

## 7 ПОСТРОЕНИЕ ПРОЦЕССОРА

Основой функционирования любого микропроцессора является взаимодействие операционного устройства (ОУ) и управляющего устройства (УУ).

Можно построить операционное устройство с таким набором узлов и такой схемой их соединения, которые обеспечат решение разнообразных задач. Такое ОУ может включать узлы, выполняющие над данными такие стандартные функции как арифметическое суммирование, дизъюнкцию, конъюнкцию, инверсию, исключающее ИЛИ, сдвиги и др. Задача, решаемая таким универсальным операционным устройством, определяется последовательностью и набором управляющих сигналов, формируемых управляющим устройством. Таким образом, для решения разнообразных задачи может быть использовано одно и тоже операционное устройство.

Существует два принципиально разных подхода к проектированию управляющего устройства МП: использование принципа схемной логики или использование принципа программируемой логики.

В первом случае при проектировании МП определяется такой порядок соединения логических схем, который обеспечивает формирование управляющих сигналов в соответствии с алгоритмом, предусмотренным исполняемой командой. Устройства построенные на принципе схемной логики, способны обеспечивать наивысшее быстродействие при заданной технологии.

Во втором случае требуемое функционирование обеспечивается занесением в память устройства определенной микропрограммы, составленной из микрокоманд, в кодах которых заложены управляющие сигналы. В зависимости от набора микропрограмм такое универсальное

управляющее устройство способно обеспечить требуемое управление операционным устройством при решении самых разных задач.

При этом наивысшее быстродействие достигается в процессорах, в которых управляющее устройство строится с использованием принципа схемной логики, а операционное устройство выполняется в виде устройства, специализированного для решения конкретной задачи.

# 7.1 СИНТЕЗ ПРОЦЕССОРА СО СПЕЦИАЛИЗИРОВАННЫМ ОПЕРАЦИОННЫМ УСТРОЙСТВОМ

Рассмотрим методику построения процессора на примере реализации устройства, выполняющего операцию умножения двоичных чисел без знака.

Проиллюстрируем на примере умножения двух двоичных чисел.

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ 1\ 0\ 1\ 1 \\ \hline \end{array}$$

Предусмотрим суммирование частичных произведений, начиная с младшего.

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ 1\ 1\ 0\ 1 \\ 0\ 0\ 0\ 0 \\ 1\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \end{array}$$

В таблице приведена последовательность выполнения операций.

| Множимое (R1) | Старшие разряды произведения (R3) | Множитель и младшие разряды произведения (R2) | Действие           |
|---------------|-----------------------------------|---|--------------------|
| 1 1 0 1       | 0 0 0 0                           | 1 0 1 1                                       | Исходное состояние |
|               | +<br>1 1 0 1                      |   | Суммирование       |
|               | 0 1 1 0 1                         |   | Сдвиг (R3) и (R2)  |
|               | +<br>1 1 0 1                      |   | Суммирование       |
|               | 1 0 0 1 1                         |   | Сдвиг (R3) и (R2)  |
|               | 0 > 1 0 0 1                       | > 1 1 1 0                                     | Сдвиг (R3) и (R2)  |
|               | 0 > 0 1 0 0                       | > 1 1 1 1                                     | Сдвиг (R3) и (R2)  |
|               | +<br>1 1 0 1                      |   | Суммирование       |
|               | 1 0 0 0 1                         |   | Сдвиг (R3) и (R2)  |
|               | 0 > 1 0 0 0                       | > 1 1 1 1                                     | Сдвиг (R3) и (R2)  |
|               | п р о и з в е д е н и е           |   |                    |

В исходном состоянии в R1 и R2 - множимое и множитель, R3 - в нулевом состоянии. Анализируется содержимое младшего разряда R2. Так как его значение "1", то 1-е частичное произведение равно множимому и оно прибавляется к содержимому регистра R3, используемому для накопления сумм частичных произведений. Далее производится сдвиг на один разряд вправо содержимого регистров R2 и R3, причем выдвигаемый при сдвиге из R3 младший разряд (не принимающий в дальнейшем участия при суммировании) передается в освобождающийся старший разряд R2. В младшем разряде R2 оказывается 2-й разряд множителя. Процесс повторяется. При нулевом значении младшего разряда R2 суммирование не производится, только сдвиг. Число циклов равно числу разрядов множителя.

#### 7.1.1 СИНТЕЗ ОПЕРАЦИОННОГО УСТРОЙСТВА

В соответствии с описанным выше процессом, для выполнения операции умножения необходимо в операционном устройстве иметь: регистры R1, R2, R3, сумматор (См) и счетчик (Сч) числа повторений.

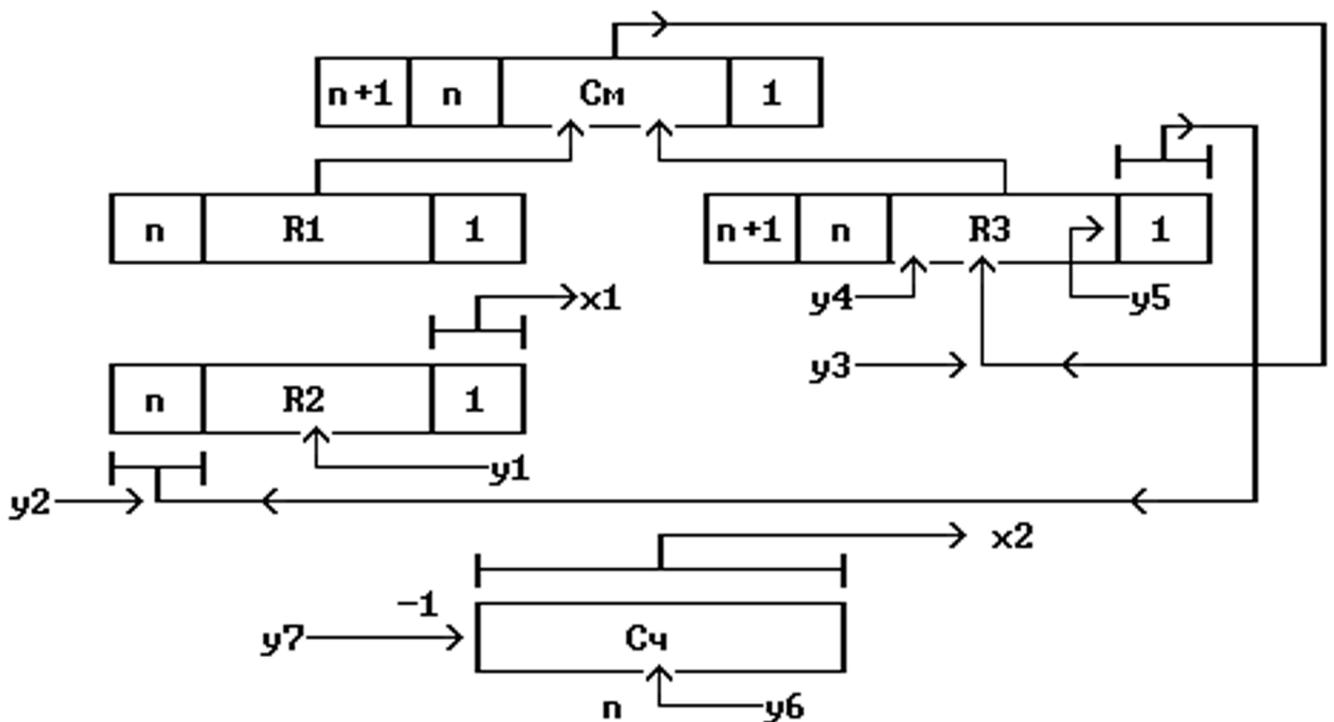


Рисунок 7.1

В R2 предусмотрена микрооперация сдвига содержимого на 1 разряд вправо, выполняемая под действием управляющего сигнала  $y_1$  и микрооперации пересылки в старший разряд этого регистра содержимого младшего разряда R3, выполняемого под действием сигнала  $y_2$ . Сумматор производит суммирование чисел, поступающих из R1 и R3. Для хранения переноса сумматор содержит  $n+1$  разряд. Результат под действием  $y_3$  принимается в R3, который, тоже должен иметь  $n+1$  разряд. Кроме того для регистра R3 предусмотрены микрооперации установки в 0 и сдвига содержимого на 1 разряд вправо, выполняемые под действием  $y_4$  и  $y_5$ . При наличии  $y_6$  счетчиком принимается установленное число  $n$ , под действием  $y_7$  выполняется микрооперация вычитания 1.

В операционном устройстве формируются следующие признаки:

$x_1$  - содержимое младшего разряда R2;

$x_2$  - результат проверки на 0 счетчика.

Приведем в условной записи список выполняемых в узлах операционного устройства микроопераций и список формируемых признаков:

$y_1$ : R2  $\leftarrow$  Сдвиг вправо R2

$y_2$ : R2[n]  $\leftarrow$  R3[1]

$y_3$ : R3  $\leftarrow$  См

$y_4$ : R3  $\leftarrow$  0

$y_5$ : R3  $\leftarrow$  Сдвиг вправо R3

$y_6$ : Сч  $\leftarrow$  n

$y_7$ : Сч  $\leftarrow$  Сч-1

-----  
 $x_1$ : R2[1] = 1

$x_2$ : Сч = 0

## 7.1.2 СИНТЕЗ УПРАВЛЯЮЩЕГО УСТРОЙСТВА НА ОСНОВЕ СХЕМНОЙ ЛОГИКИ

Разобьем на 2 этапа:

1. Построение алгоритма в микрооперациях (схема на рисунке 7.2а соответствует рассмотренному множительному устройству);

2. Построение алгоритма в микрокомандах (рисунок 7.2б).

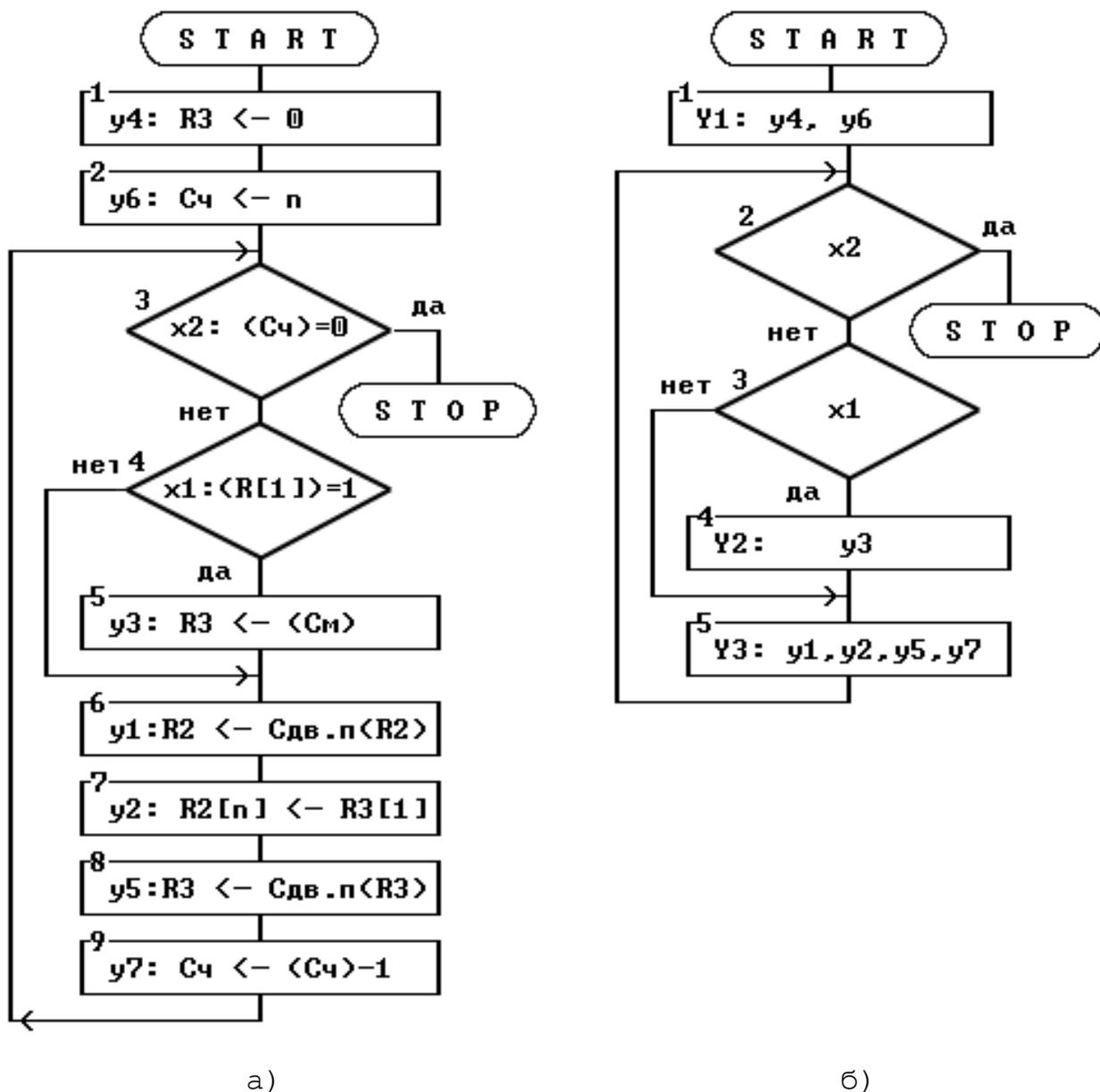


Рисунок 7.2

Для формирования микрокоманд необходимо определить, какие микрооперации могут выполняться одновременно (в одни и те же тактовые периоды). Очевидно, микрооперации  $y_4$  и  $y_6$  могут быть объединены в микрокоманду  $Y_1$ ,  $y_3$  не может быть объединена с другими микрооперациями, она представляет микрокоманду  $Y_2$ ;  $y_1, y_2,$

$y_5$ ,  $y_7$  могут выполняться одновременно и объединены в  $Y_3$ . На рисунке - схема алгоритма в микрокомандах.

Управляющее устройство - это логическое устройство последовательностного типа. Микрокоманда, выдаваемая в текущем тактовом периоде, зависит от того, какая микрокоманда выдавалась в предыдущем тактовом периоде, или, иначе, от состояния в котором находится устройство.

Можно построить структурную схему управляющего устройства. Она состоит из комбинационного узла и регистра состояния.

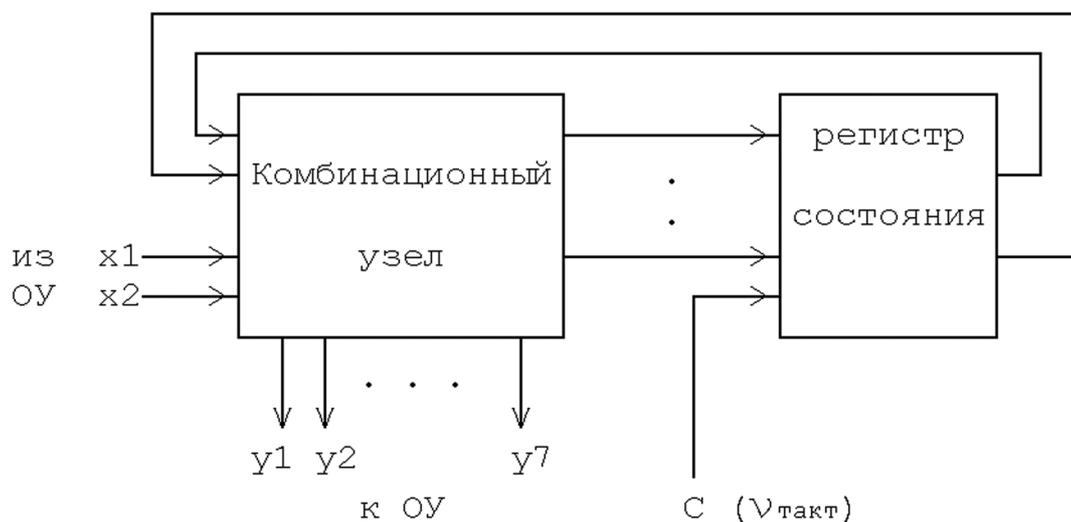


Рисунок 7.3

Регистр состояния содержит информацию о текущем шаге в ходе выполнения алгоритма команды, а комбинационный узел на основе этой информации и, при необходимости, признаков формирует набор управляющих сигналов для ОУ. Кроме того, комбинационный узел выдает информацию о следующем шаге алгоритма, которая записывается в регистр состояния под действием синхроимпульса.

## ВЫПОЛНЕНИЕ ПРОГРАММЫ (МП с УУ на принципе схемной логики)

Мы рассмотрели реализацию управляющего узла для выполнения операции умножения. Очевидно, могут быть построены подобные узлы для выполнения других операций. И если в управляющем устройстве универсального процессора предусмотреть такие узлы, то, включая тот или иной узел, можно обеспечить выполнение различных операций на одном и том же оборудовании операционного устройства.

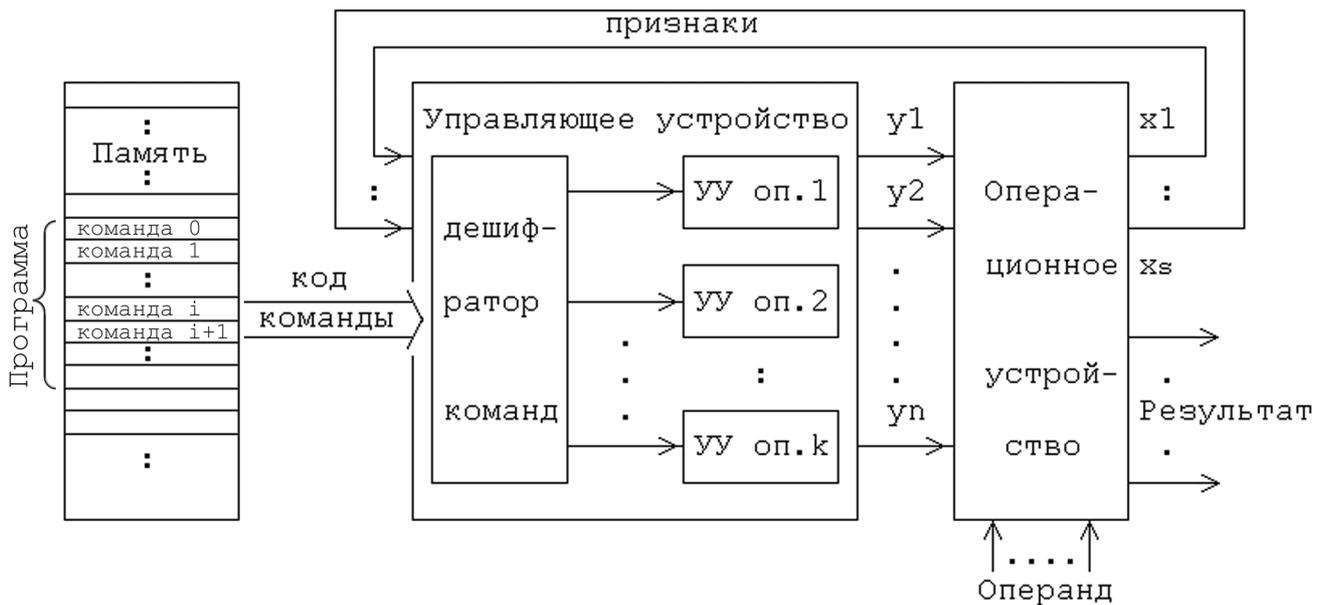


Рисунок 7.4

Операцию, подлежащую исполнению в процессоре, называют командой. С помощью дешифратора код команды преобразуется в сигнал, производящий включение управляющего узла (уу оп. k), который отвечает за выполнение соответствующей команды.

При этом алгоритм сложной задачи записывается в виде последовательности команд, которая будет соответствовать последовательности таких выполняемых простых операций. Такая последовательность команд образует ПРОГРАММУ, хранимую в памяти. Считывая последовательно из памяти команды и исполняя их в процессоре можно решить сложную задачу.

### 7.1.3 СИНТЕЗ УПРАВЛЯЮЩЕГО УСТРОЙСТВА НА ОСНОВЕ ПРОГРАММИРУЕМОЙ ЛОГИКИ.

Выше показано, что выполнение операций в МП осуществляется в виде последовательности выполняемых микрокоманд. Можно предусмотреть, отличающийся от принципа схемной логики, способ формирования в управляющем устройстве управляющих сигналов, под действием которых в операционном устройстве выполняются микрокоманды.

Управляющие сигналы  $y_1 \dots y_n$  на выходе управляющего устройства в каждом тактовом периоде имеют уровни логических 0 и 1. Таким образом, каждой микрокоманде на выходе управляющего устройства соответствует определенная кодовая комбинация. Такие кодовые комбинации можно хранить в управляющей памяти. Последовательность микрокоманд, предназначенную для выполнения некоторой операции, называют микропрограммой. При этом выполнение операции сводится к выборке из управляющей памяти микрокоманд микропрограммы и выдаче с их помощью управляющих сигналов  $y_1 \dots y_n$  в операционное устройство.

В управляющей памяти можно хранить много микропрограмм, предназначенных для выполнения различных операций. По выбранной из оперативной памяти команде в управляющей памяти находится соответствующая команде микропрограмма. Далее путем последовательного считывания микрокоманд найденной микропрограммы и их выполнения в операционном устройстве реализуется предусматриваемая командой операция.

Такой способ реализации операций называется МИКРОПРОГРАММНЫМ способом, а построенное на этом принципе устройство - управляющим устройством с ПРОГРАММИРУЕМОЙ ЛОГИКОЙ.

На рисунке 7.5 изображена структурная схема МП с управляющим устройством, построенным на принципе программируемой логики. Функции блока микропрограммного управления (БМУ) сводятся к определению адреса очередной микрокоманды (МК) в управляющей памяти (УП). Поступающая из памяти команда содержит адрес первой микрокоманды той микропрограммы, которая реализует предусмотренную командой операцию. Таким образом решается проблема поиска в УП микропрограммы, соответствующей данной команде. Адреса всех последующих микрокоманд определяются в БМУ следующим образом.

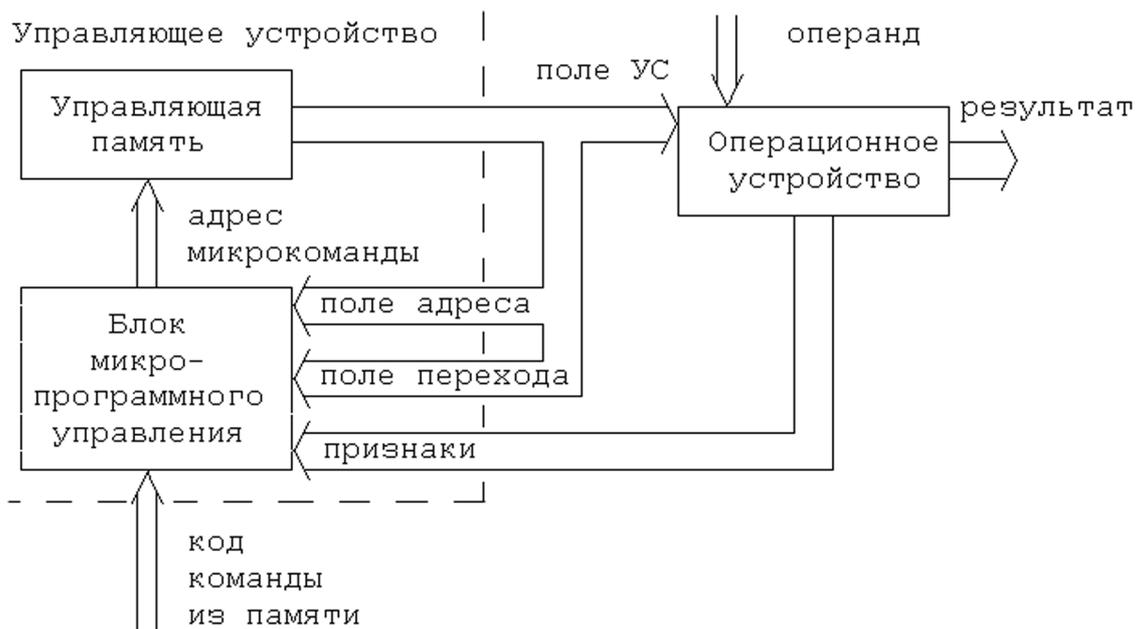


Рисунок 7.5

Рассмотрим формат микрокоманды (рисунок 7.6).



Рисунок 7.6

В формате микрокоманды предусмотрено поле адреса, которое содержит адрес очередной микрокоманды. Таким образом, считав из управляющей памяти микрокоманду, по содержанию ее поля адреса узнаем адрес следующей МК, но так можно получить адрес МК при отсутствии в алгоритме разветвлений, т.е. условных переходов (УСП). Для реализации условных переходов в микрокоманде можно предусмотреть поле переходов, в котором указывается имеет ли место условный или безусловный переход, и в случае условного перехода - на значение каких условий следует ориентироваться при определении адреса очередной МК.

Пусть поле переходов построено следующим образом: Один из разрядов поля указывает вид перехода (например: 0 - безусловный переход; 1 - условный переход).

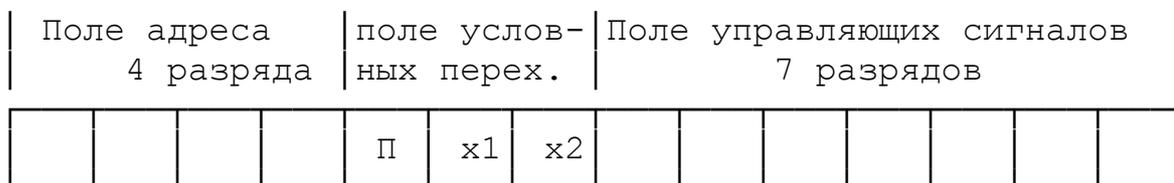
Для каждого условия в поле перехода имеется разряд, указывающий участие данного условия в определении адреса. Если условный переход осуществляется по некоторому условию, то формирование адреса очередной МК будем осуществлять замещением младшего разряда содержимого поля адреса текущей МК значением соответствующего условия. Такую операцию называют МОДИФИКАЦИЕЙ АДРЕСА.

Поле управляющих сигналов содержит кодовую комбинацию, которая

подаётся в операционное устройство в качестве управляющих воздействий.

### ПРИМЕР ПОСТРОЕНИЯ МИКРОПРОГРАММЫ

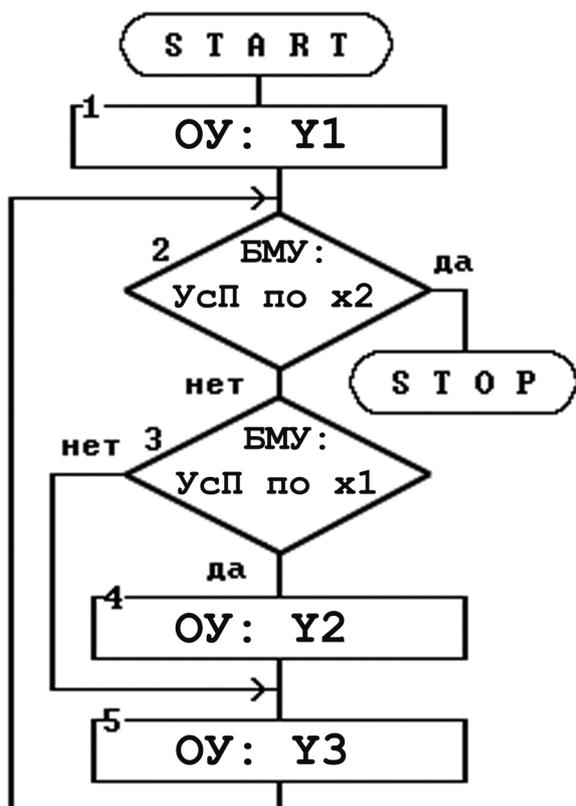
Построим микропрограмму для выполнения рассмотренной выше операции умножения. Выберем формат микрокоманды:



4-х разрядное поле адреса позволяет обращаться к любой ячейке управляющей памяти с 16 ячейками.

Поле переходов содержит 3 разряда: разряд П, наличие 1 в котором указывает на то, что имеет место условный переход; разряды x1 и x2, наличие 1 в которых определяет условие, по которому происходит условный переход.

Поле управляющих сигналов содержит 7 разрядов и обеспечивает выдачу 7 различных микроопераций.

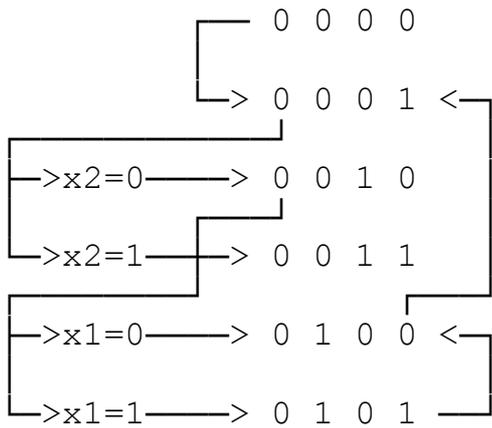


Для хранения составляемой микропрограммы используем ячейки управляющей памяти с нарастающими адресами. Ориентируясь на схему алгоритма (в командах) построим схему алгоритма в микрокомандах.

Рисунок 7.7.

Адрес упр.пам.

Содержимое ячейки (микрокоманда)



МК1 (МК ОУ:Y1; МК БМУ:БП)  
 МК2 (МК ОУ: -; МК БМУ:УсП по x2)  
 МК3 (МК ОУ: -; МК БМУ:УсП по x1)  
 STOP (выход из микропрограммы)  
 МК5 (МК ОУ:Y3; МК БМУ:БП)  
 МК4 (МК ОУ:Y2; МК БМУ:БП)

Микропрограмма приведена в таблице 7.1.

| адрес<br>ячейки | М и к р о к о м а н д а |                         |    |    |    |       |    |    |    |    |    |      |
|-----------------|-------------------------|-------------------------|----|----|----|-------|----|----|----|----|----|------|
|                 | МК БМУ                  |                         |    |    |    | МК ОУ |    |    |    |    |    |      |
|                 | поле<br>адреса          | поле услов.<br>перехода |    |    | y1 | y2    | y3 | y4 | y5 | y6 | y7 |      |
|                 |                         | П                       | x1 | x2 |    |       |    |    |    |    |    |      |
| 0000            | 0001                    | 0                       | x  | x  | 0  | 0     | 0  | 1  | 0  | 1  | 0  | МК1  |
| 0001            | 0010                    | 1                       | 0  | 1  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | МК2  |
| 0010            | 0100                    | 1                       | 1  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | МК3  |
| 0011            | -                       | -                       | -  | -  | -  | -     | -  | -  | -  | -  | -  | STOP |
| 0100            | 0001                    | 0                       | x  | x  | 1  | 1     | 0  | 0  | 1  | 0  | 1  | МК5  |
| 0101            | 0100                    | 0                       | x  | x  | 0  | 0     | 1  | 0  | 0  | 0  | 0  | МК4  |

## ВЫПОЛНЕНИЕ ПРОГРАММЫ (МП с УУ на принципе программной логики)

Мы рассмотрели реализацию микропрограммы для выполнения операции умножения. Очевидно, могут быть составлены подобные микропрограммы для выполнения других операций. И если в управляющей памяти управляющего устройства универсального процессора предусмотреть такие микропрограммы, то, выполняя ту или иную микропрограмму, можно обеспечить выполнение различных операций на одном и том же оборудовании операционного устройства (рисунок 7.8).

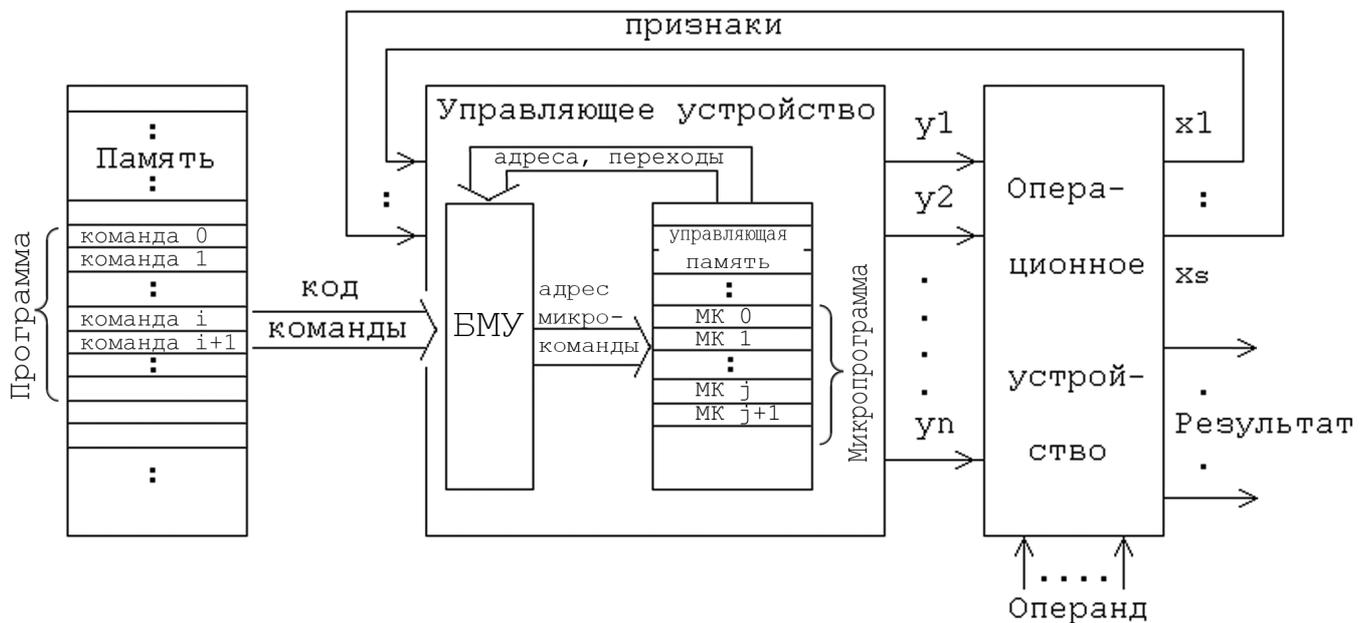


Рисунок 7.8

## 7.2 АРИФМЕТИЧЕСКО - ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ)

Комбинированная схема АЛУ выполняет арифметические и логические операции над числами в процессе межрегистровых пересылок. К входам АЛУ через буферные регистры временного хранения подключаются источники операндов в качестве которых могут выступать регистры, ячейки памяти либо порты ввода/вывода. Регистры временного хранения позволяют избежать сбоев и потери информации, когда один из регистров используется в качестве регистра - операнда и регистра - результата.

АЛУ непосредственно связано с регистром признаков, в соответствующих разрядах которого фиксируются особенности выполнения каждой операции:

- нулевой результат в аккумуляторе -  $Z$ ,
- перенос из старшего разряда -  $СУ$ ,
- знак результата -  $S$ ,
- паритет -  $P$ ,
- полуперенос -  $АС$ ;
- значение отдельных битов -  $T$ ,
- и другие признаки.

Наличие в МП регистра признаков обеспечивает выполнение условных переходов в зависимости от состояния признаков. АЛУ позволяет в процессе межрегистровых "пересылок с перекосом" выполнять операции сдвига информации на один разряд вправо или влево.

Обычно АЛУ реализует простейшие арифметические и логические операции (сложение, вычитание, сдвиги, сравнение, логическое умножение и другие операции). Все более сложные операции (умножение, деление, вычисление элементарных функций и др.) выполняются по подпрограммам.

### 7.3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

Как было показано выше, для обеспечения выполнения алгоритма, предусмотренного программой, необходимо правильно организовать порядок считывания кодов команд из памяти и выполнение их процессором. В качестве указателя адреса команды, подлежащей исполнению, используется специализированный регистр – счетчик команд (PC). Работа счетчика команд организована таким образом, что при считывании из памяти очередного байта команды, его значение автоматически увеличивается на единицу. Следовательно, вслед за текущей будет выполнена команда, расположенная в следующей ячейке памяти.

При организации ветвления алгоритма программы последовательный ход выполнения команд нарушается. Для осуществления перехода к команде, находящейся в произвольном месте программы, используют принудительную запись в счетчик команд адреса требуемой команды. Для организации программных переходов создана специальная группа команд ветвления или переходов. Переходы могут выполняться либо безусловно, либо при выполнении определенного условия, отраженного в регистре признаков, либо при вызове подпрограмм и выходе из них.

Цикл команды выполняется в течение нескольких периодов тактового генератора. В течение этого времени происходит чтение кода команды из памяти, декодирование, запрос операндов, собственно, выполнение операции, сохранение результата.

Для ускорения работы программы разработан и применяется конвейерный способ выполнения команд. Различные этапы соседних команд выполняются одновременно. При этом все предварительные действия происходят в то время, когда предыдущие команды находятся

на более поздних этапах своего цикла. Таким образом, создается впечатление, что на выполнение одной команды тратится значительно меньше время, чем на самом деле.

При наличии достаточно большого объема памяти в МП системе код адреса содержит значительное количество двоичных разрядов. Для снижения объема кода команд, использующих прямую адресацию памяти, используется метод страничной организации памяти, при котором адрес формируется из двух частей. Старшие разряды адреса хранятся в сегментном регистре, а младшие генерируются непосредственно при обращении к памяти. Полный исполнительный адрес образуется слиянием этих двух частей. Таким образом, все обращения к памяти и переходы возможны только в рамках текущей страницы. Для выхода за пределы страницы необходимо изменить содержимое сегментного регистра.

#### 7.4 СТЕКОВАЯ ПАМЯТЬ.

В МП организуется стековая память, реализующая безадресное задание операндов. В общем случае стек представляет собой группу последовательно пронумерованных ячеек памяти, снабженных указателем стека, в котором автоматически при записи и считывании устанавливается адрес последней занятой ячейки стека - вершины стека. При операции занесения в стек слово помещается в следующую по порядку свободную ячейку стека, а при считывании из стека извлекается последнее поступившее в него слово. Таким образом, в стеке реализуется принцип обслуживания "последний пришел - первый ушел". Этот порядок при обращении к стеку реализуется автоматически. Поэтому при операциях со стеком возможно безадресное задание операнда - команда не содержит адреса ячейки стека, но содержит адрес ячейки памяти или регистра, откуда слово передается в стек или куда загружается из стека.

Таким образом, стек является устройством памяти с последовательным доступом. Невозможно считать информацию, не считав стоящую перед ней.

Для организации стека в составе МП предусмотрен регистр - указатель стека (SP) и соответствующие цепи управления. Часто сам стек реализуется в виде группы последовательных ячеек оперативной памяти либо в виде специально организованной группы ячеек, не входящих ни в какой другой банк памяти.

Стековая адресация широко используется при работе с подпрограммами и в процедурах обработки прерывания. Основное назначение стека - сохранение содержимого счетчика команд при вызове подпрограмм.

## 7.5 РАБОТА УПРАВЛЯЮЩЕГО УСТРОЙСТВА МП В РЕЖИМЕ ПРЕРЫВАНИЯ

Периферийное оборудование МП - системы может запросить прерывание текущей программы у МП путем подачи сигнала запроса на вход прерывания. Сигнал прерывания может возникнуть в любой момент цикла команды. Обработка прерываний организована таким образом, что запрос прерывания фиксируется во внутреннем триггере запроса прерывания только при переходе к циклу очередной команды, и только в случае, если программным путем было разрешено прерывание. Выполнение этих условий приведет к тому, что следующей будет выполнена первая команда подпрограммы обработки запроса прерывания. Вызов подпрограммы происходит следующим образом. Текущее содержимое счетчика команд, соответствующее адресу следующей команды сохраняется в стеке. В счетчик команд загружается адрес первой команды подпрограммы обработки соответствующего прерывания - вектор прерывания. После завершения подпрограммы адрес возврата в основную программу восстанавливается в счетчик команд из стека.

Большинство МП допускают работу с несколькими источниками прерываний как внешнего, так и внутреннего происхождения. Для предотвращения конфликтов предусмотрена возможность установки уровней приоритета для различных источников.

## 7.6 РАБОТА МП В РЕЖИМЕ ПРЯМОГО ДОСТУПА К ПАМЯТИ

Управляющее устройство МП позволяет выполнять операции с прямым доступом к памяти (ПДП). При этом обмен данными осуществляется без участия процессора. Режим ПДП наиболее целесообразен при обмене блоками данных. В этом случае обмен данными между оперативной памятью и периферийным устройством производится быстрее, чем при программно - управляемом обмене. Выставляя на соответствующей линии запрос ПДП, периферийное устройство приостанавливает вычислительный процесс в МП и осуществляет управление по шинам адреса и данных. Первичный автомат при этом вырабатывает управляющие сигналы, переводящие буферные схемы шин адреса и данных МП в высокоимпедансное состояние. Таким образом реализуется принцип захвата шины МП - системой для ввода - вывода данных. Т.к. в режиме ПДП обмен данными происходит без участия МП, содержимое всех регистров МП остается неизменным.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Калабеков Б.А. Микропроцессоры и их применение в системах передачи и обработки сигналов: Учеб. пособие для вузов.- М: Радио и Связь 1988. - 368 с.

2 Каган Б.М., Сташин В.В. Основы проектирования микропроцессорных устройств автоматики.-М.:Энергоатомиздат,1987.-303 с.