6. Представление порядков чисел с плавающей запятой
7. Разрабатываемое устройство управления
8. Относительные частоты выполнения операций
9. Элементный базис
10. Разрабатываемая принципиальная схема
11. Объем адресуемой ОП (байт)
12. Способы адресации
13. Наличие адресного сумматора
14. Тип связей в устройстве управления
15. Количество адресов в команде
16. Набор команд
17. Размещение РОН
18. Количество РОН
19. Коэффициенты критерия эффективности

Задание выдано " __ " _____ 201_ г.

Срок представления проекта для проверки " ____ " _____ 201_ г.

Студент _____
(подпись)

Руководитель проекта _____
(подпись)

ПРИЛОЖЕНИЕ 2

Варианты задания на курсовой проект

Таблица 1

Предпоследняя цифра шифра	Организация сумматора АЛУ	Операции, для которых составляются микропрограммы выполнения в АЛУ	Реализация микрооперации сдвига	Разрядность данных АЛУ и ОП (байт)	Представление чисел с фиксированной запятой	Представление порядков чисел с плавающей запятой	Разрабатываемое устройство управления	Относительная частота выполнения операций				Элементный базис (серия микросхем)	Разрабатываемая принципиальная схема
								a_1	a_2	a_3	a_4		
0	Н	С _ф , М2 _ф , СЕ, ИЛ	С	2	Ц	нет	П	0.3	0.3	0.2	0.2	K133 K556	БМПУ
1	К	С _ф , Д _ф , СЕ, И	С	2	Д	нет	П	0.4	0.1	0.3	0.2	K531 K556	БМПУ
2	Н	С _ф , С _п , СЛ, Л	С	4	Л	3	А	0.4	0.3	0.15	0.15	K555	АЛУ
3	К	М _ф , М _п , СА, И	С	4	Д	3	А	0.3	0.3	0.3	0.1	K555 K556	МПА АЛУ
4	Н	Д _ф , Д _п , СЛ, ИЛ	С	2	Д	Х	А	0.3	0.2	0.25	0.25	K564	АЛУ
5	К	С _ф , М _ф , СЕ, ИЛ	Л	2	Ц	нет	П	0.5	0.1	0.3	0.1	K133 K556	БМПУ
6	Н	С _ф , Д _ф , СЕ, И	Л	4	Д	нет	П	0.4	0.1	0.3	0.2	K531 K556	БМПУ
7	К	С _ф , С _п , СЛ, Л	Л	4	Л	Х	А	0.3	0.25	0.15	0.3	K555	МПА АЛУ
8	Н	М2 _ф , М _п , СА, И	Л	2	Ц	Х	А	0.4	0.2	0.2	0.2	K176	АЛУ
9	К	Д _ф , Д _п , СЛ, НЕ	Л	4	Ц	3	П	0.4	0.3	0.2	0.1	K564	АЛУ

Таблица 2

Последняя цифра шифра	Объем адресной ОП (байт)	Способы адресации	Наличие адресного сумматора	Тип связей в устройстве управления	Количество адресов в команде	Набор команд	Размещение РОН	Количество РОН	Коэффициенты критерия эффективности				
									k_1	k_2	k_3	k_4	k_5
0	128К	И, АИ, К	есть	М	1	Ч, З, С, В, М, Д, (СЕ, ВЕ), И, ИЛ, СА, СЛ, УПН, БП, УЦ, БПВ	ОП	8	0,006	0,0004	0,1	0,004	0,025
1	64К	КР, Н, АИ	нет	М	2	П, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, ЗгР, ЗпР, БП, БПВ, УПП, УЦ	Пр	16	0,01	0,0004	0,1	0,004	0,025
2	32К	П, КР, И	есть	М	3	П, ПГ, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, ЗгР, ЗпР, БП, БПВ, УПО, УЦ	ПР	8	0,003	0,0004	0,5	0,004	0,025
3	512К	И, Д, АИ	нет	Н	2	П, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, БП, УПО, БПВ, УЦ	ОП	16	0,016	0,0005	0,2	0,008	0,04
4	256К	К, Н, И	есть	Н	3	П, ПГ, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, БП, УПП, УЦ	ОП	8	0,009	0,0008	0,2	0,008	0,04

Окончание таблицы 2

Последняя цифра шифра	Объем адресной ОЦ (байт)	Способы адресации	Наличие адресного сумматора	Тип связей в устройстве управления	Количество адресов в команде	Набор команд	Размещение ЮО	Количество ЮО	Коэффициенты критерия эффективности			
									k_1	k_2	k_3	k_4
5	128К	И, Д, КР	нет	Н	1	Ч, З, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, ЗгР, ЗпР, БП, УПП, БПВ, УЦ	Пр	16	0,004 0,05	0,001	0,1	0,008
6	64К	И, Н, П	есть	М	2	П, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, ЗгР, ЗпР, БП, БПВ, УПН, УЦ	Пр	8	0,002 0,03	0,0005	0,4	0,002
7	32К	П, К, И	нет	М	3	П, ПГ, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, БП, БПВ, УПО, УЦ	ОП	16	0,004 0,03	0,0003	0,2	0,002
8	256К	К, И, ОС	есть	Н	2	П, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, БП, УПП, БПВ, УЦ	ОП	8	0,006 0,04	0,0003	0,5	0,002
9	128К	И, Д, Н	есть	Н	3	П, ПГ, С, В, М, Д, (СЕ, ВЕ), И, Л, ИЛ, СА, СЛ, БП, БПВ, УПН, УЦ	ОП	16	0,01 0,05	0,001	0,1	0,005

* Команды СЕ и ВЕ имеются только в вариантах с предпоследней цифрой шифра 0, 1, 5 и 6. Команды С, В, М, Д в вариантах с предпоследней цифрой шифра 2, 3, 4, 7, 8 и 9 имеются в двух модификациях: с фиксированной и плавающей запятой.

Условные обозначения

Организация сумматора АЛУ: К – комбинационный, Н – накапливающий.

Микрооперации АЛУ; индекс "ф" – с фиксированной запятой, "п" – с плавающей запятой, М2 – умножение с одновременным анализом двух разрядов множителя (остальные обозначения см. "Набор команд").

Реализация микрооперации сдвига: С – в специальном блоке, Л – в любом блоке.

Представление чисел с фиксированной запятой: Ц – целое, Д – дробное.

Представление порядков чисел с плавающей запятой: З – со знаком, Х – без знака (характеристика).

Разрабатываемое устройство управления: П – блок микропрограммного управления процессора, А – автомат управления АЛУ.

Разрабатываемая принципиальная схема: БМПУ – блок микропрограммного управления, МПА АЛУ – микропрограммный автомат управления АЛУ.

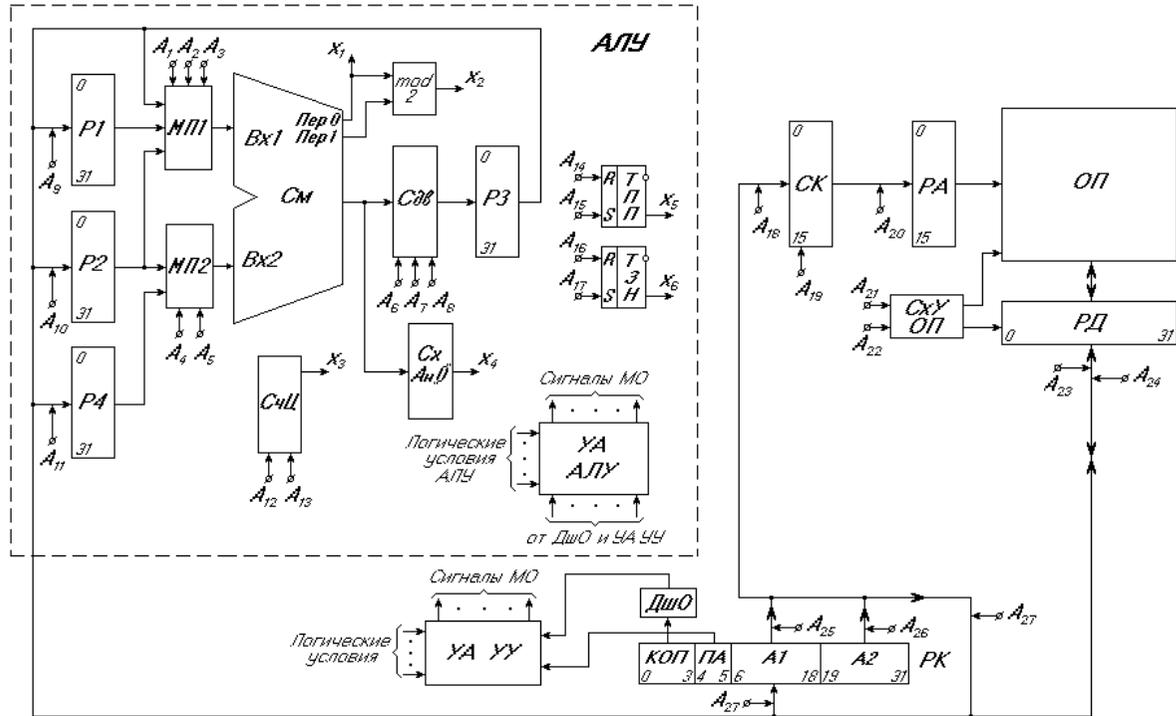
Способы адресации: АИ – автоиндексирование, Д – двойная индексация, И – индексная, К – косвенная, КР – косвенная регистровая, Н – непосредственная, ОС – относительно счетчика команд, П – прямая.

Тип связей в устройстве управления: М – магистральные, Н – непосредственные.

Набор команд: Ч – чтение, З – запись, П – пересылка слова, ПГ – пересылка группы слов, С – сложение, В – вычитание, М – умножение, Д – деление, СЕ – десятичное сложение, ВЕ – десятичное вычитание, И – логическое умножение, Л – логическое сложение, ИЛ – сумма по модулю два (исключающее ИЛИ), СА – сдвиг арифметический СЛ – сдвиг логический, ЗгР – загрузка РОН, ЗпР – запись содержимого РОН в ОП, БП – безусловный переход, БПВ – безусловный переход с возвратом (вызов подпрограммы), УПН – условный переход по нулевому результату предыдущего действия (РПД), УПО – условный переход по отрицательному РПД, УПП – условный переход по положительному РПД, УЦ – управление циклом (условный переход по индексу или счетчику).

Размещение РОН: ОП – в оперативной памяти, Пр – в процессоре.

Образец оформления
варианта структурной схемы процессора



Содержание

Введение	3
Общие указания	3
1. Содержание курсового проекта	3
2. Основные этапы работы над курсовым проектом.....	6
Структурная организация проектируемого процессора	6
1. Арифметико-логическое устройство	7
2. Оперативная память	11
3. Устройство управления процессора	14
Порядок выполнения основных этапов разработки.....	18
1. Определение структуры АЛУ и разработка алгоритмов выполнения заданных операций	18
2. Разработка структуры процессора, алгоритмов и микропрограмм выполнения команд.....	26
3. Синтез управляющего автомата АЛУ или блока микропрограммного управления процессора	35
4. Разработка принципиальной схемы и конструкции устройства	37
Библиографический список.....	38
Приложение 1	40
Приложение 2	41
Условные обозначения.....	44
Приложение 3	45

ЭВМ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА

*Методические указания
к выполнению курсового проекта
для студентов направления подготовки 230100*

Составители: М.В.Копейкин, В.В.Спиридонов, Е.О.Шумова

Печатается с оригинал макета, подготовленного кафедрой
информационных систем и вычислительной техники

Ответственный за выпуск *В.В.Спиридонов*