МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Кафедра автоматизированных систем управления

Т.М.АЛЕКСАНДРИДИ, Ю.Ю.ИСАЕВ, И.А.МОРОЗОВ

ОРГАНИЗАЦИЯ ЭВМ И СИСТЕМ
Часть 3
Функциональные устройства

Учебное пособие

Утверждено
в качестве учебного пособия
редсоветом МАДИ (ГТУ)

МОСКВА 2007

УДК-681.32
ББК-32.97

Александриди Т.М., Исаев Ю.Ю., Морозов И.А. Организация ЭВМ и систем. Часть 3. Функциональные устройства: Учебное пособие МАДИ (ГТУ). - М., 2007.- 45 с.

Рецензенты:
Доц. кафедры АСОиУ Московского Государственного университета приборостроения и информатики П.К. Круг;
доц. кафедры ИСЭМ Российской экономической академии им. Г.В. Плеханова А.И. Коников.

В учебном пособии представлены основные функциональные устройства, входящие в состав каждой ЭВМ. Рассматриваются назначение, структура, принцип действия арифметического устройства, устройства управления, оперативного запоминающего устройства.

Учебное пособие написано в соответствии с аналогичными разделами стандартной программы по дисциплине «Организация ЭВМ и систем» для специальности «Автоматизированные системы обработки информации и управления».

ВВЕДЕНИЕ

Настоящее учебное пособие предназначено для студентов направления 552800 «Информатика и вычислительная техника» и направления 654600 «Информатика и вычислительная техника» по специальности 230102 «Автоматизированные системы обработки информации и управления (АС)» и используется при изучении теоретических разделов дисциплины «Организация ЭВМ и систем», выполнении лабораторных работ, а также при курсовом и дипломном проектировании. В пособии рассмотрены назначение, принцип действия, структурные и функциональные схемы основных функциональных устройств, входящих в состав ЭВМ.

Изложены методика и примеры проектирования основных блоков арифметического устройства и устройства управления.

1. Арифметическо-логическое устройство

Арифметическо-логическое устройство (АЛУ) – блок ЭВМ, который служит для выполнения арифметических и логических операций.

Выполняемые в АЛУ операции можно условно разделить на следующие группы:

- операции двоичной арифметики для чисел с фиксированной точкой;
- операции двоичной арифметики для чисел с плавающей точкой;
- операции десятичной арифметики;
- операции индексной арифметики;
- операции специальной арифметики;
- операции над логическими кодами;
- операции над алфавитно-кодовыми полями.

Современные ЭВМ общего назначения обычно реализуют операции всех приведенных выше групп, а малые и специализированные ЭВМ часто не выполняют операции над

числами с плавающей точкой, операции десятичной арифметики и операции над алфавитно-цифровыми полями.

При необходимости эти операции выполняются специальными подпрограммами.

К арифметическим операциям относятся сложение, вычитание, умножение и деление чисел в двоичных и двоично-десятичных кодах, с фиксированной точкой и плавающей точкой.

Группу логических операций составляют операции “инверсии” (логическое НЕ), “дизъюнкции” (логическое ИЛИ) и “конъюнкции” (логическое И) над многоразрядными двоичными кодами, сравнение кодов на равенство.

Специальные арифметические операции включают в себя нормализацию, арифметический сдвиг (сдвигаются только цифровые разряды, знаковый разряд остается на месте), логический сдвиг (знаковый разряд сдвигается вместе с цифровыми разрядами).

В зависимости от кодов, используемых для представления операндов, АЛУ делятся на последовательные и параллельные. В последовательных АЛУ операнды представляются в последовательном коде, а операции производятся последовательно во времени над их отдельными разрядами. В параллельных АЛУ операнды представлены параллельными кодами и операции совершаются одновременно над всеми разрядами операндов.

АЛУ последовательного действия в настоящее время практически не применяются из-за их низкого быстродействия. Все дальнейшее изложение относится к АЛУ параллельного действия.

По способу представления чисел различают АЛУ:

- для чисел с фиксированной точкой;
- для чисел с плавающей точкой;
- для десятичных чисел.

По своим функциям АЛУ является операционным блоком, выполняющим микрооперации (МО), обеспечивающие прием из других устройств (например, памяти) операндов, их преобразование

и выдачу результатов преобразования в другие устройства. Каждая МО реализуется физическим управляющим сигналом (УС). Генерируемая устройством управления последовательность УС определяется кодом операции команды.

По структуре различают АЛУ:

- с жесткой структурой;
- с гибкой (магистральной) структурой.

1.1. АЛУ с жесткой структурой

Жесткая структура отличается тем, что связи между регистрами и функциональными узлами, выполняющими преобразование информации, однозначно реализованы при изготовлении АЛУ и не могут быть изменены в процессе эксплуатации. Эти связи соответствуют полному набору алгоритмов выполнения вычислительных и логических операций в данном АЛУ.

На рис.1.1 представлена упрощенная функциональная схема АЛУ с жесткой структурой. Основными функциональными узлами (ФУ) в схеме являются три регистра РГ (1:3) и сумматор. В регистры РГ(1) и РГ(2) поступают из памяти исходные операнды, в РГ(3) образуются результаты операций. Кроме того, любой из регистров может также выполнять операции сдвига.

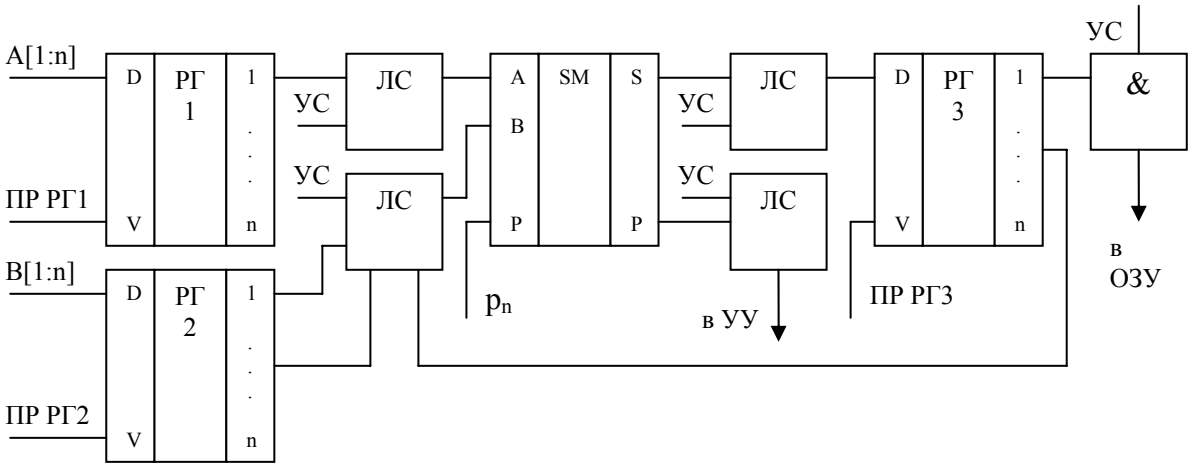

Рис. 1.1. Функциональная схема АЛУ с жесткой структурой

Связи между регистрами и сумматором реализуются с помощью логических схем (ЛС), структура которых соответствует набору алгоритмов, выполняемых данным АЛУ. Процессы обмена информацией и ее обработка осуществляются за счет поступления управляющих сигналов (УС), которые формируются в устройстве управления (УУ) в соответствии с выполняемым алгоритмом. Преимуществом данной структуры является относительная простота, вследствие чего АЛУ данного типа используются в основном при построении специализированных ЭВМ.

1.2. АЛУ с гибкой структурой

Идея гибкой структуры состоит в том, что в АЛУ имеется несколько одинаковых и равнозначных регистров, которые умеют только принимать и выдавать информацию. При этом любой из этих регистров может участвовать в каждой операции как в качестве источника, так и в качестве приемника.

На рис.1.2 представлена упрощенная функциональная схема АЛУ с гибкой или, как ее еще называют, магистральной структурой.

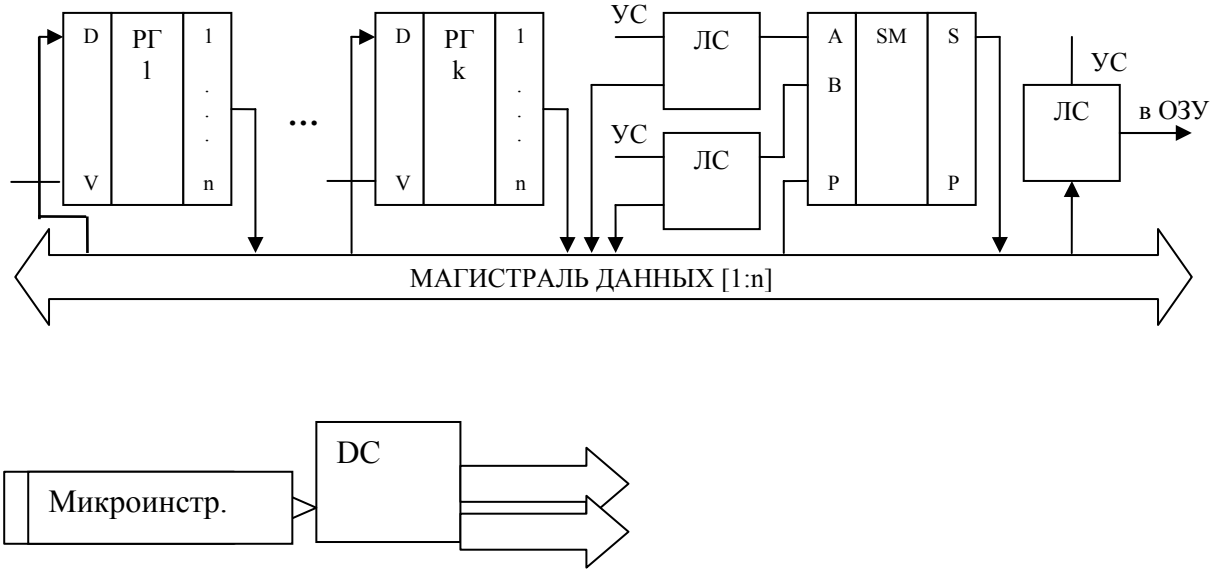

Рис.1.2. Функциональная схема АЛУ с гибкой структурой

Одной из отличительных особенностей этой структуры является наличие единой двунаправленной магистрали данных, по которой данные поступают из памяти и любого регистра, а также выдаются в память и регистры. В состав АЛУ входят несколько регистров РГ(1:к) общего назначения (РОН), а также сумматор SM и дешифратор DC. Кроме того, может быть единый блок, реализующий операции сдвигов (на схеме не показан). Функционирование АЛУ определяется микроинструкциями, которые поступают из УУ на дешифратор DC. На выходе DC появляются УС, поступающие на регистры, и ЛС - на входах и выходе сумматора в соответствии с выполняемым алгоритмом.

Преимуществом данной структуры является возможность в процессе отладки или эксплуатации изменять алгоритмы выполнения операций или добавлять новые. В большинстве современных ЭВМ используются АЛУ магистрального типа.

2. Устройство управления (УУ)

Предназначено для обеспечения работы всех узлов и устройств ЭВМ в соответствии с выполняемой программой.

Основные функции УУ:

- организация пуска и остановки ЭВМ;
- определение очередности выбора команд из оперативной памяти;
- формирование физических адресов операндов;
- формирование последовательности управляющих сигналов для выполнения арифметических, логических и иных операций при выполнении программы.

Обеспечение работы ЭВМ в различных режимах:

- автоматически выполняемая программа;
- пошаговое выполнение программы;
- режим прерывания;
- режим прямого доступа к памяти;

и т. д.

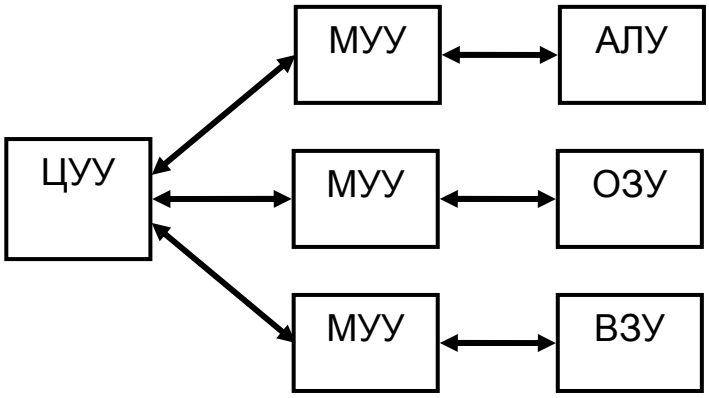

Рис. 2.1. Обобщенная структура УУ

ЦУУ - центральное УУ, которое выполняет основные функции по реализации программы.

МУУ – местное УУ (находится при каждом из устройств, входящих в состав ЭВМ). Оно реализует специфические алгоритмы, соответствующие принципам действия различных внешних устройств.

Рис.2.2. Иерархическая структура понятий при постановке задач на ЭВМ

Программа – кодированная запись алгоритма.

Команда – кодированная запись вычислительной, логической или иной операции. В устройствах ЭВМ команда физически выполняется с помощью микроопераций.

Микрооперация – некоторое простейшее преобразование данных, например, прием байта данных в регистр, инверсия переменной и т.д.

2.1. Структура команды

Структура любой команды состоит из нескольких полей, основные из которых представлены в табл.2.1.

Таблица 2.1

Код операции	Адресная часть	Код модификации

Код операции (коп) указывает, какая именно операция (арифметическая, логическая и т.п.) выполняется. Если поле «коп» состоит из «m» двоичных разрядов, то в данной системе команд может содержаться до 2^m различных операций. Любой операции соответствует свой код операции. В адресной части указываются адреса операндов.

Операнд – число, команда, код, над которыми выполняется указанная операция.

Коды модификации используются при вычислении физических адресов данных.

В зависимости от структуры адресной части различают следующие виды команд:

- безадресные (например, остановка);
- одноадресные;
- двухадресные;
- трехадресные;
- многоадресные.

Рассмотрим на примере программирования простейшего алгебраического выражения (2.1) взаимосвязь между адресностью команды и структурой соответствующего фрагмента программы.

$$\text{Дано } \gamma = \alpha + \beta \,. \qquad (2.1)$$

Данные и результат размещаются, например, в следующих условных адресах оперативного запоминающего устройства (ОЗУ).

$$\left.\begin{array}{l} \alpha \to M1 \\ \beta \to M2 \\ \gamma \to M3 \end{array}\right\} \quad адреса$$

Составим фрагмент программы для выражения (2.1) при одноадресной структуре команды. Как видно из табл. 2.2, фрагмент программы занимает три ячейки в памяти.

Таблица 2.2

Адрес	Команда	Пояснения
К	чтение М1	$\alpha \to АЛУ$
К+1	сложение М2	$\alpha + \beta$
К+2	запись М3	$\gamma - M3$

В таблице 2.3 представлена структура двухадресной команды.

Таблица 2.3

Коп	М1	М2

Фрагмент программы для выражения (2.1) при двухадресной структуре команды состоит всего из одной команды:

Сложение (М1) + (М2) → (М2). (2.2)

Как следует из (2.2), результат сложения заносится по адресу одного из слагаемых.

В табл. 2.4 представлена структура трехадресной команды.

Фрагмент программы для выражения (2.1) состоит при трехадресной структуре также всего из одной команды (2.3).

Таблица 2.4

Коп	М1	М 2	М 3

Сложение (М1) + (М2) → (М3). (2.3)

Как видно, при двух- и трехадресной структурах команды фрагмент программы для выражения (2.1) занимает только одну ячейку памяти.

Однако если принять, что ячейка оперативной памяти содержит ограниченное количество разрядов, то предпочтение имеет одноадресная структура команды, так как при этом можно адресовать больший объем памяти.

В современных ЭВМ принята переменная структура команды (за основу взята двухадресная) в зависимости от назначения команды.

Одной из основных функций ЦУУ является определение очередности выбора команды при выполнении программы или определение следующего адреса команды (САК). В каждой программе содержатся два типа участков: линейная часть и разветвление.

На линейной части программы команды располагаются по последовательным адресам оперативной памяти. При этом адрес следующей команды формируется с помощью счетчика. Фрагмент программы в таблице 2.2 является линейной частью программы. Разветвления в программе происходят при выполнении безусловного перехода (БП) и условного перехода (УП).

В командах БП и УП адрес следующей команды указан в выполняемой команде.

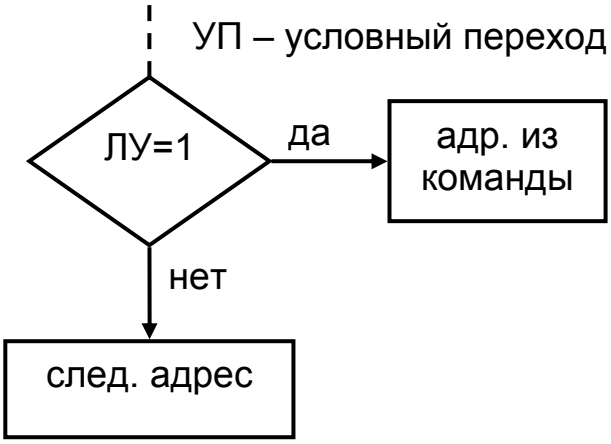

Рис.2.3. Фрагмент алгоритма программы с разветвлением

Ниже приведен алгебраический пример, для решения которого потребуется написать программу с разветвлением.

$\gamma = \alpha + \beta$

если $\gamma > 0$, то $\alpha = 2\gamma$;

если $\gamma \le 0$, то $\alpha = 2\gamma + \beta$.

2.2. Центральное устройство управления (ЦУУ)

В состав ЦУУ (рис. 2.4) входят следующие основные функциональные узлы (ФУ) и блоки.

ПУ – пульт управления с клавиатурой, посредством которой производят пуск и останов ЭВМ, задают режимы работы, вводят программу и данные.

БС - блок синхронизации, содержащий генератор импульсов, который начинает работать сразу после включения питания и выдает бесконечную последовательность синхроимпульсов. Из этих импульсов формируются в УУ все последовательности управляющих сигналов (УС), которые обеспечивают функционирование ЭВМ.

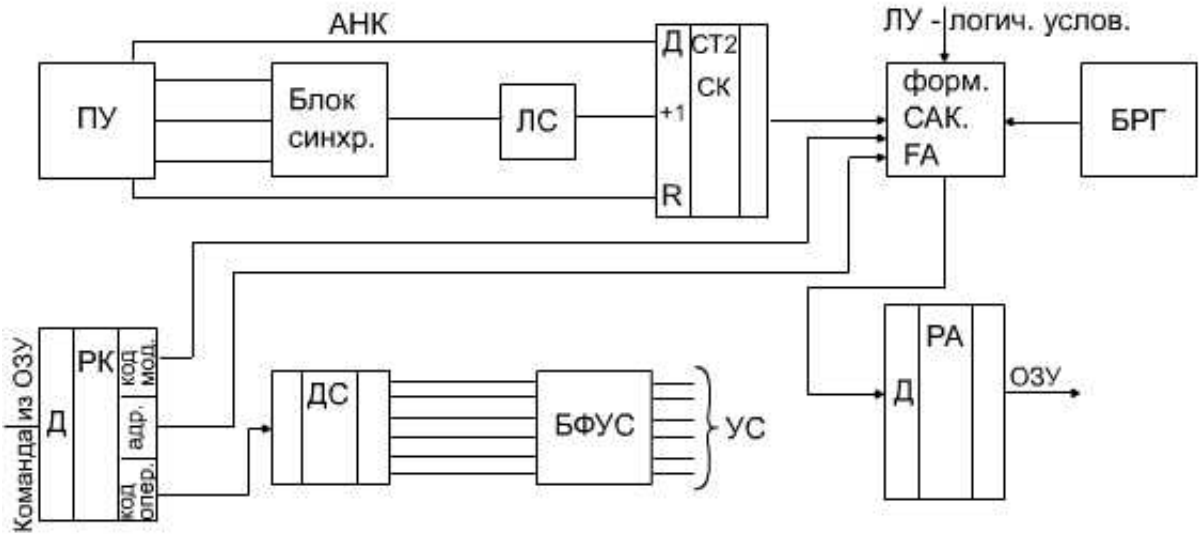

Рис. 2.4. Упрощенная структура ЦУУ

СК – счетчик команд (программный счетчик). При включении ЭВМ на вход R поступает с ПУ сигнал, который устанавливает СК:=0.

Далее с ПУ в СК поступает адрес начальной команды (АНК), с которого начинается выполнение некоторой программы.

Импульсы на счетный вход СК поступают из БС через логическую схему (ЛС), и если выполняется линейный участок программы, то при этом формируется следующий адрес команды (САК).

РК - регистр команды, в него из оперативного запоминающего устройства (ОЗУ) принимается очередная команда, которая будет выполняться.

Форм. САК - логическая схема, которая обеспечивает образование физического адреса (ФА) следующей команды при разветвлении программы. В эту логическую схему поступает информация из РК (адрес и коды модификации), а также логические условия (ЛУ) из АЛУ.

РА- регистр адреса, в него поступает сформированный ФА оперативного запоминающего устройства (ОЗУ), по которому происходит считывание команды и прием ее в РК.

Код операции попадает на дешифратор, где определяется, какая именно команда будет выполняться.

БФУС – блок формирования управляющих сигналов. Для каждой выполняемой команды он формирует свою последовательность управляющих сигналов – микроопераций (МО) , которые поступают на различные устройства и узлы ЭВМ.

На линейном участке программы САК образуется в СК при добавлении единицы к его содержимому после выполнения каждой команды.

При выполнении команд:

- безусловного (БП) перехода САК равен содержимому адресной части в команде;
- условного (УП) перехода САК формируется в зависимости от логических условий, которые могут поступать из АЛУ или других устройств.

Если ЛУ = 0, то САК = СК + 1.

Если ЛУ = 1, то САК = <адр.>, т.е. содержимому адресной части команды УП.

БРГ – блок регистров, который содержит программно-доступные и программно-недоступные регистры.

Программно-недоступные регистры используются для реализации различных вычислительных алгоритмов.

Программно-доступные регистры (регистры сегментов, регистры индикации и т.д.). С их помощью реализуется тот или иной тип организации ОЗУ.

2.3. Блок формирования управляющих сигналов

Существуют 2 основные структуры БФУС:

- жесткая;
- микропрограммная.

2.3.1. Блок формирования управляющих сигналов с жесткой структурой

Отличается тем, что формирование последовательностей управляющих сигналов осуществляется с помощью логических схем.

На рис. 2.5 представлена упрощенная функциональная схема БФУС с жесткой структурой. Основными ФУ являются двоичный счетчик тактов, дешифратор тактов (ДС), триггер управления (ТУ), формирователь управляющих сигналов (УС).

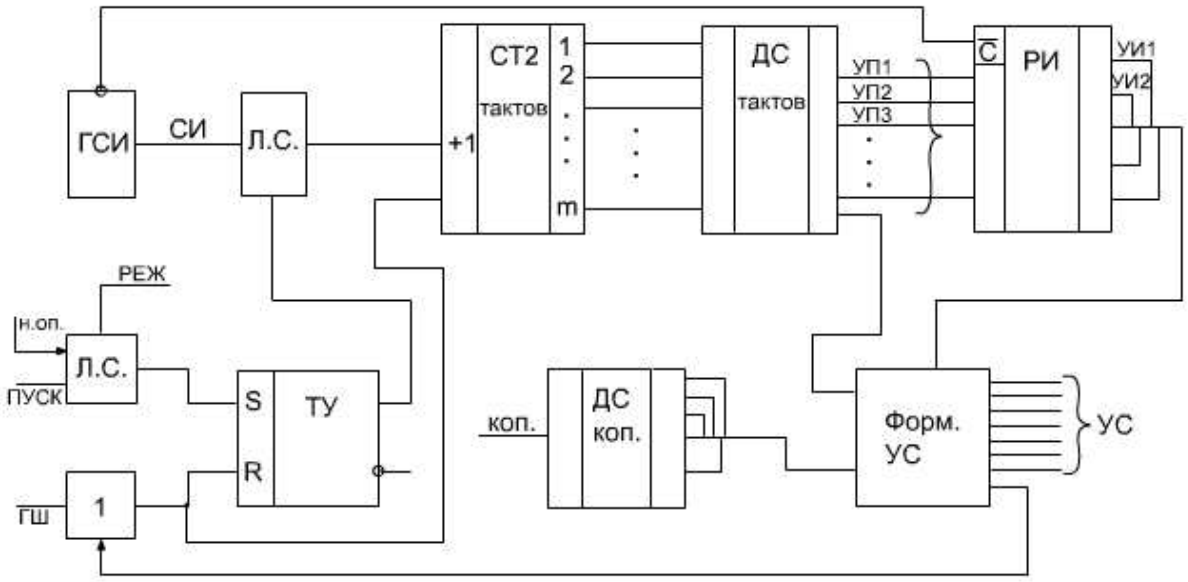

Рис. 2.5. Функциональная схема БФУС

На рис. 2.5 представлена упрощенная функциональная схема БФУС, в состав которой входят следующие функциональные узлы (ФУ) и логические схемы (ЛС):

ГСИ – генератор синхроимпульсов;

СТ2 тактов - двоичный счетчик, который подсчитывает количество тактов (импульсов), прошедших за время выполнения данной вычислительной или логической операции;

ТУ- триггер управления, который управляет логической схемой (ЛС) на входе счетчика;

ДС тактов - дешифратор, на выходах которого образуется стандартная последовательность управляющих потенциалов (УП), распределенных во времени (по номерам тактов) и в пространстве (по номерам контактов на выходе дешифратора);

ДС КОП – дешифратор кода операции определяет, какая команда будет выполняться.

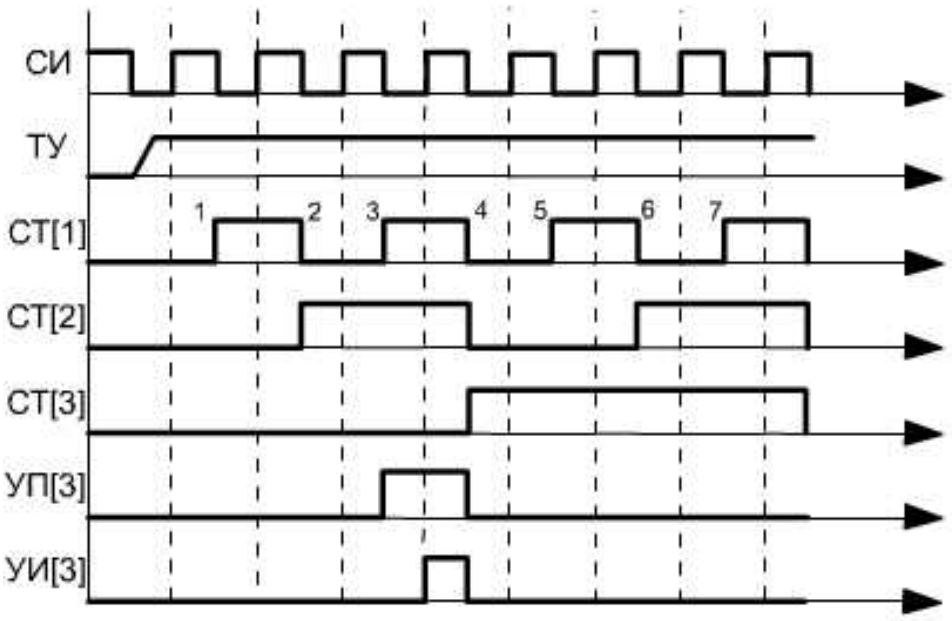

Рис. 2.6. Временная диаграмма стандартных сигналов БФУС

РИ – распределитель импульсов, который распределяет синхроимпульсы в пространстве и времени, например, управляющий импульс УИ1 появится на выходе №1 в первом такте.

При включении ЭВМ или после окончания выполнения очередной команды ТУ:=0 и СТ2:=0.

Если на вход подается сигнал Н.ОП (начало операции) или ПУСК, то ТУ устанавливается в 1, и на счетчик тактов поступают импульсы. При этом, как было показано выше, формируются две стандартных последовательности управляющих потенциалов УП(1:m) и управляющих импульсов от УИ(1:m), где m - количество тактов, необходимых для выполнения данной операции. Далее УП(1:m) и УИ(1:m) поступают на формирователь управляющих

сигналов (Форм. УС), на выходе которого образуется нужная последовательность УС, реализующая данную операцию на ЭВМ.

Для каждой команды в составе Форм. УС существует своя ЛС для формирования соответствующей последовательности УС.

2.3.2. Основные этапы проектирования БФУС жесткого типа

1) содержательное рассмотрение задачи, т.е. математическая постановка, числовые примеры;
2) построение обобщенной схемы алгоритма;
3) разработка структуры АЛУ и списка микроопераций с соответствующими характеристиками;
4) составление кодированной схемы алгоритма, т.е. привязанной к разработанной структуре АЛУ;
5) построение временной диаграммы управляющих сигналов;
6) разработка логической схемы БФУС.

2.3.3. Пример построения фрагмента БФУС с жесткой структурой для операции «сравнение модулей двух чисел»

Математическая постановка операции

$|A| < 1, |B| < 1$.

Если $|A| \geq |B|$, то признак $\omega = 1$.

Если $|A| < |B|$, то признак $\omega = 0$.

Вычитание выполняется по алгоритму ПД.

$\alpha = |A| - |B|$

$\alpha' = |A| + [-|B|]_д \quad [-|B|]_д = [-|B|]_0 + p_n$.

В результате вычитания образуется перенос из старшего разряда сумматора:

если $p_0 = 1$, то $\alpha' > 0$, $|A| \geq |B|$;

если $p_0 = 0$, то $\alpha' < 0$, $|A| < |B|$.

Строим обобщенную схему алгоритма.

На рис. 2.7 представлена так называемая обобщенная схема алгоритма.

Это означает, что данный алгоритм может быть реализован на любой структуре АЛУ. Tw - триггер, в котором должно запоминаться значение признака w.

Принимаем, что строим АЛУ параллельного действия с непосредственными связями.

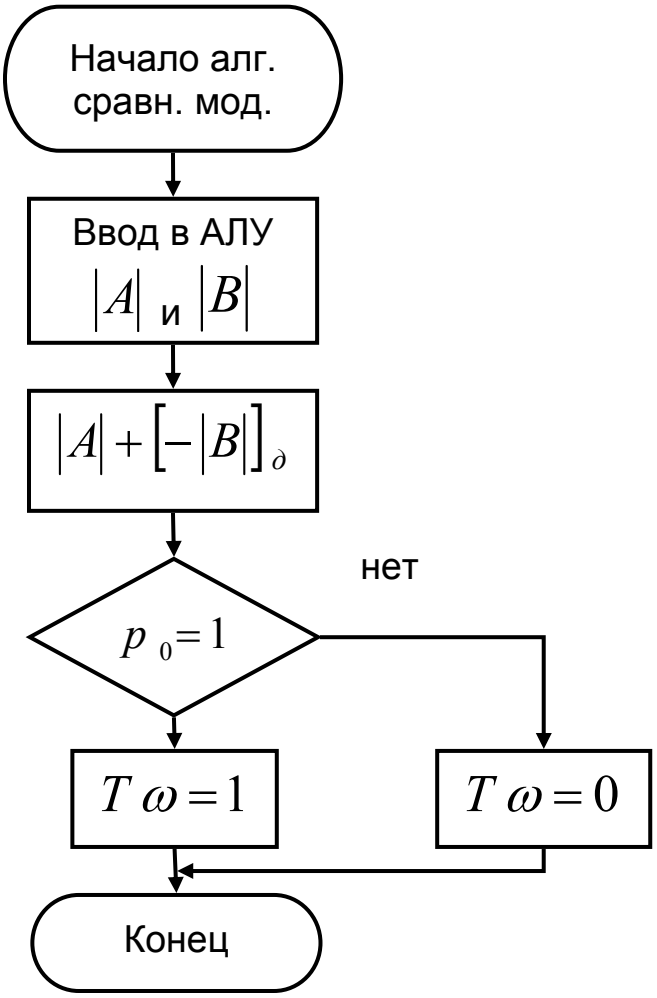

Рис. 2.7. Обобщенная схема алгоритма

Соответствующий фрагмент АЛУ представлен на рис. 2.8. Он состоит из двух регистров, сумматора параллельного действия, триггера и двух ЛС.

В РГ1 на вход «Д» поступает из ОЗУ первый операнд «А» (ОП1[0:n]), который будет принят в регистр микрооперацией ПР РГ1. Соответственно в РГ2 микрооперацией ПР РГ2 будет принят второй операнд «В» (ОП[0:n]).

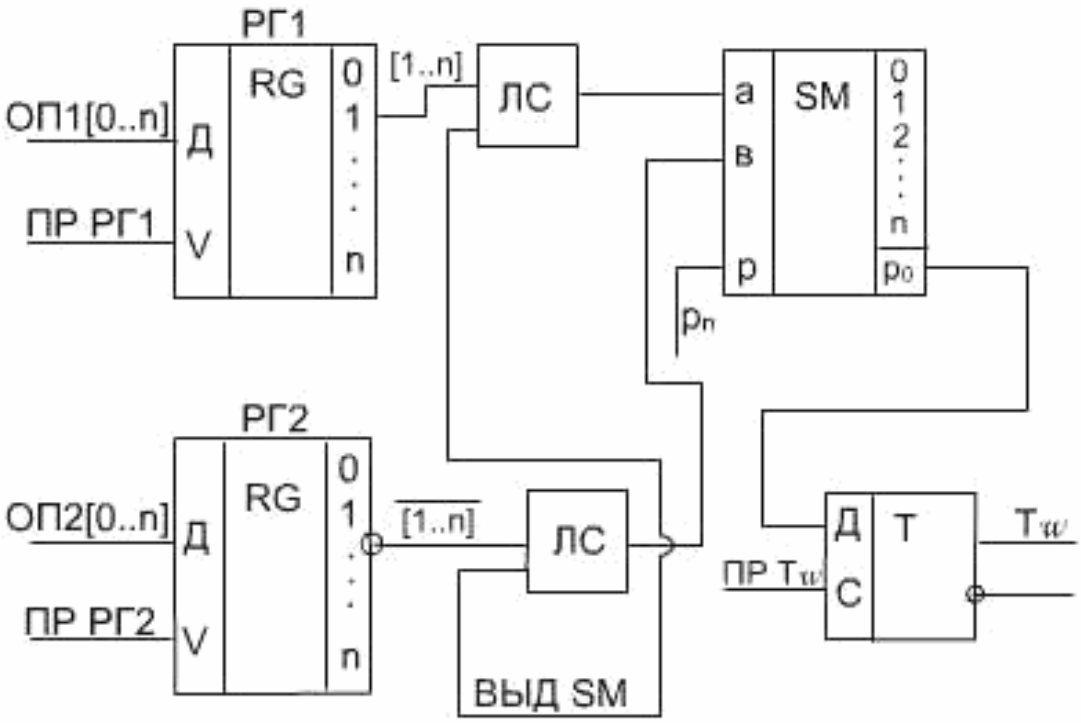

Рис. 2.8. Фрагмент АЛУ

Далее микрооперацией ВЫД SM на входы сумматора подаются «А» в прямом коде и «В» в обратном коде. Кроме того, на третий вход сумматора поступает микрооперация «перенос в младший разряд».

После этого на выходе сумматора образуются сумма (на рис. 2.8 не обозначена) и перенос из старшего разряда сумматора – p0.

Затем микрооперацией ПРTw значение переноса «p0» принимается в триггер переноса Tw. Значение Tw является

логическим признаком (флагом), который определит переход к следующей команде при разветвлении программы.

На фрагменте АЛУ на рис. 2.8 отображено всего два регистра, поскольку для реализации заданного алгоритма этого достаточно, так как не требуется определить значение разности модулей. Однако в реальном АЛУ всегда имеется третий регистр, в который принимается результат операции.

Строим кодированную схему алгоритма (рис. 2.9), “привязанную” к фрагменту АЛУ на рис. 2.8. Эта схема алгоритма содержит последовательность микроопераций (МО), которые необходимо подать на АЛУ, чтобы выполнить заданную операцию.

Напоминаем, что в действительности каждая МО выполняется соответствующим физическим УС.

Приступим к построению ЛС в составе БФУС, которая обеспечит формирование последовательности УС для выполнения рассматриваемой операции.

Принимаем, что у нас одноадресная структура команды.

Тогда получается, что операция «сравнение модулей» реализуется с помощью двух одноадресных команд (табл. 2.5).

Таблица 2.5

Чтение А	А → РГ1
Сравн. мод. В	$A + \left[- B\right]_0 + P_n$

Считаем, что операция «вычитание» выполняется по алгоритму ПД. После выполнения 1-й команды операнд «А» уже находится в РГ1 в прямом коде. Составляем фрагмент схемы БФУС для 2-й команды - «сравнение с модулем В».

В таблице 2.6 представлен перечень микроопераций с характеристиками. В 4-й графе таблицы указана длительность МО в тактах, в 5-й графе - № такта , с которого начинается данная МО. Принято, что чтение из ОЗУ происходит за три такта.

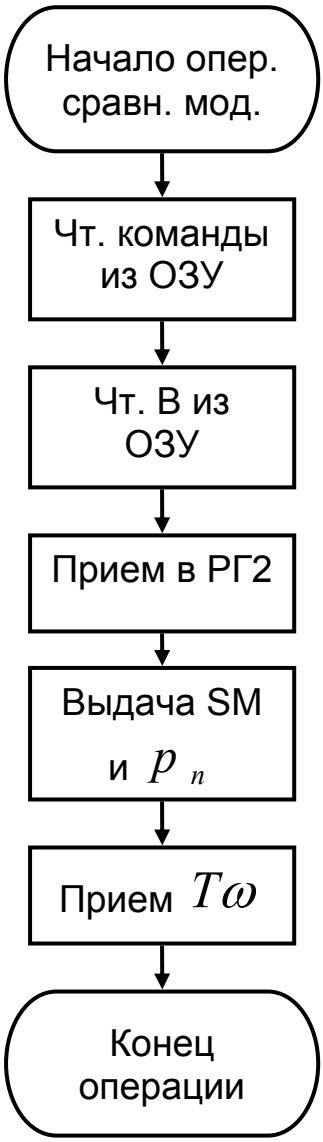

Рис. 2.9. Кодированная схема алгоритма

Таблица 2.6

№ МО	Наименование	Вид МО	Длит. (такт)	№ такта	Куда направл.
МО1	Чтение ОЗУ	импульс.	1	1-й	в ОЗУ
МО2	Чтение ОЗУ	импульс.	1	4-й	в ОЗУ
МО3	Прием в РГ2	импульс.	1	6-й	в АЛУ
МО4	Выдача SM, p_n	потенц.	4	7÷10-й	в АЛУ

Микрооперации «Выдача SM» и «p_n» являются потенциальными УС, и их длительность определяется временем суммирования. Пусть $T_{\Sigma} = 4$ такта.

МО5	Прием в $T\omega$	импульс.	1	10-й	в АЛУ
МО6	Конец операции	импульс.	1	11-й	в УУ

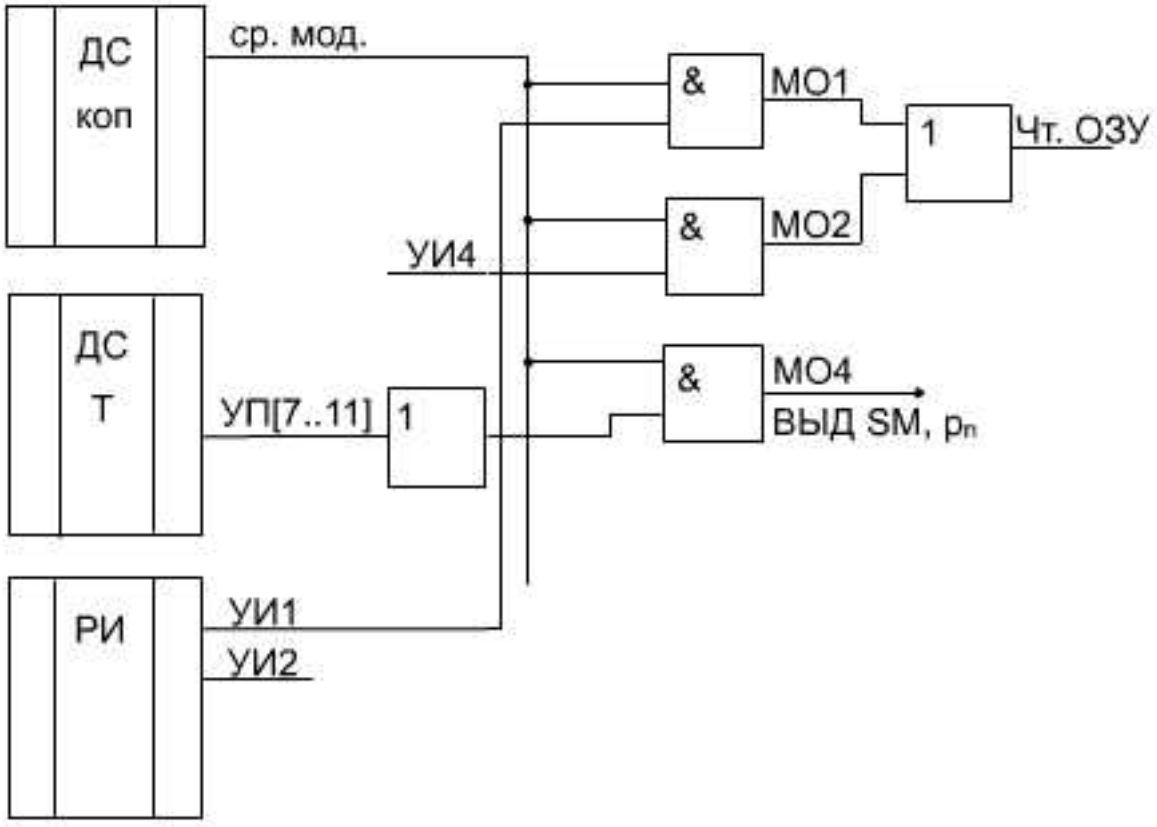

Рис. 2.10. Фрагмент БФУС с жесткой структурой

На рис. 2.10 представлен фрагмент логической схемы, которая формирует последовательность УС для реализации операции «сравнение модулей». Принцип построения ЛС достаточно прост и основан на характеристиках микроопераций, представленных табл. 2.6.

Например, МО2 относится к импульсному виду, длительность 1 такт, начинается с 4-го такта и существует, когда выполняется операция «ср. мод». Тогда ее формирование описывается следующим логическим выражением:

МО2:= «ср. мод» & УИ4,

МО4 - потенциальная переменная, существует в тактах (7-11):

МО4:= «ср. мод» & УП7 V УП8 V УП9 V УП10 V УП11.

Логические выражения для остальных МО можно составить самостоятельно.

Сформированная таким образом последовательность УС поступает на фрагмент АЛУ (рис.2.8) и обеспечивает выполнение микроопераций МО1-МО6. Результат операции сравнения соответствует состоянию триггера Tw.

2.3.4. Блок формирования управляющих сигналов микропрограммного типа

Принцип микропрограммного управления состоит в том, что алгоритм выполнения вычислительной или иной операции кодируется в виде микропрограммы аналогично тому, как алгоритм выполнения задачи кодируется в виде программы.

На рис. 2.11. представлена иерархия понятий, на которых основан принцип микропрограммного управления.

Микрокоманда - кодированная запись одной или нескольких микроопераций.

Микропрограмма – последовательность микрокоманд.

В ЭВМ микропрограммы для выполнения всех операций помещаются в постоянном запоминающем устройстве (ПЗУ).

Структура микропрограммы аналогична структуре программы (есть линейные и разветвленные участки). В табл. 2.7 представлен формат микрокоманды.

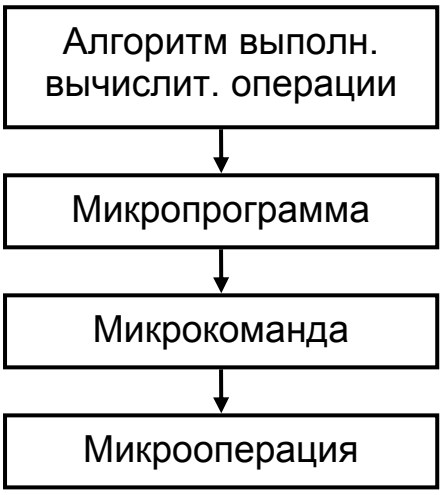

Рис. 2.11. Иерархия микропрограммирования

Таблица 2.7

МО	АМК	АР	У

МО – коды микроопераций, которые должны выполняться данной микрокомандой.

АМК – поле с адресом микрокоманды, которая должна выполняться после данной микрокоманды на линейном участке микропрограммы.

У – поле управления (коды управления), оно используется для организации разветвления в микропрограмме.

Если данная микрокоманда может повлечь разветвление, то следующий адрес микрокоманды (САМК) вычисляется на основании кодов АМК и кодов управления (У).

АР – коды адресов регистров, которые используются для управления АЛУ с магистральной структурой.

2.3.4.1. Кодирование поля МО

Существуют два способа кодирования:

- горизонтальный;
- вертикальный.

Пусть в АЛУ есть набор микроопераций:

1) гашение РГ1
2) ГШ РГ2
3) ГШ РГ3
4) ПР РГ1
5) ПР РГ2
6) контроль p_0
7) выдача SM.

Горизонтальное кодирование отличается тем, что в поле микроопераций должно быть предусмотрено столько разрядов, сколько всего имеется различных микроопераций (табл.2.8).

Таблица 2.8

Параметр	поле МО							
Наим. микрооп.	ГШ РГ1	ГШ РГ2	ГШ РГ3	ПР РГ1	ПР РГ2	контр. p_0	ВЫД ОЗУ	ВЫД SM
№ разр.	1	2	3	4	5	6	7	8
Н-р	1	1	1	0	0	0	0	0

“1” – данная микрооперация будет выполнена этой микрокомандой

Совместимые МО отличаются тем, что они могут быть выполнены одной микрокомандой.

Совместимые МО: ГШ РГ1, ГШ РГ2, ГШ РГ3.

Несовместимые МО – те, которые не могут быть закодированы в одной команде, т.е. не должны выполняться одновременно.

Например: ГШ РГ1, ПР РГ1.

Недостаток горизонтального кодирования состоит в том, что требуется слишком большое количество разрядов в поле МО, что приводит к увеличению разрядности ПЗУ.

Преимуществом этого метода является простота кодирования, наглядность.

Горизонтальное кодирование применяется в несложных специализированных цифровых управляющих устройствах.

Вертикальное кодирование отличается тем, что код поля микроопераций задается с помощью нескольких многоразрядных двоичных кодов (рис.2.12).

При кодировании микрооперации группируют в зависимости от принадлежности функциональным узлам АЛУ или другим устройствам. Такой способ используется в реальных ЭВМ.

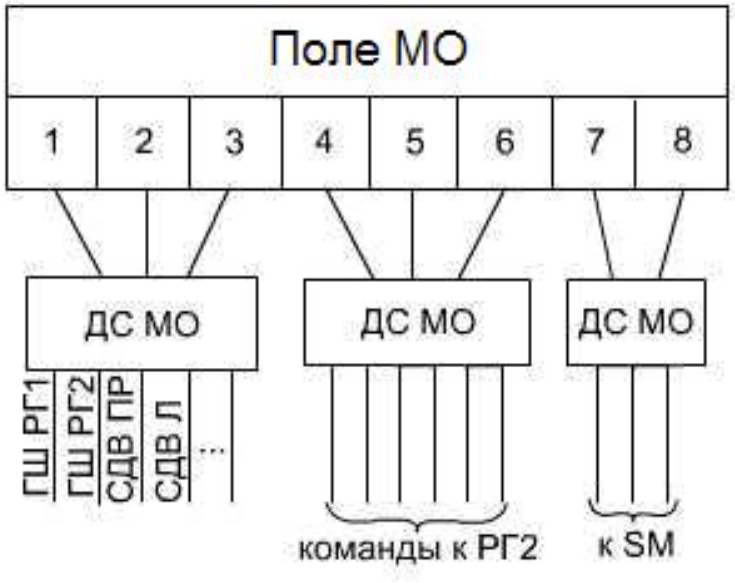

Рис. 2.12. Структура поля МО при вертикальном кодировании

2.3.4.2. Кодирование поля МО для АЛУ с магистральной структурой

В формате МО для такого АЛУ отдельно кодируется название МО, и в поле АР указывается адрес регистра, на который должна быть направлена данная МО.

Как следует из рис. 2.13, на выходе ДС МО может появиться одна из четырех МО: ГШ (гашение) - установка РГ в ноль, ПР-прием, ВЫД ОК – выдача обратным кодом, ВЫД ПК – выдача прямым кодом.

В поле АР закодированы номера регистров РГ[1:8].

Например, микрооперация ГШ РГ5 формируется так:

ГШ РГ5:= ГШ & РГ5.

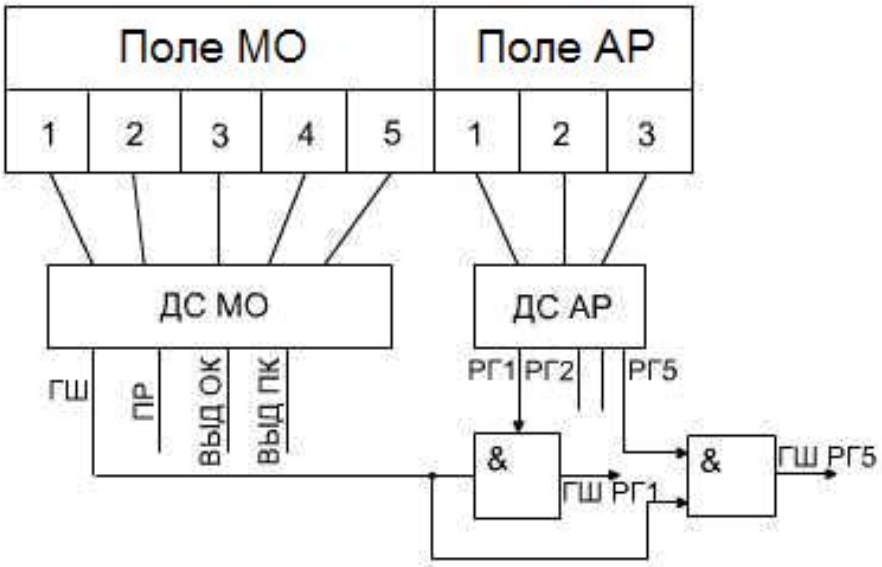

Рис. 2.13. Кодирование МО для АЛУ с магистральной структурой

Рассмотрим, как происходит формирование следующего адреса микрокоманды (САМК) при разветвлениях в микропрограмме. В этом случае формирование САМК определяется полями АМК и У.

Например, S = A ± B.

Обозначим: D: = 1 – вычитание, D: = 0 – сложение.

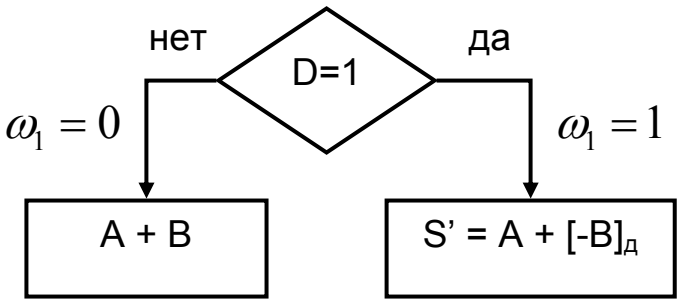

Рис. 2.14. Фрагмент алгоритма с разветвлением

На рис. 2.14 представлен фрагмент алгоритма с разветвлением.

Очевидно, что адрес САМК определяется значением переноса, т.е. признаком wi.

В общем случае следующий адрес микрокоманды при разветвлении определяется следующим логическим выражением:

САМК[1:k] := АМК[1:k] v y_1&ω_1 v y_2&ω_2 v ... v y_m&ω_m.

$k \geq m$;

y[1:m] – код управления (поле y);

ω[1..m] – логические условия.

В поле Уi ставится “1”, если данная микрокоманда определяет разветвление по логическому признаку wi.

Количество разрядов в поле У определяется количеством разветвлений, которые возможны в микропрограмме.

Пример: АМК[1:6] = 000101;

У[1:3] = 100;

ω[1:3] = 110;

следовательно, САМК[1:6] = 100101.

2.3.4.3. БФУС микропрограммного типа

На рис. 2.15 представлена упрощенная функциональная схема БФУС микропрограммного типа. Содержимое в РГкоп принимается при считывании команды из ПЗУ.

В ПЗУ находятся микропрограммы, обеспечивающие выполнение всей системы команд данной ЭВМ. На основе кода операции в ЛС формируется начальный адрес микропрограммы, который принимается в регистр адреса микрокоманды (РГ АМК).

По этому адресу из ПЗУ считывается микрокоманда и принимается в регистр микрокоманд (РГ МК). С выхода РГ МК коды поля МО поступают на дешифратор ДС МО и далее на формирователь.

Формирователь на основе кодов МО образует физические УС заданного типа длительности. УС поступают на все устройства ЭВМ в соответствии с выполняемой командой.

Коды полей АМК, У и логические признаки из АЛУ (флаги) поступают на схему «логика форм. САМК», где и образуется

следующий адрес микрокоманды. При завершении выполнения данной микропрограммы на выходе ДС МО появляется УС «конец операции», который инициирует выборку из ОЗУ следующей команды и, следовательно, новой микропрограммы.

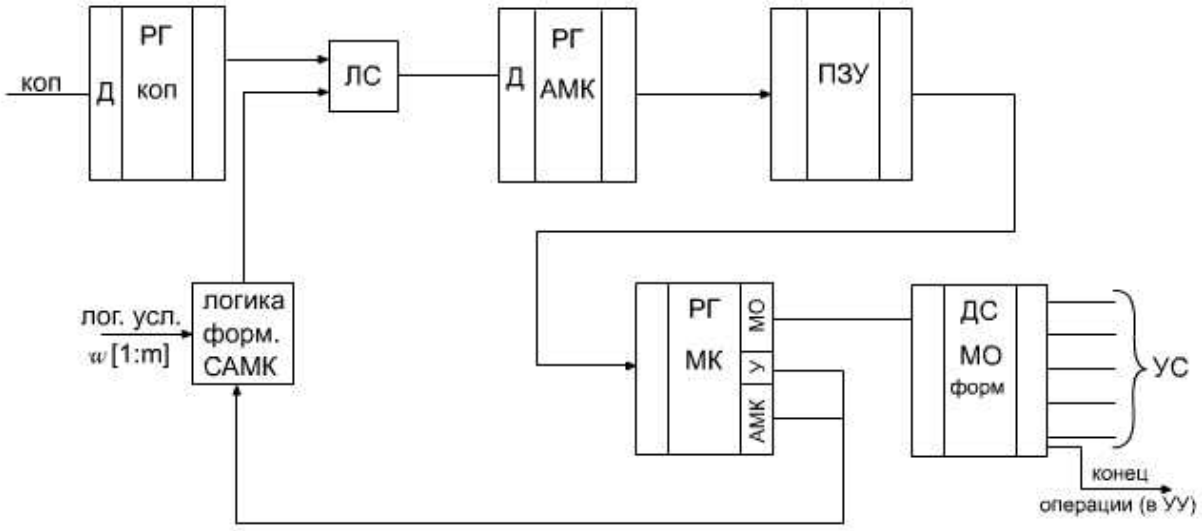

Рис. 2.15 Функциональная схема БФУС микропрограммного типа

2.3.4.4. Основные этапы разработки БФУС микропрограммного типа

1. Содержательное рассмотрение алгоритма вычислительной операции с числовыми примерами.
2. Составление обобщенного алгоритма выполнения операций.
3. Разработка структуры АЛУ и системы микроопераций
4. Составление кодированной схемы алгоритма, “привязанной” к данной схеме АЛУ.
5. Разработка формата микрокоманды, выбор разрядности полей микрокоманды.
6. Составление списка микроопераций, реализующих данный алгоритм.
7. Кодирование микрокоманд (написание микропрограммы).
8. Разработка логических схем формирования САМК, дешифратора МО и формирователя УС.

2.3.4.5. Разработка микропрограммы на примере операции “деление в прямых кодах”

Принимаем, что числа представлены в прямых кодах в форме с фиксированной запятой. Машинный алгоритм выполнения данной операции практически подобен ручному.

Основные особенности:

- операция деления выполняется над модулями чисел;
- знак результата определяется логическим способом;
- в машинном алгоритме делитель сдвигается вправо;
- операция сравнения выполняется в прямых кодах.

Выполняем числовой пример деления двоичных чисел:

$$C = \frac{A}{B}$$

$$|A| < 1, \quad |B| < 1$$

$$A = -\frac{3}{16} \qquad [A]_n = 1.0011$$

$$B = \frac{15}{16} \qquad [B]_n = 0.1111$$

$$|A| = .0011$$

$$|B| = .1111$$

$$01234 - № \ разряда$$

0 такт

$$\begin{array}{rll} & .0011 & A_0 \\ - & .1111 & B_0 \\ \hline & .0100 & |A_0| - |B_0| \end{array}.$$

Заем z = 1 → $|A_0| < |B_0| \rightarrow \quad C_0 = 0$

1 такт

$$\begin{array}{l} .\vec{0}\vec{0}11 \qquad A_1 = A_0 \\ -\big| .0111 \qquad B_1 = B_0 \cdot 2^{-1} \\ \hline .1100 \end{array}$$

z=1 $\rightarrow$ $\quad |A_1| < |B_1| \quad \rightarrow \quad C_1 = 0$

2 такт

$$\begin{array}{l} .\vec{0}\vec{0}11 \qquad A_2 = A_0 \\ -\big| .0011 \qquad B_2 = B_1 \cdot 2^{-1} \\ \hline .0000 \end{array}$$

z=0 $\rightarrow$ $\quad |A_2| \geq |B_2| \quad \rightarrow \quad C_2 = 1$

3 такт (тактов должно быть столько же, сколько разрядов)

$$\begin{array}{l} .\vec{0}\vec{0}\vec{0}0 \qquad A_3 = A_2 - B_2 \\ -\big| .0001 \qquad B_3 = B_2 \cdot 2^{-1} \\ \hline .1111 \end{array}$$

z=1 $\rightarrow$ $\quad C_3 = 0$

4 такт

$$\begin{array}{l} .0000 \qquad A_3 \\ -\big| .0000 \qquad B_4 = B_3 \cdot 2^{-1} \\ \hline .0000 \end{array}$$

z=0 $\rightarrow$ $\quad C_4 = 1$

$$|C| = .0101 \qquad \left(= \frac{5}{16} \right).$$

знак C = знак A $\oplus$ знак B = 1

$$[C]_n = 1.0101 \qquad \left(= -\frac{5}{16} \right).$$

На основе анализа данного числового примера построена схема алгоритма деления в прямых кодах, представленная на рис. 2.16.

Как видно из этого алгоритма, в начале каждого цикла выполняется сравнение модулей делимого и делителя с помощью вычитания в прямых кодах. В следующем блоке анализируется значение заема. Если заем Zi =1, то и признак Wi =1, тогда делимое не изменяется и очередная цифра частного Ci = 0.

Если же заем Zi=0, то и признак Wi=0, тогда делимое для следующего цикла определяется как разность Ai = Ai -Bi. Очередная цифра частного Ci=1. Далее в обеих ветках алгоритма происходит сдвиг вправо делителя на 1 разряд. Если содержимое счетчика циклов не равно количеству разрядов в числах, то цикл повторяется, иначе - операция закончена и результат выдается в ОЗУ.

Следующий этап - разработка структуры АЛУ в соответствии с алгоритмом на рис. 2.16. Будем строить АЛУ параллельного действия с непосредственными связями.

На рис. 2.17 приведена соответствующая функциональная схема АЛУ. В РГ1 микрооперацией ПР РГ1 принимается делимое, в РГ2 микрооперацией ПР РГ2 принимается делитель. Кроме того, так как делитель в соответствии с алгоритмом в каждом такте сдвигается вправо, на РГ2 поступают также импульсы сдвига СДВ ПР.

На входы вычитателя при поступлении микрооперации ВЫД SUB через ЛС подаются РГ1 [1:n] и инверсия РГ2 [1:n]. При этом на выходе вычитателя образуется разность Ai = Ai –Bi.

Эта разность микрооперацией ВЫД SM РГ1 выдается на вход РГ1 при значении признака Wi=0. Частное образуется в РГ3 за счет того, что в каждом такте вычитания на вход младшего разряда РГ3 поступает значение очередного разряда частного Ci = Zi, начиная со старшего разряда.

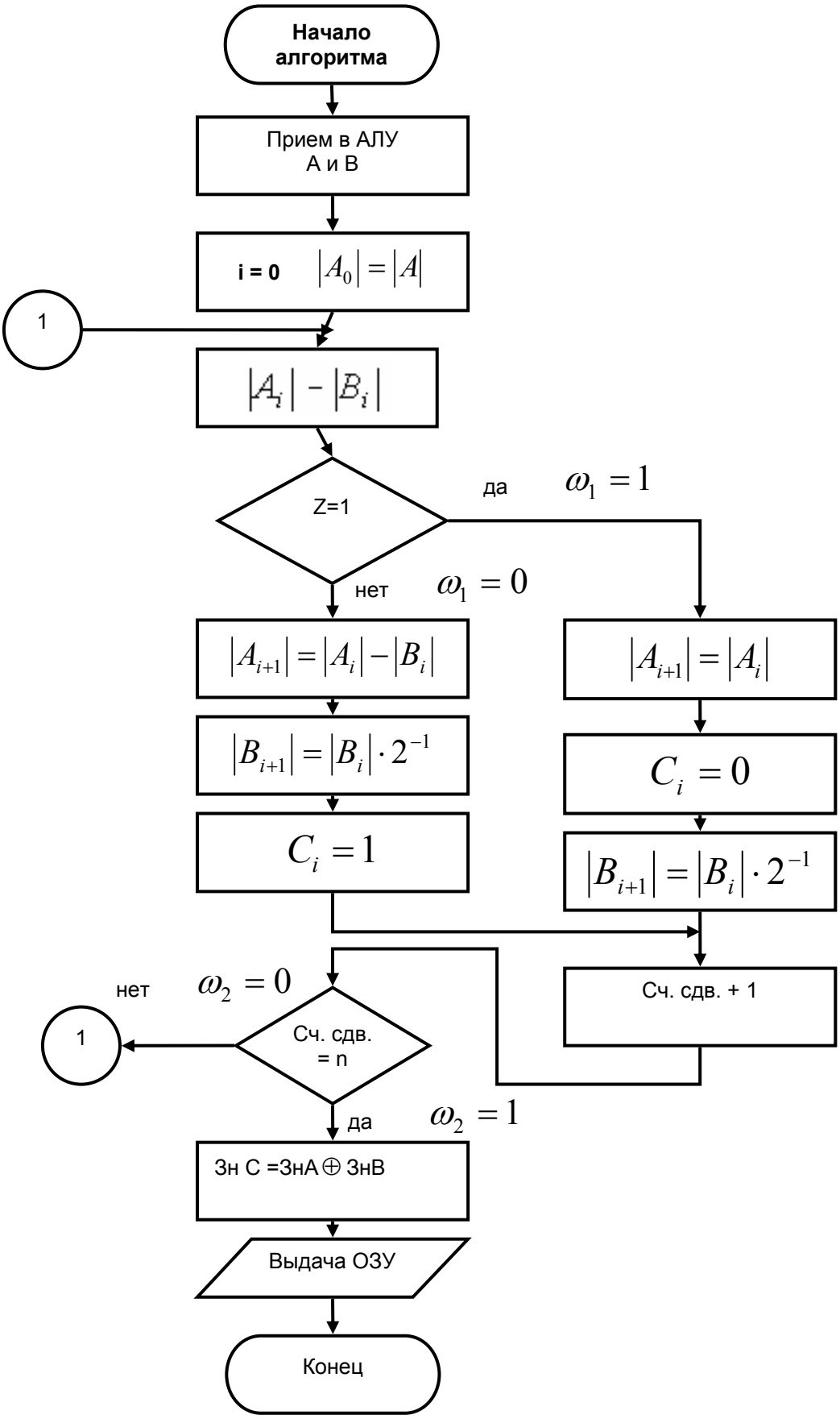

Рис. 2.16. Схема алгоритма деления в прямых кодах

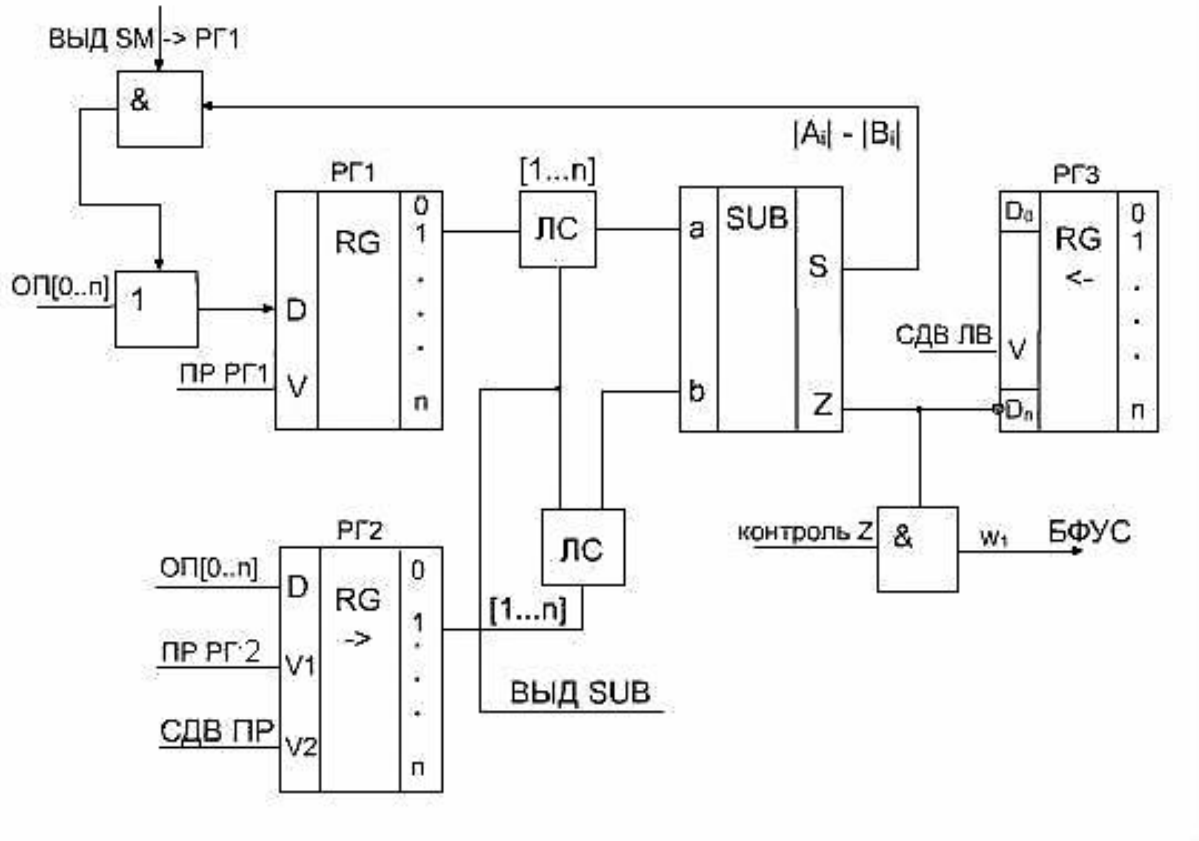

Рис. 2.17. Функциональная схема АЛУ для операции деления

Одновременно на РГ3 поступают УС СДВ ЛВ, благодаря которым при окончании операции в РГ3 устанавливается искомое значение C[1:n].

Контроль количества сдвигов выполняется с помощью счетчика СТ2, который находится в устройстве управления (рис. 2.18). Количество разрядов в счетчике выбирается по соотношению $2^m \geq n$.

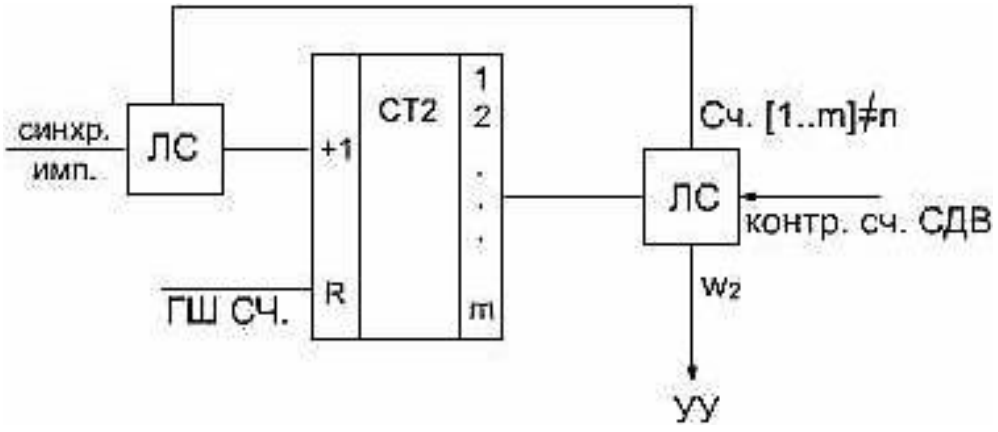

Рис. 2.18. Схема формирования и контроля числа сдвигов

Переходим к составлению микропрограммы и условимся, что принимаем двухадресную структуру команды (табл. 2.9).

Таблица 2.9

Деление	Адр. А	Адр. В

Задаем формат микрокоманды (табл. 2.10).

Таблица 2.10

МО	АМК	Поле У	
		У1	У2

Разрядность поля МО определяется полным количеством микроопераций, которое необходимо для реализации всей системы команд ЭВМ, а также принятым способом кодирования микроопераций. Разрядность поля АМК определяется необходимым объемом ПЗУ. Разрядность поля У равна максимальному числу разветвлений в отдельной микропрограмме.

Объем ПЗУ определяется в зависимости от того, сколько микропрограмм должно в нем находиться, и суммарной длиной всех этих микропрограмм.

Микропрограмма операции деления представлена в табл. 2.11.

В табл. 2.11 принята нумерация разрядов адреса ПЗУ справа, т.е. 4321. Расстановка кодов в колонках ПЗУ и АМК начинается с МК5 «контр. Z», после которой должно быть разветвление. Формирование САМК происходит за счет кода управления У1. Поэтому при w_1 =0 САМК=АМК= 1100 осуществляется переход к МК6, при w_1 =1 САМК=1101 осуществляется переход к МК7 в соответствии с алгоритмом на рис. 2.16. Аналогично расставляются коды адресов для МК8, после которой также происходит разветвление.

Таблица 2.11

N МК	Адр. ПЗУ	Переходы	МО	АМК	Поле У	
					У2	У1
1	0		ГШ СУ	0001		
2			ПР РГ1			
3			ПР РГ2			
4	1000	$\omega_2 = 0$	Выд SUB	1001		
5	1001		Контр Z	1100		1
6	1100	$\omega_1 = 0$	Выд SUB ->РГ1	1101		
7	1101	$\omega_1 = 1$	СДВ	1011		
8	1011		Контр СЧ СДВ	1000	1	
9	1010	$\omega_2 = 1$	ПР ЗН	0011		
10	11		Выд ОЗУ	0100		
11	100		конец			

Далее остается расставить коды адресов АМК и ПЗУ на линейных участках микропрограммы, используя оставшиеся комбинации кодов.

3. Запоминающие устройства

В состав современных ЭВМ входит значительное количество разнообразных запоминающих устройств (ЗУ).

Эти устройства классифицируются по ряду признаков:

1. По методу использования:

а) двусторонние, которые позволяют автоматически выполнять считывание и запись информации;

б) односторонние, которые позволяют выполнять в автоматическом режиме только считывание информации, а запись - предварительная, в процессе изготовления. Информация заносится либо механическим, либо электронным способом. Их называют постоянные запоминающие устройства (ПЗУ).

2. По назначению:

а) внутренние двусторонние ЗУ, которые в процессе выполнения программы взаимодействуют непосредственно с процессором. Их называют оперативные запоминающие устройства (ОЗУ);

б) внешние запоминающие устройства (ВЗУ), которые взаимодействуют с процессором через ОЗУ и предназначены для хранения больших массивов данных и программ.

3. В зависимости от принципа считывания информации:

а) ЗУ без разрушения информации при считывании – статические;

б) ЗУ с разрушением информации при считывании – динамические.

4. По физическим свойствам запоминающей среды:

а) полупроводниковые ЗУ, в которых бит информации запоминается на статическом триггере, эти ЗУ относятся к статическому типу;

б) ферромагнитные ЗУ, в которых принцип запоминания информации основан на изменении магнитного состояния на микроскопическом участке носителя информации;

в) емкостные ЗУ– в них бит информации запоминается в виде величины заряда конденсатора.

Ферромагнитные и емкостные запоминающие устройства являются динамическими, поскольку при считывании информация разрушается и ее надо восстановить.

6. По способу поиска информации:

а) адресные запоминающие устройства;

б) ассоциативные запоминающие устройства, в которых поиск информации выполняется по некоторому признаку.

Основные характеристики запоминающих устройств

1. Объем запоминающего устройства, который может характеризоваться двумя величинами:

а) количеством бит, которые могут храниться в запоминающем устройстве V;

б) количеством слов, которые могут храниться в запоминающем устройстве N. Слово – 1 байт [Б], тысяча байт (КБ), миллион байт (МБ) и т. д.

2. Быстродействие запоминающего устройства, которое оценивается двумя параметрами:

а) временем обращения к запоминающему устройству $t_{обр}$ – интервал времени между моментом поступления на запоминающее устройство команды чтения и моментом, когда выбранная информация будет принята в регистр слова;

б) временем выбора $t_{выб}$ – характеризует быстродействие ЗУ в составе некоторой ЭВМ. $t_{выб\ min}$ – минимальный интервал времени между двумя последовательными командами обращения к данному запоминающему устройству.

3.1. Оперативные запоминающие устройства

В зависимости от физических свойств запоминающей среды различают ферромагнитные ОЗУ и полупроводниковые ОЗУ.

Ферромагнитные ОЗУ на магнитных сердечниках широко применялись во 2-м и 3-м поколениях ЭВМ. Запоминающий элемент представляет собой ферритовое кольцо, которое обеспечивает запоминание одной двоичной цифры.

В настоящее время такие ОЗУ используются в специализированных ЭВМ (в космосе).

Преимуществом ферромагнитных ОЗУ является то, что при выключении питания или космическом облучении информация не разрушается. Однако они более сложные и дорогие, а также при построении ОЗУ больших объемов возникают трудности.

Полупроводниковые ОЗУ в настоящее время применяются очень широко в качестве основной оперативной памяти ЭВМ. Они дешевле, позволяют создавать ОЗУ большого объема, более

быстрые, но при выключении питания или космическом излучении информация пропадает.

В настоящее время используются две структуры:

- ОЗУ с двумерной структурой - тип 2D;
- ОЗУ с объемной структурой -тип 3D.

3.1.1. ОЗУ со структурой типа 2D

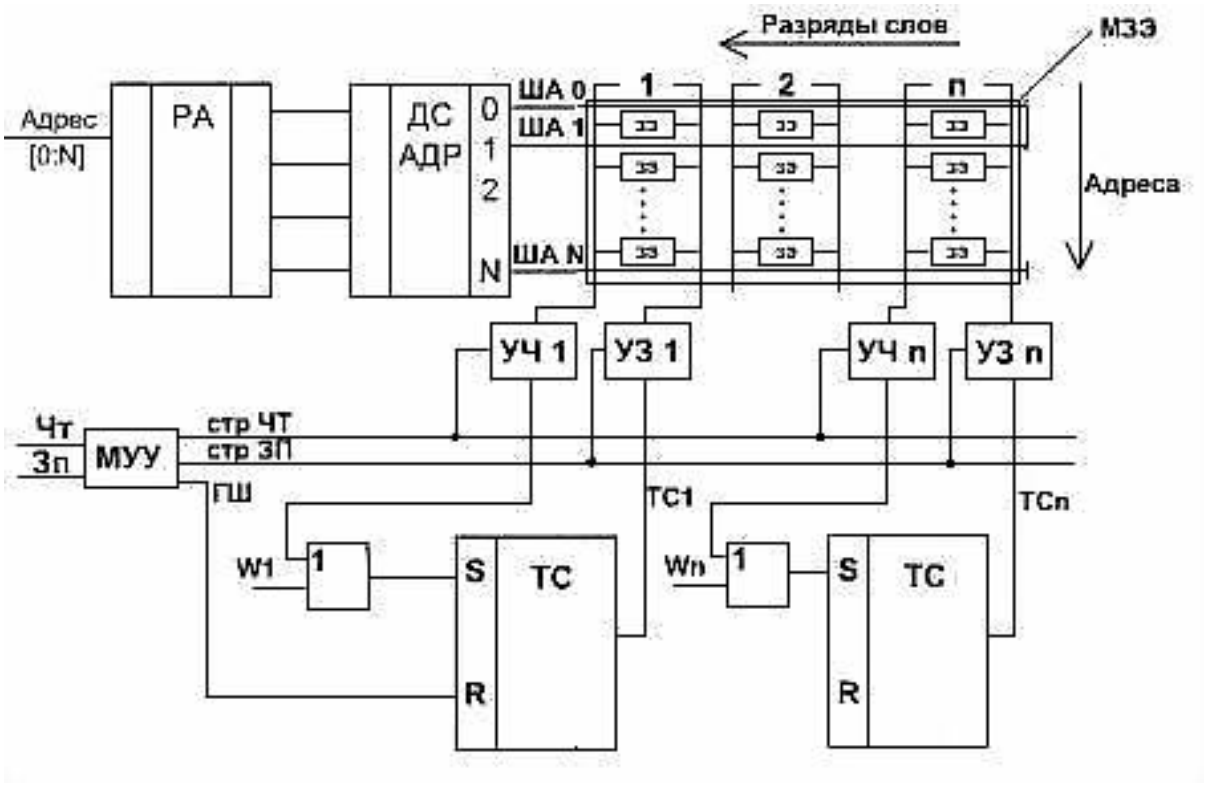

Рис. 3.1 Структура ОЗУ типа 2D

На рис. 3.1 представлена упрощенная схема ОЗУ с плоской структурой типа 2D.

Она включает в себя следующие основные функциональные узлы:

УЧ – усилитель чтения, УЗ – усилитель записи, ТС – триггер слова, РА- регистр адреса, ДС – дешифратор адреса, МЗЭ - матрица запоминающих элементов, МУУ – местное устройство управления.

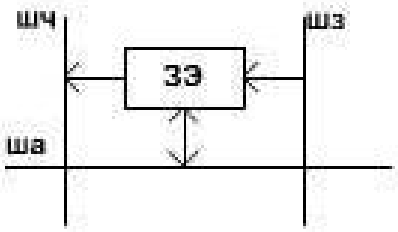

Рис. 3.2. ЗЭ – запоминающий элемент в структуре типа 2D

Основной функциональный узел – двухмерная (2D) матрица запоминающих элементов (МЗЭ). На рис.3.2 укрупнено представлен ЗЭ с подключенными шинами. Запись и считывание информации осуществляются посредством ША – шины адреса, ШЗ – шины записи и ШЧ – шины чтения.

ША подключена ко всем ЗЭ одного адреса, ЩЗ и ШЧ являются разрядными и объединяют ЗЭ одного разряда.

В режиме ЧТ поступает сигнал на соответствующую ША, при этом одновременно выбираются ЗЭ всех разрядов данного адреса и считанные сигналы поступают на УЧ всех разрядов.

При поступлении на все УЧ сигнала стр. ЧТ на их выходах появляется информация, которая принимается в ТС всех разрядов.

Опишем функционирование ТС1 с помощью логического выражения (3.1), где переменная W1 –информация, которая принимается в ТС1 в режиме записи.

ТС1 ‘: =’ ‘ЕСЛИ’ ГШ ‘ТО’ 0 ‘ИНЕСЛИ’ W1 ‘ТО’ 1 ‘ИНЕСЛИ’ УЧ1 ‘ТО’ 1 ‘ИНАЧЕ’ ТС1 (3.1)

Преимущество структуры типа 2D состоит в том, что считывание информации происходит при поступлении на него одного адресного сигнала. Это обеспечивает более высокую помехоустойчивость запоминающего устройства и более высокое быстродействие.

3.1.2. Структура ОЗУ типа 3D

Структура данного ОЗУ (рис. 3.3) называется объемной, так как в ней содержится столько МЭЭ, сколько разрядов в слове.
На одной матрице размещены ЗЭ одного разряда всех адресов. Все эти ЗЭ подключены к одной разрядной ШЧ и одной ШЗ. ША разделена пополам по числу разрядов. При этом образуются две равных группы шин: ШАх и ШАу

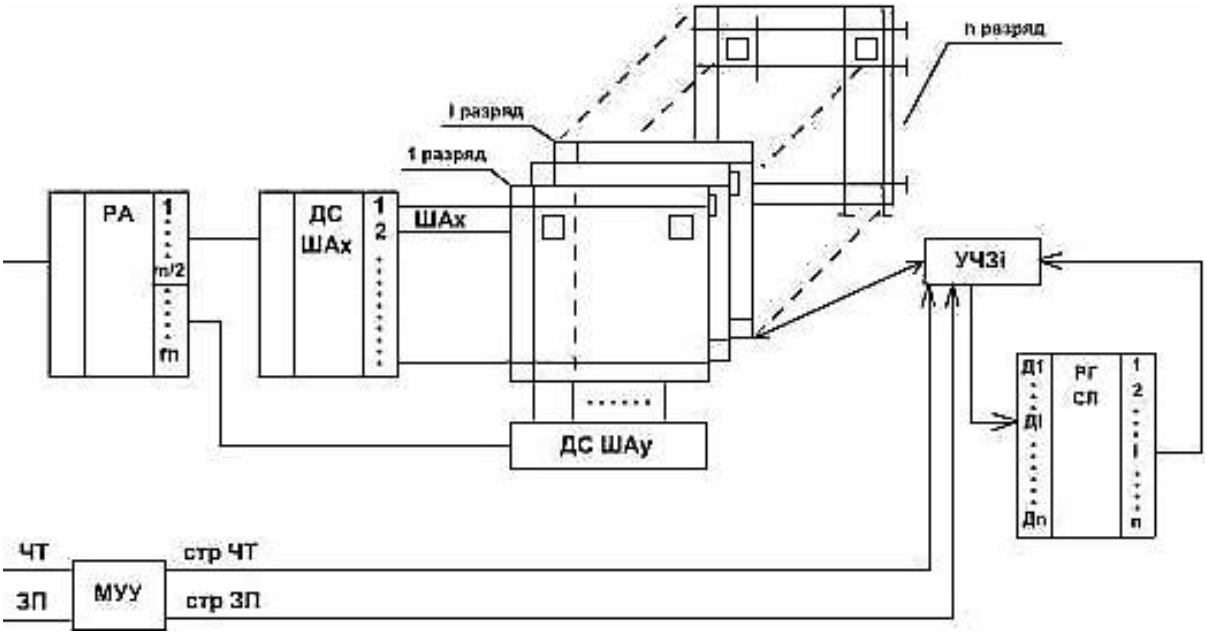

Рис. 3.3. ОЗУ по структуре типа 3D

На рис. 3.4 изображен ЗЭ с подключенными к нему шинами: ШЧ, ШЗ, ШАх, ШАу.

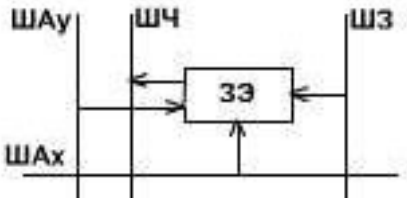

Рис. 3.4. Запоминающий элемент в структуре типа 3D

Каждый ЗЭ находится на скрещении ШАх и ШАу. Каждая из ША подключена последовательно к ЗЭ данного адреса на всех матрицах.

При операциях ЧТ и ЗП на всех матрицах одновременно выбираются ЗЭ, находящиеся на скрещении одних и тех же шин ШАх и ШАу.

Сравнительная оценка ОЗУ типа 2D и 3D

1. Построение ОЗУ по структуре типа 2D требует больше адресного оборудования

Адр. Об.$_{2D}$ ≡ N – количество слов в ОЗУ.

Адр. Об.$_{3D}$ ≡ $2 \times \sqrt{N}$

для ОЗУ больших объемов используется структура типа 3D.

2. В структуре типа 3D наличие большого количества матриц обуславливает к увеличение длины адресной шины, что приводит к колебательным процессам. Из-за этого ограничивается быстродействие запоминающего устройства данного типа (нельзя подавать короткие импульсы).

3.1.3. Полупроводниковые запоминающие устройства

Полупроводниковые запоминающие устройства используют в качестве запоминающих элементов статические триггеры типов ТТЛ, ЭСЛ, МОП. Чаще используются МОП, потому что потребляют наименьшую мощность.

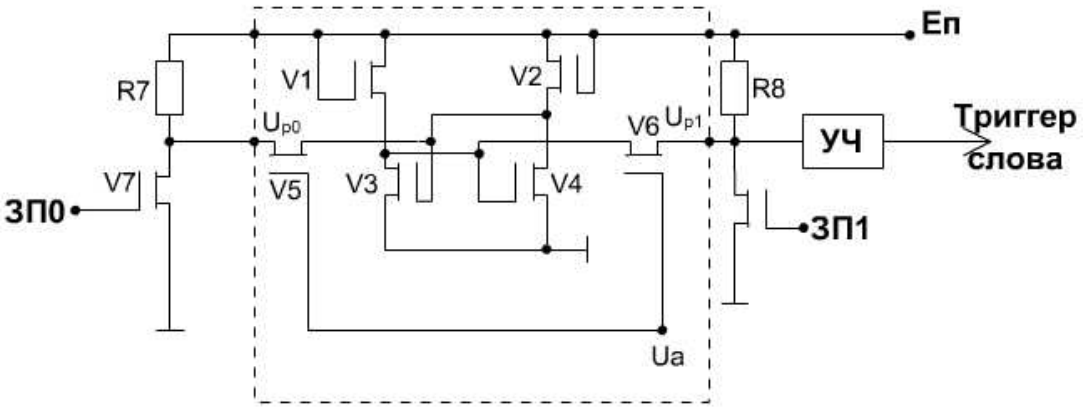

Рис. 3.5. Принципиальная схема полупроводникового ЗЭ

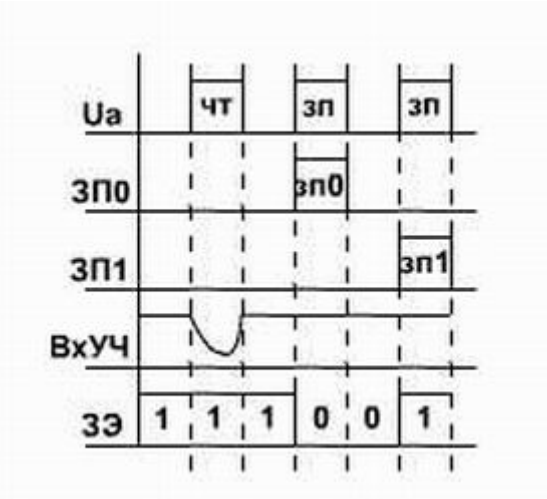

Рис. 3.6. Временная диаграмма работы ЗЭ

V3, V4 – образуют триггер.

V1, V2 – их нагрузка.

V5, V6 – выборка этого ЗЭ.

$U_в$:= 1; $U_н$: = 0; $U_в$ = Еп

Режим записи 0: на внешний усилитель собранный на V7 подается положительный сигнал “ЗП0”, а на Ua подается положительное сигнал Ua, который открывает V5 и V6; на выходе V7 появляется отрицательный сигнал, который через V5 поступает на затвор V3. Если в ЗЭ := 1, то есть V3 был открыт, т.к. на затвор V3 поступал отрицательный сигнал с R7, то V3 будет закрываться при этом открывается V4 и ЗЭ переходит в состояние 0.

(ЗЭ := 0).

Под положительным сигналом на V8 и положительным сигналом на Ua дальнейшее происходит аналогично для ЗП0.

3.1.4. Ассоциативные запоминающие устройства (АЗУ)

Отличаются тем, что поиск информации происходит не по адресу, а по некоторому признаку, что ускоряет процесс поиска нужной информации; позволяет провести поиск информации по указанному признаку.

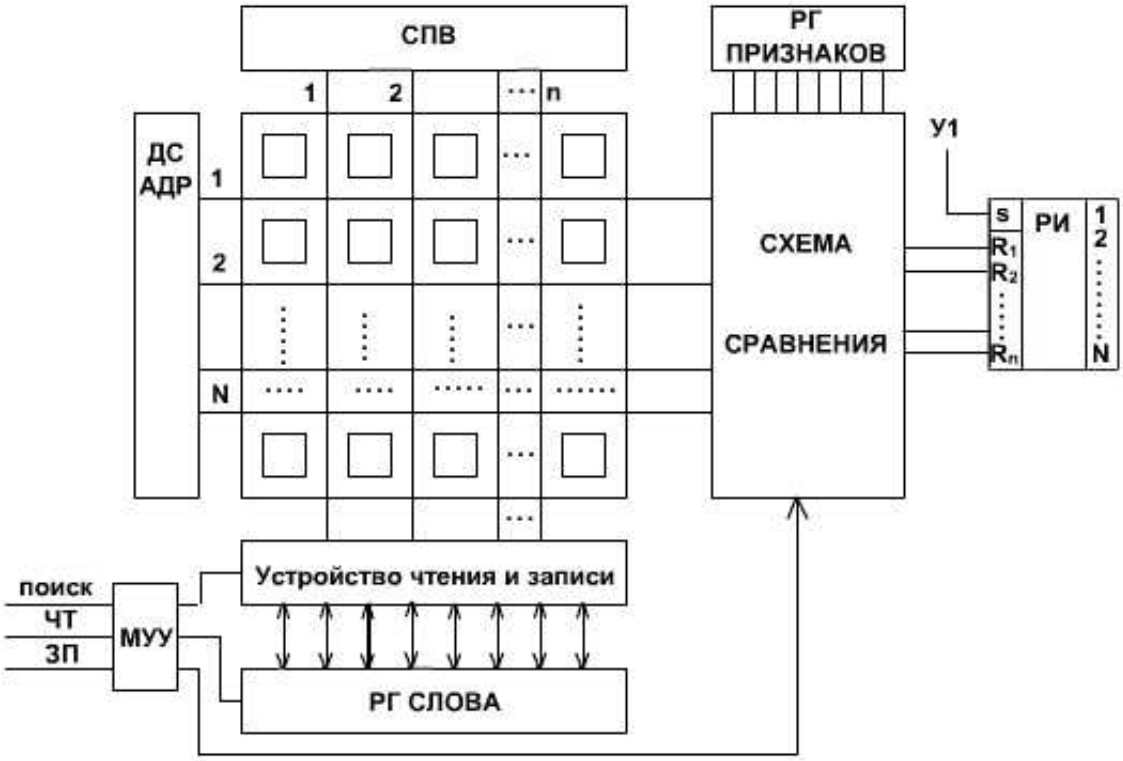

Рис. 3.7. Функциональная схема ассоциативного ОЗУ

СПВ – схема поразрядной выборки

СПВ обеспечивает одновременную выборку содержимого всех разрядов одного адреса.

РИ – регистр индикатор.

РИj ‘:=’ ‘ЕСЛИ’ У1 ‘ТО’ 1 ‘ИНЕСЛИ’ (ЗЭj1 ⊕ П1) v (ЗЭj2 ⊕ П2) v (ЗЭj3 ⊕ П3)… (ЗЭjn ⊕ Пn) ‘ТО’ 0 ‘ИНАЧЕ’ РИj.

В процессе поразрядной выборки во всех адресах одновременно и последовательно сравнивается содержимое ЗЭ с признаком, заданным в РГ ПРИЗНАКОВ. Если по какому-то адресу содержимое ЗЭ отдельных разрядов не совпадает с признаком, то на выходе схемы сравнения данного адреса появляется сигнал, сбрасывающий в 0 триггер соответствующего адреса в РИ. Только в тех адресах, где содержимое всех разрядов совпало с состоянием РГ ПРИЗНАКОВ, сигнал на выходе схемы сравнения равен 0. Соответственно триггер РИ, РИj остается в состоянии 1, это означает, что данный адрес соответствует искомому признаку.

После нахождения адреса считывание осуществляется обычным адресным способом. Запись информации осуществляется только по свободному адресу (признак свободной ячейки).

ОГЛАВЛЕНИЕ

Тамара Миновна Александриди
Юрий Юрьевич Исаев
Игорь Андреевич Морозов

ОРГАНИЗАЦИЯ ЭВМ И СИСТЕМ
Часть 3
Функциональные устройства ЭВМ

Учебное пособие

Редактор Н.П. Лапина
Технический редактор Е.К. Евстратова

Тем. план 2007 г., п. 19

Подписано в печать		Формат 60x84/16
Печать офсетная	Усл. печ. л. 2,8	Уч.-изд. л 2,3
Тираж 300 экз	Заказ	Цена 20 руб.

Ротапринт МАДИ (ГТУ) 125319, Москва Ленинградский просп., 64