

**Варламова Екатерина, ИУ7-51Б**

**РК по ЭВМ**

**3 вопрос**

**3. Способы адресации: непосредственная, прямая, регистровая, неявная, косвенная, косвенная регистровая**

Существует несколько форматов команд:

Форматы команд.

Операционная часть	Адресная часть
--------------------	----------------

1. Четырехадресная команда.

КОП	1 операнд	2 операнд	результат	Адр след ком.
-----	-----------	-----------	-----------	---------------

2. Трехадресная команда

КОП	1 операнд	2 операнд	результат
-----	-----------	-----------	-----------

3. Двухадресная команда.

КОП	1 операнд	2 оп-д/результат
-----	-----------	------------------

Характерна для CISC-архитектуры

4. Аккумуляторная архитектура

КОП	1 операнд
-----	-----------

Второй операнд хранится в аккумуляторе. Данный формат команд характерен для RISC-архитектур.

5. Нульоперандная команда.

КОП
-----

Рисунок 1 Форматы команд

При этом КОП – код операции входит в операционную часть, а в адресную часть входит способ адресации и сам адрес/операнд. Рассмотрим несколько способов адресации.

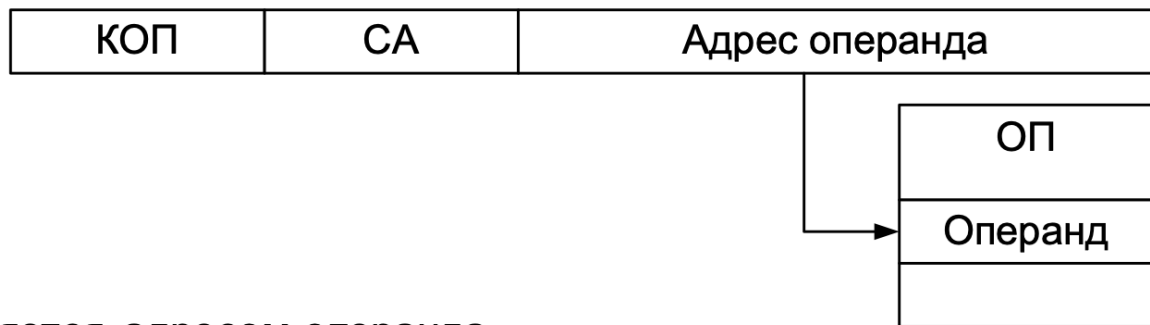
**Непосредственная адресация**

КОП	СА	Непосредственный операнд
-----	----	--------------------------

При такой адресации операнд находится прямо в команде, поэтому его размер ограничен размером команды. К преимуществам можно отнести высокую скорость выполнения (так как не нужно обращаться к памяти).

Пример: `mov ax, 0;`

## Прямая адресация



является адресом операнда

При такой адресации команда хранит в себе адрес операнда, по которому происходит обращение в ОП. Однако при такой адресации переносимость программы может быть невысокой из-за явного указания адреса ячейки, к которой необходимо выполнить обращение.

Пример: `mov ax, a` ; `a` – переменная, которая преобразуется в адрес

## Неявная адресация



При такой адресации операнд следует из кода операции. При такой адресации команды занимают мало места, однако только таких команд недостаточно для построения системы. Пример: команда `div`.

## Регистровая адресация

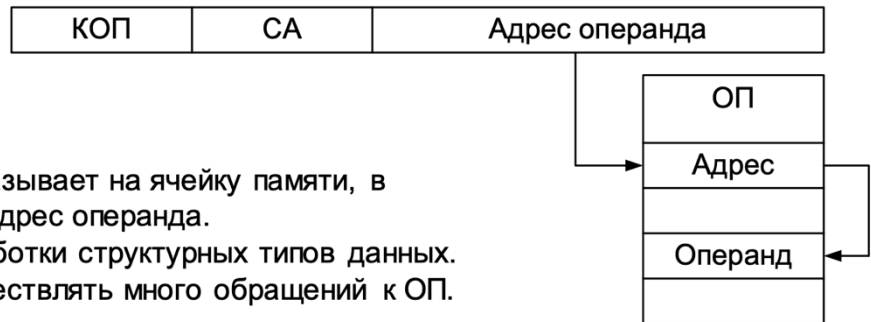


При такой адресации обращение происходит не к ОП, а к регистрам, что достаточно сильно ускоряет выполнение команды по сравнению с подобной командой при прямой адресации. К сожалению, чем быстрее память, тем меньше её ёмкость, поэтому за быстроедействие обращений к регистрам приходится платить их небольшим количеством. Пример: `mov ax, bx`

## Косвенная адресация

### Косвенная адресация

Адрес в команде указывает на ячейку памяти, в которой находится адрес операнда.  
(+) удобна для обработки структурных типов данных.  
(-) приходится осуществлять много обращений к ОП.

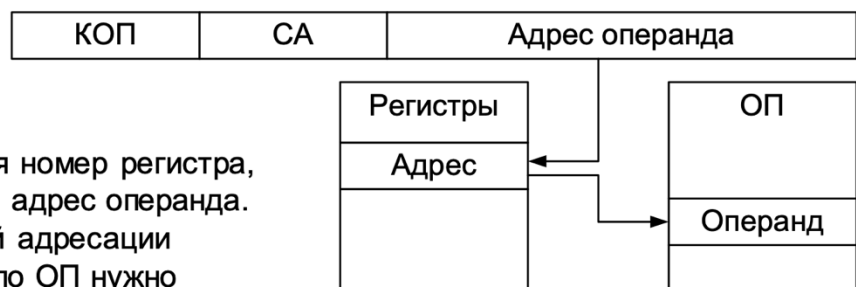


При такой адресации команда хранит в себе адрес ячейки, которая хранит в себе адрес ячейки операнда. В asm обычно реализуется с помощью квадратных скобок. Из-за большого количества обращений к ОП команда выполняется долго.

## Косвенная регистровая адресация

### Косвенная регистровая адресация

В команде содержится номер регистра, в котором содержится адрес операнда.  
(+) быстрее косвенной адресации  
(-) для перемещения по ОП нужно менять содержимое регистра



При такой адресации выполнение похоже на выполнение команды при обычной косвенной адресации, однако вместо первого обращения к памяти происходит обращение к регистру, в котором уже хранится адрес операнда из ОП. Такой вариант адресации, очевидно, быстрее предыдущего из-за быстродействия регистровой памяти. Пример: `mov ax, [bx]`