Overview

80x86 Instructions categories

80x86 instructions can be (roughly) divided into eight different categories:

1) Data movement instructions

   mov lea les push pop pushf popf

2) Conversions

   cbw cwd xlat

3) Arithmetic instructions

   add inc sub dec cmp neg mul imul div idiv

4) Logical shift rotate and bit instructions

   and or xor not shl shr rcl rcr

80x86 Instructions categories

5) I/O instructions

in out

6) String instructions

movs stos lods

7) Program flow control instructions

jmp call ret j* (conditional jumps)

8) Miscellaneous instructions

clc stc cmc

Assembly to machine code

- Object Code for assembly instruction
  - Hex coded
  - <Opcode> [<Operand1>,…]
  - {Mnemonic , Operands} => Hex Opcode
  - Operand(s) => Hex encoded
- Addressing Modes

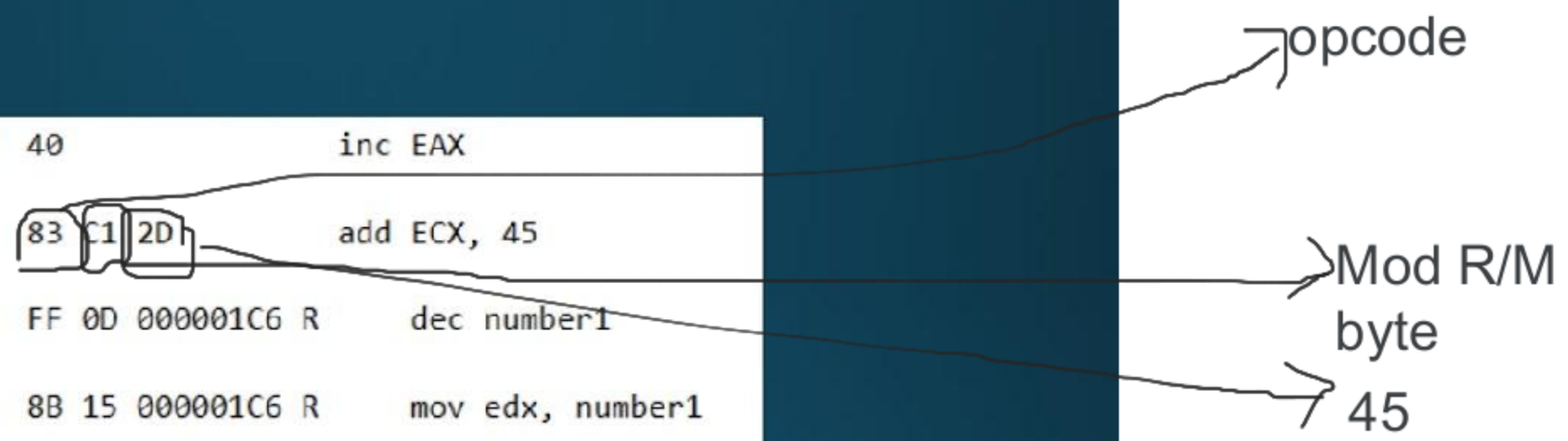
Mode	Location of Data
Immediate	Within the instruction
Register	In the register specified in the instruction
Memory	Memory Direct: At memory address specified in the instruction. Register Indirect: At memory address stored in the register that is specified in the instruction.

Mnemonic to Opcode

- Mnemonic is more general, assembly instruction
- Opcode is more specific, hex code
- Opcode is chosen by the assembler
- One→Many mapping between assembly instruction and opcode
- {Mnemonic , Operands} => Opcode
  - mov eax, number => opcode is A1
  - mov sum, eax => opcode is A3
  - mov eax, 0 => opcode is B8

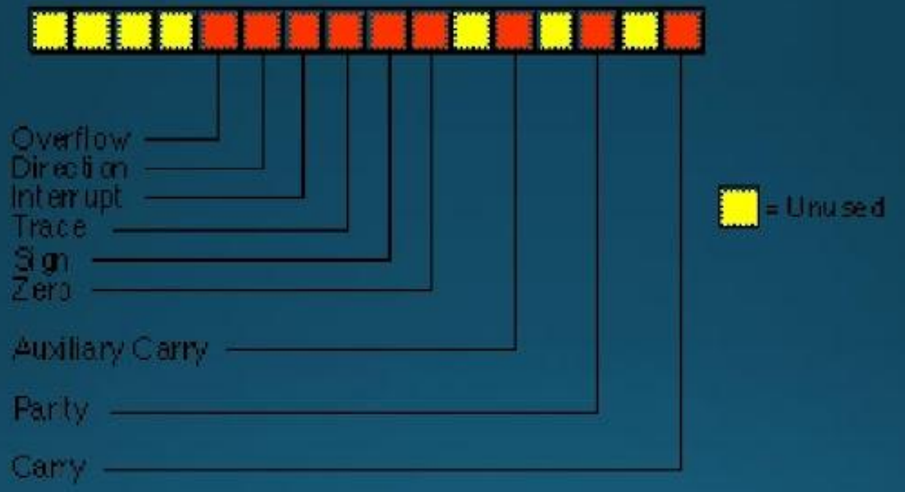
Machine/Object code generation

```
43    00000000  40                    inc EAX
44
45    00000001  83 C1 2D              add ECX, 45
46
47    00000004  FF 0D 000001C6 R      dec number1
48
49    0000000A  8B 15 000001C6 R      mov edx, number1
50
51    00000010  83 2D 000001C6 R      sub number1, 30
52          1E
53
54    00000017  A3 000001C6 R         mov number1, eax
55
56
57    0000001C  C3                    ret
```



FLAGS Register

- Maintains the current operating mode of the CPU
- And some instruction state information



- Will revisit later…