SNS COLLEGE OF ENGINEERING

Kurumbapalayam (PO), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE NAME: 19IT301 COMPUTER ORGANIZATION AND ARCHITECTURE

II YEAR/ III SEM

Unit 3 : Processor and Pipeling

Topic 1: Fundamental Concepts

Overview

- Instruction Set Processor (ISP) or Processor – Executes machine instructions and coordinates other units
- Central Processing Unit (CPU)
- High performance processors
  - Pipelining
  - Superscalar processing
- A typical computing task consists of a series of steps specified by a sequence of machine instructions that constitute a program.
- An instruction is executed by carrying out a sequence of more rudimentary operations.

Fundamental Concepts

- Processor fetches one instruction at a time and perform the operation specified.
- Instructions are fetched from successive memory locations until a branch or a jump instruction is encountered.
- Processor keeps track of the address of the memory location containing the next instruction to be fetched using Program Counter (PC).
- Instruction Register (IR)

3 steps to execute an Instruction

1. Fetch the contents of the memory location pointed to by the PC. The contents of this location are loaded into the IR (fetch phase).

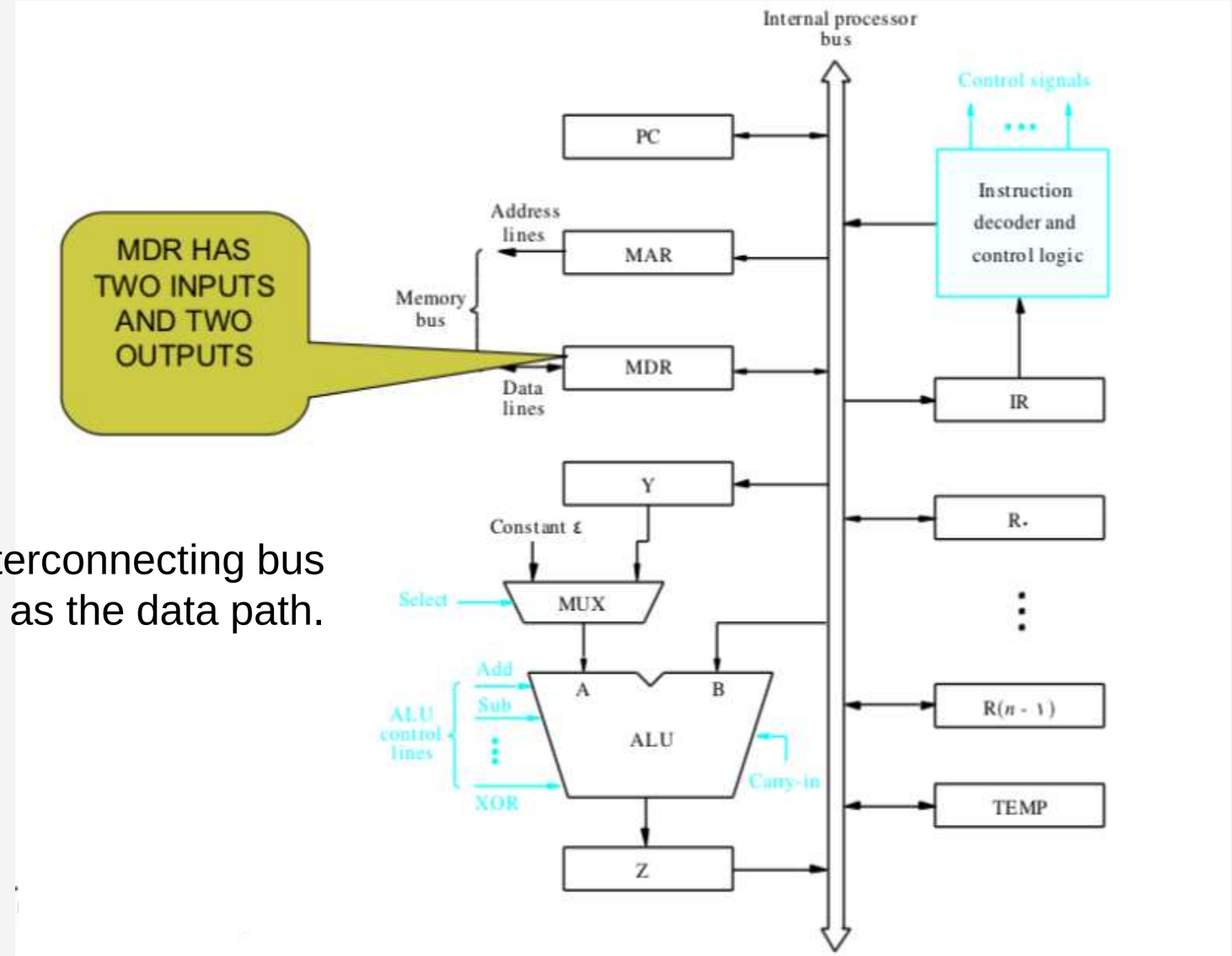
   $$IR \leftarrow [[PC]]$$

2. Assuming that the memory is byte addressable, increment the contents of the PC by 4 (fetch phase).

   $$PC \leftarrow [PC] + 4$$

3. Carry out the actions specified by the instruction in the IR (execution phase).

Single-bus Processor organization

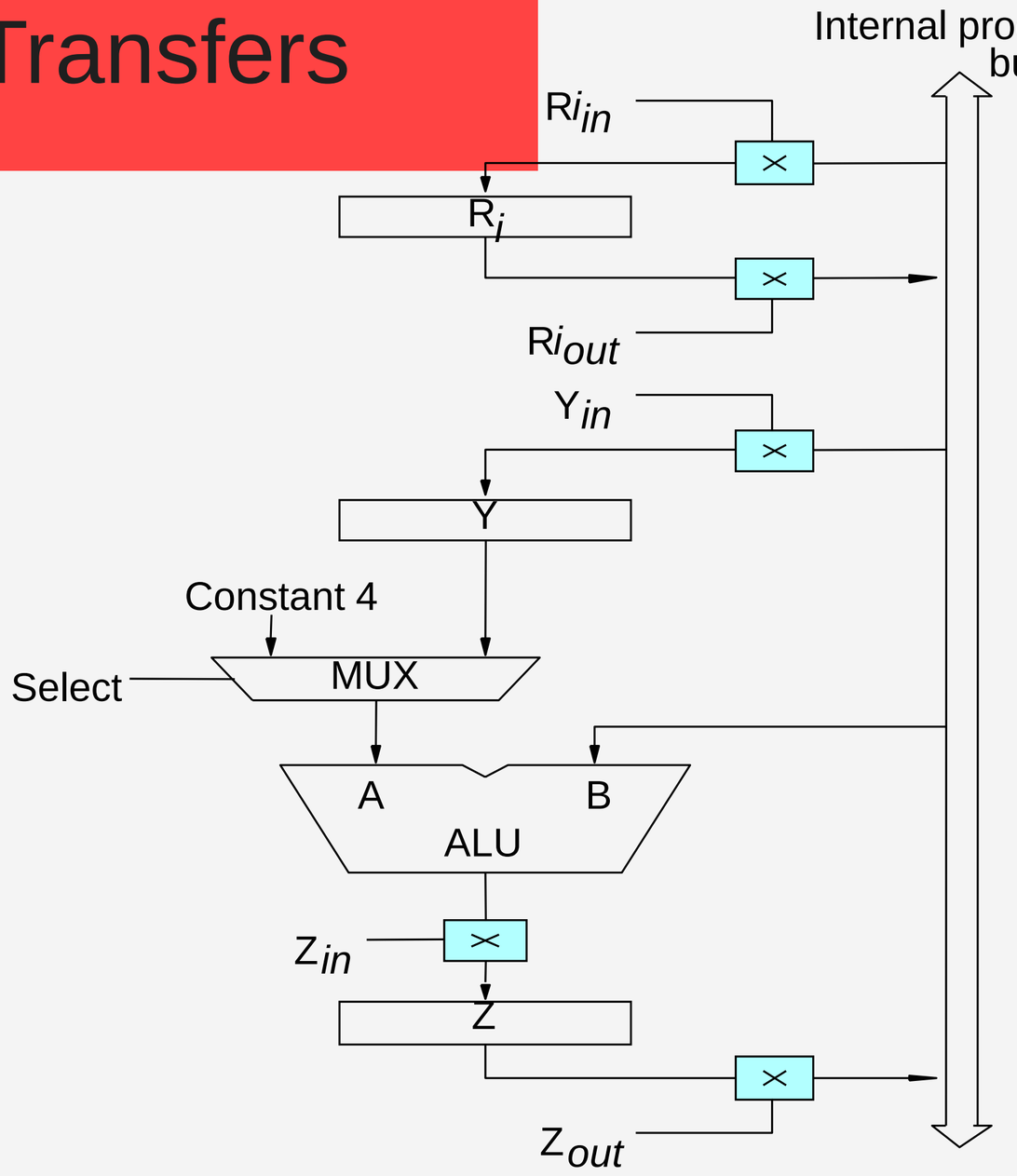


Data Path:
The registers, ALU and interconnecting bus are collectively referred to as the data path.

Few operations to execute an instruction

- Fetch the contents of a given memory location and load them into a processor register.
- Transfer a word of data from one processor register to another or to the ALU.
- Perform an arithmetic or a logic operation and store the result in a processor register.
- Store a word of data from a processor register into a given memory location.

Register Transfers



Input and output gating for the registers

Register control signals

- The input and output of register R*i* are controlled by signals is Ri_{in} and Ri_{out}
- Ri_{in} is set to 1 – data on the bus are loaded into R*i*
- Ri_{out} is set to 1 – the contents of register are placed on the bus.
- Ri_{out} is set to 0 – the bus can be used for transferring data from other registers

Example:

Transfer the contents of R1 to R4.

- Enable output of register R1 by setting $R1_{out}$=1. This places the contents of R1 on the processor bus.
- Enable input of register R4 by setting $R4_{in}$=1. This loads the data from the processor bus into register R4

Performing an Arithmetic or Logic Operation

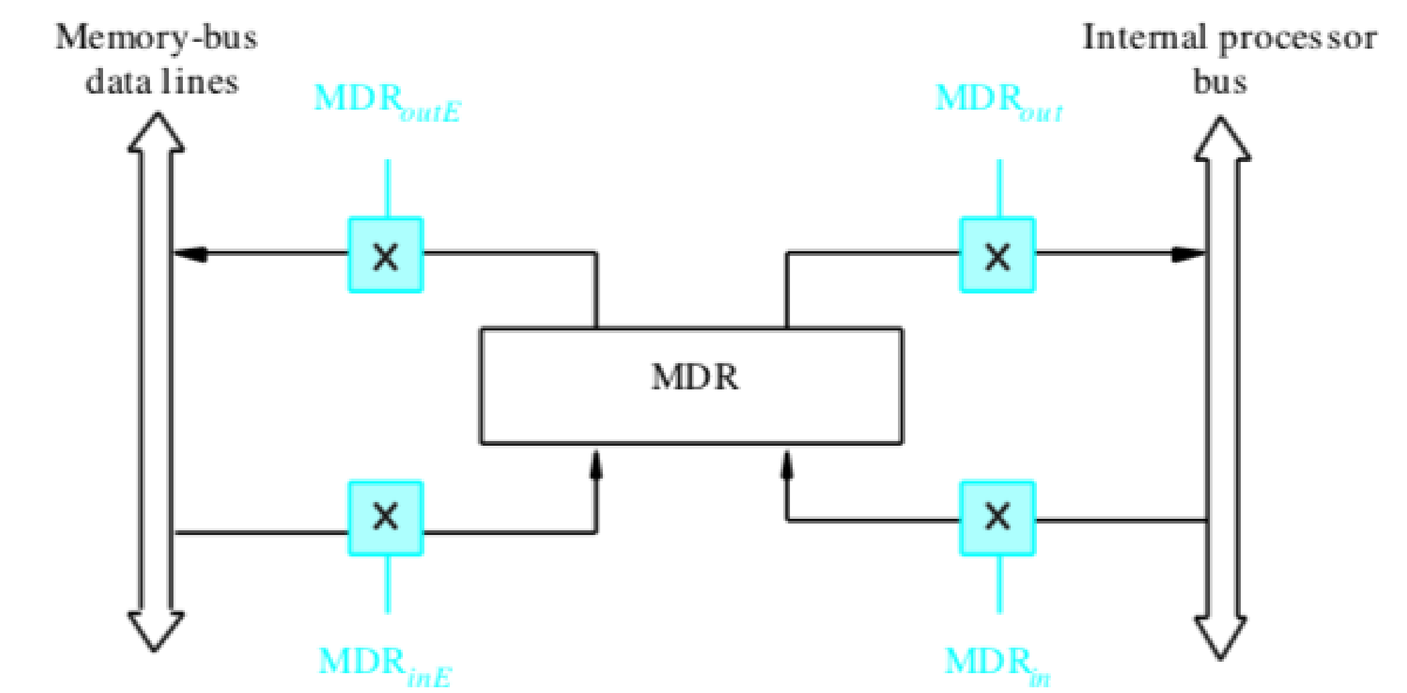
- The ALU is a combinational circuit that has no internal storage.
- ALU gets the two operands from MUX and bus. The result is temporarily stored in register Z.
- What is the sequence of operations to add the contents of register R1 to those of R2 and store the result in R3?

  i.e. R3 ← R1 + R2

  1. $R1_{out}$, Y_{in}
  2. $R2_{out}$, SelectY, Add, Z_{in}
  3. Z_{out}, $R3_{in}$

  * Signals given in any step are activated others are inactive for the clock cycle

Fetching a Word from Memory



Connection and control signals for register MDR

Processor:

- transfers address into MAR
- issue Read operation
- data into MDR

Fetching a Word from Memory

- The response time of each memory access varies (cache miss, memory-mapped I/O,…).
- To accommodate this, the processor waits until it receives an indication that the requested operation has been completed (Memory-Function-Completed, MFC).

Example:

Move (R1), R2

- MAR ← [R1]
- Start a Read operation on the memory bus
- Wait for the MFC response from the memory
- Load MDR from the memory bus
- R2 ← [MDR]

Signals activated for Move (R1), R2:

- $R1_{out}$, MAR_{in}, Read
- MDR_{inE}, WMFC
- MDR_{out}, $R2_{in}$

Storing a word in memory

- Address is loaded into MAR
- Data to be written loaded into MDR
- Write command is issued

Example: Move R2,(R1)

- $R1_{out}$, MAR_{in}
- $R2_{out}$, MDR_{in}, Write
- MDR_{outE}, WMFC

Thank You