

Машинные команды. Выполнение программы.



Регистры общего назначения – сумматор, регистр данных, адресный регистр и т.д.

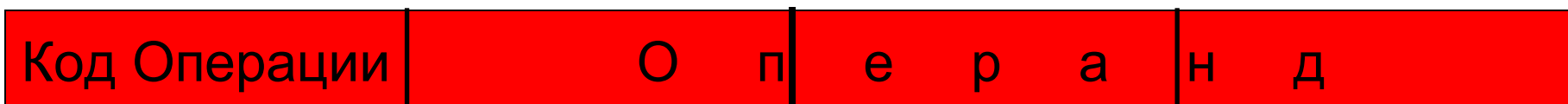
Счетчик команд

Ячейки памяти

Регистр команд

Наша вымышленная машина имеет 16 (16-разрядных) регистров общего назначения, (16-разрядный) регистр команд и (8-разрядный) регистр счетчика команд. Размер памяти – 256 байт.

Формат машинной команды (наш RISC процессор):



Размер машинной команды – одно слово (2 байта).

Старшие 4 бита выделены для кода операции, остальные биты представляют операнд. Кодировка операнда зависит от типа операции.

Система команд:

Команды передачи данных:

- Память -> Регистр - *загрузить* (*load, ld*) (2)
- Регистр -> Память - *сохранить* (*store, st*) (3)
- Команды ввода-вывода (*in, out*) (0, 1)

Арифметико-логические команды:

- сложение, вычитание, умножение и деление (*add, sub, mul, div*) (4, 5, 6, 7)
- логические операции (*and, or, not* и *xor*) (8, 9, A, B)
- операции сдвига и циклического сдвига (*sl, sr, slc, src*)

Команды управления:

- команды безусловного и условного перехода (*je, jg* и т.д.) (C, D)

1	1	1	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---

Регистр А

0	1	1	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Логический сдвиг вправо

1	1	1	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Циклический сдвиг вправо

1	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Логический сдвиг влево

1	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Циклический сдвиг влево

1	1	1	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Арифметический сдвиг вправо

1	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Арифметический сдвиг влево

Пример 1:

Код	
Операции	Операнд
0011	0101 1110 0111
3	5 E 7

Код операции 3 означает сохранение содержимого регистра, в данном случае регистра 5, в ячейке памяти с адресом E7.

Используя мнемокод эту команду можно записать –

ST 5, 231

При выполнении программы устройство управления непрерывно повторяет *машинный цикл*: а) в командный регистр загружается команда, хранящаяся по адресу, указанному в счетчике команд, и в счетчик команд загружается адрес следующей команды; б) расшифровывает команду; в) активирует схему для выполнения необходимой операции.

Пример 2:

1. LD	1, 120	0010	0001	0111	1000
2. LD	2, 122	0010	0010	0111	1010
3. JE	2, +6	1100	0010	0000	0110
4. DIV	1, 2,4	0111	0100	0001	0010
5. ST	4,120	0011	0100	0111	1000
6. STOP					

1. Загрузить содержимое ячейки 120 (0x78) в первый регистр.
2. Загрузить содержимое ячейки 122 (0x7A) во второй регистр.
3. Если содержимое второго регистра равно нулю, то добавить к счетчику команд 6.
4. Разделить содержимое первого регистра на содержимое второго регистра и результат занести в четвертый регистр.
5. Сохранить содержимое четвертого регистра в ячейке с адресом 120 (0x78)
6. Остановка

Процессоры 80x86 являются CISC – процессорами.

Код операции имеет разную длину.

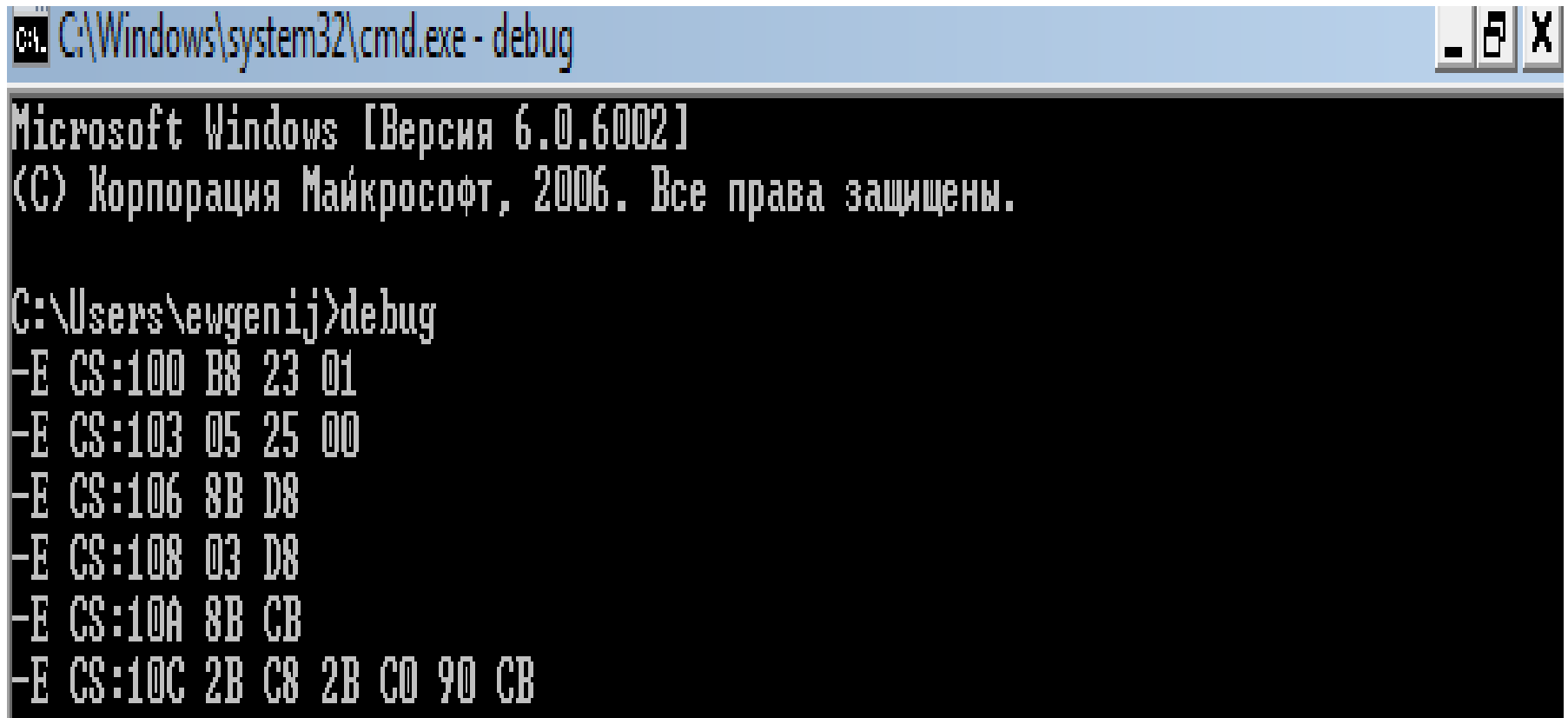
Одной операции может соответствовать несколько кодов в зависимости от контекста.

Сегментированная память (сегментные регистры).

Регистры флагов – флаг переполнения, флаг переноса, флаг нуля и т.д.

B82301	Загрузить значение 0x123 в AX
052500	Добавить значение 0x25 к AX
8BD8	Переслать содержимое AX в BX
03D8	Прибавить содержимое AX к BX
8BCB	Переслать содержимое BX в CX
2BC8	Вычесть содержимое AX из AX
90	Нет операции
CB	Возврат

Ввод команд в память компьютера



```
C:\Windows\system32\cmd.exe - debug

Microsoft Windows [Версия 6.0.6002]
(C) Корпорация Майкрософт, 2006. Все права защищены.

C:\Users\ewgenij>debug
-E CS:100 B8 23 01
-E CS:103 05 25 00
-E CS:106 8B D8
-E CS:108 03 D8
-E CS:10A 8B CB
-E CS:10C 2B C8 2B C0 90 CB
```


Отображение участка памяти, содержащего код программы:

```
-D CS:100
17E9:0100 B8 23 01 05 25 00 8B D8-03 D8 8B CB 2B C8 2B C0 .#..%.....+.+.
17E9:0110 90 CB 00 00 00 00 00 00 00-00 00 00 00 34 00 D8 17 .....4...
17E9:0120 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
17E9:0130 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
17E9:0140 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
17E9:0150 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
17E9:0160 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
17E9:0170 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

Пошаговое выполнение программы:

-R

```
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=17E9 ES=17E9 SS=17E9 CS=17E9 IP=0100  NU UP EI PL NZ NA PO NC
17E9:0100 B82301          MOV     AX,0123
```

-I

```
AX=0123 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=17E9 ES=17E9 SS=17E9 CS=17E9 IP=0103  NU UP EI PL NZ NA PO NC
17E9:0103 052500          ADD     AX,0025
```

-I

```
AX=0148 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=17E9 ES=17E9 SS=17E9 CS=17E9 IP=0106  NU UP EI PL NZ NA PE NC
17E9:0106 8BD8          MOV     BX,AX
```

-T

```
AX=0148  BX=0148  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=17E9  ES=17E9  SS=17E9  CS=17E9  IP=0108  NU UP EI PL NZ NA PE NC
17E9:0108 03D8          ADD     BX,AX
```

-T

```
AX=0148  BX=0290  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=17E9  ES=17E9  SS=17E9  CS=17E9  IP=010A  NU UP EI PL NZ AC PE NC
17E9:010A 8BCB          MOV     CX,BX
```

-

-T

```
AX=0148  BX=0290  CX=0290  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=17E9  ES=17E9  SS=17E9  CS=17E9  IP=010C  NU UP EI PL NZ AC PE NC
17E9:010C 2BC8          SUB     CX,AX
```

-T

```
AX=0148  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=17E9  ES=17E9  SS=17E9  CS=17E9  IP=010E  NU UP EI PL NZ AC PE NC
17E9:010E 2BC0          SUB     AX,AX
```

-T

-I

AX=0000 BX=0290 CX=0148 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=17E9 ES=17E9 SS=17E9 CS=17E9 IP=0110 NU UP EI PL ZR NA PE NC
17E9:0110 90 NOP

-I

AX=0000 BX=0290 CX=0148 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=17E9 ES=17E9 SS=17E9 CS=17E9 IP=0111 NU UP EI PL ZR NA PE NC
17E9:0111 CB RETf

-I