x86 Instruction Encoding
Part-1

Lecture # 36

Encoding and Decoding Instructions

- One of the skills we need to master in this course is the encoding and decoding of instructions.

- Encoding refers to translating an assembly-language instruction into machine code.

- Decoding instructions by the decoder within microprocessor generates control signals for the data path.

- Dis-assembly refers to translating machine code into an assembly-language instruction.

Instruction Format

Opcode	'mod r/m' byte	Offset/Displacement	Immediate data

Opcode Except for some few instructions that have a two-byte opcodes, most machine language instructions for the Intel 8086 processor have a one-byte opcode. The primary role of the opcode is to specify the operation to be performed by the CPU. But, smaller encoding fields may be defined within the opcode byte to provide other information to the CPU such as the size of operands (8-bit or 16-bit), where the result should be stored, or that a 16-bit immediate data is represented in the instruction using 8 bits.

Instruction Format

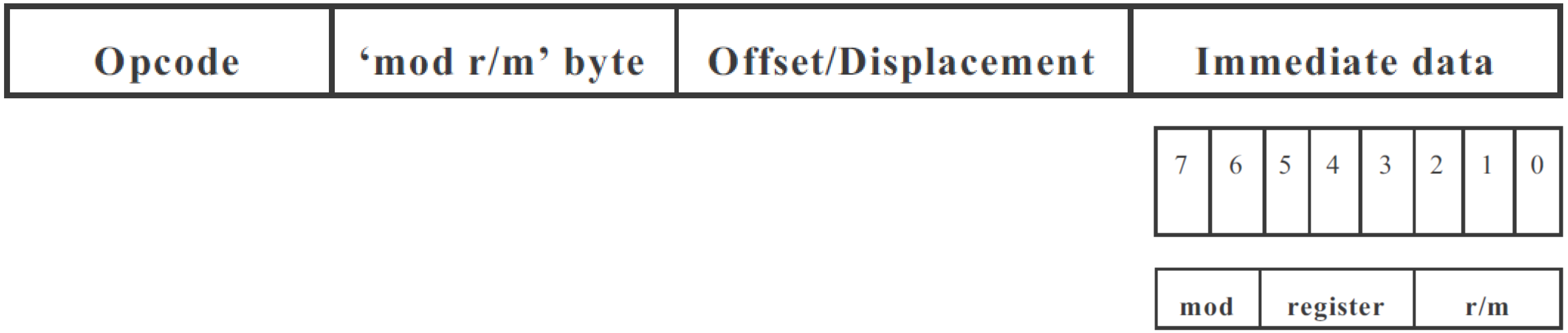
Opcode	**‘mod r/m’ byte**	**Offset/Displacement**	**Immediate data**

‘mod r/m’ byte this byte consists of three fields: the *mod field* (bits 7 and 6); the *register field* (bits 5, 4 and 3); and the *r/m (register/memory) field* (bits 2, 1 and 0) as follows:

7	6	5	4	3	2	1	0

mod	**register**	**r/m**

Instruction Format



- the **mod** and the **r/m fields** are used to let the CPU know whether or not there is a memory location operand, and how to get its offset when it is the case.
- For some instructions, the **register field** is used as an opcode extension. Otherwise, it is used with the **r/m field** to hold the codes of register operands.

Instruction Format

Opcode	'mod r/m' byte	Offset/Displacement	Immediate data

Offset/Displacement This field is present only if there is an operand in memory and its offset is specified in direct addressing mode or in an indirect addressing mode with a displacement.

An offset is specified here with its low and high bytes swapped. A one-byte displacement is a two's complement binary integer, whereas a word displacement is an unsigned binary integer. A word displacement is also specified with its high and low bytes swapped.

Instruction Format

Opcode	**‘mod r/m’ byte**	**Offset/Displacement**	**Immediate data**

Immediate Data this field is present only if there is an operand in immediate addressing mode. Immediate data are two’s complement binary integers and a word immediate data is specified with its low and high bytes swapped.