Для адресации *S* ячеек памяти адресная часть одного операнда должна иметь число разрядов
$n_A \geq \log_2 S.$

Это требование находится в противоречии с желанием иметь малую разрядность команды и возможностью использовать большое адресное пространство. Можно:

- Использовать регистры общего назначения. Это существенно повышает быстродействие, но только в том случае, если операнд используется многократно. Существует ограничение на количество регистров общего назначения.
- определять один или несколько операндов неявным образом. Это означает использование не трехадресной команды, а двух- одно- и безадресных команд.

Адресный код –это информация об адресе команды, содержащаяся в команде. **Исполнительный адрес** – это номер ячейки памяти, к которой производится обращение.

Способы адресации

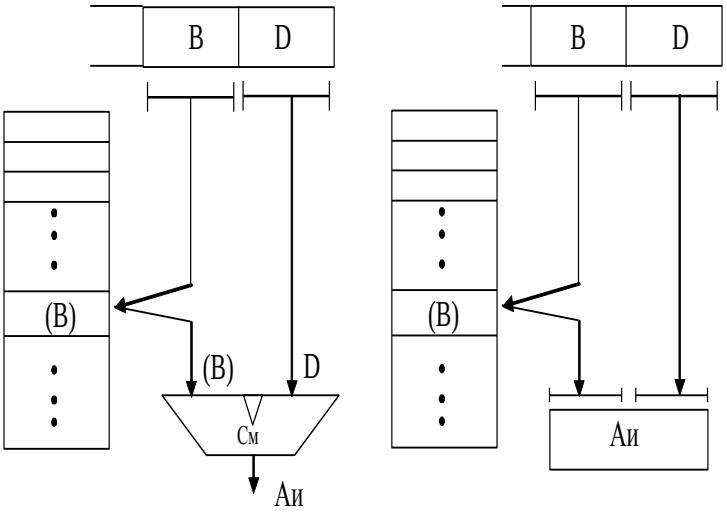
Подразумеваемый операнд	В команде не содержится в явном виде указаний об адресе операнда. Операнд подразумевается и фактически задается кодом операции команды. INC, DEC.
Подразумеваемый адрес.	В команде нет явных указаний об адресе участвующих операндов или адреса, по которому помещается результат операции, но этот адрес подразумевается. Например, A1 = A1+A2
Непосредственная адресация.	В команде содержится не адрес операнда, а непосредственно операнд. Удобно при использовании различного рода констант. Но число значений ограничено размером поля.
Прямая адресация.	Исполнительный адрес = адресный код(доступ к глобальным переменным, команда имеет доступ только к одному и тому же адресу памяти).
Регистровая адресация.	В качестве операнда используется содержимое регистров процессора.
Косвенная адресация.	«Адресация адреса» - адресный код команды указывает адрес ячейки памяти, в которой находится адрес операнда или команды. *Регистровая косвенная адресация* - если в качестве адресного кода используется регистр (*указатель)*.
Индексная адресация	**Модификация команд (адресных частей)** - автоматическое изменение в соответствующих командах их адресных частей согласно расположению в ОП обрабатываемых операндов после каждого выполнения цикла. В современных ЭВМ используется механизм *индексации.* Вводятся дополнительные *индексные регистры*. В формате команды вводится поле X для указания индексного регистра. **Аи = Ак + (Реги) * [ОП] + Аб** **Аи** - исполнительный адрес, **Ак** - адресный код команды, **Реги** - содержимое индексного регистра, **ОП** - размер операнда, **Аб** - базовый адрес.
Стековая адресация	Стековая память реализует безадресное задание операндов. *Стек* -группа последовательно пронумерованных регистров *(аппаратный стек)* или ячеек памяти, снабженных указателем стека (регистром SP), в котором автоматически при записи и считывании указывается номер (адрес) последней занятой ячейки стека *(вершины стека)*. При выполнении операции записи в стек слово помещается в следующую ячейку стека, а при считывании из стека последнее поступившее в него слово. Таким образом, в стеке реализуется дисциплина обслуживания «последний пришел – первый ушел» (LIFO). Указанное правило при обращении к стеку реализуется автоматически, и поэтому при операциях со стеком возможно безадресное задание операндов. Однако при такой структуре команд возникают сложности с построением команд передачи управления и работы с периферийными устройствами.

Относительная адресация или базирование.

Аи - исполнительный адрес, Ак - адресный кода команды, Аб - базовый адрес: Аи = Аб + Ак.

Для хранения базовых адресов в машине могут быть использованы специальные регистры или ячейки памяти (базовые регистры). В команде выделяется поле B для указания номера базового регистра. Это позволяет при меньшей длине адресного кода обеспечить доступ к любой ячейке памяти. Число разрядов в базовом регистре

выбирается так, чтобы можно было адресовать любую ячейку ОП, а Ак используют для представления короткого «смещения» (D). Смещение D определяет положение операнда относительно начала массива, задаваемого базовым адресом Аб.



а) б)

Рис. 4.21

Аи = (B) + D,

где (B) – содержимое регистра с номером B.

Относительная адресация обеспечивает так называемую *перемещаемость* программ, т.е. возможность перемещения программ в памяти без изменений внутри самой программы.

Способы адресации команд перехода

1. Прямая адресация, когда целевой адрес просто полностью включается в команду.
2. Косвенная и регистровая адресация позволяют программа вычислить целевой адрес, помещать его в регистр, а затем переходить по этому адресу. Такой способ дает максимальную гибкость, поскольку целевой адрес вычисляется во время выполнения программы, но предоставляет огромные возможности для появления ошибок, которые практически невозможно найти.
3. Индексная адресация, при которой известно смещение от регистра.
4. Относительная адресация по счетчику команд. В этом случае для получения целевого адреса смещение (со знаком), находящееся в самой команде прибавляется к программному счетчику.