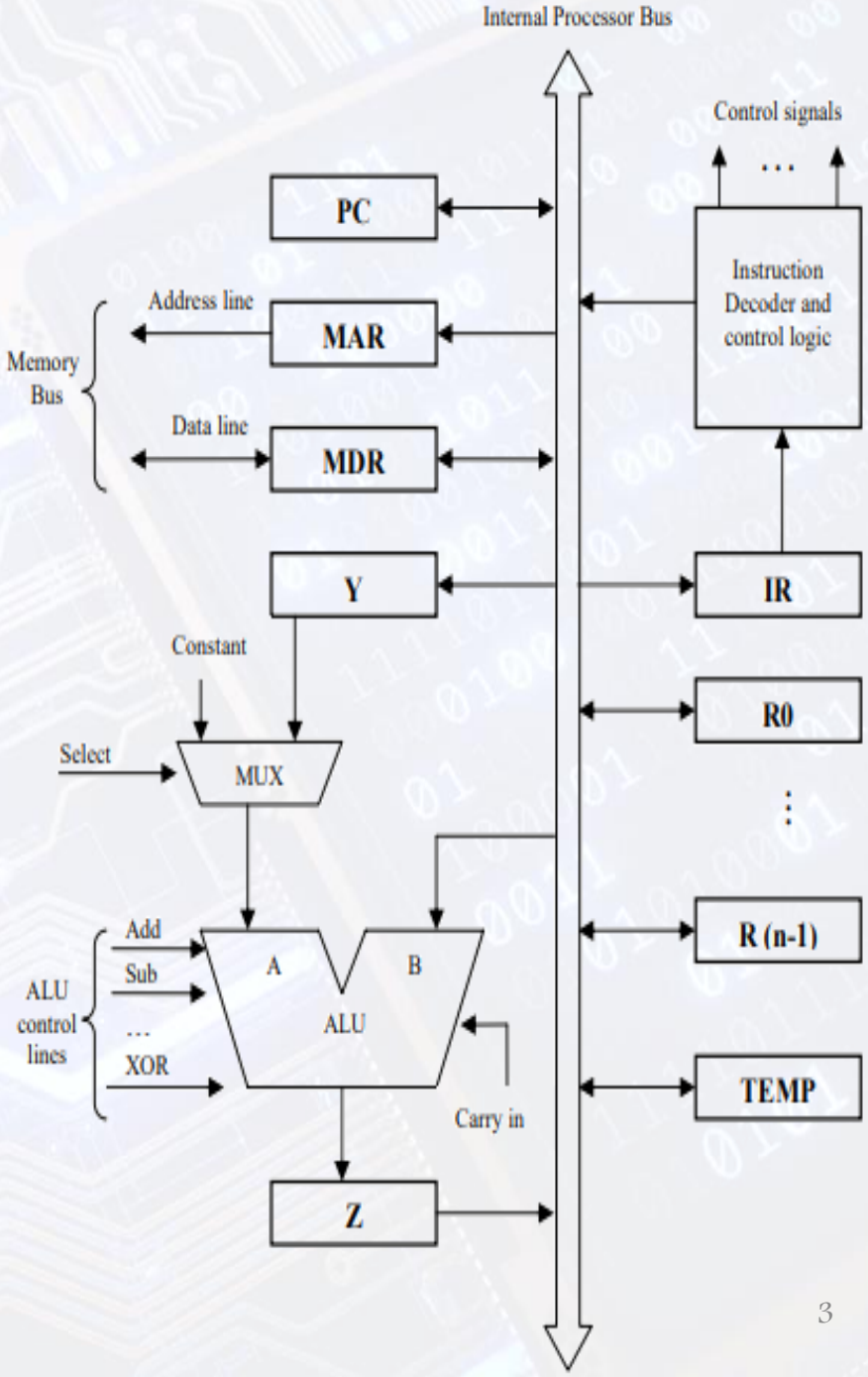
DataPath

Module 3~ Part C

Introduction

- Movement of data within a processor from one component to another
- Registers, ALU, and the interconnecting bus collectively form the datapath
  - ALU
    - Performs arithmetic and logical operations
    - Contains control lines to select one of the possible ALU operations
  - Registers
    - General purpose registers to store data, memory address register (MAR), memory data register (MDR), program counter (PC), instruction register (IR), temporary registers (optional)
  - Buses
    - Internal (one or more) and external to carry the data from one component to another

Single Bus Organization of Datapath



Basic Steps in Instruction Execution

- Fetch the contents of the memory location pointed to by the PC. Load these contents into the IR to be interpreted.

$$IR \leftarrow [PC]$$

- Assuming that the memory is byte addressable, increment the contents of the PC by 4.

$$PC \leftarrow PC + 4$$

- Carry out the actions specified by the instruction.

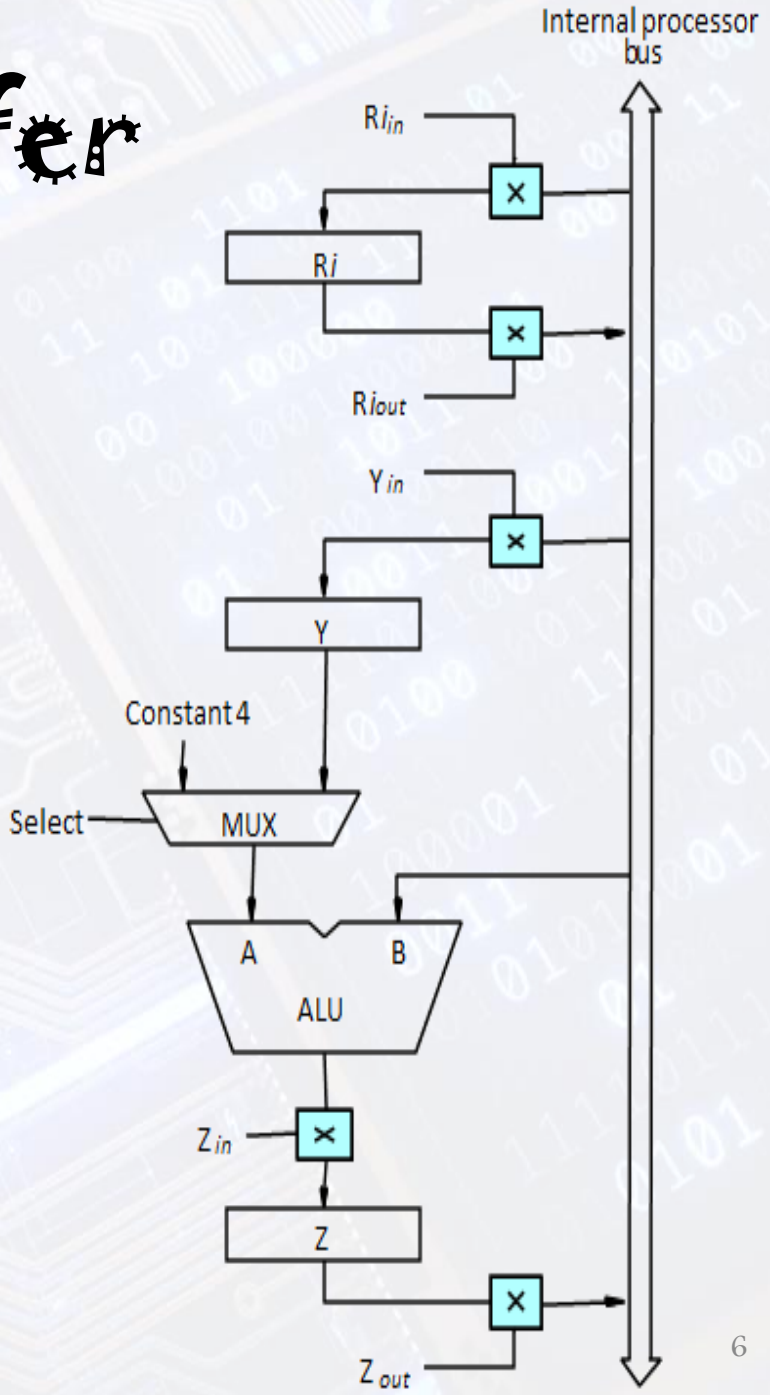
Basic Steps in Instruction Execution

- Most instructions involve the following operations:
  - Transfer a word of data from one processor register to another or the ALU
  - Perform an arithmetic or logical operation and store the result in a processor register
  - Fetch the contents of a memory location and load them into a processor register
  - Store a word of data from a processor register into a memory location

Register Transfer

MOV R1, R4

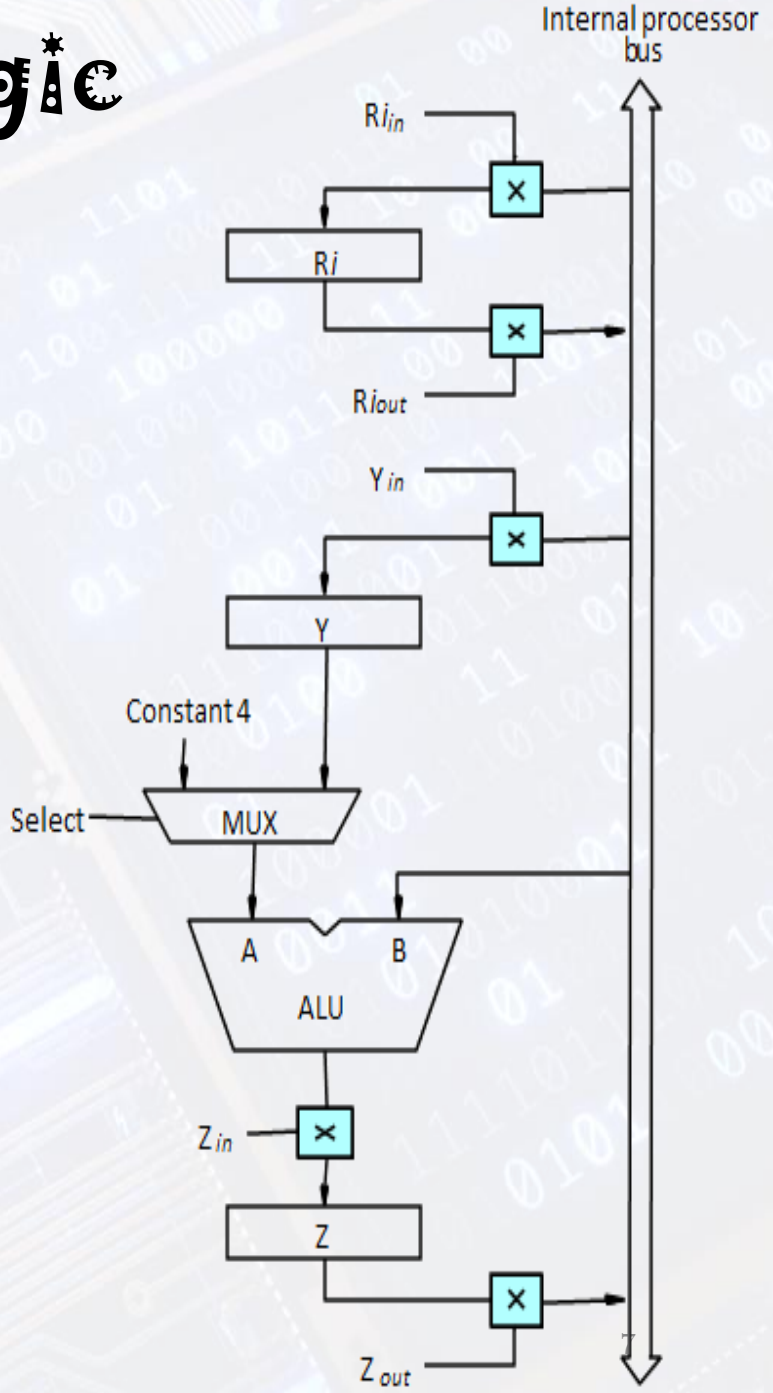
- Enable the output of $R1$ by setting $R1_{out}$ to 1. This places the contents of $R1$ on the processor bus.
- Enable the input of the register $R4$ b setting $R4_{in}$ to 1. This loads data from the processor bus into register $R4$.



Arithmetic and Logic Operations

ADD R1, R2, R3

- $R1_{out}, Y_{in}$
- $R2_{out}, SelectY, Add, Z_{in}$
- $Z_{out}, R3_{in}$



Memory Read and Write

- Memory Read

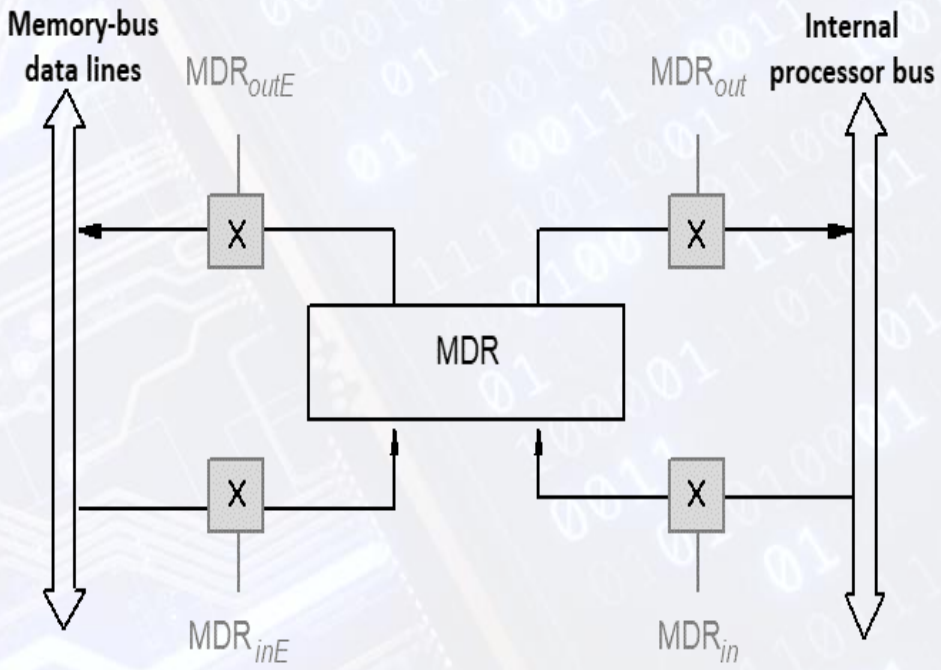
MOV (R1), R2

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

- Memory Write

MOV R2, (R1)

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



Execution of a Complete Instruction

ADD (R3), R1

PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}

Z_{out}, PC_{in}, Y_{in}, WMFC

MDR_{out}, IR_{in}

$R3_{out}$, MAR_{in}, Read

$R1_{out}$, Y_{in}, WMFC

MDR_{out}, SelectY, Add, Z_{in}

Z_{out}, $R1_{in}$, End

Multi Cycle Data Path Architecture

Why Multi Cycle Data Path ?

- In a single cycle data path architecture , exclusively single data word could be moved through the bus in a given clock cycle
- Because of this no of steps needed to execute the inst increases
- To decrease the no of steps needed to execute the inst and to transfer more than one word in a clock cycle we go for multicycle

How Multi Cycle Data Path Works

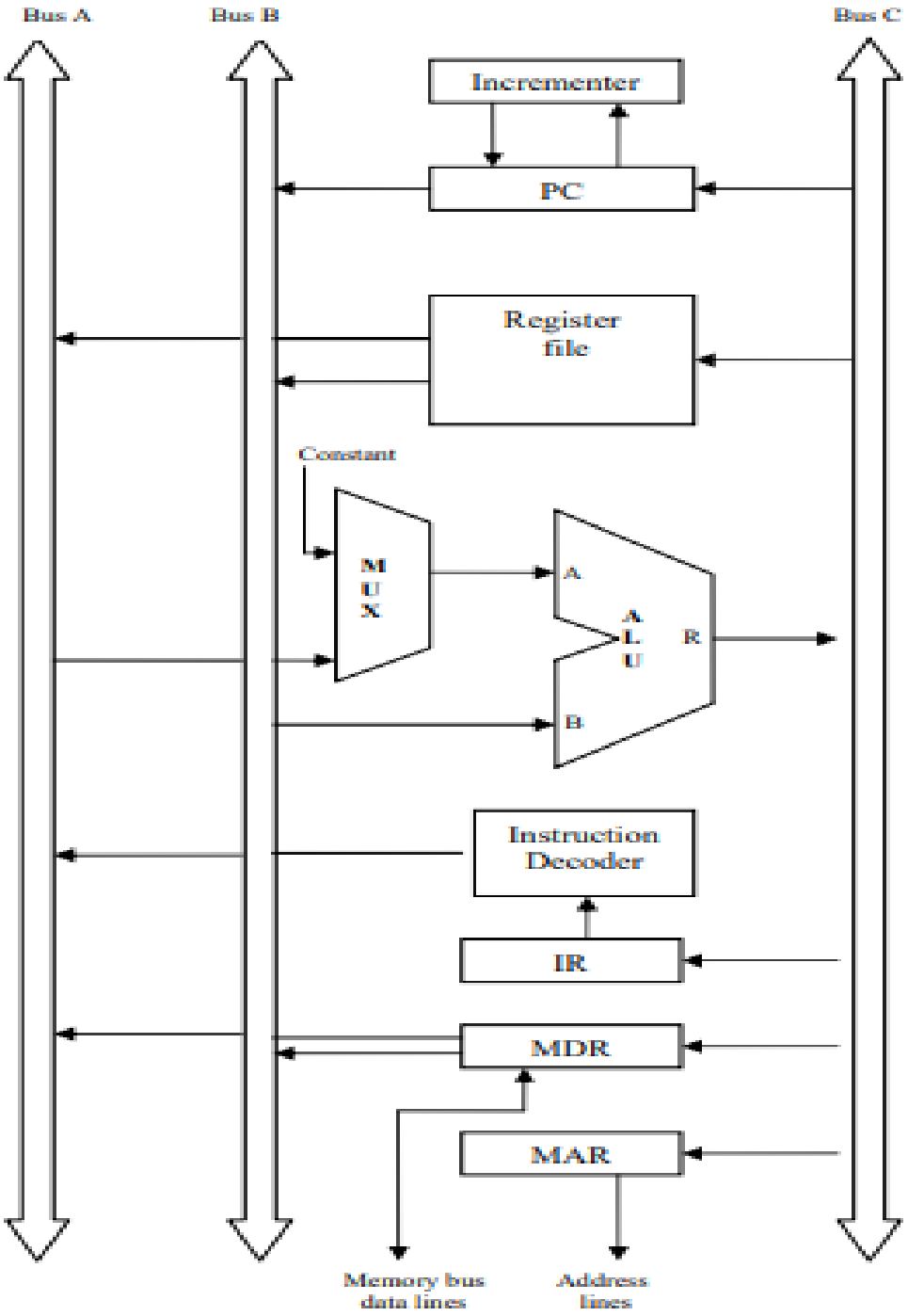
- Three buses are used to link reg and ALU of the CPU

- All GPR ,

  $R_1, R_2 \ldots R_n$ are presented in one block known as reg files

- Figure 2. shows the register files has three ports

Figure 2. Multi Cycle Data Path Architectur



How Multi Cycle Data Path Works

- One input and two output ports
- Therefore data of three registers are possible to access in single clk cycle
- Through Bus C, the value could be put in one reg
- Data from two regs is available through Bus A and Bus B

How Multi Cycle Data Path Works

- Bus A and B are used to move the source operands to i/ps of the ALU A and B
- After ALU process is executed the resultant is moved to destination operand through the bus C
- Separate incremental unit is provided to increment the value of PC after every instruction is executed

Execution of Instruction using Multi Cycle Data Path Add R_1, R_2, R_3 Control Sequence

- The inst adds the values of register R_2 & R_3 and stores the resultant in R_1

Step 1: PC_{out}, R=B, MAR_{in}, Read, IncPC

Step 2: WMFC

Step 3: MDR_{out}, IR_{in}, R=B

Step 4: $R_{2\,out}$, $R_{3\,out}$, Add, $R_{1\,in}$, End

Multi Cycle Data Path Add R_1, R_2, R_3 Control Sequence Explanation

- Step 1: The value of the PC are moved to MAR by means of Bus B to begin Read operation. PC→MAR

- Parallelly PC is incremented point towards the next instruction PC→PC+1

Multi Cycle Data Path Add R_1, R_2, R_3 Control Sequence Explanation

- Step 2: The processor waits for WMFC signal from the memory
- Step 3: The inst code is moved from MDR to IR MDR→ IR
- Step 4: The inst decoder decodes the IR contents

Multi Cycle Data Path Add R_1, R_2, R_3 Control Sequence Explanation

- Step 4: Two values from reg R_2 & R_3 are made accessible at inputs A and B of ALU by means of Bus A & B

- By activating the Add signal two inputs are added

- Through Bus C the resultant is stored in R1