LOGICAL AND SHIFT MICROOPERATION

WHAT IS LOGIC MICROOPERATION

- Logic microoperation specify binary operation for strings of bit stored in registers.
- These operation consider each bit of the register separately and treat them as binary variables.For example,

$P: R1 \leftarrow R1 \oplus R2$

1010	Content of R1
1100	Content of R2
0110	Content of R1 after P=1

LIST OF LOGIC MICROOPERATION

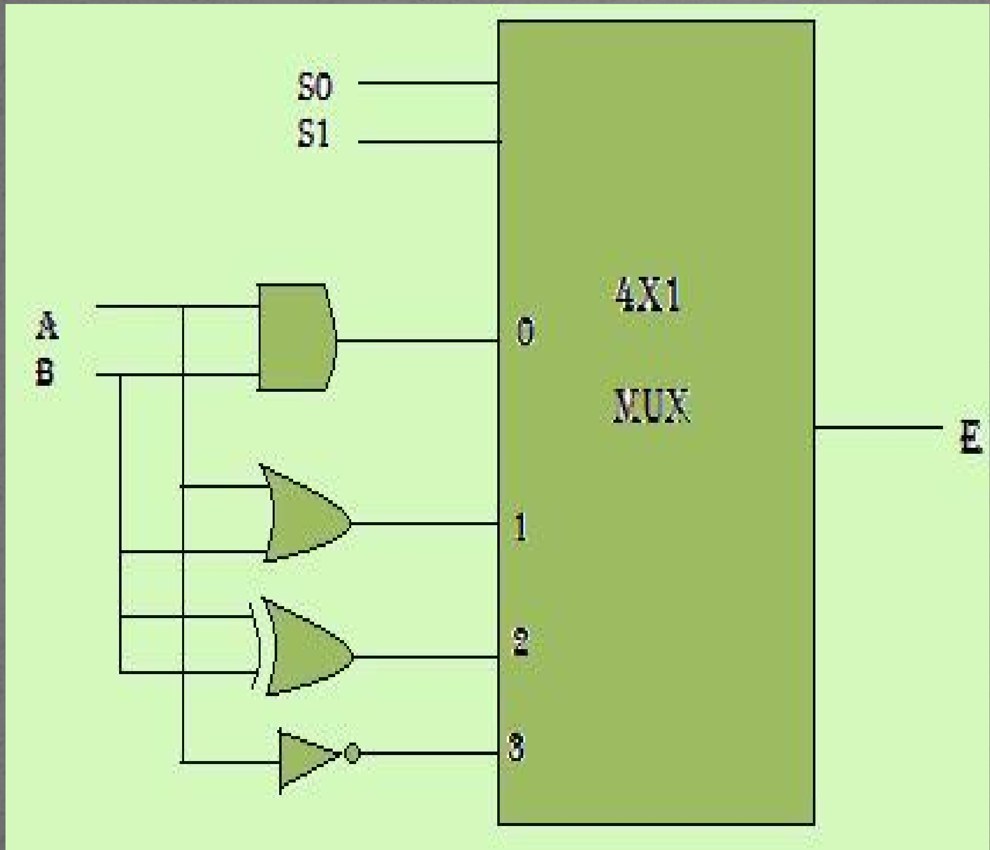
TRUTH TABLE FOR 16 FUNCTION OF TWO VARIABLES

X	Y	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
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 MICROOPERATION

Boolean Function	Microoperation	Name
F0=0	$F \leftarrow 0$	Clear
F1=xy	$F \leftarrow A \wedge B$	And
F2=xy'	$F \leftarrow A \wedge \overline{B}$	
F3=x	$F \leftarrow A$	Transfer A
F 4=x'y	$F \leftarrow \overline{A} \wedge B$	
F5=y	$F \leftarrow B$	Transfer B
F6=$x \oplus y$	$F \leftarrow A \oplus B$	Exclusive-OR
F7=(x+y)	$F \leftarrow A \vee B$	OR
F8=(x+y)'	$F \leftarrow \overline{A \vee B}$	NOR
F9=$(x \oplus y)'$	$F \leftarrow \overline{A \oplus B}$	Exclusive-NOR
F10=y'	$F \leftarrow \overline{B}$	Complement B
F11=x+y'	$F \leftarrow A \vee \overline{B}$	
F12=x'	$F \leftarrow \overline{A}$	Complement
F13=x'+y	$F \leftarrow \overline{A} \vee B$	
F14=(xy)'	$F \leftarrow \overline{A \wedge B}$	NAND
F15 =1	$F \leftarrow$ all 1's	Set to all 1's

HARDWARE IMPLEMENTATION



S1 S0	OUTPUT	OPERATION
0 0	E=A·VB	AND
0 1	E=AλB	OR
1 0	$E=A\oplus B$	XOR
1 1	$E=\overline{A}$	COMPLEMENT

SOME OTHER FUNCTION

- **SELECTIVE SET:**

  it sets the bit's to 1 in register A where there are 1,s in register B. 0's in B will not be affected. Logic OR operation is followed.example

  1010 A before

  1100 B(logic operant)

  1110 A after

- **SELECTIVE COMPLEMENT :**

  it complements bits in A where there are corresponding 1's in B.example

  1010 A before

  1100 B

  0110 A after

it can be seen selective complement can be done by Exclusive –OR

- **SELECTIVE CLEAR:**

it clear the bit to 0 in A where there are corresponding 1's in B.example

1010 A before

1100 B

0010 A after

(it can be obtained by microoperation AB')

- **MASKING:**

it is similar to selective clear except that the bit of A is cleared where there corresponding 0's.

1010 A before

1100 B

1000 A after

- **<u>INSERT :</u>**

it inserts a new value into a group of bits.

This is done by first *masking* and then ORing with the value.Example

0110 1010 A before

0000 1111 B

0000 1010 A after

then insert a new value

0000 1010 A before

1001 0000 B(insert)

1001 1010 A after

SHIFT MICROOPERATION

- Shift microoperation are used for serial transfer of data.
- The content of the register can be shifted to left or the right.
- At the same time of bits shifted to the left or right, the first flip flop receive its binary information from the serial input.
- There are three types of shift:

I. Logical shift
II. Circular shift
III. Arithmetic shift

LOGICAL SHIFT

- A *logical shift* is one that transfer 0 through the serial input. The bit transferred to the end position through the serial input is assumed to be zero.
- Example:

  R1 ← shl R1 (1 bit shift to the left)

  R2 ← shr R2(1 bit shift to the right)

CIRCULAR SHIFT

- The circular shift(also known as rotate operation) circulates the bits of the register around the ends without the loss of information.
- This is accomplished by the connecting the serial output of the register to the serial input.
- Example:

  R1← cil R1(shifts left)

  R2← cir R2(shifts right)

ARITHMETIC SHIFT

- An arithmetic shift is a microoperation that shifts signed binary number to the left or right.
- An arithmetic shift left multiplies a signed binary no. by 2 and shift right divides by 2.
- The signed bit remains unchanged whether it is divided or multiplied by 2.

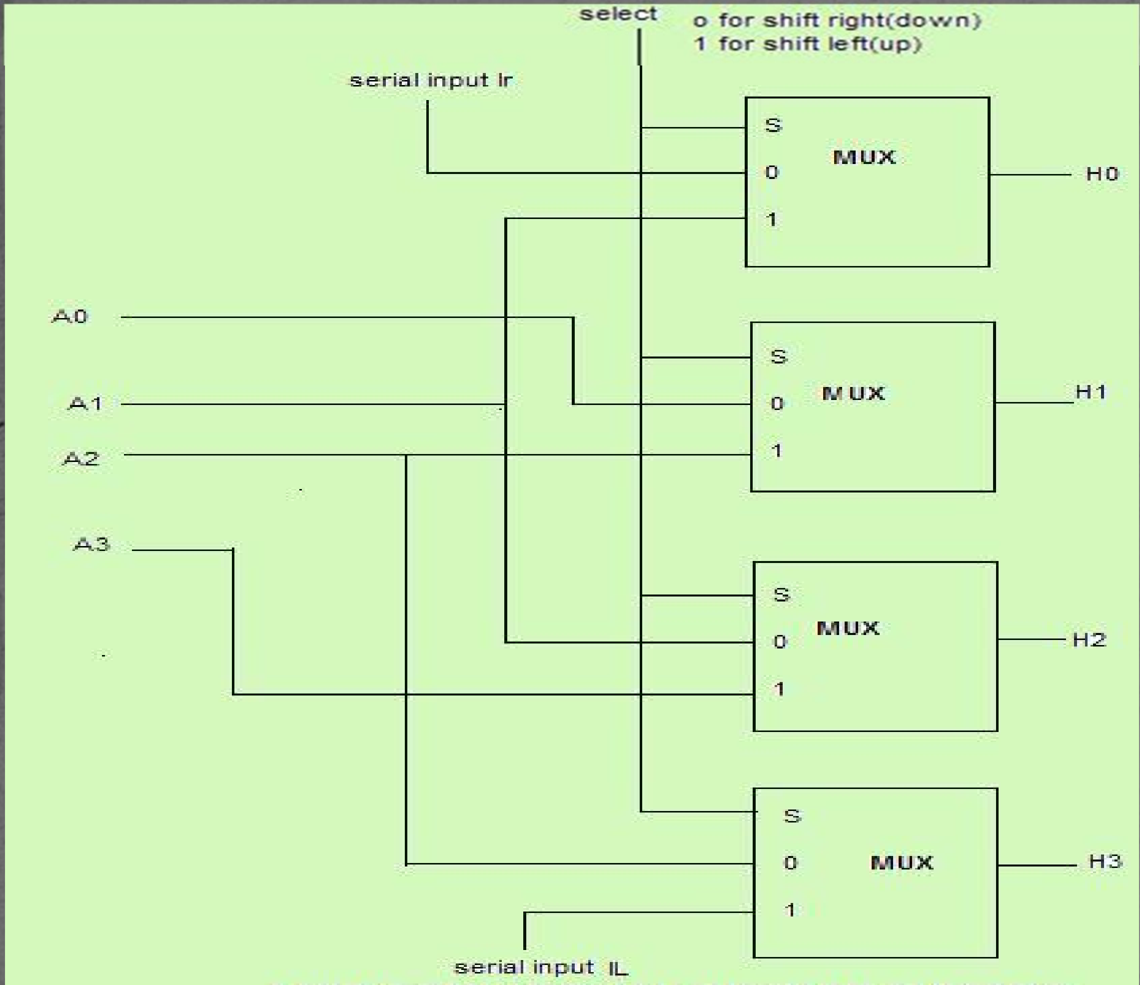
ARITHMETIC SHIFT

- The arithmetic shift right leaves the sign bit unchanged and shift the no.(including the sign bit) to the right the bit Rn-1 remain unchanged and R0 is lost.
- The arithmetic shift left insert a 0 into R0 and shifts all the other bits to the left. The initial bit of Rn-1 is lost and replaced by the bit from Rn-2.A sign reversal occurs if the bit in Rn-1 changes in the value after shift.
- An over-flow flip-flop Vs can be used to detect an arithmetic shift left overflow.

$$V_s = R_{n-1} \oplus R_{n-2}$$

- If Vs=0,there is no over flow, if Vs=1 there is overflow and a sign reversal takes place.

HARDWARE IMPLEMENTATION



4 BIT COMBITIONAL CIRCUIT SHIFTER

FUNCTIONAL TABLE

SELECT	OUTPUT			
S	H0	H1	H2	H3
0	IR	A0	A1	A2
1	A1	A2	A3	IL