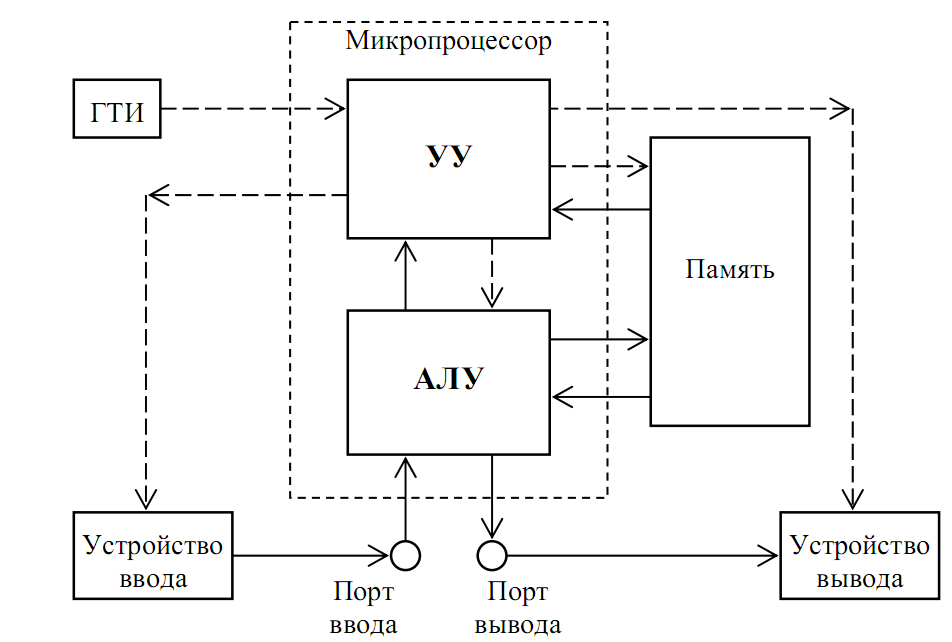
**6. Устройства ввода /вывода**

Через эти устройства осуществляется контакт компьютера с внешним миром. Они являются буферами для преобразования информации с тех языков и тех скоростей, на которых работает компьютер, к тем, которые воспринимает человек или другая связанная с компьютером система. ***Устройство ввода*** получает из внешнего мира данные и команды, которые поступают в память. ***Устройство вывода*** получает вычисленные результаты и передает их человеку-оператору или другой системе. ***Устройства ввода и вывода*** представляют собой периферийные устройства машины. В качестве примеров можно назвать цифровое табло или клавиатуру. Точки контакта между устройствами ввода/вывода и микропроцессором называются портами ввода/вывода. ***Порты ввода/вывода*** – это просто параллельные или последовательные регистры, которые также имеют свои адреса, так что к одному микропроцессору может быть подключено несколько устройств ввода/вывода. Характерная особенность цифрового компьютера состоит в том, что вся информация хранится и обрабатывается в дискретном виде, т. е. в виде конечных чисел. Часто возникает необходимость сопряжения компьютера с другой системой, не способной обрабатывать дискретную информацию. Недискретная информация называется аналоговой или непрерывной. В таких случаях приходится осуществлять преобразование из цифровой формы в непрерывную и обратно. Устройства ввода/вывода, осуществляющие соответствующие преобразования, называются аналого-цифровыми и цифро-аналоговыми преобразователями.

На рис. 1.1 устройства ввода/вывода показаны подключенными к арифметическому устройству. Это не единственно возможный способ подключения. Часто для достижения большей производительности системы желательно позволить устройствам ввода/вывода обращаться к памяти непосредственно, а не через арифметическое устройство. В таком случае говорят о прямом доступе к памяти (DMA – direct memory access). Организация прямого доступа к памяти обеспечивается в компьютерах с помощью специальных функциональных блоков, называемых контроллерами прямого доступа.



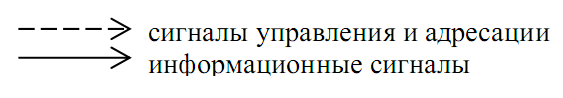


рис.1.1 Типовая стр.схема микрокомпьютера

**7. Шины, обмен информацией в микрокомпьютере с помощью шин**.

Отдельные блоки микрокомпьютера связаны друг с другом с помощью шин. ***Шина***  представляет собой совокупность линий, по которым передается информация от любого из нескольких источников к любому из нескольких приемников. Одна из распространенных в микрокомпьютерах структур шин представлена на рис. 1.5. На рисунке представлены шины трех типов.

***Шина адреса (ША)*** – однонаправленная, т. е. информация по ней передается только в одном направлении. Эта шина служит для передачи адреса от микропроцессора к памяти, вводному или выводному устройству.

***Шина данных (ШД)*** – двунаправленная, т. е. информация по ней может передаваться в обоих направлениях, и она служит для передачи данных.

Наконец, ***шина управления (ШУ)*** состоит из линий, по которым передаются тактовые, синхронизирующие сигналы, а также информация о состоянии (статусе) устройств. По шине управления могут передаваться следующие сигналы: «запись», «чтение», «ввод», «вывод» и «готовность». Часть линий в управляющей шине – однонаправленные, часть – двунаправленные. Поэтому на рисунке направленность этой шины никак не обозначена.

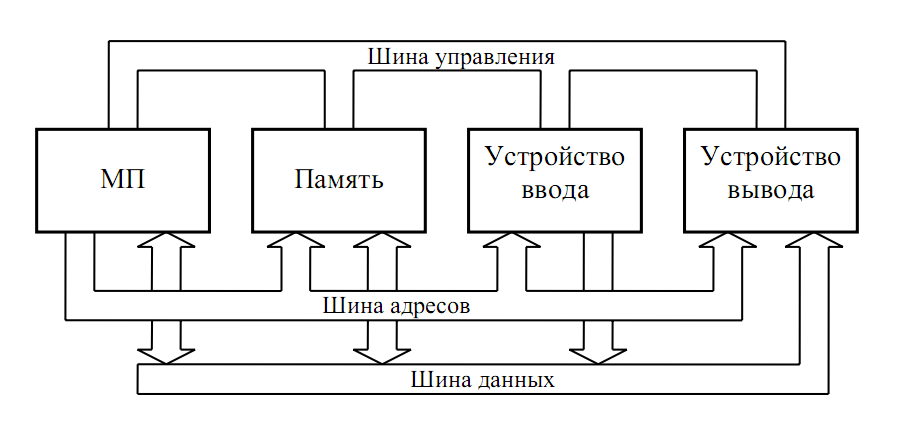


рис.1.5 Шины типичного микрокомпьютера

**8. Структура и формат команды**

Для того чтобы быть правильно проинтерпретированной устройством управления, команда должна иметь определенную ***структуру***, т.е. содержать определенные логические части, называемые полями, каждая из которых несет определенную информацию. Взаимное расположение полей в команде с указанием длин полей в битах называют ***форматом команды***. У микропроцессоров разных типов форматы команд различны. Однако есть информация, которая должна присутствовать в команде в любом случае. Наиболее важное значение имеет код операции ***(КОП)*** и в некоторых командах адрес. ***Код операции*** – это совокупность двоичных цифр, которые однозначно определяют операцию, выполняемую в процессе интерпретации команды. ***Адресная часть команды*** (если она присутствует) указывает на ячейки (например в памяти), к которым нужно обратиться, выполняя команду. Например, если выполняется операция сложения, адресная часть команды может указывать на ячейку, где находится второе слагаемое. Но адресный код не обязателен – в некоторых командах он может и отсутствовать.

Структура команд первых процессоров была более громоздкой (рис. 1.3). В команде указывались адреса обоих операндов, адрес, куда записывался результат и адрес следующей команды. Для извлечения такой команды приходилось несколько раз обращаться к памяти, что естественно приводило к значительному снижению быстродействия. За счет введения программного счетчика избавились от части, в которой указывался адрес следующей команды. Так как один из операндов, как правило, после выполнения некоторого действия не нужен, то стали записывать результат на его место, что позволило исключить еще одно поле. С введением аккумулятора в АЛУ приняли, что один из операндов находится в нем, и отпала необходимость указывать адрес этого операнда. Таким образом, в процессе эволюции компьютеров сократили длины команд и за счет этого увеличили их быстродействие.

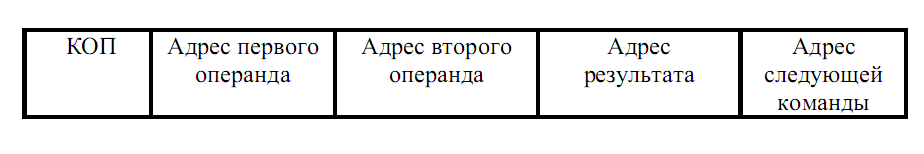


рис.1.3 Стр-ра команд первых процессоров

**9. Стек, принцип организации, назначение**

Место, где запоминается содержимое программного счетчика, обычно представляет собой стек. В общем случае стек – это совокупность регистров, которые принимают и выдают информацию в соответствии с правилом «последним вошел, первым вышел» (LIFO – last-in first-out). Это означает, что только вершина стека, где находится последний его элемент, непосредственно доступна извне. В зависимости от конструкции компьютера стек может находиться в устройстве управления или быть частью памяти. Принцип работы стека отражен на рис. 1.8.

Если стек может хранить несколько адресов (значений программного счетчика), имеется возможность выполнять вложенные подпрограммы. Таким образом, из одной подпрограммы можно сделать переход на другую подпрограмму или даже на саму себя – это называется рекурсией. Очевидно, что глубина вложений подпрограмм зависит от емкости стека. Если емкость стека будет полностью заполнена, а в него запишется еще одно значение, то значение, записанное первым, будет потеряно.

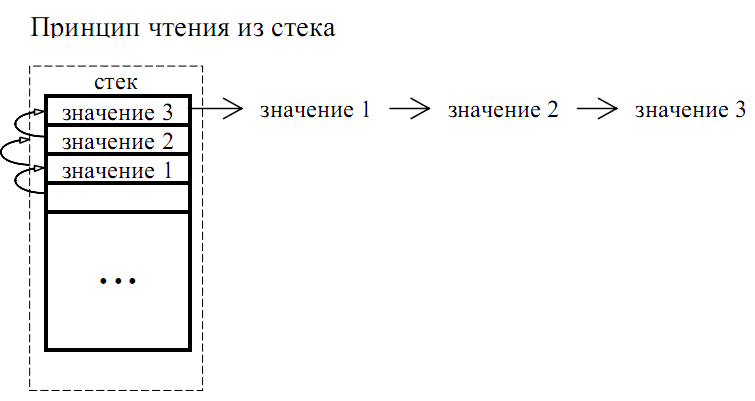
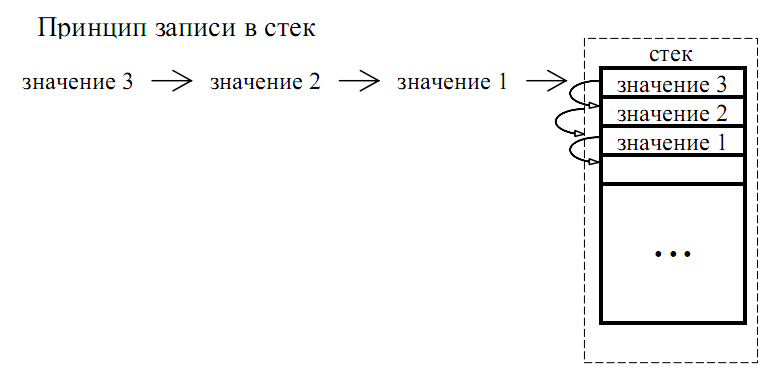
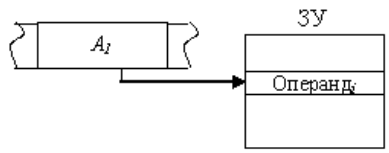


рис.1.8. Работа стека в режиме чтения /записи

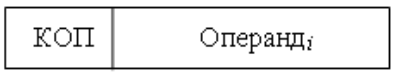
**10. Способы адресации: прямая, непосредственная, регистровая, косвенная**

***Способами адресации*** называются различные режимы интерпретации поля адреса команды. ***Адресный код*** – это информация об адресе операнда, содержащаяся в команде. ***Исполнительный адрес*** – номер ячейки памяти, к которой производится фактическое обращение.

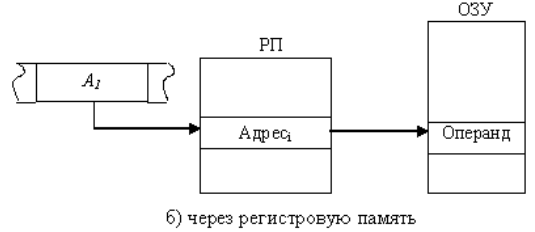
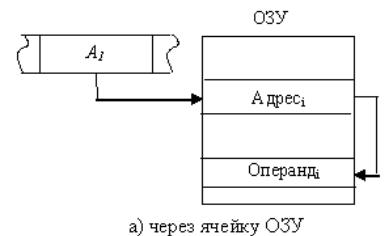
***Прямая адресация***: исполнительный адрес совпадает с адресной частью команды. В команде прямо задаётся адрес операнда

:

***Непосредственная адресация***: в команде содержится не адрес операнда, а непосредственно сам операнд:



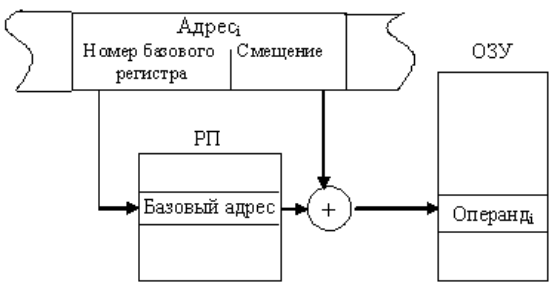
***Косвенная адресация***. В этом случае в адресном поле команды указан адрес ячейки памяти, где хранится адрес операнда, а не сам операнд.Разновидностью косвенной адресации является косвенно-регистровая адресация, при которой в адресном поле команды указывается номер регистра, в котором находится адрес ячейки с операндом.



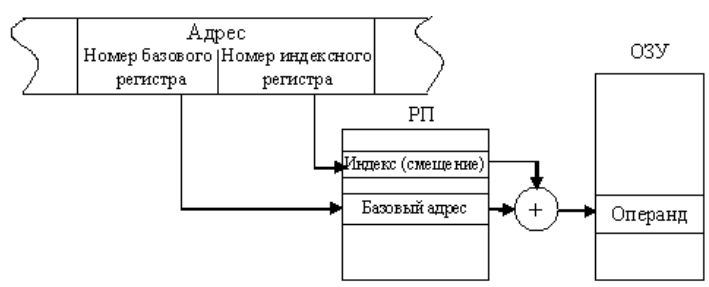
***Неявная адресация***: в команде не содержится явных указаний об адресе участвующего в операции операнда, но этот адрес подразумевается и фактически задается кодом операции. В качестве места для хранения неявно адресуемого операнда чаще всего используется аккумулятор. Относительная адресация. Этот способ используется тогда, когда память логически разбивается на блоки, называемые сегментами. В этом случае адрес ячейки памяти содержит две составляющих: адрес начала сегмента (базовый адрес) и смещение адреса операнда в сегменте. Адрес операнда определяется как сумма базового адреса и смещения относительно этой базы:

**Операндi = (базаi + смещениеi).**

Адресное поле команды состоит из двух частей (рис. 2.12): в одной указывается номер регистра, хранящего базовое значение адреса (начальный адрес сегмента), а в другом адресном поле задается смещение, определяющее положение ячейки относительно начала сегмента. Именно такой способ представления адреса обычно и называют ***относительной адресацией***.



Первая часть адресного поля команды также определяет номер базового регистра, а вторая содержит номер регистра, в котором находится смещение (рис. 1.13) Такой способ адресации чаще всего называют ***базово-индексным***.



***Стековая адресация***. Во всех современных процессорах аппаратно поддерживается стек, то есть область оперативной памяти, предназначенная для временного хранения любой информации. Для стека характерны две операции: 1) запись в стек, 2) чтение из стека. Запись и чтение производятся через вершину стека, таким образом, что последнее записанное в стек значение читается из него первым. В современных ЭВМ чаще используют понятие не прямого адреса, а смещения, что связано с использованием сегментной памяти