Chapter 4

8051 Family Microcontrollers Instruction Set

Lesson 1

Machine code, Opcode, Operand and Assembly Instruction

Machine Codes

Machine Code

- Each distinct executable instruction has a distinct byte(s) at an address(es) and that defines the instruction

Example- 8051 Machine code 00H

- It defines NOP instruction
- Let current PC = 1000H. Let 00H is stored at that address.
- Processor when fetches machine code 00H, it is interpreted as NOP.
- NOP means no operation; PC sets to new value and processor fetches next from 1001H.

Example- 8051 Machine code 68H

- It defines MOV A, R0 instruction
- Let current PC = 1001H. Let 68H be stored at that address.
- Processor when fetches machine code 68H, it interprets as MOV.
- MOV A,R0 means move (copy) R0 bits at R0 into A.

Machine Code

- Codes stored in the memory are called machine codes.
- Program consists of the machine codes that correspond to the instructions.

Assembly Mnemonics

Why Assembly Mnemonics?

- Machines codes for a program difficult to remember and too lengthy to write

Assembly Mnemonics

- An assembly instruction in a program relatively easier to remember
- Eases the assembly program writing in

Example- 8051 Assembly Mnemonic: ADD *A*, R1

- ADD *A*, R1 means that add into *A* the byte at the R1 register

Example- A program in high level language

```
x = (a + b+ c) * d
```

Same program in Assembly language

1. Let R0 ← a; R1 ← b; R2 ← c;

2. Let R3 ← d;

3. R4 and R5 16-bits ← x lower and higher bits, respectively

Assembly Instructions

```
MOV A, R0;
ADD A, R1;
ADDCA, R2;
MOV B, R3;
MUL A, B
MOV R4, A
MOV R5, B
```

Same program in machine codes

1000H: 68H

1001H: 29H

1002H:3AH

1003H: 8BH, F0H

1005H: A4H

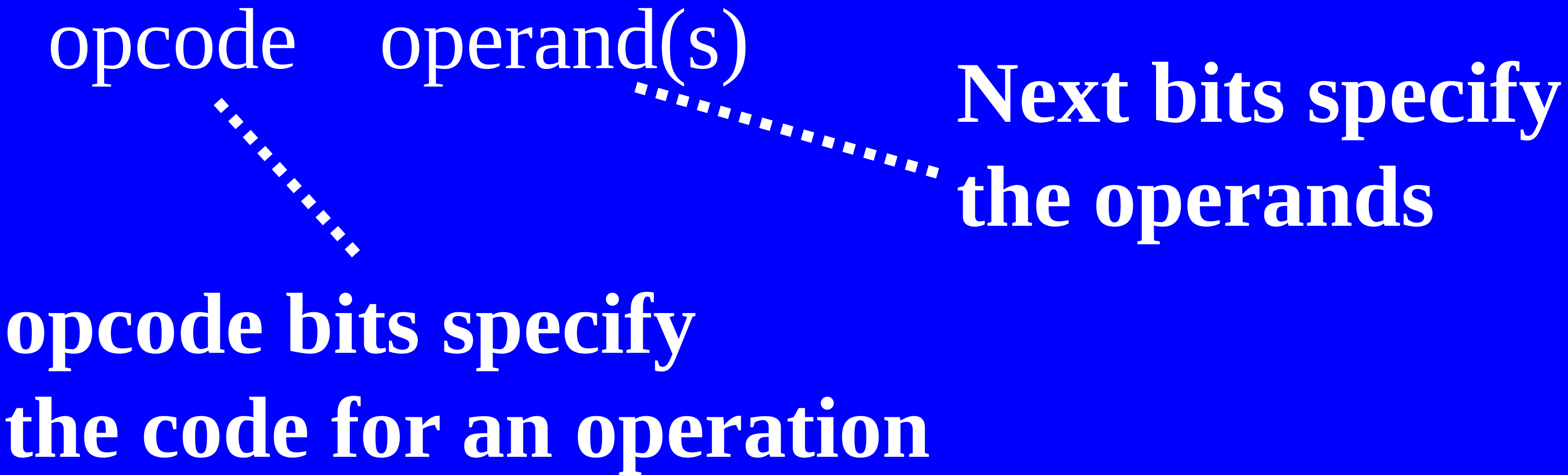
1006H: FCH

1007H:ADH, F0H

Address for the code given before the colon. Code after the colon.

Opcode and operand in an assembly instruction

Assembly instruction: components

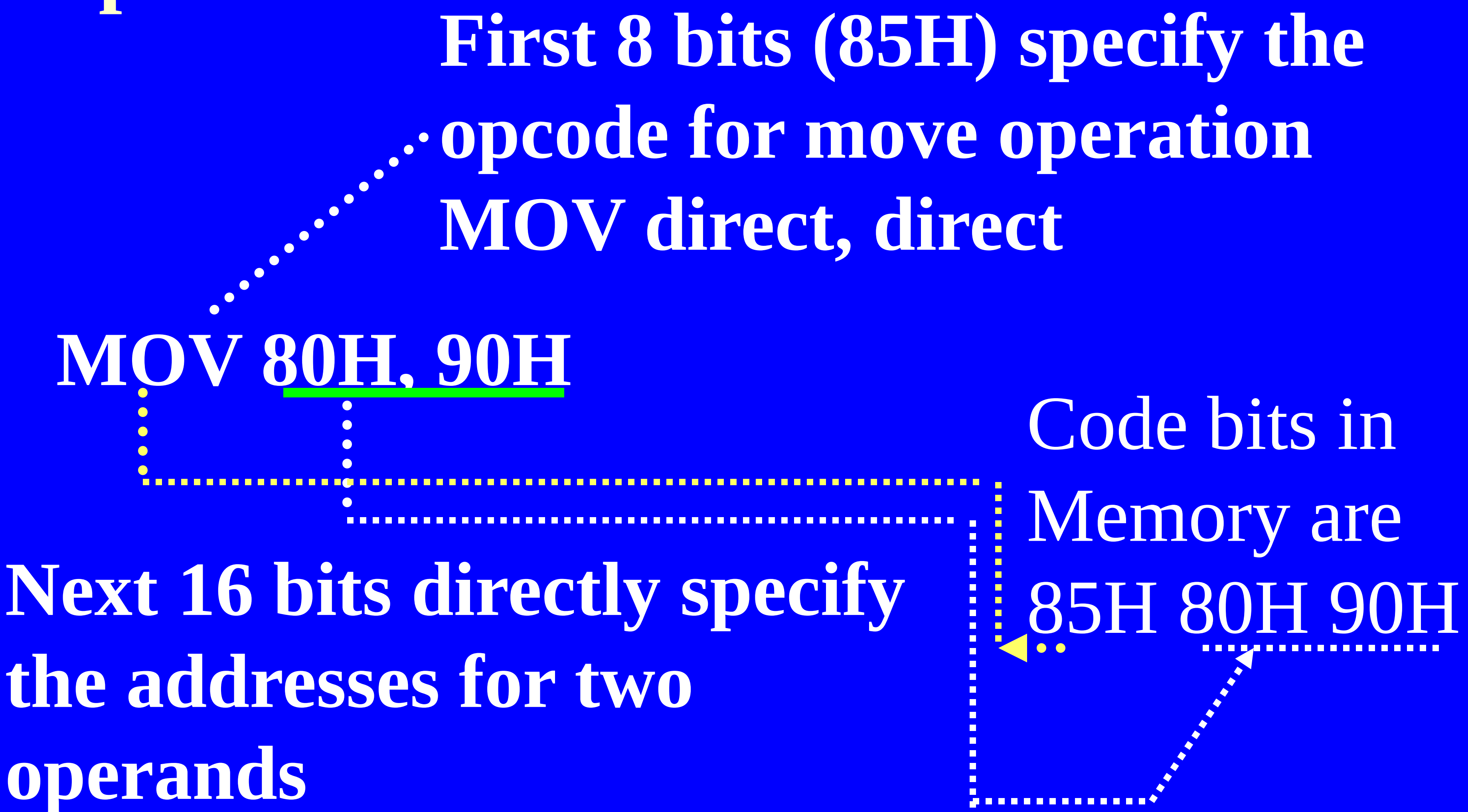


opcode operand(s)

Next bits specify the operands

opcode bits specify the code for an operation

Assembly instruction: opcode and operands



First 8 bits (85H) specify the opcode for move operation MOV direct, direct

MOV 80H, 90H

Code bits in Memory are 85H 80H 90H

Next 16 bits directly specify the addresses for two operands

STEP 1

MOV A, @Ri

1. Fetch Opcode 5 bits

clock

cycle (s)

Time

2. Fetch 3 bits for getting the operand address from R1 For transferring from a pointed address byte into A register,

STEP 2

IR gets Code bits 11100 111 from Memory

MOV A, @R1

3 bits specifies a register, the byte at that indirectly points and specify the address for operand

[Sign @ means R1 is a pointer]

Summary

We learnt

- The program is in machine codes inside the machine (memory associated with CPU)
- User program can be in assembly instructions
- User program can be in high level language such as C

We learnt

- Assembly instruction translates to opcode and operand(s) for each instruction