

**1. ВВЕДЕНИЕ**

**Базовая структура микропроцессорной системы**

Обобщенная структурная схема микропроцессорного устройства (МПУ) изображена на рис. 1.1. В общем случае, работа МПУ сводится к следующей последовательности действий: получение данных от различных внешних устройств (ВУ) (с клавиатуры, из каналов связи, внешних запоминающих устройств и др.), обработка данных и выдача результатов обработки на ВУ. При этом данные от ВУ, подлежащие обработки, могут поступать и в процессе их обработки.

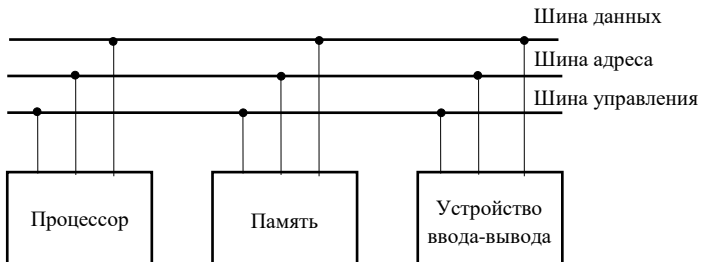


Рис.1.1. Обобщенная структурная схема микропроцессорного устройства

Для выполнения этих действий в МПУ кроме микропроцессора предусмотрены следующие устройства: память, предназначенная для хранения программ, определяющих работу микропроцессора, и различных данных (исходных данных, промежуточных и конечных результатов обработки данных в микропроцессоре), устройства ввода-вывода, обеспечивающие обмен данными различных ВУ с микропроцессором и памятью. Кроме того, имеются линии связи, объединенные по функциональному назначению в шины: шина данных, адресная шина и шина управления.

В ходе работы микропроцессор выдает на шину адреса номер (адрес) ячейки памяти, в которой хранится очередная команда, а по шине управления в память поступают сигналы, обеспечивающие чтение содержимого ячейки памяти. Запрошенная команда выдается на шину данных, откуда она принимается в МП. Далее команда расшифровывается, и если данные, действие над которыми преду-

сматривает команда, находясь в регистрах МП, то он приступает к выполнению указанной в команде операции. Если при расшифровке команды выясняется, что участвующие в операции данные находятся в памяти, то МП опять выдает на адресную шину адрес ячейки, хранящей эти данные, в сопровождении с управляющим сигналом чтения данных, и после выдачи их из памяти на шину данных считывает и выполняет указанную в команде операцию. После завершения выполнения текущей команды на шину адреса выдается адрес следующей команды и описанный процесс повторяется.

В процессе работы микропроцессорного устройства может потребоваться выдача результата обработки данных на внешние устройства, либо прием данных от них. Для этого группа ВУ подключается к шинам МПУ через устройство ввода-вывода (устройство сопряжения), обеспечивающее необходимые уровни сигналов и управляющее процессом обмена данными в соответствии с так называемым протоколом обмена данными. Если устройства сопряжения программируемые, то до начала обмена МП должен выдать туда информацию о режимах, используемых при передаче, направлениях передачи данных и другие установки. Затем, в момент, когда потребуется, например, передать поступающие из ВУ данные в оперативную память, МП, выполняя команду ввода данных, подает на устройство ввода-вывода соответствующие управляющие сигналы. Данные из ВУ принимаются в регистр устройства ввода-вывода, откуда они поступают на шину данных. Далее, эти данные с шины принимаются микропроцессором, после чего в процессе выполнения другой команды они передаются в оперативную память. Аналогично происходит обмен данными в обратном направлении.

Описанные способы обмена данными предполагают, что моменты обмена известны заранее уже на этапе программирования, и в программе предусматриваются в определенных местах соответствующие команды. Однако существуют ситуации, например, в случае аварии, когда обмен данными должен определяться самим ВУ. В этих случаях ВУ, подавая в МП определенные сигналы, переводит его в режим так называемого прерывания. В этом режиме МП прекращает выполнение основной программы и переходит к

исполнению команд другой, находящейся в памяти программе обработки прерывания, обеспечивающей обмен данными по запросу ВУ. После окончания выполнения такой программы МП возвращается к продолжению основной программы.

Эти способы обеспечивают низкую скорость обмена данными и применять их целесообразно лишь с низкоскоростными ВУ. При работе с высокоскоростными ВУ используется так называемый режим прямого доступа к памяти (ПДП). В этом режиме МП отключается от шин, предоставляя их в распоряжение ВУ для непосредственного обмена данными с памятью (без участия МП). Обмен при этом организуется специальным контроллером ПДП. В режиме ПДП ВУ для большей эффективности обменивается не одиночными данными, а блоками данных. Перед обменом в таком режиме, МП предварительно в контроллер ПДП помещает информацию, необходимую для управления обменом: адрес ячейки, откуда или куда начинается передача данных, размер блока данных и др. В процессе обмена контроллер ПДП выдает на шину адрес ячейки памяти, данные и управляющие сигналы, т.е. заменяет собой таким образом микропроцессор. После завершения передачи заданного размера блока данных контроллер ПДП извещает об этом МП и тот, в свою очередь, восстанавливает связь с шинами и продолжает выполнение основной программы.

## **2. ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОЦЕССОРА**

### **Общая структура процессора**

Процессор предназначен для обработки данных, он состоит из двух устройств - операционного и управляющего (см. рис. 2.1). Операционное устройство (ОУ) - устройство, в котором выполняются определенные операции. Оно включает в себя в качестве узлов регистры, сумматор, каналы передачи данных, мультиплексоры для коммутации каналов, шифраторы, дешифраторы и другие узлы, в зависимости от заданных требований к процессору.

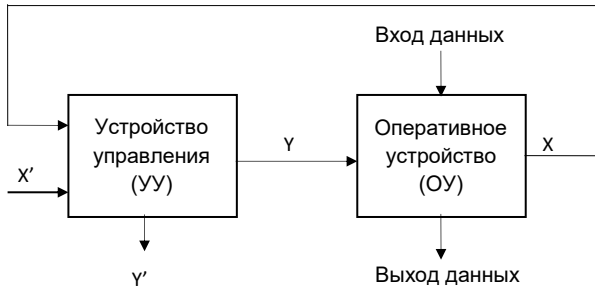


Рис.2.1. Общая структура процессора

Управляющее устройство (УУ) координирует действия узлов ОУ. Оно вырабатывает в определенной временной последовательности управляющие сигналы  $Y(X, X')$ , под действием которых в узлах ОУ выполняются требуемые действия. Процессор имеет внешние выходы, по которым поступают данные и в обработанном виде выдаются для дальнейшего использования, а также входные и выходные линии управления  $X'$  и  $Y'$ , соответственно, связанные с внешними устройствами. На рис. 2.1 все изображенные линии обозначают шины, которые могут иметь различную ширину (различное число линий).

Последовательность функционирования ОУ распадается на последовательность элементарных действий в его узлах. Такими элементарными действиями могут быть:

- 1) установка регистра в некоторое состояние:  
 $R1 \leftarrow 0;$
- 2) инвертирование разрядов содержимого регистра:  
 $R2 \leftarrow \underline{R2};$
- 3) пересылка содержимого одного регистра в другой:  
 $R1 \leftarrow R2;$
- 4) сдвиг содержимого регистра влево:  
 $R1 \leftarrow \text{Сдвл}(R1);$
- 5) счет, при котором число в регистре возрастает на 1:  
 $\text{Сч} \leftarrow (\text{Сч})+1;$

б) сложение:

$$R2 \leftarrow (R2) + (R1);$$

7) сравнение содержимого регистра на равенство с некоторым числом: результат - либо 0, либо 1;

8) некоторое логическое действие: поразрядное выполнение операции или конъюнкции, или дизъюнкции, или др.

Каждое такое элементарное действие, выполняемое в одном из узлов операционного устройства в течение одного тактового периода, называется микрооперацией. В определенные тактовые периоды одновременно могут выполняться несколько микроопераций. Такая совокупность одновременно выполняемых микроопераций называется микрокомандой. А весь набор микрокоманд, предназначенный для выполнения определенной операции - микропрограммой.

Если в ОУ предусматривается возможность выполнения  $n$  различных операций, то из УУ выходят  $N$  управляющих цепей, каждая из которых соответствует определенной микропрограмме. И для выполнения некоторой микропрограммы достаточно по управляющей цепи, соответствующей данной операции, подать сигнал, например, напряжение уровня логической 1. Для сокращения числа управляющих цепей, выходящих из УУ (когда оно выполнено отдельно от ОУ), операции могут определенным образом кодироваться.

### **Проектирование устройства управления**

Существует два подхода к проектированию управляющего устройства: использование схемной логики (аппаратный) и использование программируемой логики (микропрограммный).

В первом случае, из набора интегральных микросхем или, используя микроэлектронную технологию, создаются такие схемы, которые обеспечивают требуемое функционирование УУ. В качестве их узлов используются комбинационные и последовательностные схемы для выработки требуемых управляющих сигналов.

Комбинационные схемы строятся на основе логических элементов. Эти схемы характеризуются отсутствием памяти, и поэто-

му их выходные сигналы определяются только входными (рис. 2.2). Алгоритм работы устройств на основе комбинационных схем может быть представлен в виде таблиц соответствия, содержащей значения выходных сигналов для всех возможных комбинаций значений входных сигналов.

Последовательностные схемы отличаются от комбинационных схем, прежде всего, наличием памяти. Их выходные сигналы являются функцией не только входных сигналов, но и функцией внутреннего состояния, в котором устройство находилось до поступления входных сигналов. На рис. 2.3 показана логическая организация этих схем. Память в них может быть любого типа - здесь она представляет собой совокупность триггеров  $T_1, T_2, \dots, T_p$ . Такая архитектура является универсальной и может рассматриваться в качестве модели любой цифровой системы независимо от ее сложности.

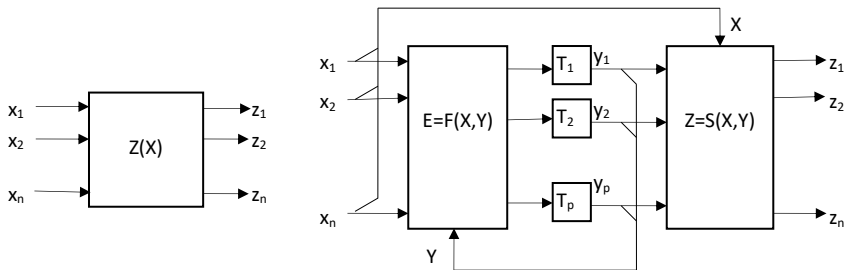


Рис. 2.2. Комбинационная схема      Рис. 2.3. Последовательностная схема

Связь между входными и выходными сигналами описывается следующим образом. Имеется  $n$  двоичных входов  $x_1 \dots x_n$ , т.е. существует  $2^n$  комбинаций, каждая из которых может рассматриваться как вектор:

$$X = [x_1, x_2, \dots, x_n], \text{ где } x_i - \text{либо } 0, \text{ либо } 1.$$

Имеется  $m$  двоичных выходов  $z_1 \dots z_m$ . Выходной вектор имеет вид:

$$Z = [z_1, z_2, \dots, z_m], \text{ где } z_i - \text{либо } 0, \text{ либо } 1.$$

Выходной вектор памяти, определенный на выходе триггеров задается в следующем виде:

$$Y = [y_1, y_2, \dots, y_p], \text{ где } y_i - \text{ либо } 0, \text{ либо } 1.$$

Связь между векторами может быть представлена следующими функциями:

$$\text{выходов } Z = S(X, Y);$$

$$\text{возбуждения } E = F(X, Y)$$

$$\text{переходов } Y' = G(X, Y)$$

Функция переходов G, определяющая выходной вектор памяти - это не булевская функция, описывающая комбинационную схему, а уравнение следующего состояния. Y' - это состояние памяти, в которое она перейдет из текущего состояния Y при поступлении на вход схемы сигналов.

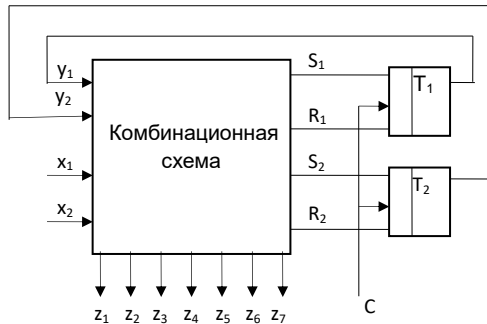


Рис. 2.4. Структурная схема управляющего устройства на основе схемной логики для процессора умножения целых чисел

Устройства, построенные на основе таких схем, обладают наивысшим быстродействием, однако узкая специализация и, следовательно, малосерийность делает их дорогостоящими. На рис. 2.4 изображено управляющее устройство простого процессора, выполняющего умножение n-разрядных целых чисел без знака.

Другой подход предполагает построение устройства управления с использованием одной или нескольких больших интегральных схем (БИС) универсального устройства, в котором требуемое функционирование обеспечивается занесением в управляющую

память устройства определенной микропрограммы. В этом случае число БИС, необходимых для построения УУ, небольшое, и это обеспечивает целесообразность их выпуска промышленностью массовыми тиражами.

В этом случае управляющие сигналы записываются в управляющую память в виде микрокоманд. При этом, выполнение операции сводится к последовательной выборке из управляющей памяти микрокоманд и выдачи их в операционное устройство в виде сигналов  $z_1...z_m$ . В управляющей памяти можно хранить много микропрограмм, предназначенных для выполнения различных операций. Таким образом устроены современные микропроцессоры, выполняющие сотни операций. По выбранной из оперативной памяти инструкции (команды) в управляющей памяти находится соответствующая микропрограмма, микрокоманды которой последовательно считываются в ОУ для выполнения заданной операции.

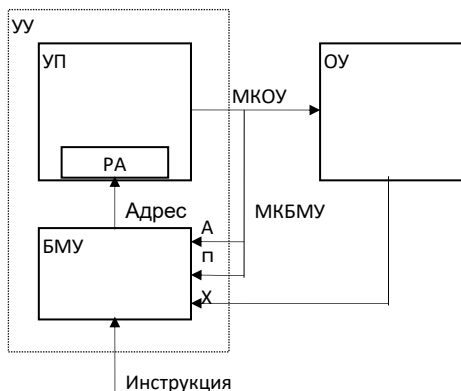


Рис. 2.5. Структурная схема процессора с микропрограммным способом управления

На рис. 2.5 изображена структурная схема процессора с управляющим устройством, построенным на основе программируемой логики. Блок микропрограммного управления (БМУ) определяет адрес очередной микрокоманды (МК) в управляющей памяти (УП). Поступающая из оперативной памяти инструкция содержит адрес первой микрокоманды реализуемой операции. Адреса всех последующих микрокоманд определяется в БМУ.



Поле адреса				Поле условных переходов			Поле управляющих сигналов						
				П	x <sub>1</sub>	x <sub>2</sub>	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>	Z <sub>4</sub>	Z <sub>5</sub>	Z <sub>6</sub>	Z <sub>7</sub>
МК БМУ							МК ОУ						

Рис.2.6. Формат микрокоманды для операции умножения целых чисел без знака

В формате микрокоманды, изображенной на рис. 2.6, имеется поле адреса, которое содержит адрес очередной МК. Причем, при разветвлении программы этот адрес может модифицироваться в зависимости от значения условия. Для этого имеется поле условных переходов, в котором в разряде вида перехода П указывается, имеет ли место условный или безусловный переход (0 - безусловный, 1 - условный), а в разрядах условий x<sub>1</sub>x<sub>2</sub> находятся их значения. Формирование адреса осуществляется замещением младшего разряда содержимого поля адреса текущей микрокоманды значением соответствующего условия, если разряд перехода П указывает на необходимость его изменения.

Так как на выполнение микрокоманд, модифицирующих адреса переходов, затрачиваются тактовые периоды, быстродействие процессора ухудшается по сравнению с аппаратным управлением.

**Конвейерный способ организации управления**

Быстродействие процессора сильно зависит от длительности тактового периода, за время которого должен быть сформирован адрес очередной МК, микрокоманда должна быть прочитана из управляющей памяти, а затем выполнена в операционном устройстве. На рис. 2.7 приведена временная диаграмма работы процессора с микропрограммным управлением.

В момент времени t<sub>1</sub> происходит прием в регистр адреса РА УП (см. рис.2.5) сформированного адреса следующей микрокоманды. Далее адрес из РА выдается на внутреннюю шину адреса и поступает непосредственно в память, где происходит чтение очередной микрокоманды. Процесс чтения завершается к моменту времени t<sub>2</sub>, и с этого момента в ОУ поступает микрокоманда и начинается процесс ее выполнения, который завершается к моменту t<sub>3</sub>. В момент t<sub>4</sub> полученный результат фиксируется в соответствующем

регистре ОУ. В БМУ в интервале  $t_3...t_4$  под действием МК БМУ происходит формирование адреса очередной микрокоманды и фиксация его в момент  $t_4$  в регистре адреса УП.

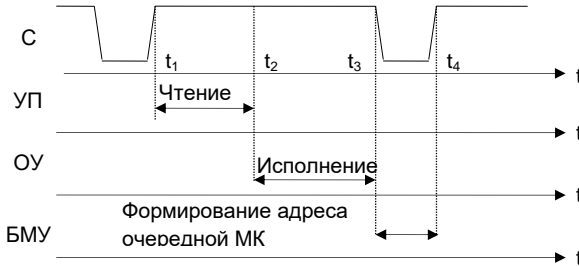


Рис. 2.7. Временная диаграмма работы процессора с микропрограммным управлением

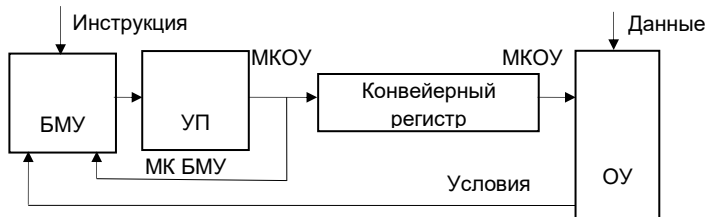


Рис. 2.8. Структура процессора с конвейерным регистром

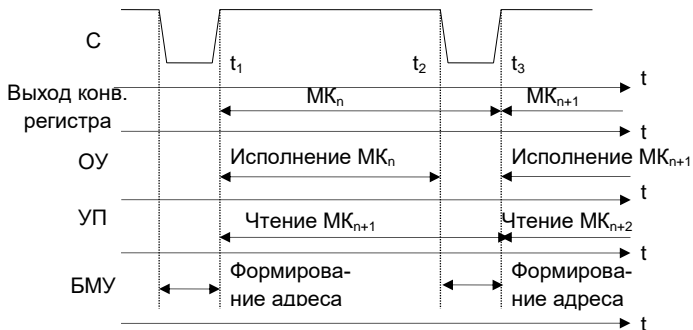


Рис. 2.9. Временная диаграмма работы процессора с конвейерным регистром

Последовательный способ чтения и исполнения микрокоманды задает длительность тактового периода, определяющего быстродействие процессора. Этот недостаток может быть устранен использованием конвейерного способа чтения и исполнения микрокоманды. При этом способе осуществляется параллельное чтение и исполнение микрокоманды. В процессе исполнения в ОУ  $n$ -той микрокоманды в УП производится чтение  $(n+1)$ -ой МК, и в том же тактовом периоде в БМУ формируется адрес  $(n+2)$ -ой МК. Реализация этого способа требует использования конвейерного регистра, изображенного на рис. 2.8. Временная диаграмма работы процессора с конвейерным регистром показана на рис. 2.9.

В интервале времени  $t_1...t_3$  из конвейерного регистра выдается  $n$ -я МК ОУ, считанная из УП на предыдущем такте. В интервале времени  $t_1...t_2$  она исполняется в ОУ. В этом же такте производится чтение из УП  $(n+1)$ -я МК по адресу, сформированному в БМУ в предыдущем такте. В интервале времени  $t_2...t_3$  в БМУ формируется адрес  $(n+2)$ -й МК.

Введение конвейерного регистра усложняет микропрограмму, так как на период времени выработки условия в ОУ для формирования следующего адреса микрокоманды, выполняемой по условию, в микропрограмму необходимо вводить холостые микрокоманды, не изменяющие состояния ОУ по мере их поступления туда из конвейерного регистра.

При использовании управляющего устройства с конвейерным регистром длительность тактового периода может оказаться той же, что и в устройстве, построенным на принципе схемной логики. Однако и в этом случае на выполнение того же алгоритма остается все же больше тактовых периодов.