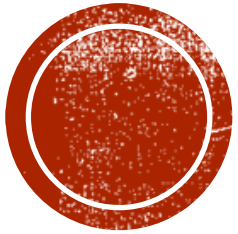
COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code : CSE 2151

Credits : 04



CONTROL UNIT

MODULE 4

CONTROL UNIT

- **Basic concepts**
  - Fundamentals of Control Unit
  - Register transfer notations and descriptions
  - Buses

- **Design methods**
  - Hardwired approach
  - Microprogramming

INTRODUCTION

- CPU is viewed as a collection of two major components:
    - Processing section
    - Control Unit
- Control unit's responsibility is to drive the associated processing hardware by generating a set of signals that are synchronized with the master clock.
- In order to carry out a task, the CU must generate a set of control signals in a predefined sequence governed by the hardware structure of the processing section.

INTRODUCTION (CONTD.)

- Inputs to CU are:
  - Master clock
  - Status information from processing section
  - Command signals from external agent (like RESET, ABORT)
- Outputs produced by CU
  - Signals that drive the processing section and responses to an external environment.
- Control unit undertakes the following responsibilities:
  - **Instruction interpretation:** (CU read instructions, recognizes the instruction type, gets operands and route to appropriate functional units of Processing Unit (PU), necessary control signals are then issued to the PU to perform desired operation)
  - **Instruction sequencing:** CU determines the address of next instruction to be executed and loads it on to PC.

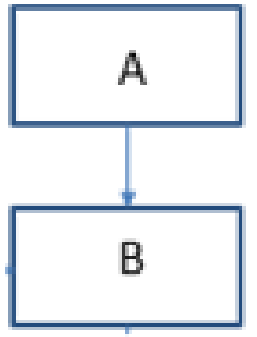
BASIC CONCEPTS

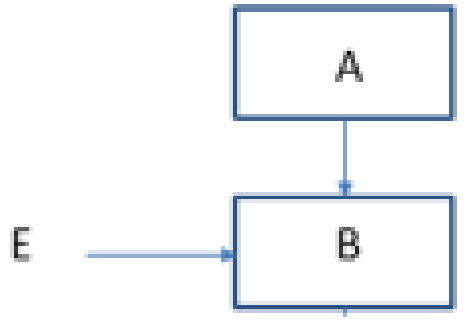
- Basis for CU design are register transfer operations
  1. 8-bit info moved from Register A to Register B.
     - Such operation is described as B←A
     - Declaring registers: Declare registers A[8], B[8], PC[16] ;
  2. Register can be defined as a portion of some other register.
     - Assigning higher order byte of 16-bit PC: Declare subregisters PCHI[8]= PC[15-8] ;
  3. Assigning individual bits
     - B[0]=A[7] means MSB of A is copied to LSB of B.
  4. Normally two inputs are associated with each register:
     i. Enable input (E) or control input controls the data flow from A to B
     ii. Data input

     Register B is loaded with A only when E is held high else contents of register remain the same.

     Such conditional transfer is expressed as

     E: B←A



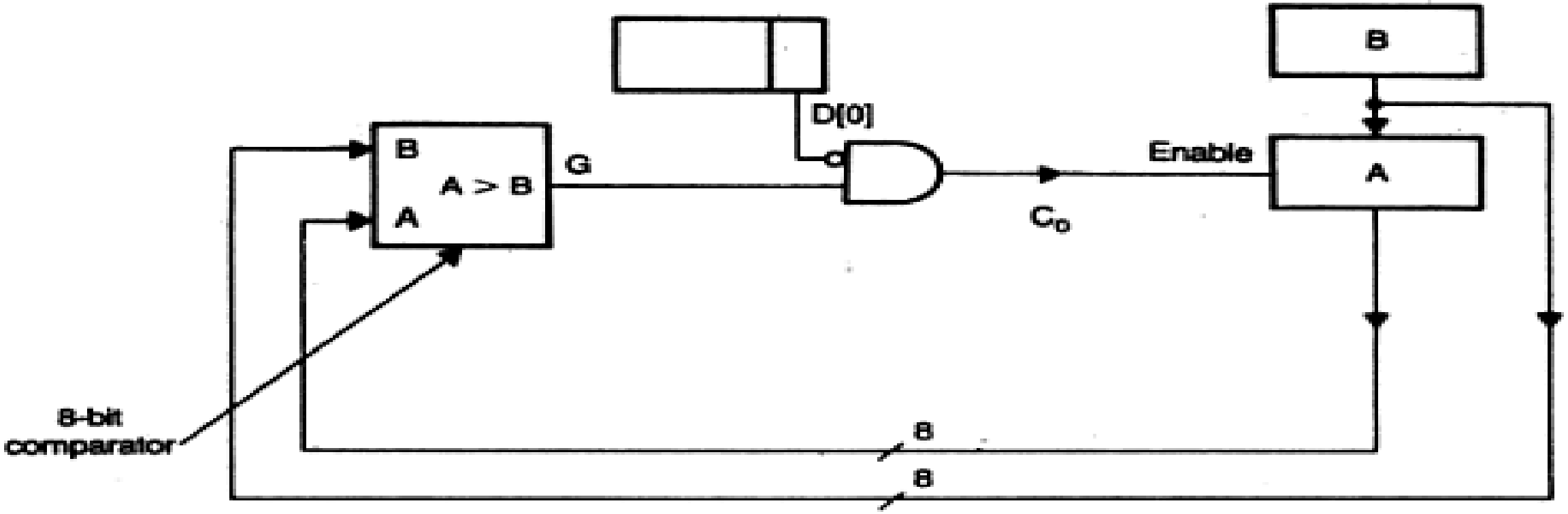


BASIC CONCEPTS (CONTD.)

5. Control input can be a function of more than one variable.

   IF A>B and D[0]=0 THEN A←B

- Comparator : if A>B, the output G from the comparator is set to high
- Conditional transfer: C_0: A←B ; where $C_0 = G \wedge D[0]'$



BASIC CONCEPTS (CONTD.)

6. To perform register transfer operation that involves selection.

If x=0 and t=1, then

$A \leftarrow B$

else

$A \leftarrow D$

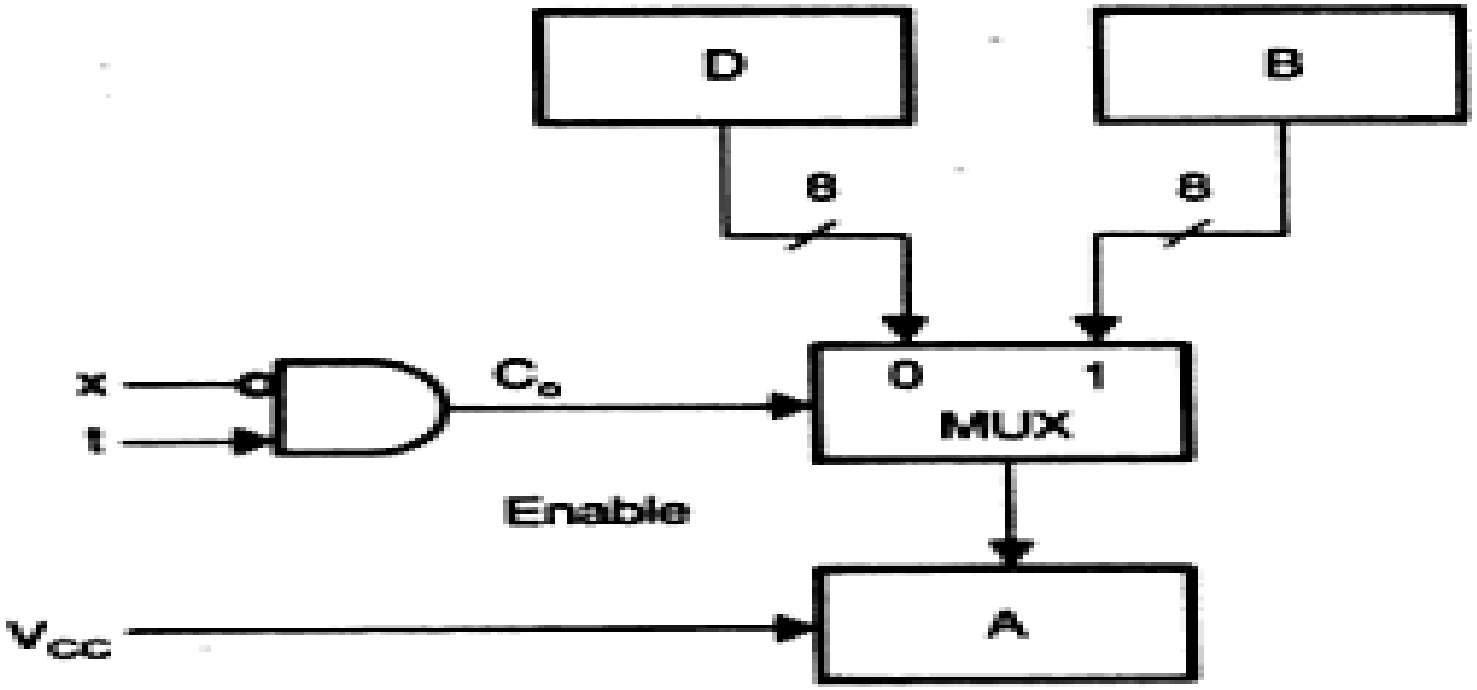
Such transfer is expressed as

$C_0 : A \leftarrow B$;

$C_0' : A \leftarrow D$;

Where $C_0 = x't$ and

$C_0' = (x't)' = x + t'$



BASIC CONCEPTS (CONTD.)

- The other register transfer operations are
- D←A’; Transfer the complement of A to D.
- A← A+1; Increment the content of A by 1.
- A← A-1; Decrement the content of A by 1.
- D← A V B; A OR B, store result in D
- D← A Λ B; A AND B, store result in D
- LSR(A); Logical shift right
- ASR(A); Arithmetic shift right
- LSL, ASL, ROR, ROL
- A$Q – used to concatenate A and Q
  - ASR(A$Q) ;

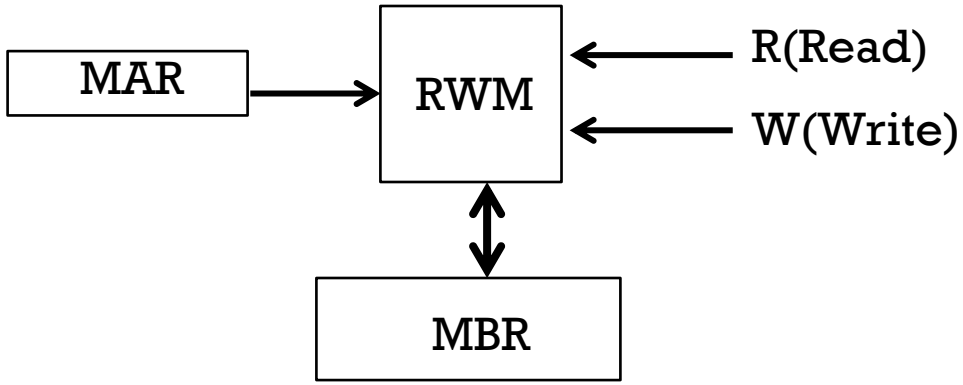
BASIC CONCEPTS (CONTD.)

- Whenever RWM is a part of processing section , MBR and MAR are associated with RWM unit.

MAR holds the address of desired memory word and MBR as buffer register in all data transfer operations.

- R: MBR← M((MAR))
- W: M((MAR))← MBR

The line b/n RWM and MBR is bidirectional bus, and it can be easily implemented using tristate buffers.



BASIC CONCEPTS (CONTD.)

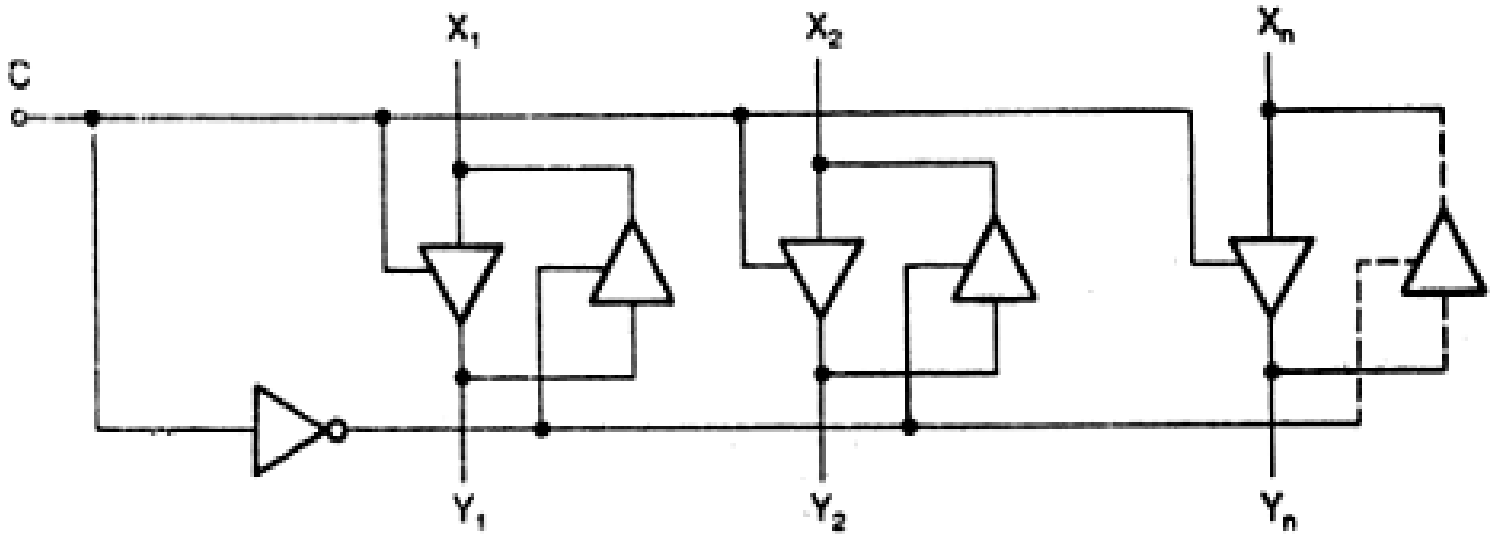


Fig: Bidirectional Data bus

- When C=1, X to Y
- When C=0, Y to X

BASIC CONCEPTS (CONTD.)

i. Declare buses Inbus[4] and outbus[4]; //4-bit buses
ii. A=inbus; // data of inbus is transferred to Reg A when next clock arrives
iii. Outbus = B[7:4] //Higher order 4 bits of 8-bit register B are made available on the outbus for one clock period.

Rate at which computer performs operations (such as A← A+M, A←A Λ B) is determined by bus structure.

BUSES

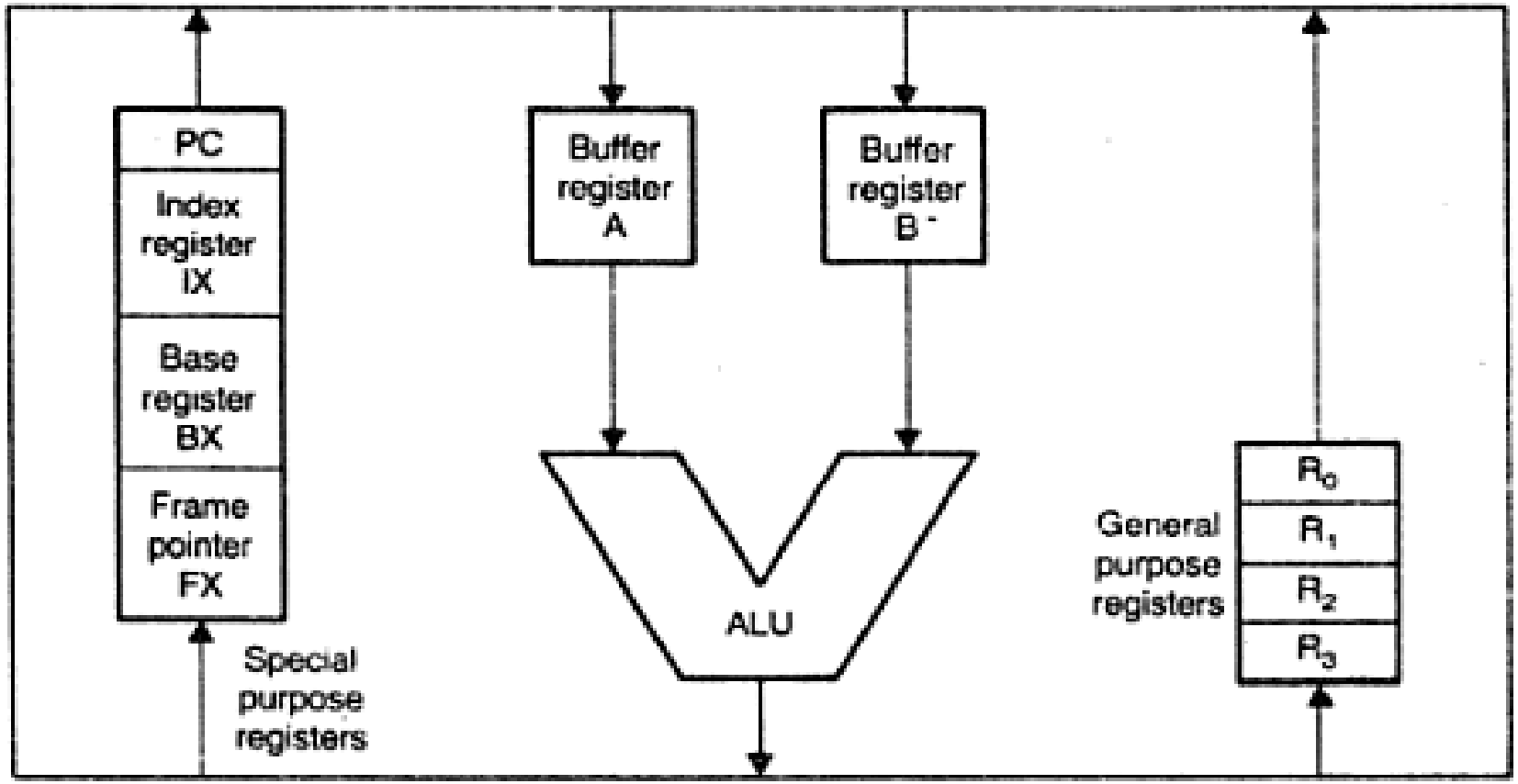
- Several types of bus structure within the CPU:
  i) Single-bus oriented ALU
  ii) Two-bus oriented ALU
  iii) Three-bus oriented ALU

BUSES

- Single-bus oriented ALU

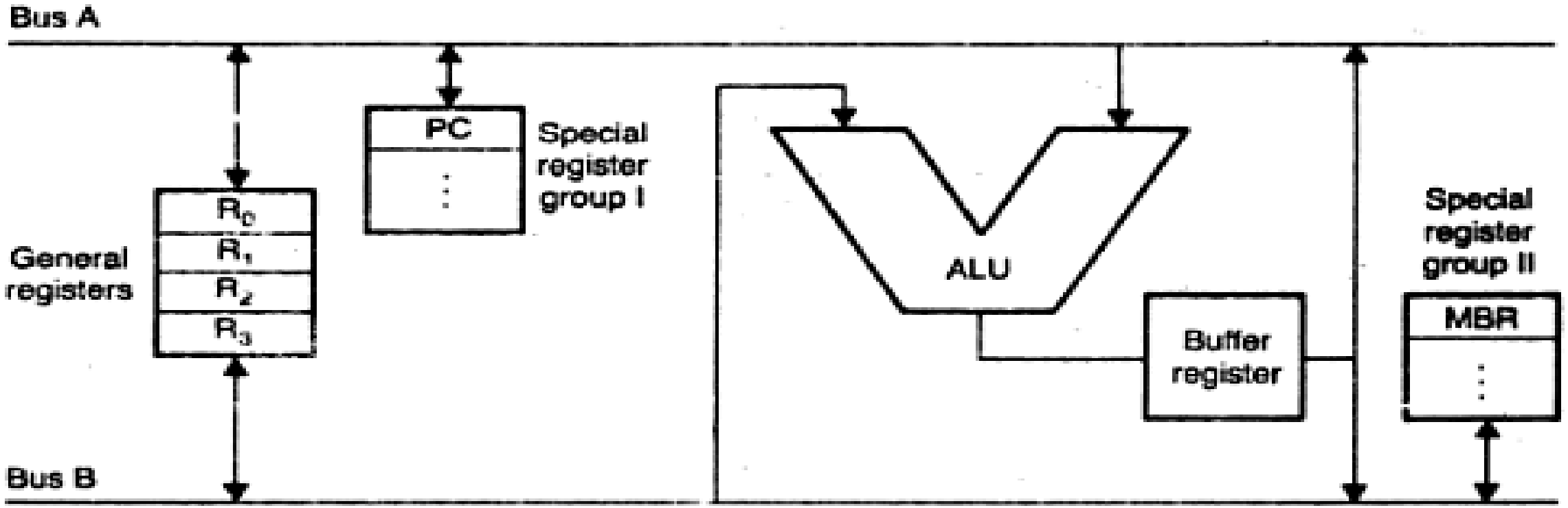
Disadvantages:

- Affects speed of execution of a typical 2 operand memory
- Increases the number of states in control logic. Hence more HW may be required to design control unit



BUSES

- Two-bus oriented ALU

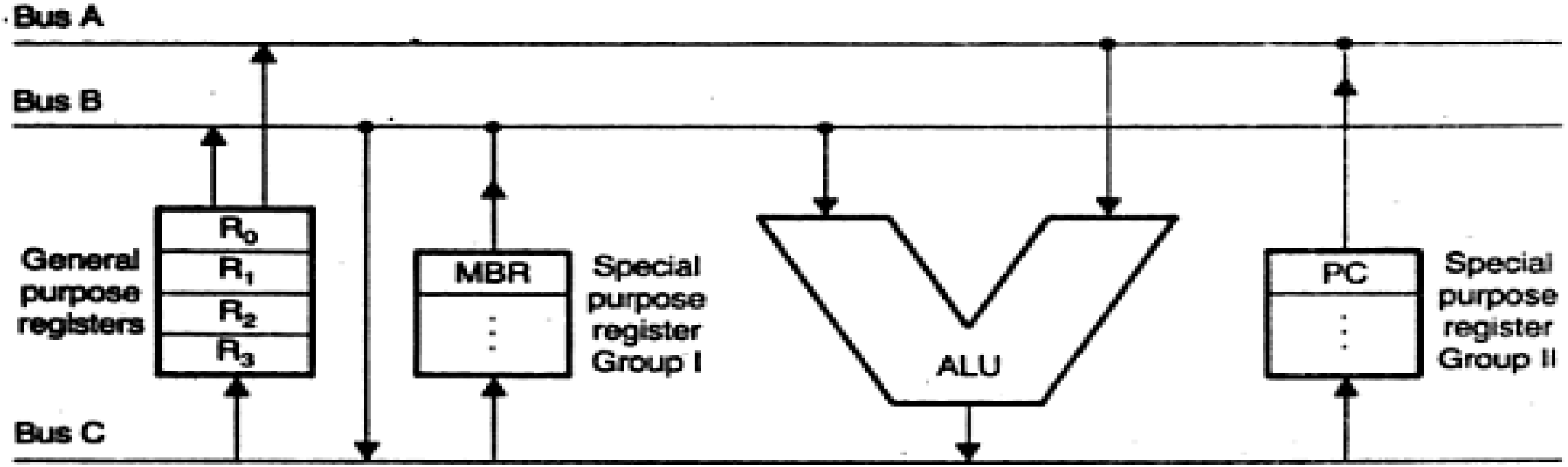


BUSES

- Two-bus oriented ALU
- Output Buffer register is used to prevent collision of the buses
- 1st cycle: loading operands and storing result in O/P buffer
- 2nd cycle: result in O/P buffer is pushed to bus(destination). The contents of buffer register can be gated to either bus A or bus B.

BUSES

- Three-bus oriented ALU



TOPICS COVERED FROM

- Textbook 3:
  - Chapter 4: 4.1, 4.2