

Лекция 7. Устройство управления

УУ - часть цифрового вычислительного устройства, предназначенная для формирования последовательности управляющих сигналов, которые определенным образом распределены во времени.



Лекция 7. Устройство управления

Выполнение команд происходит в виде последовательности операций (фаз).

Элементарную функциональную операцию, выполняемую за один тактовый интервал и приводимую в действие управляющим сигналом, называют *микрооперацией*. Приставка «микро» означает, что каждая операция очень проста и реализуется очень быстро.

Каждая фаза – набор микроопераций, который образует *микропрограмму*.

Лекция 7. Устройство управления

Совокупность микроопераций, выполняемых в одном такте, называют *микрокомандой* (МК).

Микрооперации: передача данных между регистрами (пересылки), передача данных из регистра на внешнюю магистраль, управление действиями АЛУ.

Устройство управления решает 2 задачи:

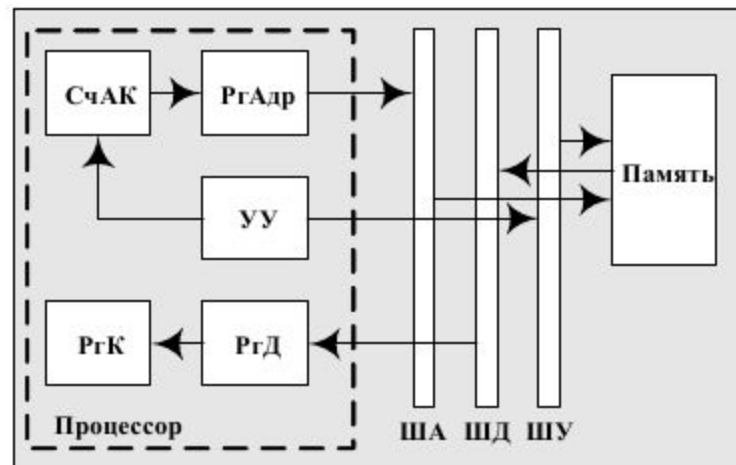
- Организация выполнения процессором микроопераций в нужной последовательности;
- Формирование управляющих сигналов, необходимых для выполнения каждой микрооперации.

Лекция 7. Устройство управления

Микрооперации УУ на фазе извлечения машинной команды

1. 1. $R_{Г\text{Адр}} \leftarrow (C_{ч\text{АК}})$;
2. $\left\{ \begin{array}{l} 2. R_{Г\text{Д}} \leftarrow \text{Память}; \\ 3. C_{ч\text{АК}} \leftarrow (C_{ч\text{АК}}) + I; \end{array} \right.$
3. 4. $R_{Г\text{К}} \leftarrow (R_{Г\text{Д}})$.

1. 1. $R_{Г\text{Адр}} \leftarrow (C_{ч\text{АК}})$;
2. 2. $R_{Г\text{Д}} \leftarrow \text{Память};$
3. $\left\{ \begin{array}{l} 3. C_{ч\text{АК}} \leftarrow (C_{ч\text{АК}}) + I; \\ 4. R_{Г\text{К}} \leftarrow (R_{Г\text{Д}}). \end{array} \right.$



Лекция 7. Устройство управления

Микрооперации УУ на фазе извлечения машинной команды

Для синхронизации операций используют единый тактовый генератор. Длительность каждого шага должна быть одинаковой. Каждая микрооперация должна быть реализована в течение 1 такта.

При группировании микроопераций существует 2 правила:

- Необходимо сохранять правильную последовательность действий;
- Группирование не должно приводить к конфликту использования ресурсов.

Лекция 7. Устройство управления

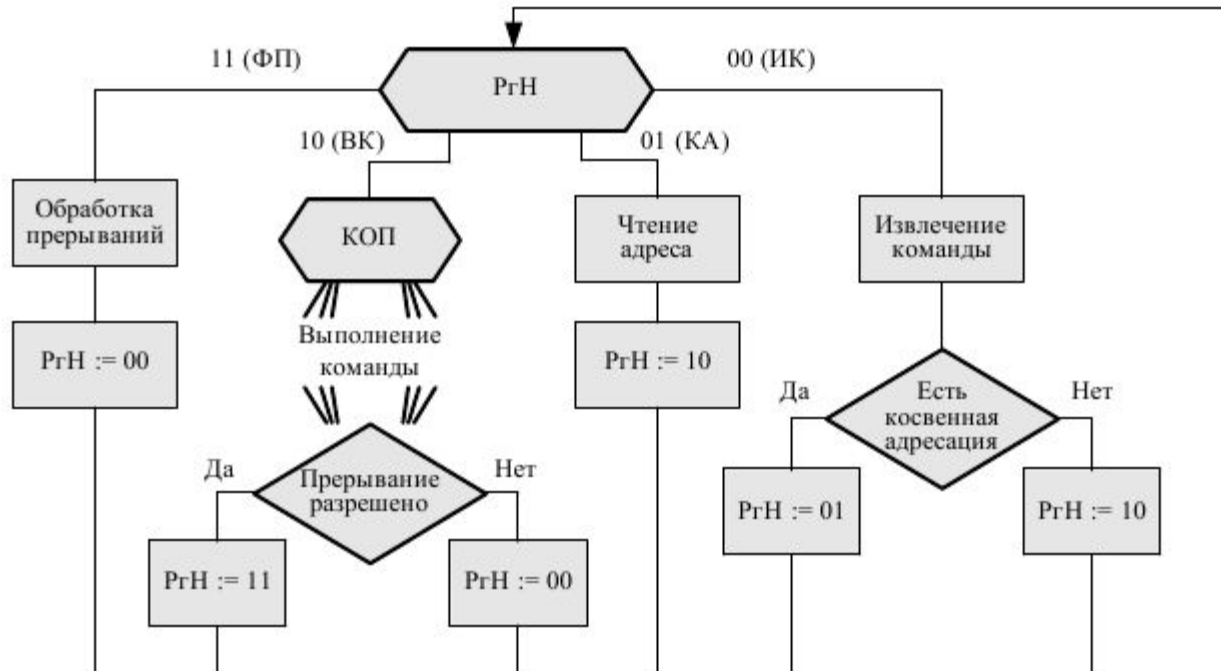
Управление циклом обработки машинной команды

ЦОК состоит из 4 фаз. В УУ вводим двухразрядный регистр номера фазы цикла (PгН), который принимает следующие значения:

- 00 - фаза извлечения команды,
- 01 - фаза косвенной адресации,
- 10 - фаза выполнения команды,
- 11 - фаза прерывания.

Лекция 7. Устройство управления

Управление циклом обработки машинной команды



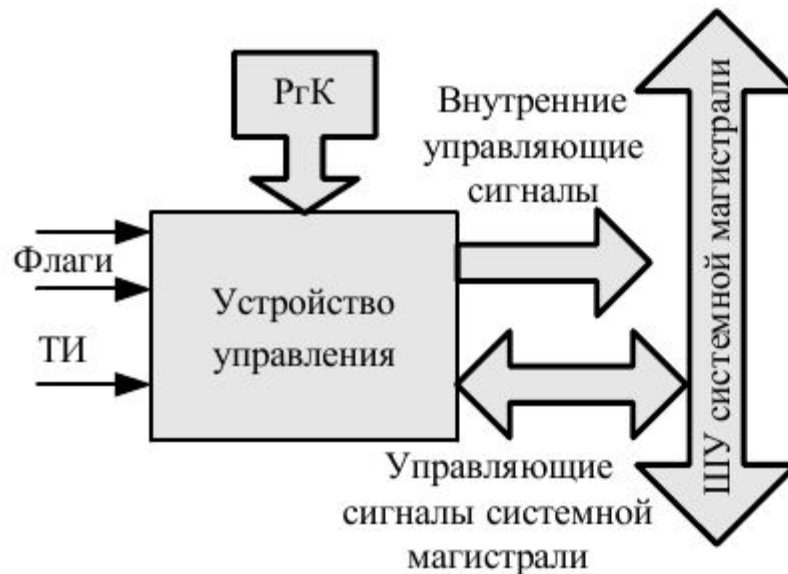
Лекция 7. Устройство управления

Процесс проектирования устройства управления

Три стадии:

1. Определение базовых функциональных элементов процессора;
2. Составление перечня микроопераций, которые процессор будет выполнять;
3. Определение требуемых от УУ действий для организации выполнения последовательности микроопераций.

Лекция 7. Устройство управления: управляющие сигналы



Для функционирования УУ должно получать со входа определенную информацию о текущем состоянии системы и формировать выходные управляющие сигналы, которые задают дальнейшее поведение системы.

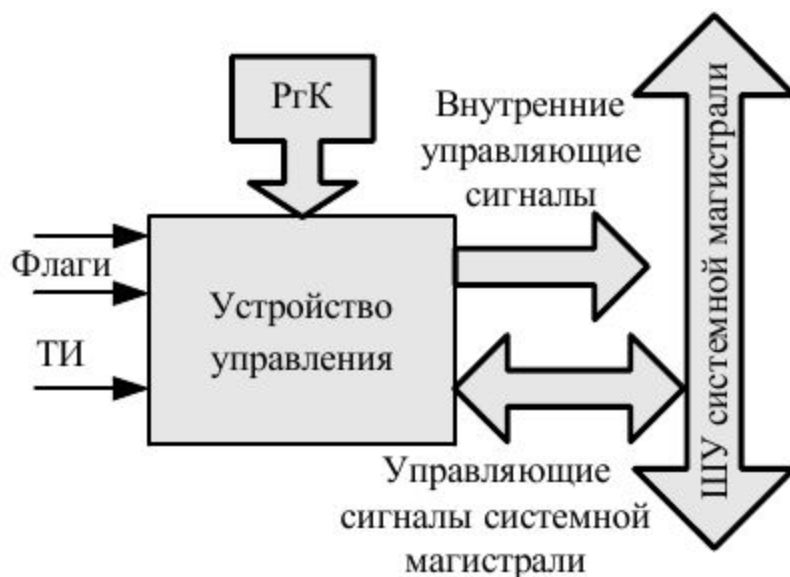
Лекция 7. Устройство управления: управляющие сигналы



Управляющие сигналы:

1. *Тактовые импульсы* синхронизируют выполнение всех операций во времени.
2. *Поле КОП* регистра команд определяет, какую последовательность микроопераций должно сформировать УУ.

Лекция 7. Устройство управления: управляющие сигналы



Управляющие сигналы:

3. *Флаги* необходимы для определения текущего состояния процессора и результата выполнения последней операции в АЛУ.
4. *Сигналы ШУ* системной магистрали, включающие запросы на прерывания и подтверждения от внешних устройств

Лекция 7. Устройство управления: управляющие сигналы

Управляющие сигналы, формируемые УУ:

1. УС, распространяемые внутри процессора:

- сигналы передачи данных из одного регистра в другой;
- сигналы запуска определенных операций в АЛУ.

2. *Сигналы, передаваемые по ШУ* системной магистрали:

- сигналы, управляющие работой памяти;
- сигналы, управляющие работой модулей ввода/вывода.

Лекция 7. Устройство управления: управляющие сигналы

Управление на фазе извлечения машинной команды:

1. Пересылка содержимого СчАК в РгАдр (УС С2).

2. Чтение слова из

памяти и

приращение кода

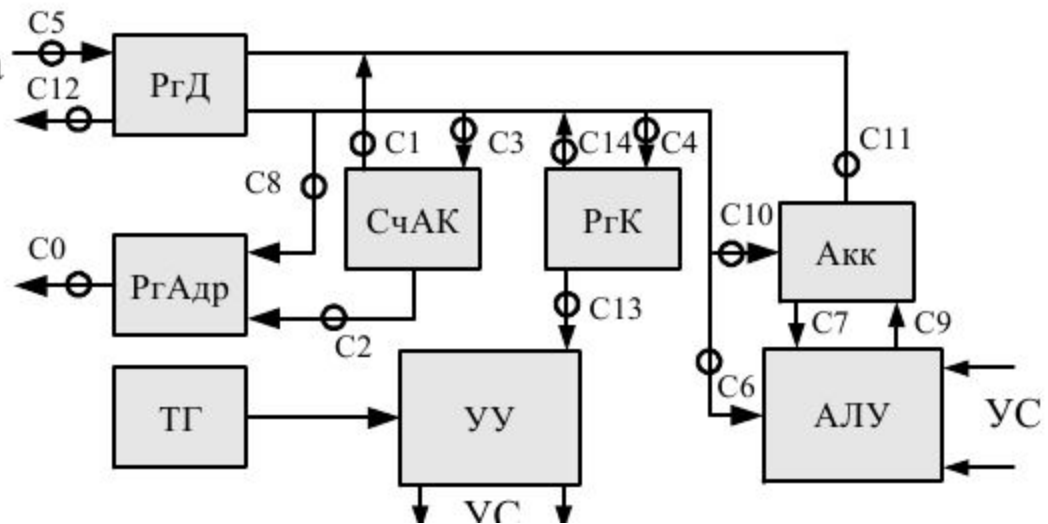
адреса СчАК

(- УС С0;

- УС Read;

- УС С5;

- УС С3).



Лекция 7. Устройство управления: управляющие сигналы

Управление циклом обработки команды:

Извлечение команды	$R_{Г\text{Адр}} \leftarrow (C_{\text{чАК}})$ $R_{Г\text{Д}} \leftarrow \text{Память}$ $C_{\text{чАК}} \leftarrow (C_{\text{чАК}}) + I$ $R_{Г\text{К}} \leftarrow R_{Г\text{Д}}$	C2 C5, Read C3 C4
Косвенная адресация	$R_{Г\text{Адр}} \leftarrow R_{Г\text{К}}(\text{Адр})$ $R_{Г\text{Д}} \leftarrow \text{Память}$ $R_{Г\text{К}}(\text{Адр}) \leftarrow R_{Г\text{Д}}(\text{Адр})$	C14, C8 C5, Read C4
Обработка прерываний	$R_{Г\text{Д}} \leftarrow R_{Г\text{К}}$ $R_{Г\text{Адр}} \leftarrow \text{Адр_сохр}$ $C_{\text{чАК}} \leftarrow \text{Адр_п/прогр}$ $\text{Память} \leftarrow (R_{Г\text{Д}})$	C1 C12, Write

Лекция 7. Устройство управления: управляющие сигналы

Управление циклом обработки команды:

УУ функционирует, зная только исполняемую команду и свойства результата выполнения предыдущей команды.

УУ не имеет доступа к значениям обрабатываемых данных и полученных результатов.

Лекция 7. Устройство управления: управляющие сигналы

АЛУ и все регистры процессора подключены к единой внутренней магистрали. Группы вентилях на входе и выходе каждого элемента обеспечивают подключение.

Достоинство: минимизация коммутационного пространства

Суммирование значения из памяти со значением аккумулятора:

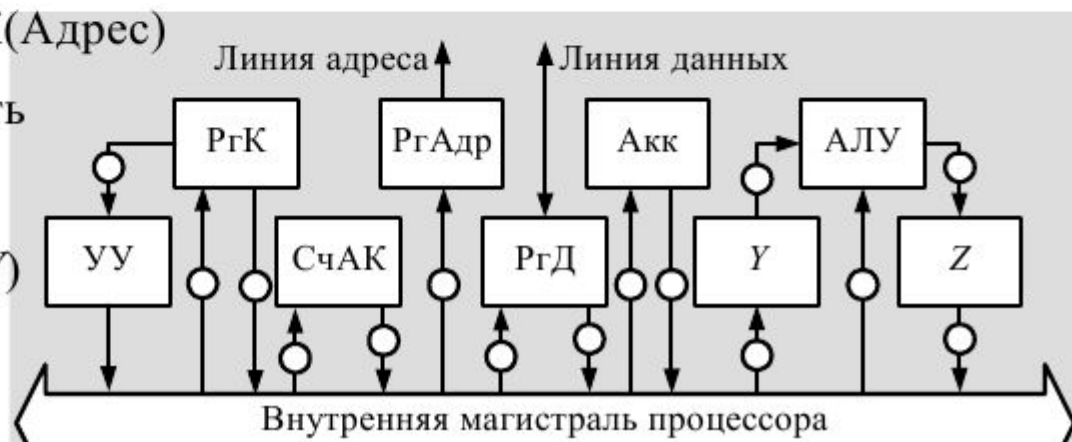
$t1: R_{Г\text{Адр}} \leftarrow R_{Г\text{К}}(\text{Адрес})$

$t2: R_{Г\text{Д}} \leftarrow \text{Память}$

$t3: Y \leftarrow (R_{Г\text{Д}})$

$t4: Z \leftarrow (\text{Акк}) + (Y)$

$t5: \text{Акк} \leftarrow (Z)$



Лекция 7. Устройство управления

Устройство управления



С жесткой логикой

В виде конечного автомата, формирующего выходные сигналы управления в зависимости от текущего состояния и значений осведомительных входных сигналов

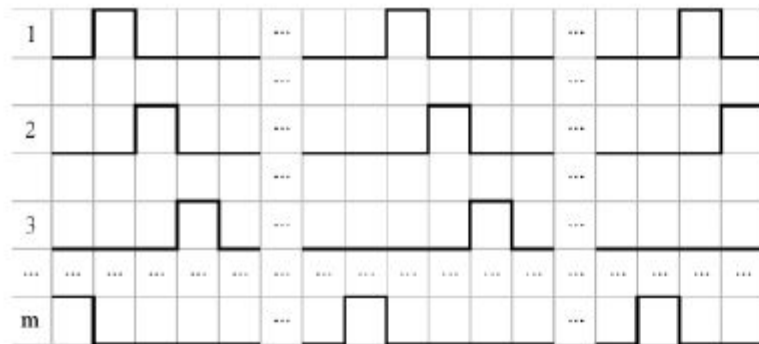


Микропрограммного управления

В виде устройства выбора из микропрограммной памяти микрокоманд, содержащих набор управляющих сигналов, в нужной последовательности и загрузки их в регистр микрокоманд.

Лекция 7. Устройство управления: с жесткой логикой

- **датчик сигналов** вырабатывает последовательность импульсов, равномерно распределенную во времени (m - количество тактов, за которое выполняется самая длинная операция);



Лекция 7. Устройство управления: с жесткой логикой

- *дешифратор кода операций*, декодирует код операции команды, присутствующей в данный момент в РгК, и возбуждает одну выходную шину, соответствующую данной операции; этот сигнал использует блок управления операциями для выработки нужной последовательности управляющих сигналов.



Лекция 7. Устройство управления: с жесткой логикой

- *блок управления операциями*

осуществляет выработку управляющих сигналов $УС_1, УС_2, \dots, УС_N$ в соответствии со значениями флагов, активного сигнала с дешифратора кода операции и номера такта с датчика сигналов.



Лекция 7. Устройство управления: с жесткой логикой

Пример построения УУ для управляющего сигнала C5.

P Q

0 0 - фаза извлечения

0 1 - фаза косвенной адресации

1 0 - фаза выполнения

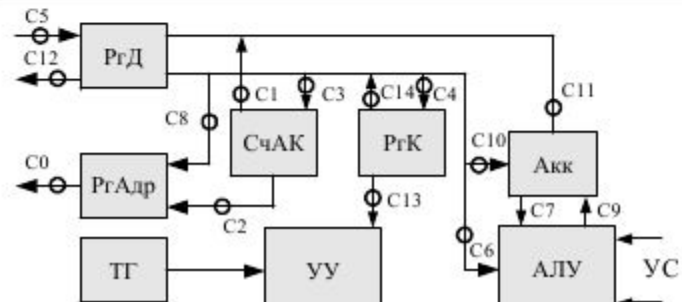
1 1 - фаза обработки прерываний

$$C5 = \bar{P} \bar{Q} T_2 + \bar{P} Q T_2 .$$

Если C5 должен быть установлен при выполнении команд(ы), то

$$C5 = \bar{P} \bar{Q} T_2 + \bar{P} Q T_2 + P \bar{Q} T_3 Op.$$

Извлечение команды	РгАдр ← (СчАК) РгД ← Память СчАК ← (СчАК) + I РгК ← РгД	C2 C5, Read C3 C4
Косвенная адресация	РгАдр ← РгК(Адр) РгД ← Память РгК(Адр) ← РгД(Адр)	C14, C8 C5, Read C4
Обработка прерываний	РгД ← РгК РгАдр ← Адр_сохр СчАК ← Адр_п/прогр Память ← (РгД)	C1 C12, Write



Лекция 7. Устройство управления: с жесткой логикой

Этапы синтеза

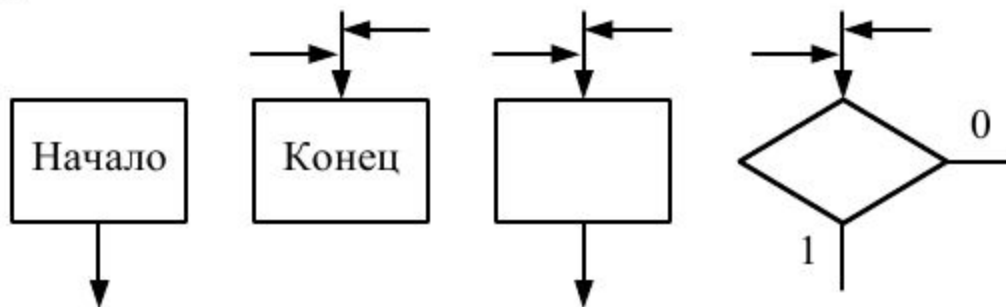
1. Выбор типа логических и запоминающих элементов, в базисе которых будет реализован управляющий автомат;
2. Кодирование состояний автомата, с учетом множества реализуемых процессором функций и сигналов управления операционного автомата;
3. Синтез комбинационной схемы, формирующей выходные управляющие сигналы.

Внутреннюю логику УУ описывают как булеву функцию входных сигналов (систему логических функций) с последующей оптимизацией.

Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

Граф-схема алгоритма (ГСА) – ориентированный связный граф, содержащий одну начальную вершину A_0 , одну конечную вершину A_k и произвольное конечное множество условных $\{\mathbf{P} = \{p_1, \dots, p_F\}\}$ и операторных $\{\mathbf{A} = \{A_1, \dots, A_G\}\}$ вершин.



Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

ГСА удовлетворяет следующим условиям:

1. Входы и выходы вершин соединяют друг с другом дугами, направленными всегда от выхода ко входу.
2. Каждый выход соединен точно с одним входом.
3. Любой вход соединен как минимум с одним выходом.
4. Любая вершина графа лежит по крайней мере на одном пути из начальной вершины к конечной вершине.

Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

ГСА удовлетворяет следующим условиям:

5. Один из выходов условной вершины может быть соединен с ее входом, что недопустимо для операторной вершины.
6. В каждой условной вершине записывают один из элементов множества логических условий $\{\mathbf{X} = |x_1, \dots, x_L|\}$. В разных условных вершинах можно записывать одинаковые элементы множества \mathbf{X} .

Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

ГСА удовлетворяет следующим условиям:

7. В каждой операторной вершине записывают оператор (микрокоманду) Y_t – подмножество множества микроопераций $\mathbf{Y} = \{y_1, \dots, y_N\}$. $Y_t = \{y_{t1}, \dots, y_{tu}, \dots, y_{tU_t}\}$, $y_{tu} \in \mathbf{Y}$, $u=1, \dots, U_t$. Разрешено записывать в различных операторных вершинах одинаковые подмножества множества операций.

Если в операторных вершинах ГСА стоят разные операторы, то их можно отождествить с записанным в них оператором (в вершине A_i записан оператор Y_i) и использовать Y_i вместо обозначения A_i .

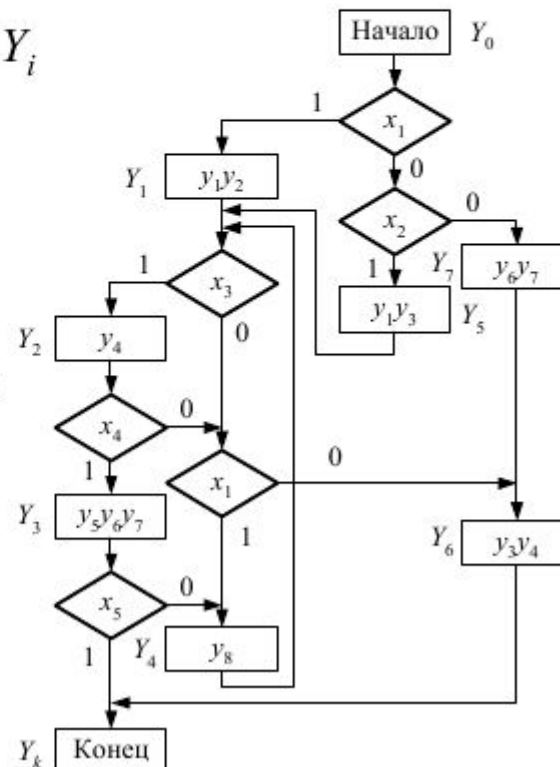
Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

Пусть ГСА имеет путь из вершины Y_i
($i = 0, 1, \dots, T$) в вершину Y_j
($j = 1, \dots, T+1$):

$$Y_i p_{i1}^{e_{i1}} \dots p_{ir}^{e_{ir}} \dots p_{iR}^{e_{iR}} Y_j,$$

проходящий только через условные
вершины p_{i1}, \dots, p_{iR} ; $e_{ir} \in [0, 1]$ –
символ, приписанный выходу
условной вершины.



Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

Каждому такому пути соответствует

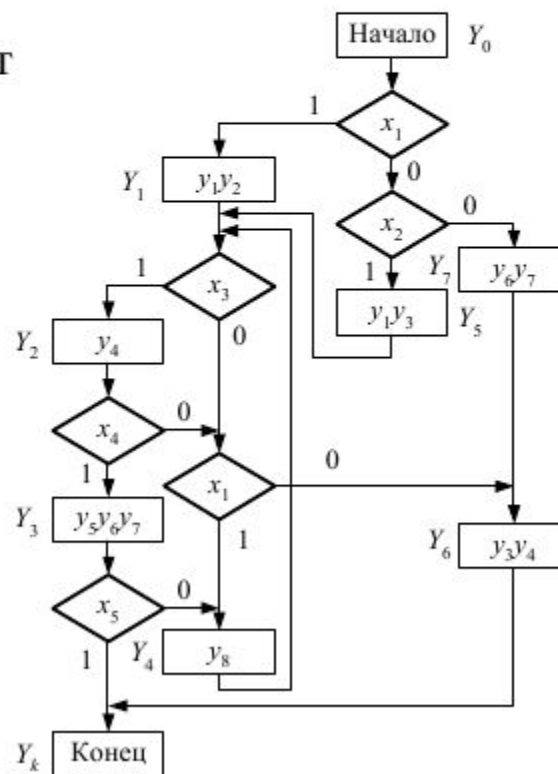
конъюнкция:

$$\alpha_{ij} = x_{i1}^{e_{i1}} \dots x_{iR}^{e_{iR}} = \prod_{r=1}^R x_{ir}^{e_{ir}},$$

где x_{ir} – логическое условие,

записанное в условной вершине p_{ir}

$$x_{ir}^0 = \bar{x}_{ir}, x_{ir}^1 = x_{ir}.$$



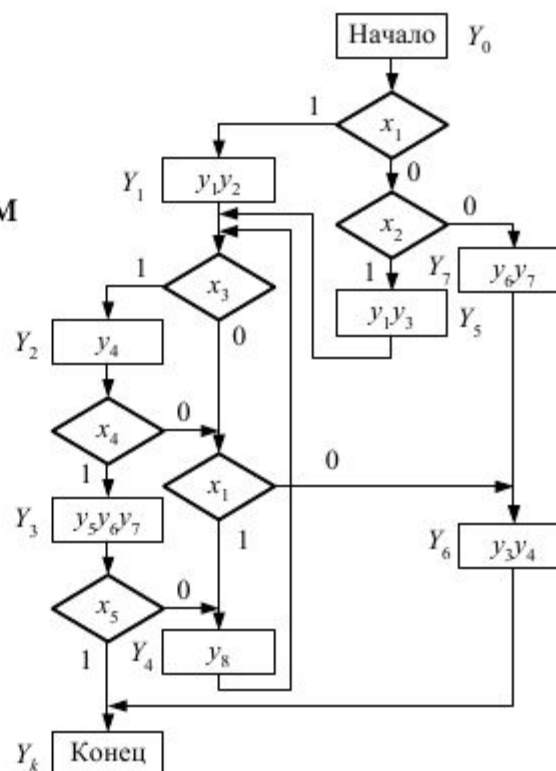
Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

Если между вершинами Y_i и Y_j H путей, то α_{ij} равно дизъюнкции конъюнкций, соответствующих всем путям:

$$\alpha_{ij} = \sum_{h=1}^H \alpha_{ir}^h$$

α_{ij} – функция перехода от оператора (микрокоманды) Y_i к оператору (микрокоманде) Y_j .



Лекция 7. Устройство управления: с жесткой логикой

Кодирование состояний

Обычно при проектировании цифровых устройств предварительно составляют *содержательные граф-схемы алгоритмов*, в которых внутри условных и операторных вершин записывают не элементы множеств X и Y , а логические условия и микрооперации в содержательных терминах.

Пример. Для операции деления S / Z . АЛУ содержит сумматор S , где находится делимое, регистр Z (делитель) и регистр Y , где сохраняют частное. СТ – счетчик тактов, ТП – триггер переполнения.

Лекция 7. Устройство управления: с жесткой логикой

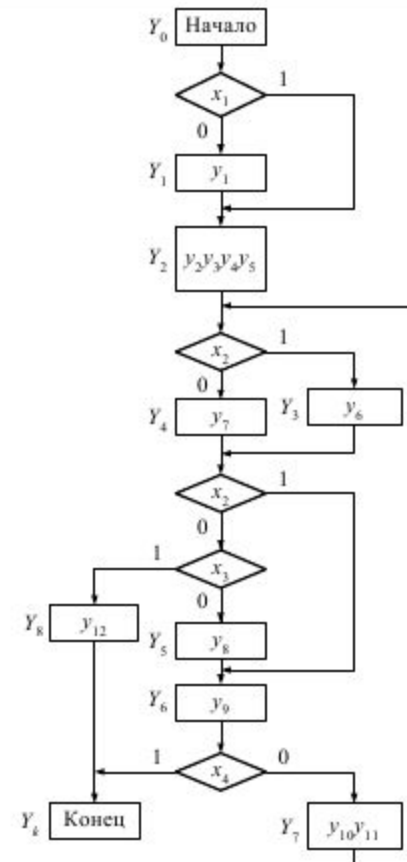
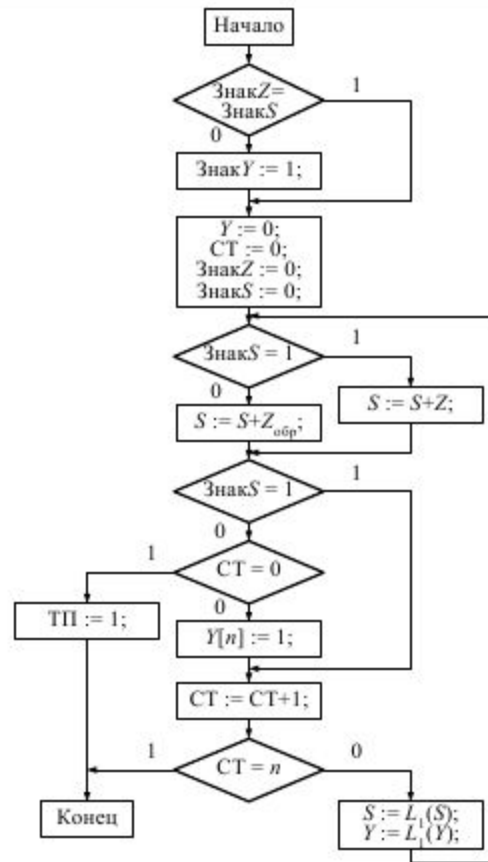
Условные вершины:

- x_1 : Знак S = Знак Z
- x_2 : Знак S = 1
- x_3 : $CT = 0$
- x_4 : $CT = n$

Операторные вершины:

- y_1 : Знак $Y := 1$;
- y_2 : $Y := 0$;
- y_3 : $CT := 0$;
- y_4 : Знак $Z := 0$;
- y_5 : Знак $S := 0$;
- y_6 : $S := S + Z$;
- y_7 : $S := S + Z_{обр}$;
- y_8 : $Y[n] := 1$;
- y_9 : $CT := CT + 1$;
- y_{10} : $S := L_1(S)$;
- y_{11} : $Y := L_1(Y)$;
- y_{12} : $ТП := 1$.

© С. Г. Мосин, 2007



Лекция 7. Устройство управления: с жесткой логикой

Недостатки:

1. Большое количество логических выражений и переменных в них;
2. С возрастанием сложности снижается быстродействие;
3. Необходимость заново проектировать управляющий автомат при расширении системы команд процессора.

Достоинства:

1. Максимальное быстродействие;
2. Компактная реализация на аппаратном уровне.

Лекция 7. Устройство управления: микропрограммное управление

Основная идея: для инициализации любой микрооперации достаточно сформировать определенный сигнал управления на соответствующей линии управления, т.е. активизировать эту линию. Этот процесс можно представить с помощью назначения двоичных состояний – логической единицы (активное значение линии) и логического нуля (пассивное значение линии).

Для указания микроопераций, выполняемых за один такт, формируют управляющее слово, в котором каждый бит соответствует одной определенной управляющей линии.

Лекция 7. Устройство управления: микропрограммное управление

Такое управляющее слово называют *микрокомандой*.

Последовательность микрокоманд, реализующих определенную фазу цикла обработки машинных команд, называют *микропрограммой*.

Микропрограммы хранят в *микропрограммной памяти (МПП)*.

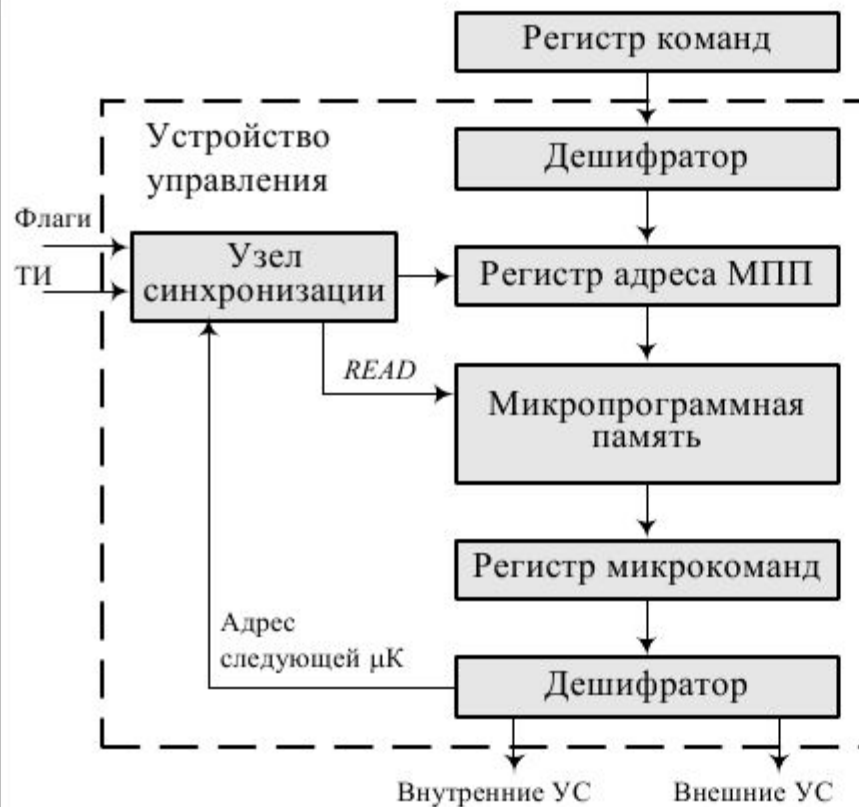
Формирование управляющих сигналов сводят к последовательному извлечению микрокоманд из МПП и коммутации выходов регистра микрокоманд с линиями управления.

Лекция 7. Устройство управления: микропрограммное управление

В общем случае *микрокоманда* включает следующие поля:

- поле внутренних сигналов управления (по 1 разряду на каждый управляющий сигнал);
- поле внешних управляющих сигналов;
- поле условий, в котором указывают код условия перехода;
- поле адреса следующей микрокоманды при условном переходе.

Лекция 7. Устройство управления: микропрограммное управление



- *Регистр команд* содержит поле КОП;
- *Регистр адреса МПП* содержит адрес следующей МК;
- *МПП* включает весь набор микропрограмм;

Лекция 7. Устройство управления: микропрограммное управление



- *Регистр* μK содержит текущую микрокоманду;
- *Узел синхронизации* загружает адрес в регистр адреса МПП и формирует сигнал чтения МПП.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды

В зависимости от значений флагов АЛУ и кода условий микрокоманды в регистре микрокоманд существует три возможных варианта формирования адреса следующей μ К:

- Извлечь μ К, размещенную по следующему адресу в МПП. Для этого необходимо увеличить на 1 содержимое регистра адреса МПП;
- Перейти на новую микропрограмму. Содержимое поля адреса текущей микрокоманды загружают в регистр адреса.
- Перейти на микропрограмму выполнения определенной машинной команды. В регистр адреса загружают адрес МПП, соответствующий КОП в регистре команд.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды

Первый вариант встречается чаще всего и обусловлен естественным порядком следования микрокоманд при выполнении микропрограмм.

Второй вариант используют только один раз в цикле обработки машинной команды по окончании фазы извлечения команды.

Третий вариант связан с нарушением естественного порядка следования микрокоманд в силу наличия условных и безусловных переходов в микропрограммах.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды

Выбор средств управления последовательностью микрокоманд оказывает влияние на два базовых параметра УУ:

- размер микрокоманды;
- время формирования адреса микрокоманды, т.е. на быстродействие УУ.

При формировании адреса следующей μ К УУ анализирует следующую информацию:

- содержимое текущей μ К;
- состояние флагов;
- содержимое регистра команд.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды

Методы формирования разделяют на три категории в соответствии с форматом представления адресной информации в микрокоманде:

- микрокоманда с двумя полями адреса;
- микрокоманда с единственным полем адреса;
- микрокоманда переменного формата.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды



Микрокоманда с двумя полями адреса:

Мультиплексор в зависимости от значения сигнала «Выбор адреса» передает на выход либо код одного из двух полей адреса микрокоманды, либо адрес микрокоманды, соответствующий КОП в регистре команд.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды



Микрокоманда с единственным полем адреса:

Мультиплексор в зависимости от значения сигнала «Выбор адреса» передает на выход либо код поля адреса микрокоманды, либо содержимое регистра адреса, увеличенное на 1, либо адрес микрокоманды, соответствующий КОП в регистре команд.

Лекция 7. Устройство управления: микропрограммное управление

Формирование адреса следующей микрокоманды



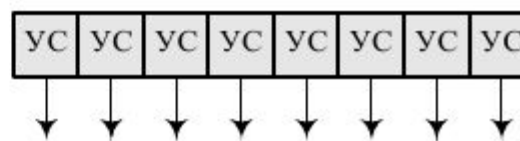
Микрокоманда переменного формата:

Специальный разряд определяет вариант формата $\mu\text{К}$. В одном оставшиеся разряды – управляющие сигналы, в другом – часть разрядов определяют условие ветвления, а другая часть задает адрес перехода.

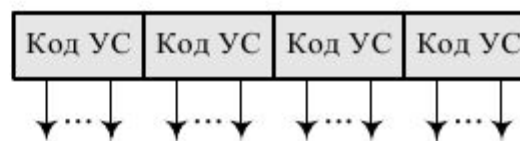
Лекция 7. Устройство управления: микропрограммное управление

Классификация микрокоманд:

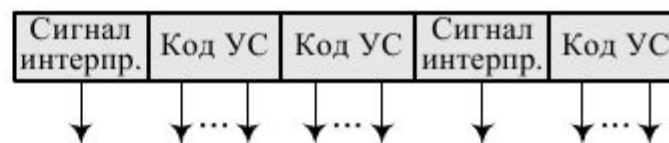
- *горизонтальные и вертикальные* относят к длине $\mu\text{К}$;



- *жесткие и мягкие* отражают особенности аппаратных средств;



- *прямые и косвенные* определяют правила интерпретации полей $\mu\text{К}$.



Лекция 7. Устройство управления: микропрограммное управление

Кодирование микрокоманд обеспечивает сокращение длины $\mu\text{К}$, что приводит к необходимости меньшей по объему МПП, или увеличению объема микропрограммного обеспечения, хранимого в МПП.

Как кодировать $\mu\text{К}$? Пусть УУ должно формировать в совокупности K внутренних и внешних УС. Следовательно можно выделить K битов управляющего слова, что обеспечивает кодирование 2^K возможных комбинаций.

Лекция 7. Устройство управления: микропрограммное управление

Однако не все комбинации используются:

- два источника информации не могут быть одновременно подключены через вентили к одному и тому же приемнику;
- один и тот же регистр не может быть одновременно и источником и приемником информации;
- только один вариант УС может быть передан в каждый момент времени в АЛУ;
- только один вариант внешних сигналов в каждый момент времени может быть передан на внешние устройства.

Лекция 7. Устройство управления: микропрограммное управление

Таким образом, возможно активизировать лишь Q состояний, причем $Q < 2^K$, т.е. для кодирования понадобится $\log_2 Q < K$, что обеспечивает самую компактную форму кодирования.

Но на практике она не применима по следующим причинам:

- сложность программирования;
- требует включения в состав устройства сложных, а значит менее быстродействующих дешифраторов.

Компромисс: увеличивать количество разрядов для представления управляющего слова по сравнению с минимально необходимым $\log_2 Q$.

Лекция 7. Устройство управления: микропрограммное управление

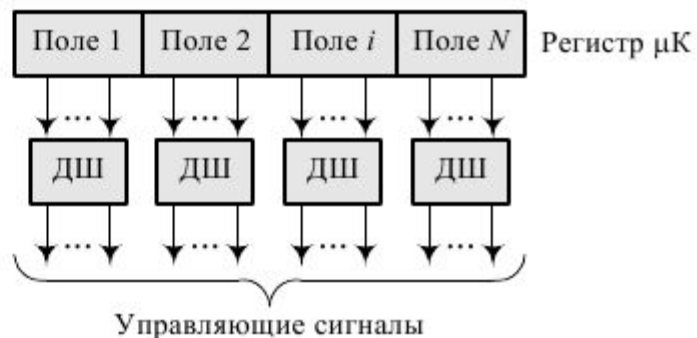
Технология кодирования микрокоманд

Базовый метод кодирования.

Управляющее слово $\mu\text{К}$
представляют в виде набора
полей.

Каждое поле содержит код,
определяющий состояние
группы управляющих сигналов.

Если N полей, то одновременно активизируются N функций.



Лекция 7. Устройство управления: микропрограммное управление

Проектирование формата микрокоманд:

- 1) Разделить поле управляющего слова $\mu\text{К}$ на отдельные поля. Каждое поле отвечает за определенный набор операций, причем операции разных групп могут быть активизированы одновременно.
- 2) Операции внутри каждой группы должны быть взаимно исключаящими, т.е. только одна из них должна активизироваться при выполнении любой $\mu\text{К}$.

Лекция 7. Устройство управления: микропрограммное управление

Разбиение микрокоманд на поля:

- 1) Метод функционального кодирования** предполагает составление перечня элементарных функций. Данный перечень разбивают на группы, причем функции в каждой группе должны быть взаимно исключающими. Каждой группе ставят в соответствие определенное поле $\mu\text{К}$.
- 2) Метод ресурсного кодирования** подразумевает выделение полей $\mu\text{К}$ каждому устройству из множества независимых ресурсов, из которых состоит компьютер.