Register Transfer and Microoperations

Digital system:-

- A digital system is an interconnection of digital hardware modules.
- The modules are constructed from digital components such as registers, decoders, arithmetic elements, and control logic.
- The various modules are interconnected with common data and control paths to form a digital computer system.

Microoperation:

- Register contains data.
- The operations executed on data stored in registers are called microoperations.
- The result of the operation may replace the previous binary information of a register or may be transferred to another register.
- Example of microoperations:
  - shift
  - count
  - clear
  - load

Register transfer Language:-

The symbolic notation used to describe the microoperation transfe

Register Transfer

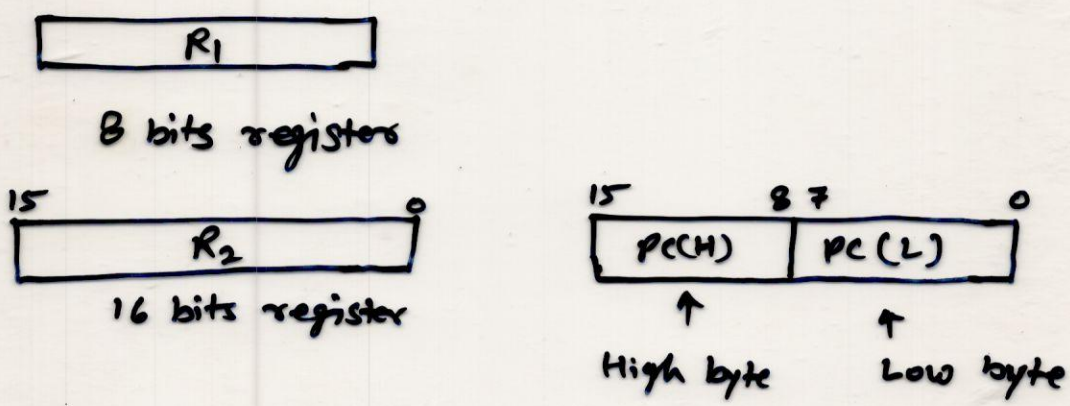
Different types of register:-

- MAR : Memory Address Register
- PC : Program Counter
- IR : Instruction Register
- R_1, R_2 : Processor register.

Information transfer from one register to another is designated in symbolic form by means of a replacement operator.

$$R_2 \leftarrow R_1$$

denotes a transfer of the content of Register R_1 into register R_2.



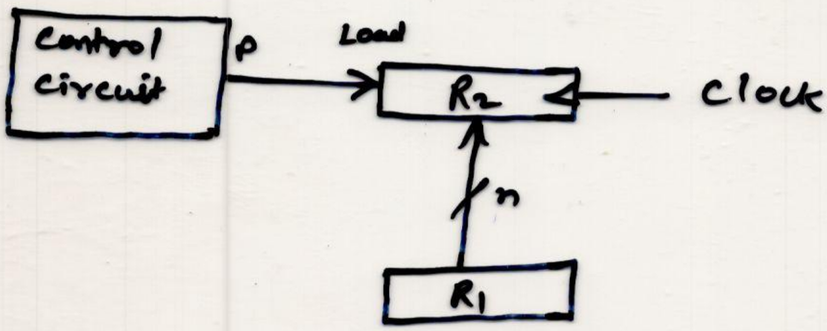
(Fig. Block diagram of register)

Control function

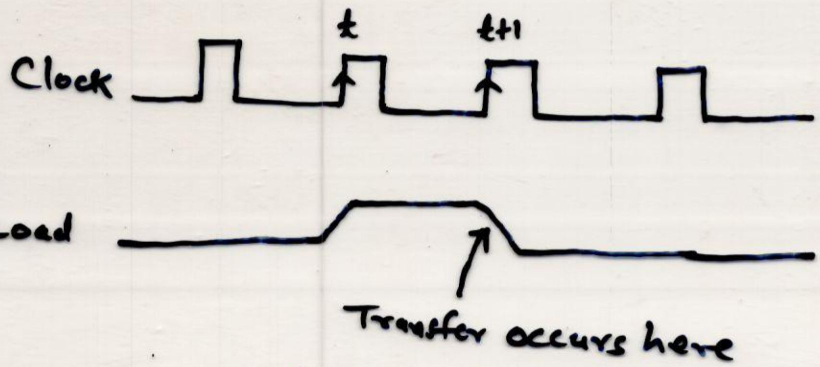
A control function is a Boolean variable that is equal to 0 or 1. The control function is included in the statement as follows:

$$P : R_2 \leftarrow R_1$$

The transfer operation be executed by the hardware only if $P = 1$.



((a) Block diagram)



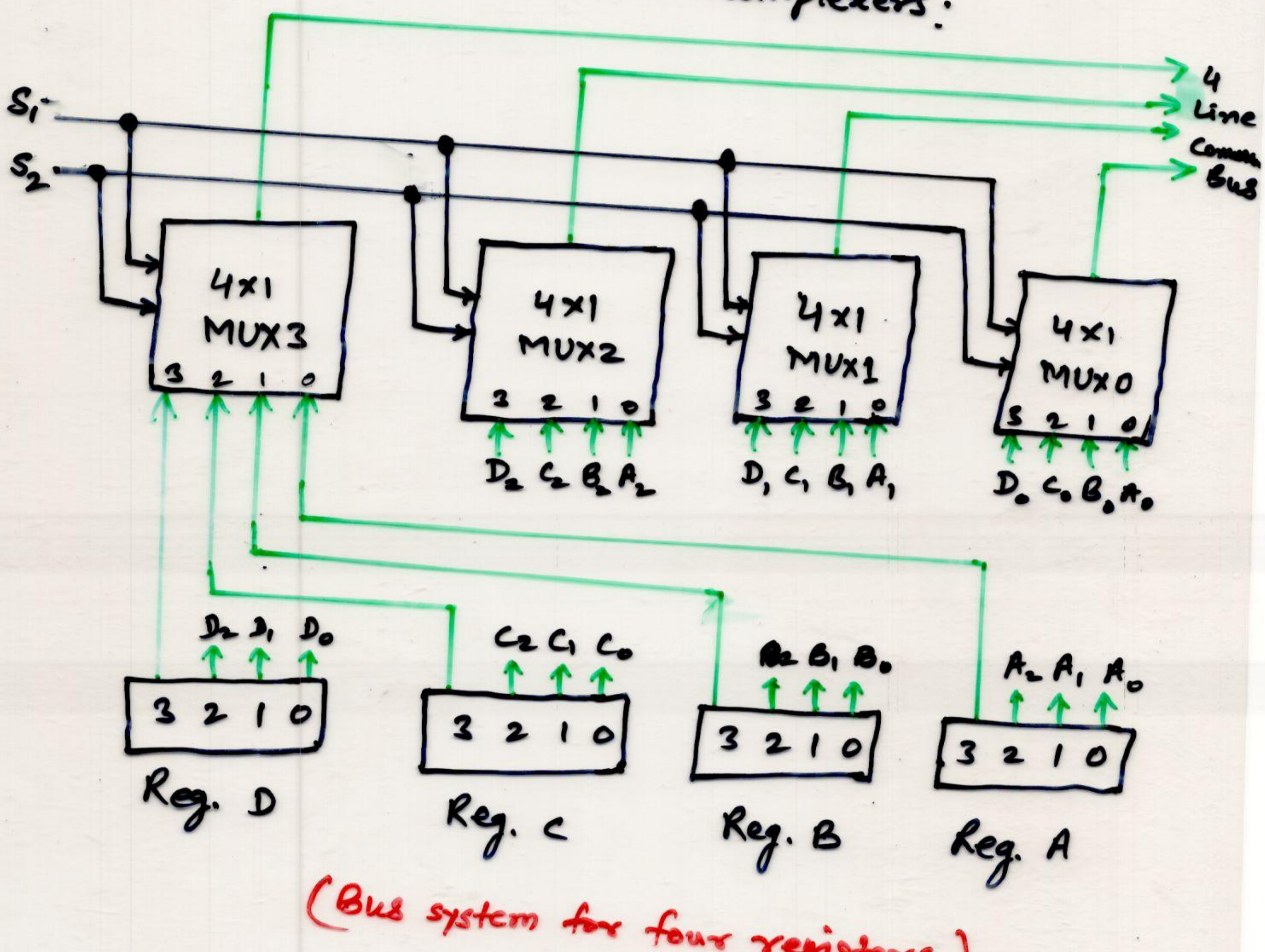
((b) Timing diagram)

- The n (no. of bits) outputs of register R_1 are connected to the n inputs of register R_2.
- Register R_2 has a load input that activated by the control variable P.

Bus :

A bus consists of a set of common lines, one for each bit of a register, through which binary information is transferred one at a time.

Construction of bus system with multiplexers:



(Bus system for four registers.)

· A bus system will multiplex K registers of n bits each to produce an n-line common bus.

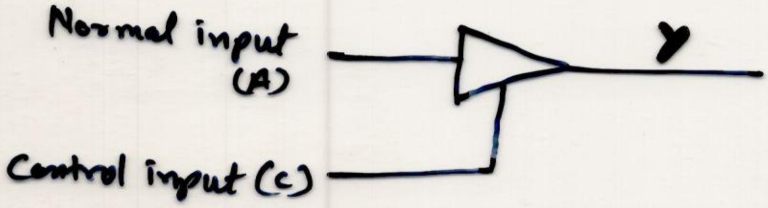
Number of multiplexers = n

The size of multiplex ~~and~~ must be K×1

Three-state Bus Buffers

A bus system can be constructed with three-state gates instead of multiplexers. A three-state gate is a digital circuit that exhibits three states. Two of the states are signals (1 or 0) and third state is a high-impedance state.

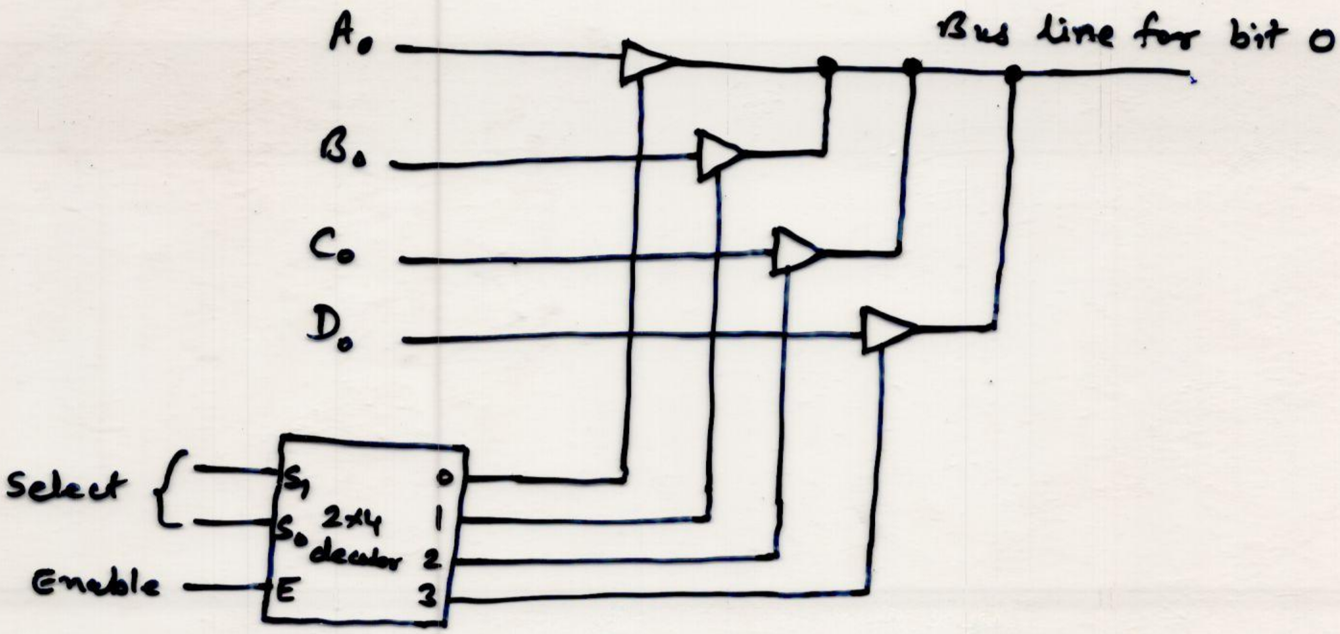
The high-impedance state behaves like a open circuit.



Output $Y = A$ if $C=1$

High-impedance if $C=0$

(Graphic symbol for three-state buffer)



(Bus line with three-state buffer)

Memory Transfer :

Operations:

1. Read :- The transfer of information from a memory word to the outside environment.
2. Write :- The transfer of new information to be stored into the memory.

Memory read:

$$\text{Read: } DR \leftarrow M[AR]$$

where: $DR \rightarrow$ Data Register
$M \rightarrow$ Memory word
$AR \rightarrow$ Address Register

Memory write:

$$\text{Write: } M[AR] \leftarrow R_1$$

Types of Microoperations

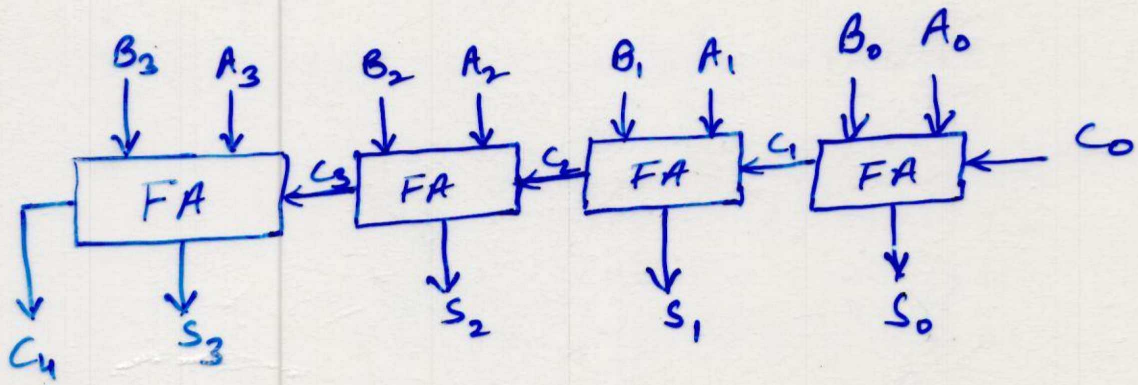
1. Register transfer microoperations
2. Arithmetic microoperations
3. Logic microoperations
4. Shift microoperation

Arithmetic Microoperation

The basic arithmetic microoperations are addition, subtraction, increment & decrement.

Symbolic designation	Description
$R3 \leftarrow R_1 + R_2$	Contents of R_1 plus R_2 transferred to R_3
$R3 \leftarrow R_1 - R_2$	Contents of R_1 minus R_2 transferred to R_3
$R_2 \leftarrow \overline{R_2}$	Complement the contents of R_2 (1's com)
$R_2 \leftarrow \overline{R_2} + 1$	2's complement contents of R_2 (subtract)
$R_1 \leftarrow R_1 + 1$	increment
$R_1 \leftarrow R_1 - 1$	Decrement

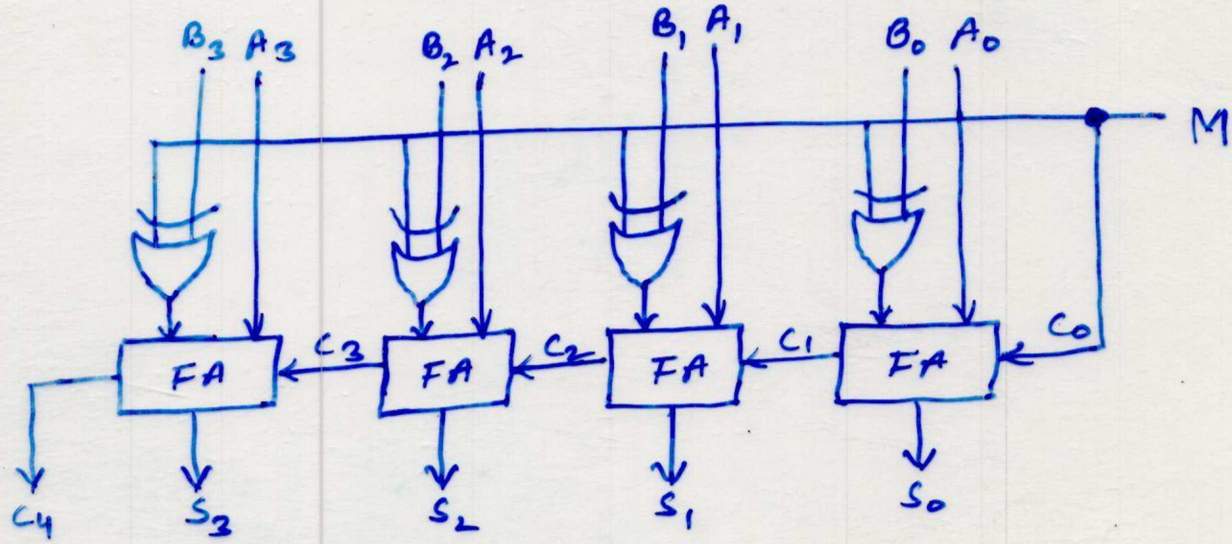
4-bit binary adder :-



where C_0 - input Carry
C_4 - output Carry
S - output

An n-bit binary adder requires n full adders.

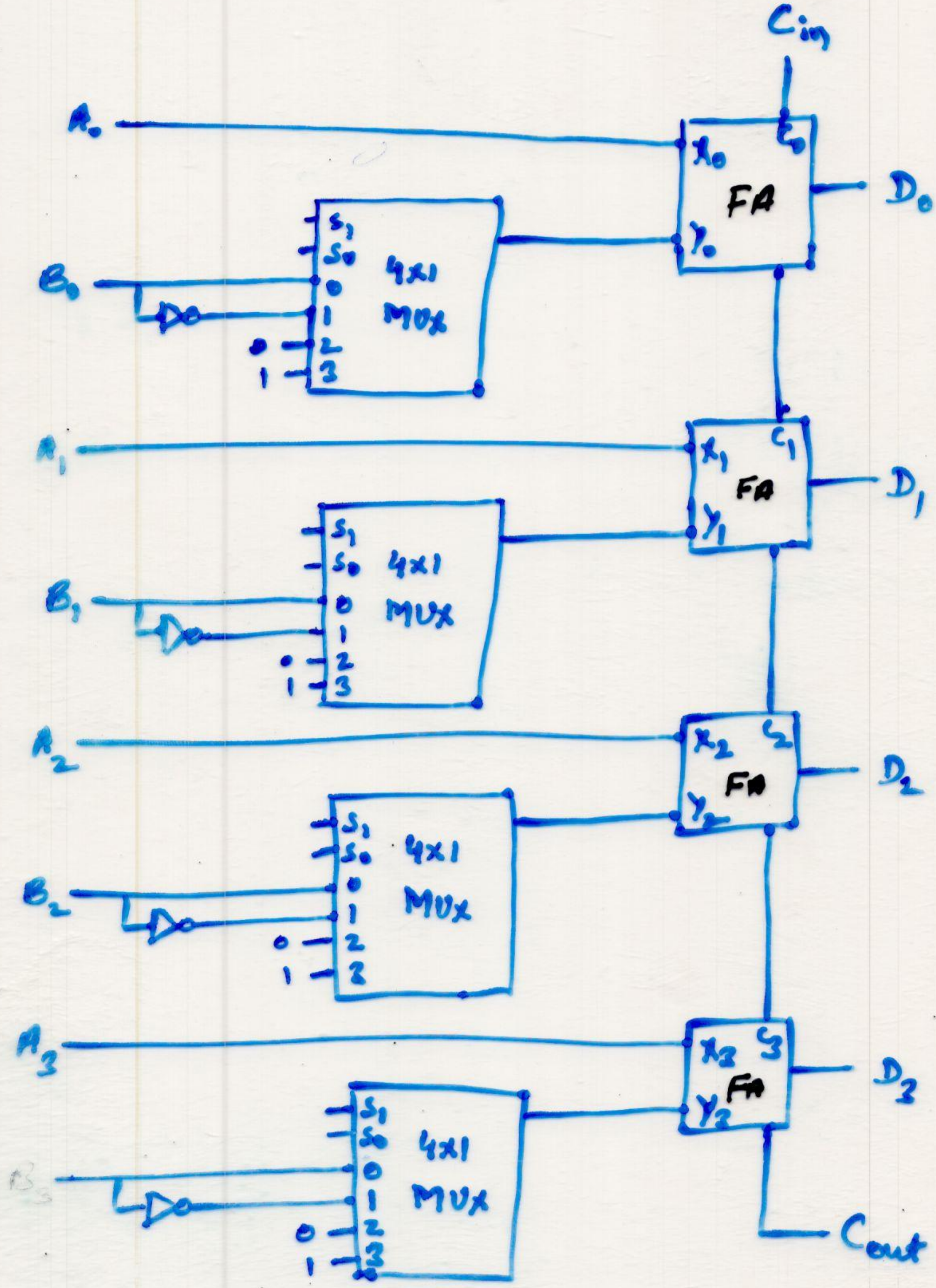
4-bit adder-subtractor :-



When $M=0$ the circuit is adder
$M=1$ the circuit is subtractor

$B \oplus 0 = B$
$B \oplus 1 = B'$

Arithmetic Circuit



4-bit arithmetic circuit

Arithmetic Circuit Function Table

Select			Input	Output	Microoperation
S_1	S_0	C_{in}	Y	$D = A + Y + C_{in}$	
0	0	0	B	$D = A + B$	Add
0	0	1	B	$D = A + B + 1$	Add with carry
0	1	0	$\bar{B}$	$D = A + \bar{B}$	Subtract with borrow
0	1	1	$\bar{B}$	$D = A + \bar{B} + 1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A + 1$	Increment A
1	1	0	-1	$D = A - 1$	Decrement A
1	1	1	-1	$D = A$	Transfer A

Note: when $S_1 S_0 = 11$, all 1's are inserted ~~to~~ into the Y input of the adder to produce the decrement operation $D = A - 1$, because a number with all 1's is equal to 2's complent of 1 (2's complement of 0001 is 1111).

$F = A + 2\text{'s complement} = A - 1.$

Logic Microoperation

Logic microoperation specify binary operations for strings of bits stored in registers.

Truth tables for 16 Functions of Two Variables

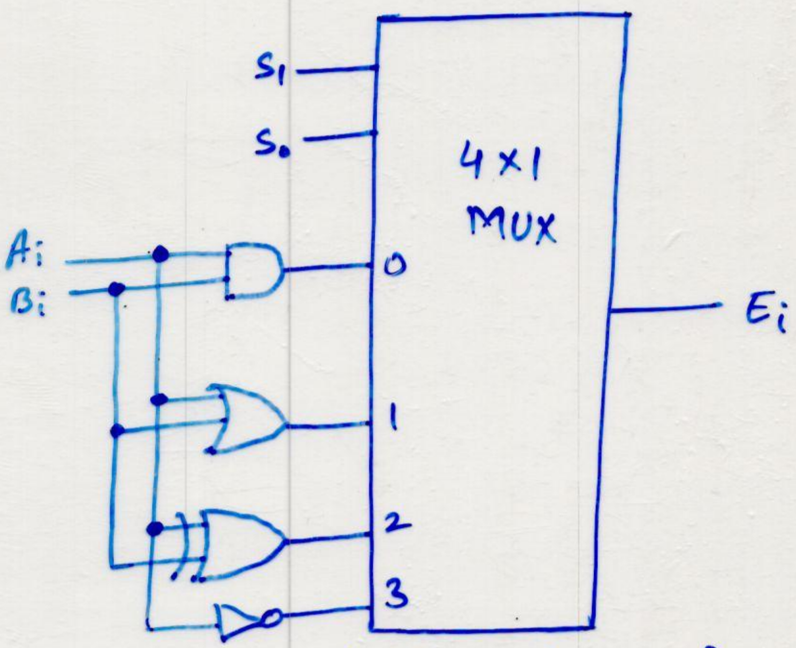
x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Sixteen Logic microoperations

Boolean Function	Microoperation	Name
$F_0 = 0$	$F \leftarrow 0$	Clear
$F_1 = xy$	$F \leftarrow A \wedge B$	AND
$F_2 = xy'$	$F \leftarrow A \wedge \overline{B}$	
$F_3 = x$	$F \leftarrow A$	Transfer A
$F_4 = x'y$	$F \leftarrow \overline{A} \wedge B$	
$F_5 = y$	$F \leftarrow B$	Transfer B
$F_6 = x \oplus y$	$F \leftarrow A \oplus B$	Exclusive-OR
$F_7 = x + y$	$F \leftarrow A \vee B$	OR
$F_8 = (x+y)'$	$F \leftarrow \overline{A \vee B}$	NOR
$F_9 = (x \oplus y)'$	$F \leftarrow \overline{A \oplus B}$	Ex-NOR
$F_{10} = y'$	$F \leftarrow \overline{B}$	Complement B
$F_{11} = x + y'$	$F \leftarrow A \vee \overline{B}$	

Hardware Implementation of logic microoperations

There are 16 logic microoperations, most computers use only for - AND, OR, XOR and complement - from which all others can be derived.



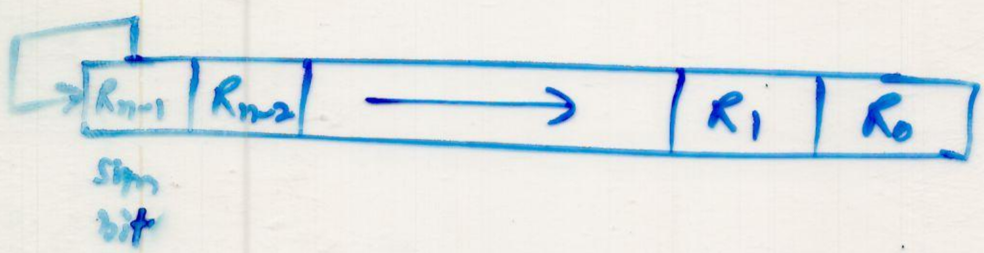
(block diagram)

S_1	S_0	output	operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \overline{A}$	Complement

(Function table)

Shift Microoperations

1. **Logical shift** :- A logical shift is one that transfers 0 through the serial input.
2. **Circular shift** :- The circular shift circulates the bits of the register around the two ends without loss of information.
3. **Arithmetic shift** :- An arithmetic shift is a micro-operation that shifts a signed binary number to the left or right.



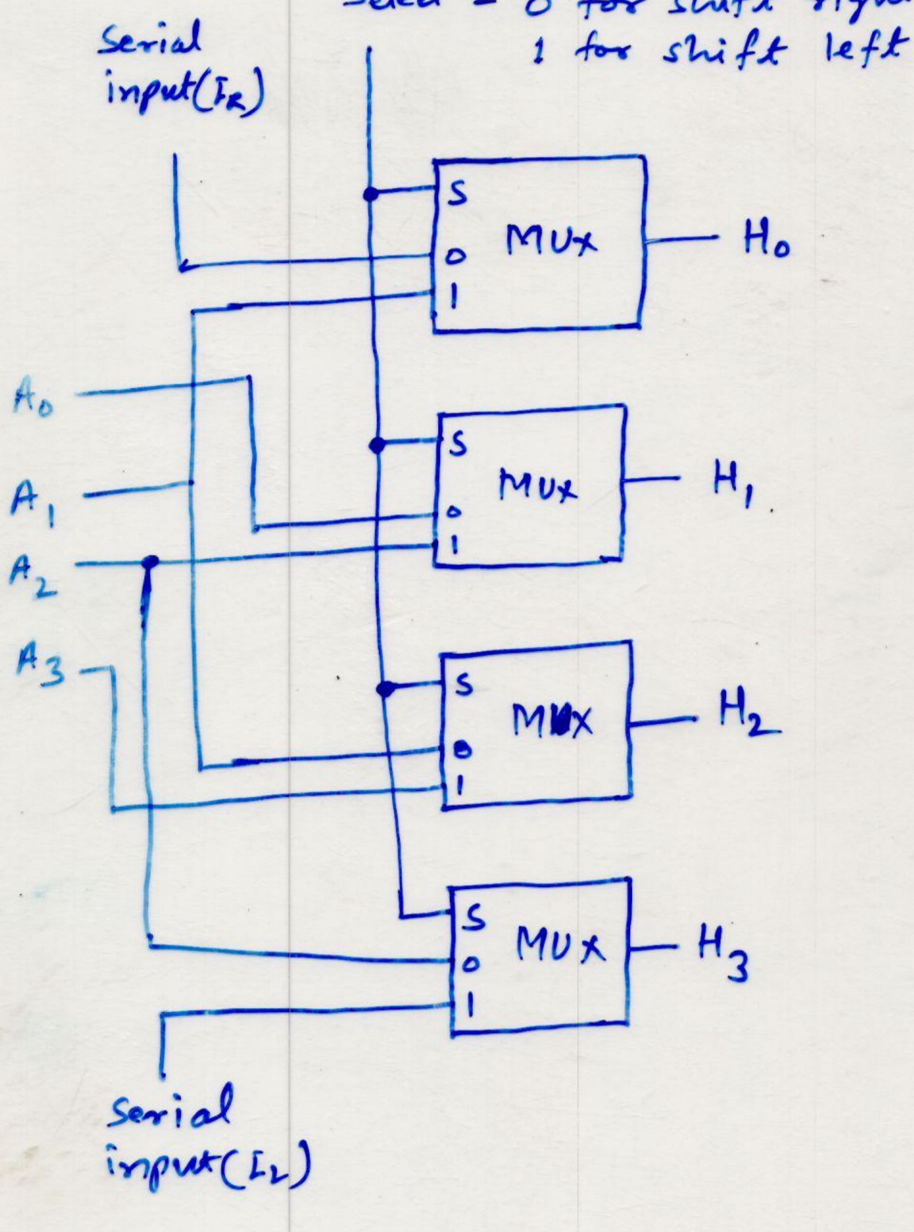
R_{n-1} — Sign bit

R_{n-2} to R_0 — hold the number

Shift Microoperation

Symbolic designation	Description
$R \leftarrow shl\ R$	shift-left register R
$R \leftarrow shr\ R$	shift-right register R
$R \leftarrow cil\ R$	circular shift-left Register R
$R \leftarrow cir\ R$	circular shift-right Reg R
$R \leftarrow ashl\ R$	Arithmetic shift ...

4-bit combinational circuit shifter



Function table

select S	output H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L