

Basic Processing Unit

1

Central Processing Unit (CPU)

- Examine **internal structure of processor** and how it performs the tasks of **fetching**, **decoding** and **executing** the instructions of a program

```

                SUB    R0, R0, R0; R0 has the value of index i
Loop_Begin:    CMP    R0, N
                JEQ    Loop_End
                ADD    A[R0], B[R0], C[R0]
                INC    R0
                JMP    Loop_Begin
Loop_End:

```

- To execute a program, the processor fetches one instruction at a time and performs the operation specified
- Instructions are fetched from the successive memory locations until the branch instruction is encountered

2

2

Central Processing Unit (CPU)

- **Program counter (PC):**
 - Holds the next instruction to be fetched
 - Helps the processor to keep track of the address of the memory location of next instruction to be fetched
 - After fetching, the content of PC is updated to point to the next instruction in the sequence
 - Branch instruction may load different value into the PC
- **Instruction register (IR):**
 - Instruction to be executed is loaded into the IR

3

3

Execution of an Instruction

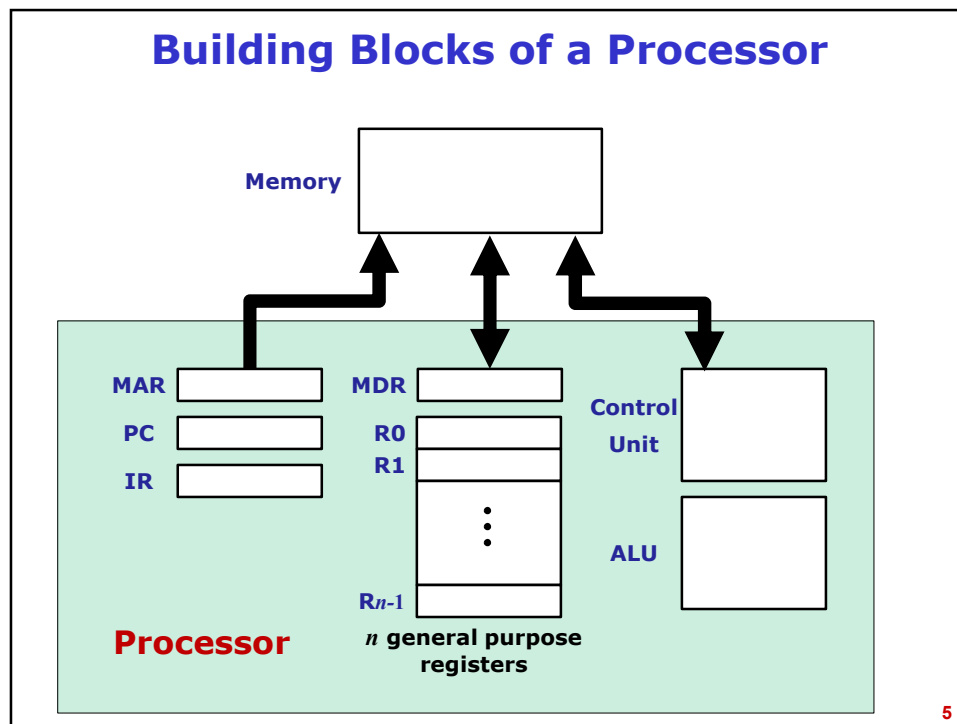
- Suppose each instruction is 4 bytes in length and stored in one memory word
- Assume that memory is byte addressable
- Steps to execute an instruction:
 - **Step1:** Fetch the content of the memory location pointed to by the PC and are load into IR

$$IR \leftarrow [[PC]]$$
 - **Step2:** Increment the content of the PC by 4

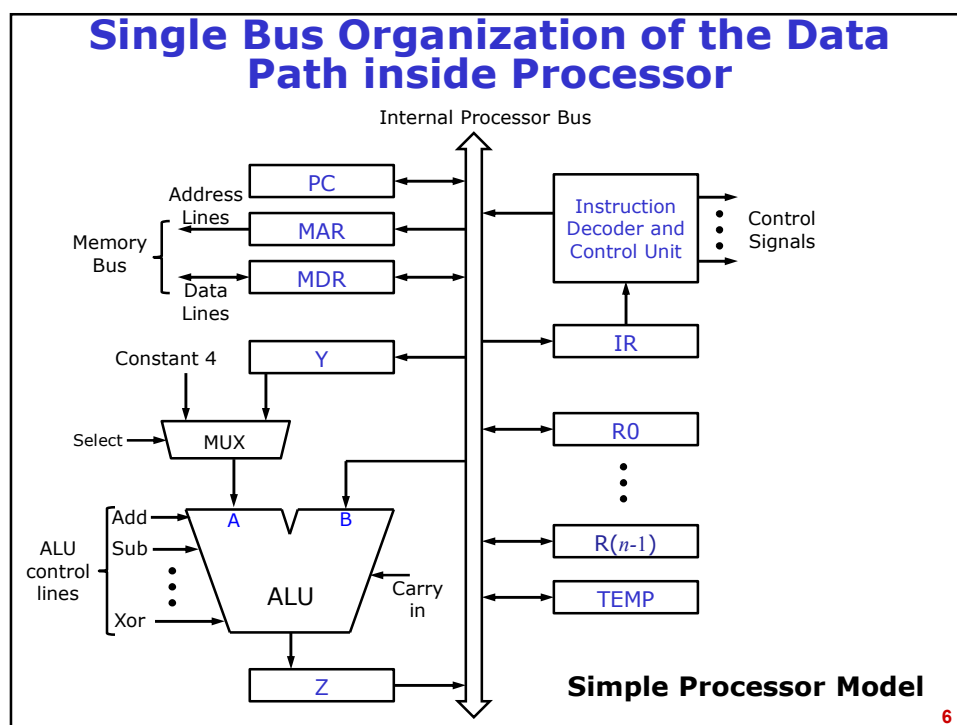
$$PC \leftarrow [PC] + 4$$
 - **Step3:** Carry out the actions specified by the instruction in the IR
- Step1 and Step2 repeated each time to fetch the instruction. They are referred to as **Fetch Phase**
- Step3 is referred as **Execute Phase**

4

4



5



6

Operations During Instruction Execution

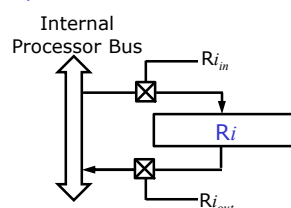
- Instructions are executed by performing **one or more** of the following operations:
 - Transfer a word of data from one processor register to another or to ALU
 - Perform arithmetic or logic operation and store the results in the processor register
 - Fetch the content of a given memory location and load them into a processor register
 - Store a word of data from a processor register into a given memory location
- All the operations and data transfer within processor take place within a time period defined by **processor clock**
- Control signals that govern a particular transfer are asserted at the start of the clock cycle
- Each action is performed in **one or more clock cycles**

7

7

Register Transfers

- Transfer a word of data from one processor register to another or to ALU
- Example: **MOV R1, R4** Data is transferred from R4 to R1



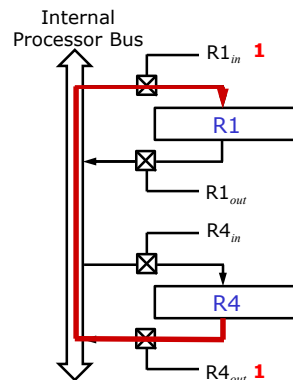
- For each register, two control signals are used
 - One to place the content of the register on the bus
 - Another to load the data on the bus into the register
- Input and output of register R_i are connected to bus via switches controlled by the signals $R_{i_{in}}$ and $R_{i_{out}}$
 - $R_{i_{in}} = 1$: Data on the bus loaded into R_i
 - $R_{i_{out}} = 1$: Data in R_i placed on the bus

8

8

Register Transfers

- Example: `MOV R1, R4`



- $R4_{out}$, $R1_{in}$
- Processor completes its internal data transfer in 1 clock cycle

9