Управляющие конструкции

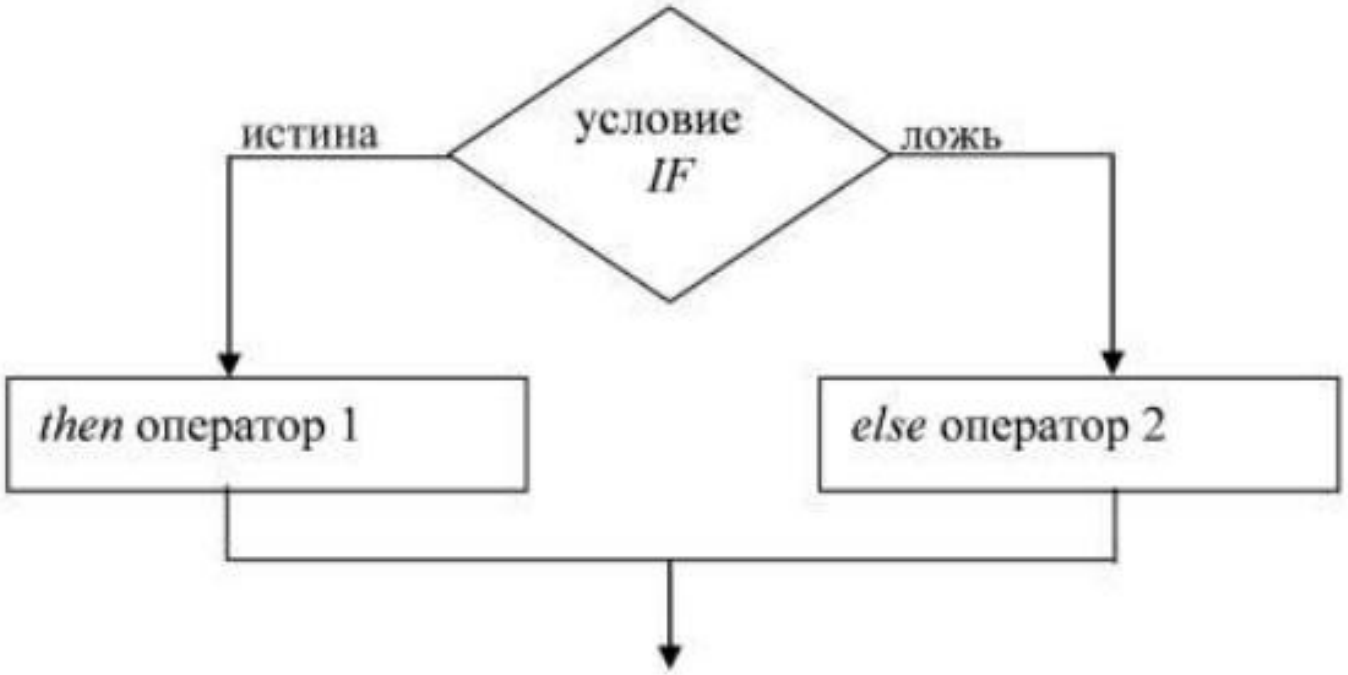
Последовательное выполнение команд

Что это?

- Каждая программа состоит из одной или нескольких последовательностей отдельных элементарных команд.
- Последовательность здесь означает участок программы, где команды выполняются одна за другой, без любых переходов.
- Если программа не содержит других конструкций, кроме последовательности элементарных команд, она называется линейной.

Конструкция «IF THEN» — выбор пути

Что это?



- В языке ассемблера механизм выбора реализован посрздством команд сравнения, условного и безусловного переходов.

Команды CMP и TEST

- Команды CMP и TEST используются для сравнения двух операндов.
- Операндами могут быть как регистры, так и адреса памяти, размер операнда — 8, 16 или 32 бита.

Команды CMP и TEST

- `CMP o1 , o2`
- Команда CMP — это сокращение от «compare», «сравнить».
- Она работает подобно SUB: операнд o2 вычитается из ol.
- Результат нигде не сохраняется, команда просто изменяет регистр признаков.
- Команда CMP может использоваться как для сравнения целых беззнаковых чисел, так и для сравнения чисел со знаком.
- Команда TEST работает подобно CMP, но вместо вычитания она вычисляет поразрядное И операндов.

Например…

```
cmp   ax, 4                ;сравниваем AX со значением 4
cmp   dl,ah                ;сравниваем DL с AH
cmp    [diameterl] ,ax     ,-сравниваем переменную "diameterl" с AX
```

Команда безусловного перехода — **JMP**

- Самый простой способ изменить последовательность выполнения команд заключается в использовании команды jmp
- Это команда безусловного перехода.
- Она перезаписывает указатель команд (регистр IP или CS), что заставляет процессор «переключиться» на выполнение команды по указанному адресу.
- Формат команды таков:

```
JMP операнд
```

Куда пойдем?

- **метка** — это идентификатор, заканчивающийся двоеточием.
- Во время компиляции он будет заменен точным адресом согласно его позиции в программе.

```
mov ax,4          ;AX = 4
new_loop:         ;метка new_loop
mov bx, ax        ;копируем AX в BX
```

Чтобы перейти к метке new__loop из другого места программы, используйте команду:

```
jmp new_loop      ;переходим к new_loop
```

Условные переходы

- В языке ассемблера имеется множество команд условного перехода
- Имена этих команд различаются в зависимости от условия перехода
- Работают эти команды одинаково: если условие истинно, выполняется переход на указанную метку, если нет, то процессор продолжит выполнять программу со следующей команды.
- Общий формат команд условного перехода следующий:
- `Jx метка_назначения`

Сводная таблица

Сводная таблица команд условного перехода

	o1==o2 o1=o2	o1!=o2 oK>o2	o1>o2	oKo2	o1=<o2	o1>=o2
Инструкции для беззнаковых чисел	JE(JZ)	JNE(JNZ)	JA(JNBE)	JB(JNAE)	JNA(JBE)	JNB(JAE)
	Jump, если равно Jump, если 0	Jump, если не равно Jump, если не 0	Jump, если больше Jump, если не меньше или равно	Jump, если меньше Jump, если не больше или равно	Jump, если не больше Jump, если меньше или равно	Jump, если не меньше Jump, если больше или равно
Инструкции для чисел со знаком	JE(JZ)	JNE(JNZ)	JG(JNLE)	JL(JNGE)	JNG(JLE)	JNL(JGE)
	Jump, если равно Jump, если 0	Jump, если не равно Jump, если неО	Jump, если больше Jump, если не меньше или равно	Jump, если меньше Jump, если не больше или равно	Jump, если не больше Jump, если меньше или равно	Jump, если не меньше Jump, если больше или равно

Как это запомнить?

- Чтобы лучше запомнить имена команд, запомните несколько английских слов:
- equal — равно,
- above — больше,
- below — ниже,
- zero — ноль,
- greater — больше,
- less — меньше.
- Таким образом, JE — Jump if Equal (Переход, если Равно), JNE — Jump if Not Equal (Переход, если Не Равно), JA — Jump if Above (Переход, если больше)

Например…

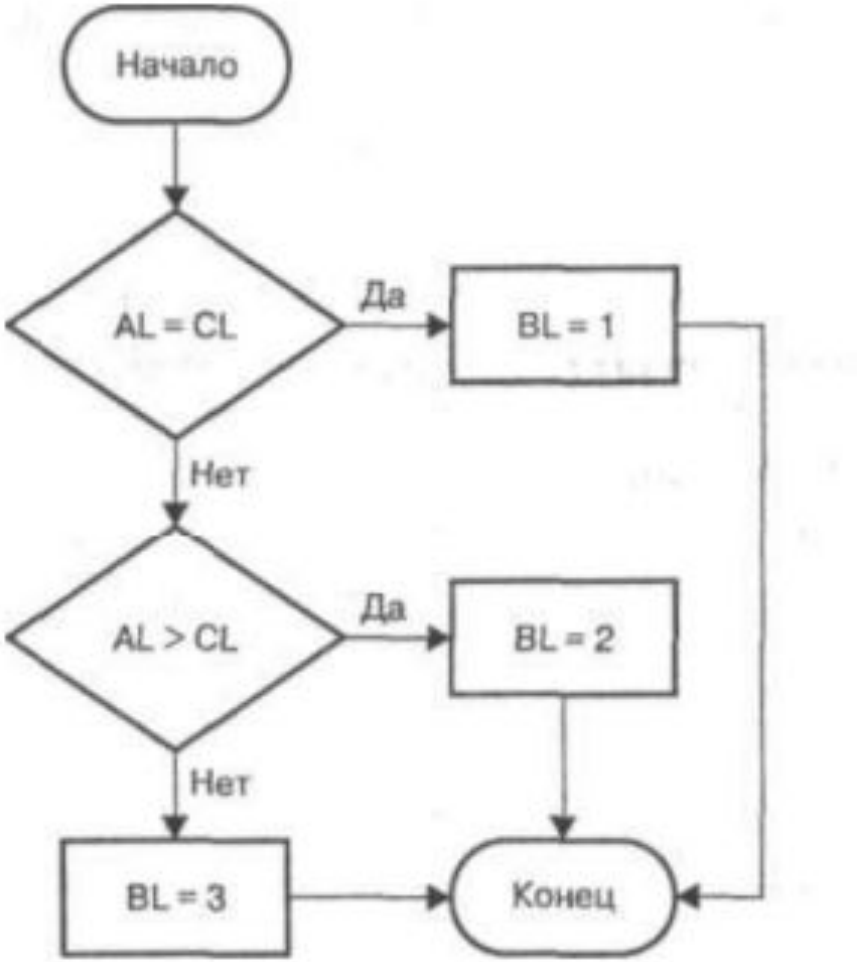
Теперь рассмотрим, как реализовать конструкцию IF-THEN на языке ассемблера. В нашем простом случае мы перейдем к метке if_three, если регистр AX содержит значение 3.

Прежде всего мы должны проверить, есть ли в регистре AX тройка. Для этого используем команду CMP:

```
cmp ax,3          ;сравниваем AX с 3
```

Для проверки равенства применим команду JZ, как показано в таблице команд условного перехода:

```
jz is_three       ;переходит к "is_three", если AX = 3
```


Начало
AL = CL
Да
BL = 1
Нет
AL > CL
Да
BL = 2
Нет
BL = 3
Конец

```
cmp al,cl          ;сравниваем AL и CL
jz write_1         ;переходим к write_1, если AL = CL
cmp al,cl          ;сравниваем AL и CL
ja write_2         ;переходим к write_2, если AL > CL
mov bl,3           ;последний случай — сразу загружаем 3 в BL
end_if:            ;просто метка, символизирующая конец IF

write_1:           ;метка write_1
mov bl,1           ;BL = 1
jmp end_if         ;переходим к end_if
write_2:           ;метка write_2
mov bl,2           ;BL = 2
jmp end_if         ;переходим к end_if
```

1. Требуется вычислить значение формулы:

- *e=a-(b+c-1)+(-d)*.
- Все числа являются 8-битными целыми со знаком.

2. Вычислить значение выражения (53+8)*2 + (150-60)/3. Полученный результат записать в регистр DX.