# Chapter 04: Instruction Sets and the Processor organizations

Lesson 04:

**Type of Operands in the instructions**

# Objective

- To understand type of the operands in instructions

# Type of operands

# Common Ways

- 4-ways— the instructions of an instruction set compute the address of the operands

1. At a register
2. At a memory address specified in instruction
3. At a memory address pointed by a register
4. The immediate operand (part of instruction after the opcode bits)

# 1) Register operand

- Operand is at a register—

MOV r0, r1; source and destination for register bits transfer both register operands

# 2) Memory address operand

- Operand is at a memory address—

LD r0, M (X); source operand is memory address X and destination operand is register operand

# 3) Indirect Memory reference address operand

- Operand is at a memory address pointed by a register—

ST {r1 [M (X)]}, r2; source operand is register r2 and destination operand is a memory address X pointed by the register r1

# Memory address as operand

# Register to memory or Memory to Register Operation Formats

- *I* = operation ($M_0$,.., $M_i$, $r_0$,.., $r_j$) or I = operation ($r_0$,.., $r_j$, $M_0$,.., $M_i$,)
- i = 0, 1 or 2 and j = 0, 1 or 2
- M─ means a memory address and r means a register
- $M_i$ ─ means memory content at address i, $ADDR_i$

# Memory address among $2^n$ locations

1) Memory address short or at a page or segment of memory – A memory address represented by few bits than *n*

# MUL $r_0$, M [$addr_1$].

- Direct memory by short address — If register $r_2$ is specified by 4-bits and memory addresses by 20 bits (a short address for 32-bits), then MUL $r_2$, M [$addr_1$] can be four-bytes ($m$ = 4) long when the opcode field can have 32 – 4 – 20 = 8 bits. [$r_3$ is implicit.] (r2-r3) pair ← r2 × M [$addr_1$].

# Memory address among $2^n$ locations

2) Memory address long - A memory address represented by all address bits represented by *n* bits

# MUL $r_2$, $r_3$, M [addr$_1$]

- Absolute long memory address─ If register pair $r_2$-r$_3$ is specified by 4-bits each and memory addresses by 32 bits, then MUL $r_2$, $r_3$, M [addr$_1$] can be six-bytes ($m$ = 6) long when the opcode field can have 48 – 8 – 32 = 8 bits. r2-r3 ← r2 × M [addr$_1$]

# Memory address among $2^n$ locations

3) Relative address— A memory address represented by relative displacement (plus or minus) from the next instruction address

# Memory address among $2^n$ locations

4) Offset address– A memory address represented by an offset with respect to a base address, the offset plus base address bits displacement (plus or minus) from next instruction address

# Memory address among $2^n$ locations

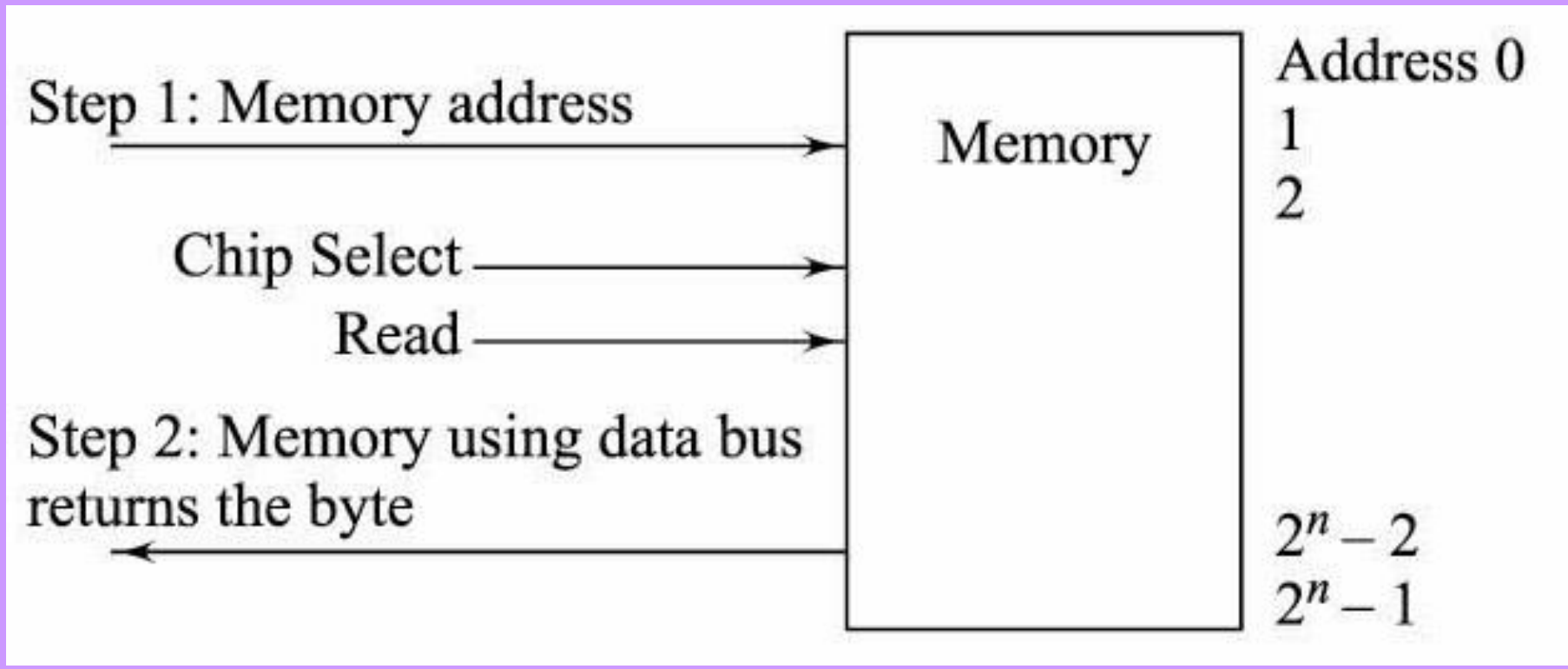
5) Base plus relative or offset, Index plus relative or offset, Base plus index plus relative or offset

# Memory Locations and Addresses and Fetch operation

# $2^n$ Locations Each having one byte

- Let $n$ = 20, then memory addresses are between 0 and $2^{20} - 1$ ($=2^{10} \times 2^{10} - 1 = 1024 \times 1024 - 1 = 1048575$)
- Memory = 1 MB as 1 MB = 1K × 1K B = $2^{10} \times 2^{10}$) as per convention for expressing the total memory locations

# Memory location identified by an address between $2^n - 1$



# Summary

# We Learnt

- Type of operands

End of Lesson 04 on

**Type of Operands in the instructions**