

## CHAPTER 1-4.

### 7

# MICROOPERATION AND DESIGN OF ARITHMETIC LOGIC UNIT

7.3

## 7.1 MICROOPERATION

The operation performed on the data stored inside the register is called microoperation. The examples of microoperation are add, shift, load and clear. The operation performed on the data stored in register is faster than the data stored in memory.

## 7.2 REGISTER TRANSFER LANGUAGE (RTL)

The symbolic representation to specify microoperation is called register transfer language. This is the compact and short description method to represent the behavior of an instruction and the organization of computer. The required hardware circuit performs the microoperation and transfers the result to one of the register. The statement  $R2 := R1$  represents that the content of register R1 is transferred to R2. This statement can be represented using register transfer language as  $R2 \leftarrow R1$ . The register transfer language method to represent microoperation with control signal is shown ahead.

$PT : R2 \leftarrow R1$

Here P is the control variable generated by control unit and T is the specific time signal. That means this microoperation is executed when control and timing signal both are 1.

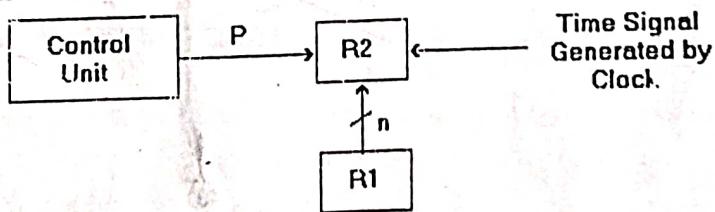


Figure 7.1 Register Transfer Operation

9/2010

## CH :- 4 Register Transfer Lang.

4"

Digital System :- A DS is an interconnection of digital H/W modules that accomplish a specific info-processing Tasks. Digital Components such as Reg., decoders, arithmetic elements & Control logic<sup>etc</sup> are interconnected with common data and control paths to form a digital Comp. System.

Digital Systems are characterised in Terms of:-

- (i) Registers they contain
- (ii) Op<sup>n</sup> that can be performed on the data stored on them.

Micro-op<sup>n</sup> :- The op<sup>n</sup> performed on the data stored inside the register is called Micro-op<sup>n</sup>.

Examples :- Add, Shift, Load & Clear.

The op<sup>n</sup> performed on the data stored in Registers is faster than the data stored in memory.

Bidirection Shift Register can perform shift Right & shift Left Micro- op<sup>n</sup>.

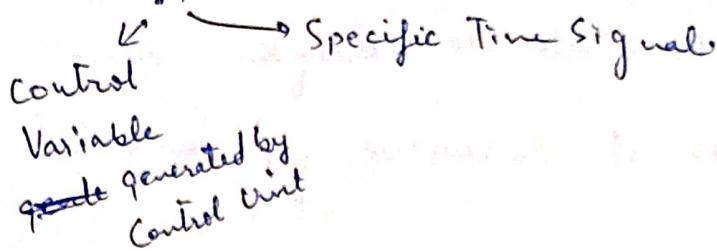
Register Transfer Lang :- The Symbolic Notation used to describe the micro-op<sup>n</sup> among registers is Called a RTL. It is a convenient tool for describing the internal organization of digital computers in concise & precise manner.

$$\swarrow \\ R_2 := R_1$$

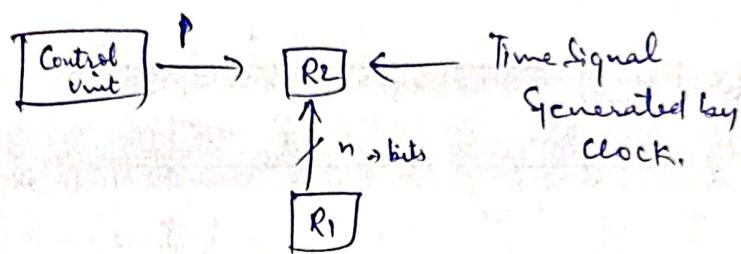
$$RTL : \rightarrow R_2 \leftarrow R_1.$$

The Register Transfer Language Method to represent Micro-op<sup>n</sup> with control signal.

pt:  $R_2 \leftarrow R_1$



That means this micro-op<sup>n</sup> is executed when control & Timing Signal both are 1.



Reg. Transfer Op<sup>n</sup>.

Prob :- Represent the following cond. Control statement by

Two Register Transfer statements with Control fx's.

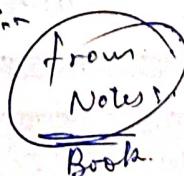
if ( $P=1$ ) then ( $R_2 \leftarrow R_3$ ) else if ( $Q=1$ ) then  
( $R_1 \leftarrow R_3$ )

Sol :-

P:  $R_2 \leftarrow R_3$

$P'Q$ :  $R_1 \leftarrow R_3$

\* Basic Symbols:-



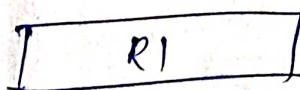
## Basic Symbols for Reg. Transfer.

representation of Registers,-

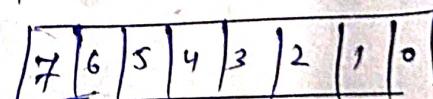
Comp. registers are designed by capital letters sometimes followed by numerals to denote the # of the register).

For Example:-

- The register that holds an address for the memory unit is designated by the name MAR
- The PC register that holds the add. of next inst to be executed.
- IR that holds the inst under execution
- AC → Result Storage.
- R1 → Processor Register.
- The individual flip flops in an n-bit register are numbered in seq. from 0 to n-1, starting from 0 in the RM position & increasing the no's toward the left



(a) Representing  
a Register



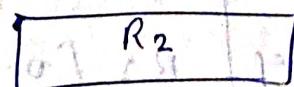
(b) showing individual  
bits of a 8 bit

(c)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

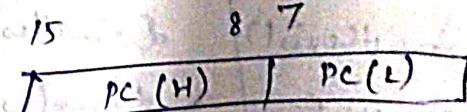
0

(d)



No. of bits

of 16 bit



low byte &  
High byte.

## Examples

Figure 6

$R_2(0-1)$ ,  $R_2(4)$

Denotes a Rep.

Min:

$\rightarrow$

Capital delta &

$\rightarrow$

Parenthesis ()

$\rightarrow$

Arrows  $\leftarrow$

Colon :

$\rightarrow$

Colon :

$\rightarrow$

Comma ;

$\rightarrow$

Separate 2 Micro op's.

$\rightarrow$

$A = B$ ,  $B = C$

## Parallel Micro-Op :-

Place the op<sup>n</sup> on the same line &

Separate with Commas. Same Clock Cycle.

Control fx : op<sup>1</sup>, op<sup>2</sup>.....

Eg :- P: R3 ← R5, MAR ← IR

Serial Micro-Op :- Just Sequence the op<sup>n</sup> on subsequent lines. Each op<sup>n</sup> takes a clock cycle.

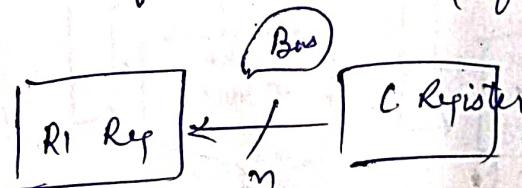
Control fx 1 : op 1.

Control fx 2 : op 2.

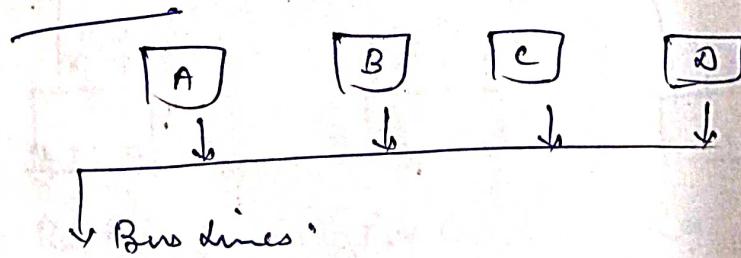
Bus & Memory

Transfers :-

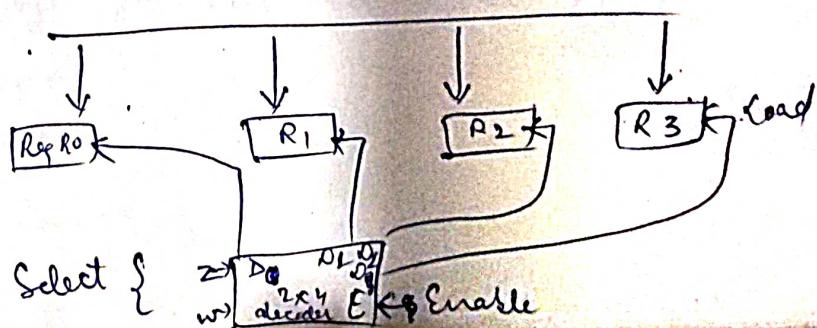
Registers  
Line  
Cost  
↓  
Realistic Approach  
for Large Systems.



From Reg to Bus :-



From Bus to Reg :-



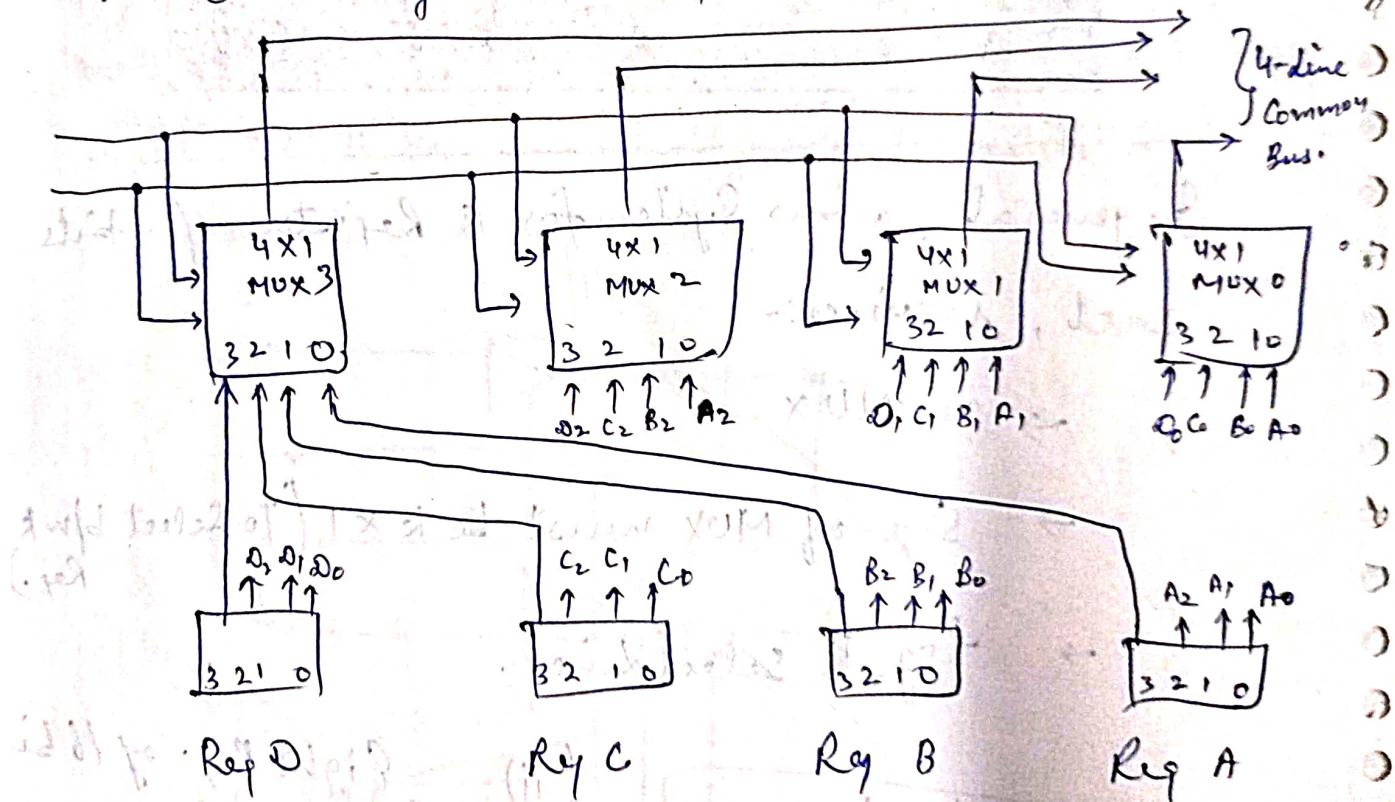
Bus :- It is a path (of a group of wires) over which info is transferred from any several sources to any of several destination.

✓

Control Signals determine which Reg is selected by bus during each transfer.

BUS → C , RI ← Bus.

The way of Constructing a Common bus System is with Multiplexers. Multiplexers select source register whose binary Info is then placed on the bus. The bus lines are connected to inputs of all Dest. Reg. To transfer the info from Bus to one of dest-reg, Load Control S/I/P of correxp Register is activated & contents of bus are placed in Reg. Dest. Reg



Construction of a bus Sys. for four 4-bit Registers:-

- four 4-bit Reg.
- four 4x1 MUX.
- 2 Selection lines.

In general in a bus system for  $k$  Registers of  $n$  bit each, requires:-

$\rightarrow$   $n \text{ MUX}$

$\rightarrow$  Size of MUX must be  $k \times 1$  (To select  $n$  bits)  
 $\log_2 k$  selection.

(1)

Eight Reg. of 16 bits

$\rightarrow$  16 MUX

$8 \times 16$

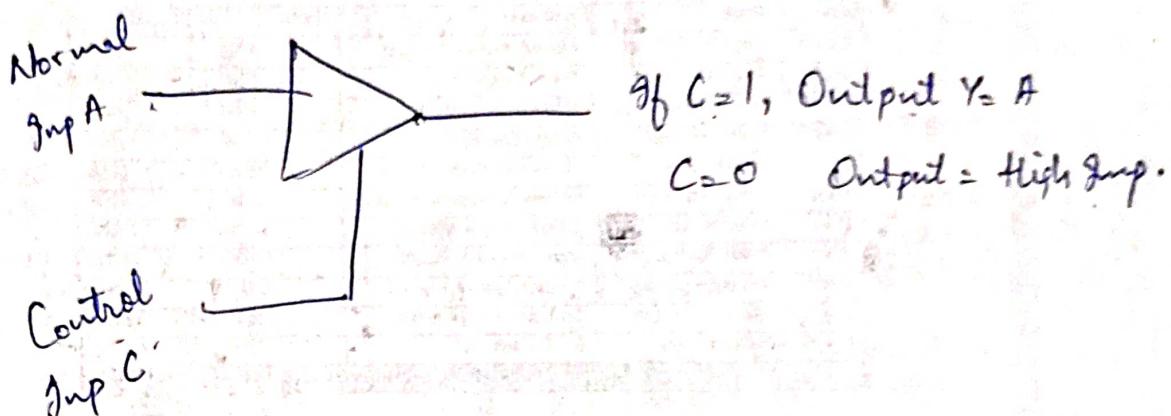
With 3 Selection bits ( $2^3$ )

$S_1$	$S_0$	Reg.
0	0	A
0	1	B
1	0	C
1	1	D

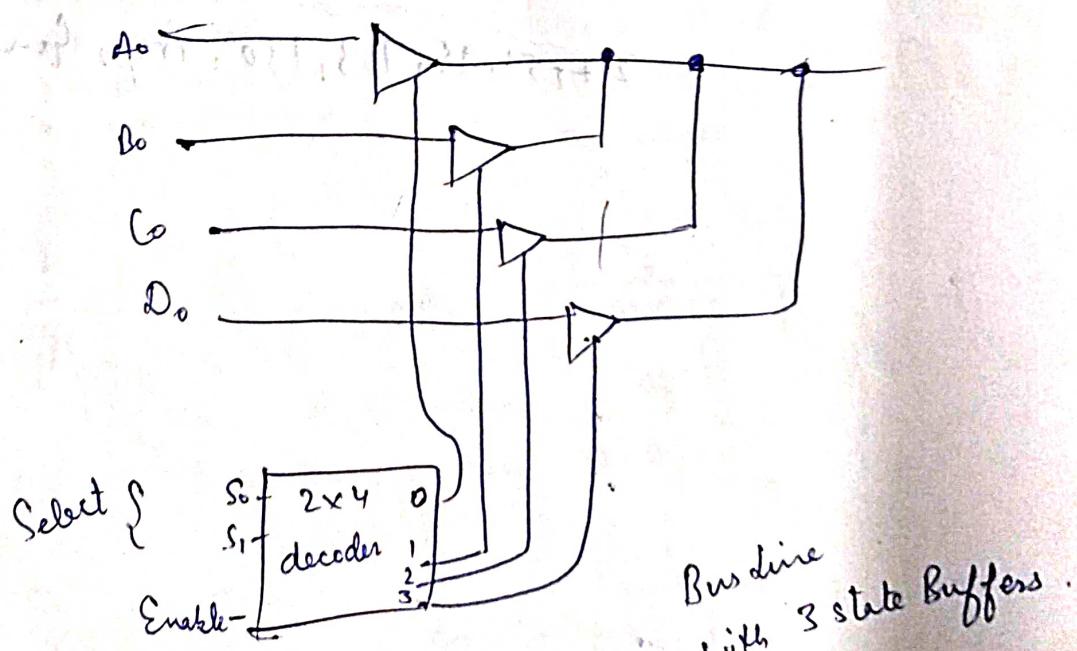
## Three State Bus Buffer:-

It is a digital circuit that exhibits three states. Two of the states are signals eq. to logic 1 & 0.

Third state is a high impedance state. High Imp. state behave like an open circuit, which means that output is disconnected & does not have logic signif.



High Imp. state of 3 state gate provides a special feature not available in other gates. It enables a large no of 3 state gate o/p to be connected with 1 wire to form a common bus line.



To construct a common bus for  $k$  registers of  $n$  bits each using three-state buffers, we need :

- $n$  circuits with  $k$  buffers in each circuit
- 1 decoder with  $\log_2 k$  select lines to select between the  $k$  registers

*Example*

## ◆ 6.7 MEMORY TRANSFER

(Memory is a collection of storage cells together with the associated circuits needed to transfer information in and out of storage. The memory stores binary information in groups of bits called words.)

A word is a group of bits that can move in and out of the storage as a unit. A group of eight bits is called a byte. Most computer memories use words whose number of bits is a multiple of 8. Thus a 16-bit word contains two bytes and a 32-bit word is made up of four bytes.

Each word (storage cell) in the memory is referred by its address. If a memory contains  $M$  words then their address varies from 0 to  $M-1$ . The number of words in a memory is usually a power of 2 i.e.  $M = 2^k$  so that the address varies from 0 to  $2^k - 1$ .

The two main operations of memory are Memory Read & Memory Write.

✓ **Memory Read** : The transfer of information from memory to the outside environment is called a Memory read operation. One memory word is read at a time.

✓ **Memory write** : The transfer of new information to be stored into the memory is called a memory write operation. One memory word is written at a time.

A memory unit of  $M (= 2^k)$  words of  $n$  bits each, needs the following :

- $n$  input data lines
- $n$  output data lines
- $k$  address lines
- A Read control signal
- A write control signal

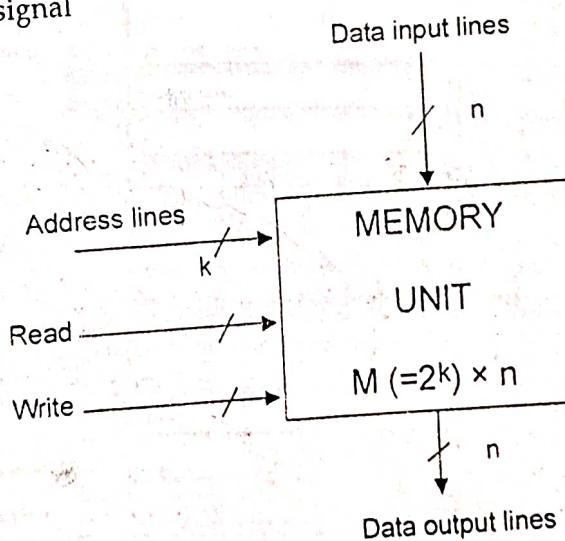


Fig. 6.8. Memory Unit

Memory is usually accessed by putting the desired address in a special register called the Memory Address register (MAR or just AR) for both memory read and memory write operations. When memory is accessed, the contents of the AR are put on the memory unit's address lines.

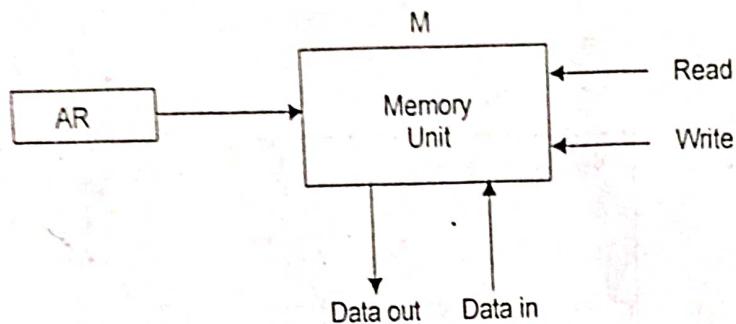


Fig. 6.9. Memory with address register (AR)

To represent memory transfers :

- Designate a memory word by the letter M
- Designate the address register by AR and the data register by DR
- The read operation in the register transfer language notation look like:

Read:  $DR \leftarrow M[AR]$

This causes the following to occur :

- + The contents of AR are sent to the memory address lines.
- + The Read control input is set to 1
- + The contents of specified address are put on the memory's output data lines.
- + The contents are finally sent over the common bus to be loaded in to the DR (the Load control input of DR is set to 1).
- The write operation in register transfer language is stated as:

Write:  $M[AR] \leftarrow R1$

This causes the following to occur

- + The contents of AR (address) are sent to the memory unit's address lines.
- + The Write control input is set to 1.
- + The contents of Register R1 (by setting the appropriate control signals of the common bus) are sent over the common bus to the data input lines of memory.
- + The contents finally get loaded into the specified address in the memory.

## EXERCISE

- Q1. Show the block diagram of the hardware that implements the following register transfer statement:

$yT_2 : R2 \leftarrow R1, R1 \leftarrow R2$

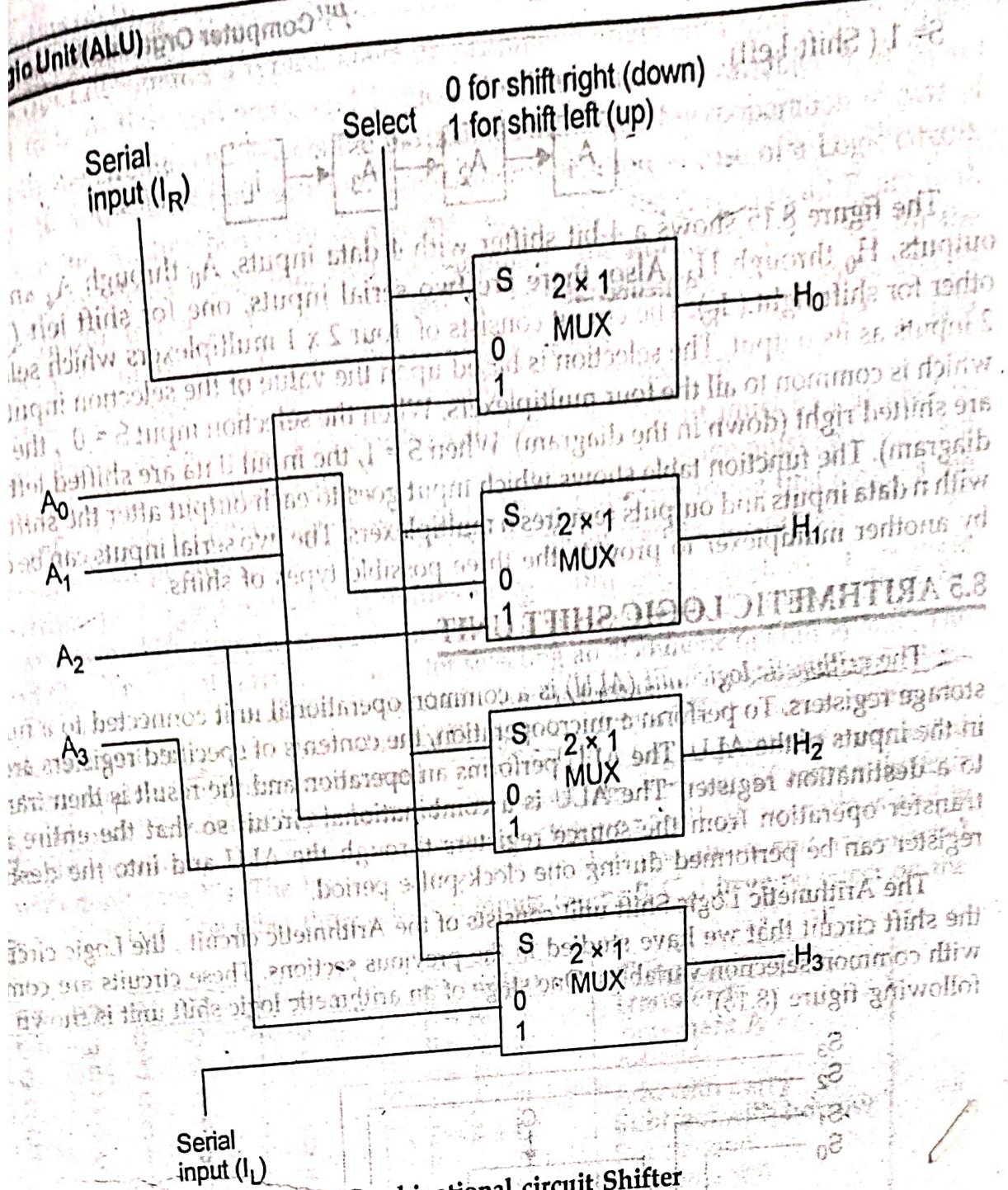
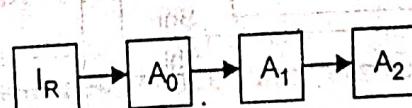


Fig. 8.15 : Combinational circuit Shifter

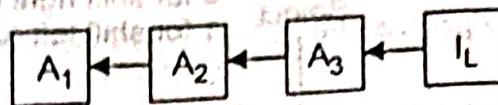
Select	Output			
	$S$	$H_0$	$H_1$	$H_2$
0	$I_R$	$A_0$	$A_1$	$A_2$
1	$A_1$	$A_2$	$A_3$	$I_L$

Table 8.8 : Function Table for Combinational circuit shifter



$S = 0$  (Shift Right)

$S = 1$  (Shift Left)



The figure 8.15 shows a 4-bit shifter with 4 data inputs,  $A_0$  through  $A_3$  and outputs,  $H_0$  through  $H_3$ . Also, there are two serial inputs, one for shift left ( $I_L$ ) other for shift right ( $I_R$ ). The circuit consists of four  $2 \times 1$  multiplexers which select 2 inputs as its output. The selection is based upon the value of the selection input which is common to all the four multiplexers. When the selection input  $S = 0$ , the data are shifted right (down in the diagram). When  $S = 1$ , the input data are shifted left (up in the diagram). The function table shows which input goes to each output after the shift. A 4-bit shifter with  $n$  data inputs and outputs requires  $n$  multiplexers. The two serial inputs can be controlled by another multiplexer to provide the three possible types of shifts.

## 8.5 ARITHMETIC LOGIC SHIFT UNIT

The arithmetic logic unit (ALU) is a common operational unit connected to a storage registers. To perform a microoperation, the contents of specified registers are loaded into the inputs of the ALU. The ALU performs an operation and the result is then transferred to a destination register. The ALU is a combinational circuit so that the entire transfer operation from the source registers through the ALU and into the destination register can be performed during one clock pulse period.

The Arithmetic Logic Shift unit consists of the Arithmetic circuit, the Logic circuit, and the shift circuit that we have studied in the previous sections. These circuits are interconnected with common selection variables. One stage of an arithmetic logic shift unit is shown in the following figure (8.16).

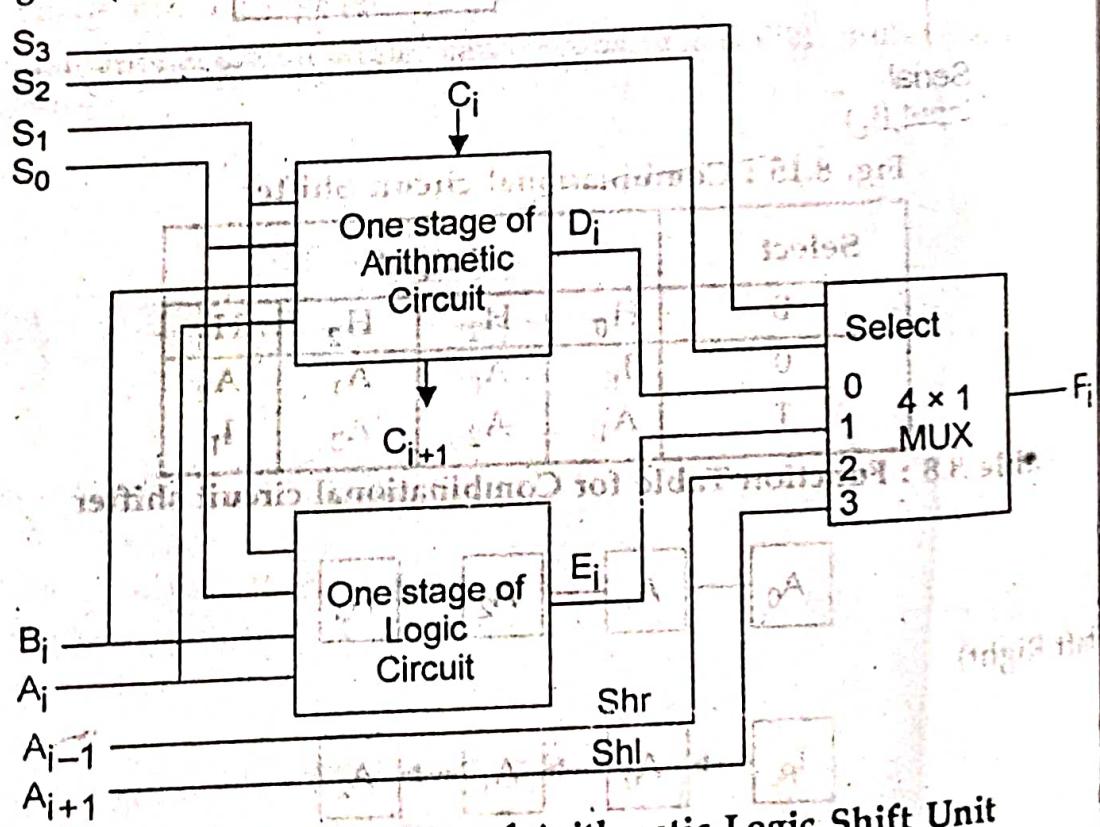


Fig. 8.16. One stage of Arithmetic Logic Shift Unit

ALU  
r, shl

Combined Arithmetic, Logic & Shift circuit.

" " → Typical stage

$A_i B_i$  are applied to both units.

Particular op<sup>n</sup> is selected with  $S_0 \& S_1$

4x1 Mux select  $4w A_i \& 4w E_i$

ja in MUX are selected with  $S_3 \& S_2$

| p  $A_{i-1}$  for shr

$A_{i+1}$  for shl

14 op<sup>n</sup> are performed i.e. 8 Arithmetic, 4 logic, 2 shift

Op<sup>n</sup> is selected with  $S_3 S_2 S_1 S_0$  & Cin.

Cin is used for Arithmetic Op<sup>n</sup> only

8 Arithmetic op<sup>n</sup>  $S_3 S_2 = 00$

$Cin S_0 S_1 > \frac{2^3}{2} = 8$   
Combinations

4 logic op<sup>n</sup>  $S_3 S_2 = 01$

(Cin is under don't care during logic & shr op<sup>n</sup>.

2 shift op<sup>n</sup>  $S_3 S_2 = 10$  shr

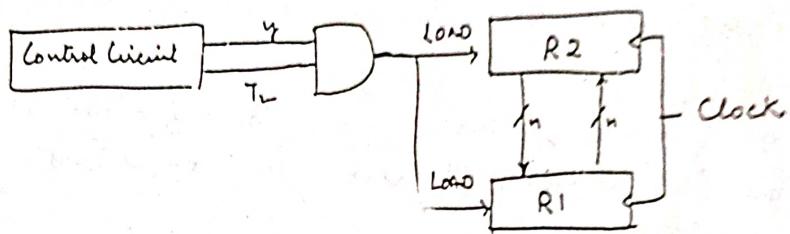
$S_3 S_2 = 11$  shl.

Examples

Title Sheet No: 3

Q1:- Show the block diagram of hardware that implements the following Register Transfer statement:

Sol:-  $y T_2 : R_2 \leftarrow R_1 \quad R_1 \leftarrow R_2.$



Sorry?

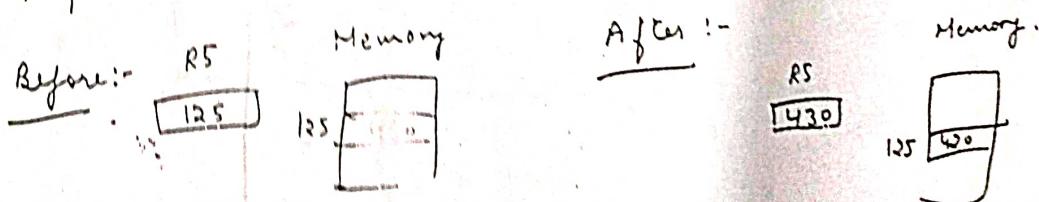
Q2:- The following transfer statements specify a memory. Explain the memory op<sup>n</sup> in each case.

(a)  $R_2 \leftarrow M[AR]$  (b)  $M[AR] \leftarrow R_3$  (c)  $R_5 \leftarrow M[R_5]$

(a)  $R_2 \leftarrow M[AR]$  :- The statement specifies that the contents of memory location whose address is stored in AR are transferred to R2. Hence, it is a Memory Read Operation & the contents are send in to R2.

(b)  $M[AR] \leftarrow R_3$  :- The statement specifies a Memory Write Op where the contents of Reg R3 are written in the memory loc specified by AR.

(c)  $R_5 \leftarrow M[R_5]$  :- It specifies a Memory Read Op where the contents of the memory location whose address is given by Register R5 are transferred to Register R5.



K reg. of n bits

A digital Computer has a common bus system for 16 registers of 32 bits each. The bus is constructed with multiplexers.

- (a) How many selection inputs are there in each multiplexer?
- (b) What size of multiplexers are needed?
- (c) How many multiplexers are there in the bus?

Ans:- The common bus has 16 registers, so the multiplexer has to select 1 among 16 registers.

$$MUX = 16 \times 1$$

a) Selection Control I/p for  $16 \times 1$  MUX are 4

$$\therefore 2^4 = 16$$

b) Size of Multiplexers =  $16 \times 1$

c) No. of Multiplexers needed = 32 ( $\because n=32$ )

Q. Draw the diagram of a bus system using three state buffers & a decoder.

Inputs		Output Selected
$S_1$	$S_0$	
0	0	A
0	1	B
1	0	C
1	1	D

4 Registers of 4 bit each

