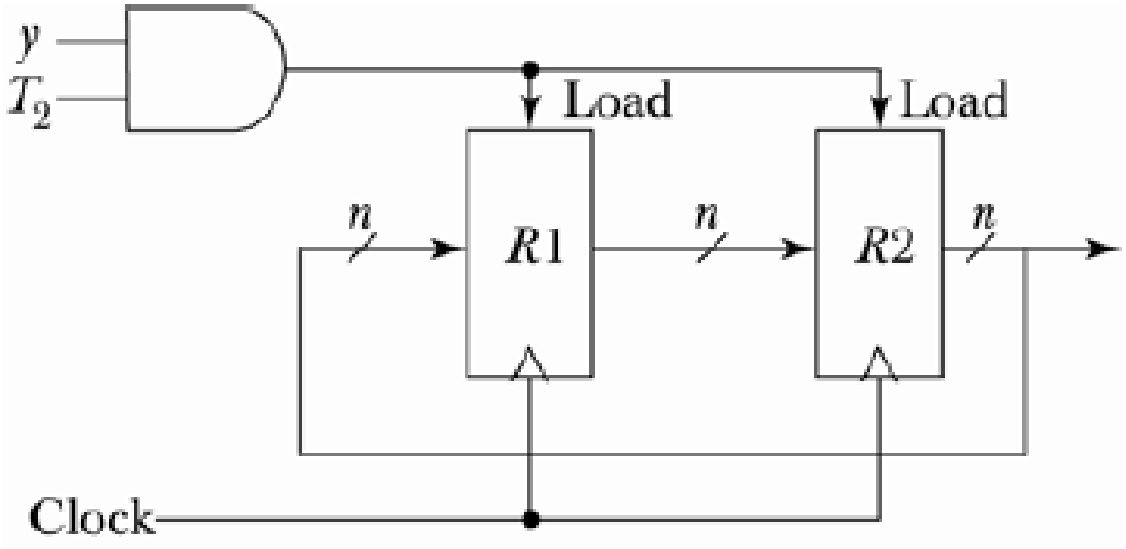
4-1. Show the block diagram of the hardware (similar to Fig. 4-2a) that implements the following register transfer statement:

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

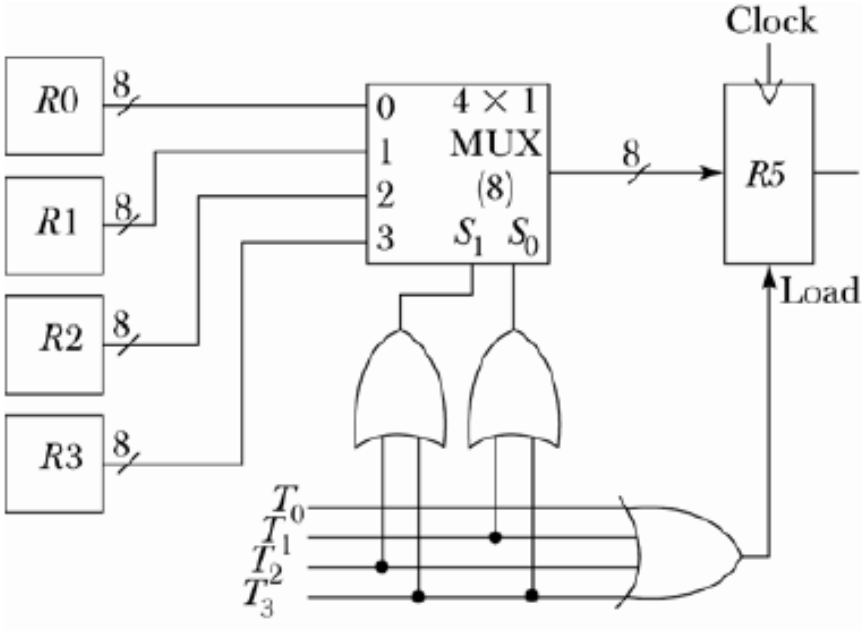
Part 2
Slide 12



4-2. The outputs of four registers, $R0$, $R1$, $R2$, and $R3$, are connected through 4-to-1-line multiplexers to the inputs of a fifth register, $R5$. Each register is eight bits long. The required transfers are dictated by four timing variables T_0 through T_3 as follows:

$$T_0: \quad R5 \leftarrow R0$$
$$T_1: \quad R5 \leftarrow R1$$
$$T_2: \quad R5 \leftarrow R2$$
$$T_3: \quad R5 \leftarrow R3$$

The timing variables are mutually exclusive, which means that only one variable is equal to 1 at any given time, while the other three are equal to 0. Draw a block diagram showing the hardware implementation of the register transfers. Include the connections necessary from the four timing variables to the selection inputs of the multiplexers and to the load input of register $R5$.



$T_0T_1T_2T_3$	S_1	S_0	R_3	load
0 0 0 0	X	X	0	
1 0 0 0	0	0	1	
0 1 0 0	0	1	1	
0 0 1 0	1	0	1	
0 0 0 1	1	1	1	

$$S_1 = T_2 + T_3$$
$$S_0 = T_1 + T_3$$
$$\text{load} = T_0 + T_1 + T_2 + T_3$$

4-3. Represent the following conditional control statement by two register transfer statements with control functions.

If ($P = 1$) then ($R1 \leftarrow R2$) else if ($Q = 1$) then ($R1 \leftarrow R3$)

P: R1 ← R2

P'Q: R1 ← R3

4-6. 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?

(a) 4 selection lines to select one of 16 registers.

(b) 16 × 1 multiplexers.

(c) 32 multiplexers, one for each bit of the registers.

4-7. The following transfer statements specify a memory. Explain the memory operation in each case.

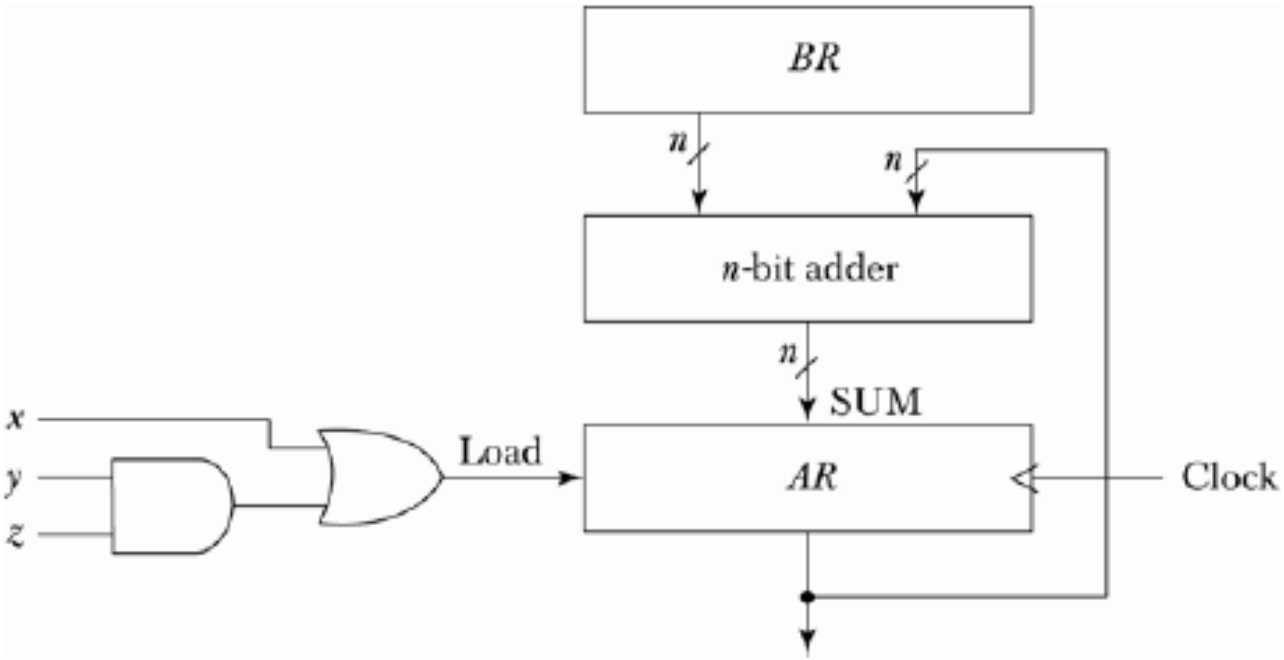
a. $R2 \leftarrow M[AR]$
b. $M[AR] \leftarrow R3$
c. $R5 \leftarrow M[R5]$

(a) Read memory word specified by the address in AR into register R2.
(b) Write content of register R3 into the memory word specified by the address in AR.
(c) Read memory word specified by the address in R5 and transfer content to R5 (destroys previous value)

4-8. Draw the block diagram for the hardware that implements the following statements:

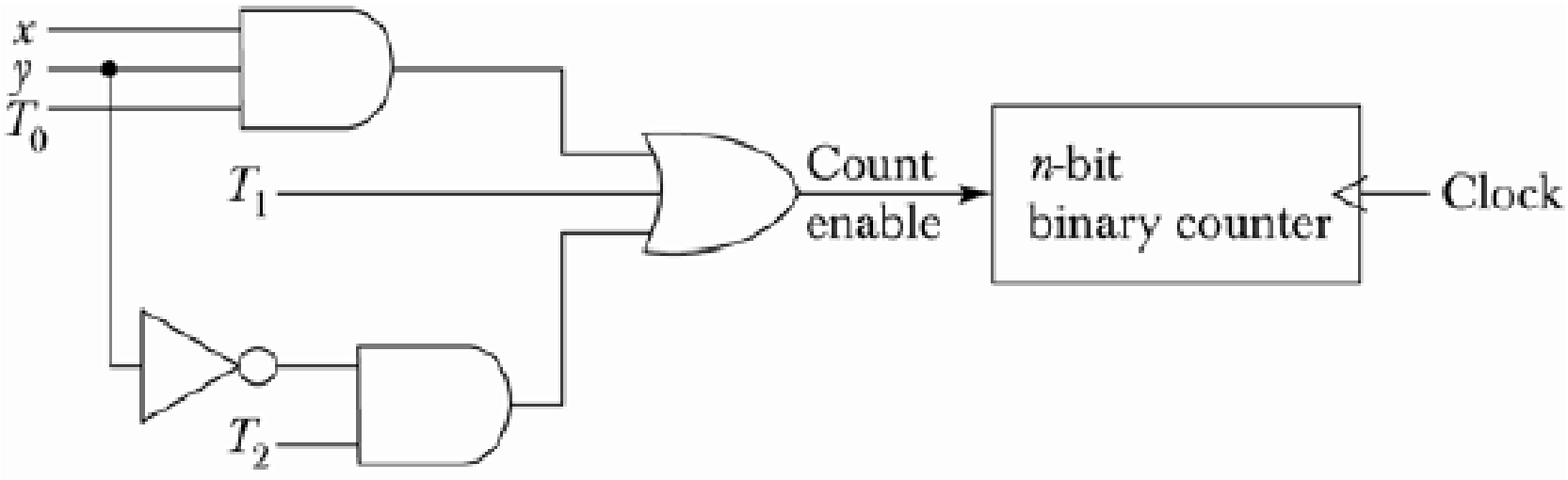
$$x + yz: \quad AR \leftarrow AR + BR$$

where AR and BR are two n-bit registers and x, y, and z are control variables. Include the logic gates for the control function. (Remember that the symbol + designates an OR operation in a control or Boolean function but that it represents an arithmetic plus in a microoperation.)



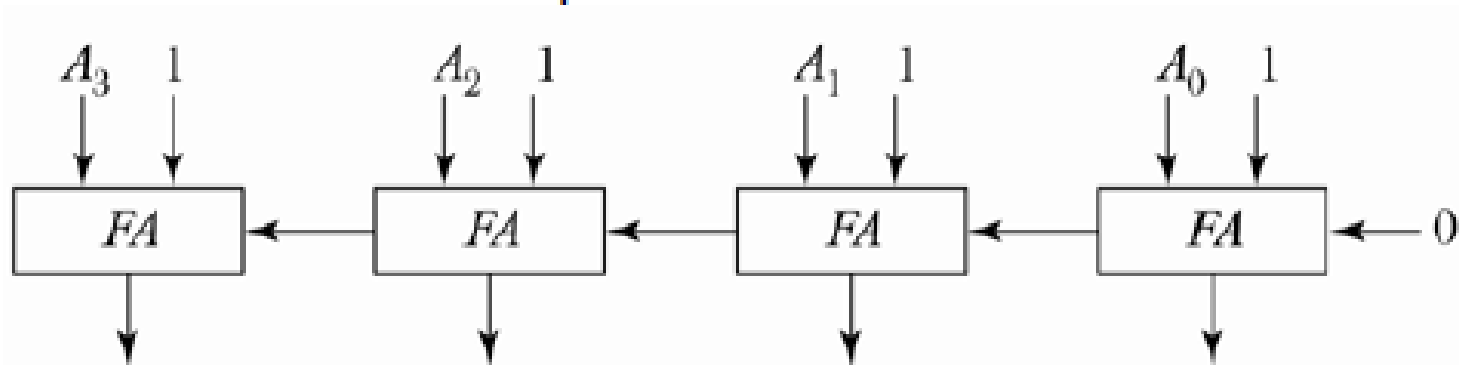
4-9. Show the hardware that implements the following statement. Include the logic gates for the control function and a block diagram for the binary counter with a count enable input.

$$xyT_0 + T_1 + y'T_2: \quad AR \leftarrow AR + 1$$

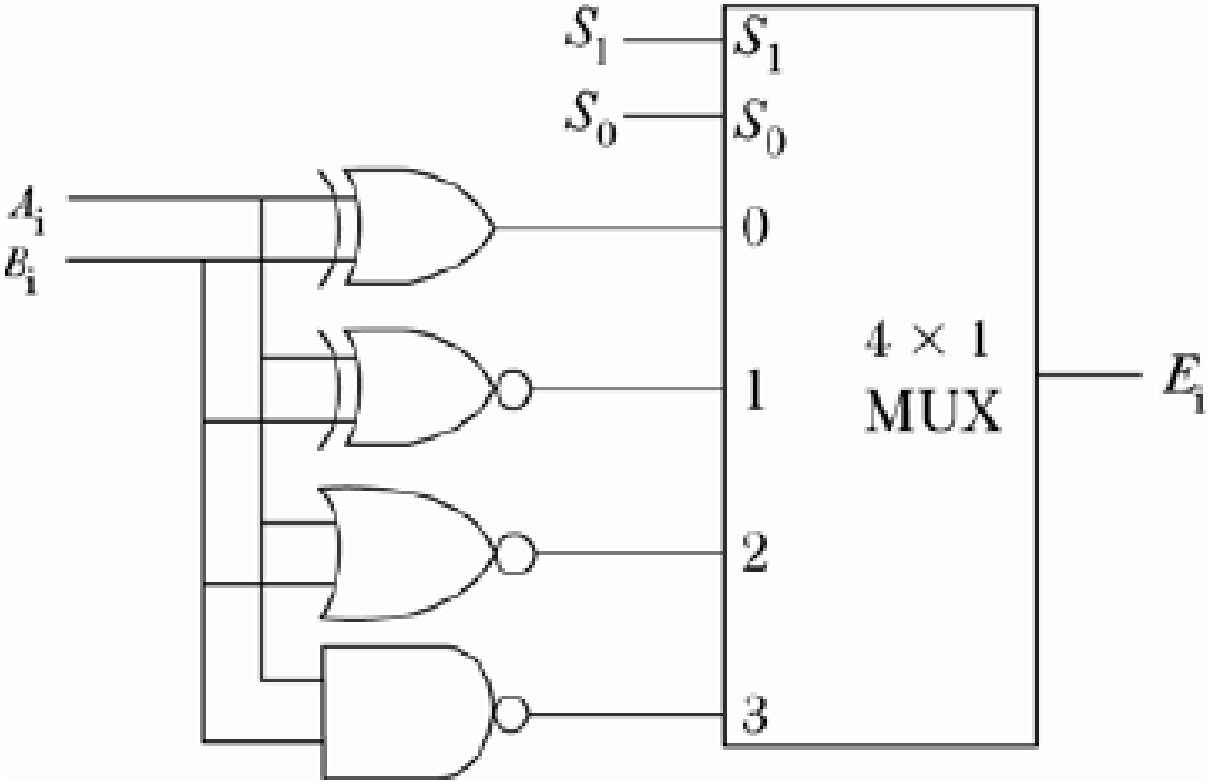


4-13. Design a 4-bit combinational circuit decrementer using four full-adder circuits.

A – 1 = A + 2's complement of 1 = A + 1111



4-17. Design a digital circuit that performs the four logic operations of exclusive-OR, exclusive-NOR, NOR, and NAND. Use two selection variables. Show the logic diagram of one typical stage.



4-18. Register *A* holds the 8-bit binary 11011001. Determine the *B* operand and the logic microoperation to be performed in order to change the value in *A* to:

a. 01101101

b. 11111101

(a) A = 11011001

B = 10110100 $\oplus$

A $\leftarrow$ A $\oplus$ B 01101101

A = 11011001

B = 11111101 (OR)

11111101

A $\leftarrow$ AVB

4-19. The 8-bit registers *AR*, *BR*, *CR*, and *DR* initially have the following values:

$$AR = 11110010$$
$$BR = 11111111$$
$$CR = 10111001$$
$$DR = 11101010$$

Determine the 8-bit values in each register after the execution of the following sequence of microoperations.

$AR \leftarrow AR + BR$	Add *BR* to *AR*
$CR \leftarrow CR \wedge DR,\ BR \leftarrow BR + 1$	AND *DR* to *CR*, increment *BR*
$AR \leftarrow AR - CR$	Subtract *CR* from *AR*

(a) AR = 11110010
BR = 11111111(+)
AR = 11110001 BR = 11111111 CR = 10111001 DR= 1110 1010

(b) CR = 10111001 BR = 1111 1111
DR = 11101010 (AND) +1
CR = 10101000 BR = 0000 0000 AR = 1111 0001 DR = 11101010

(c) AR = 11110001 (–1)
CR = 10101000
AR = 01001001; BR = 00000000; CR = 10101000; DR = 11101010

4-20. An 8-bit register contains the binary value 10011100. What is the register value after an arithmetic shift right? Starting from the initial number 10011100, determine the register value after an arithmetic shift left, and state whether there is an overflow.

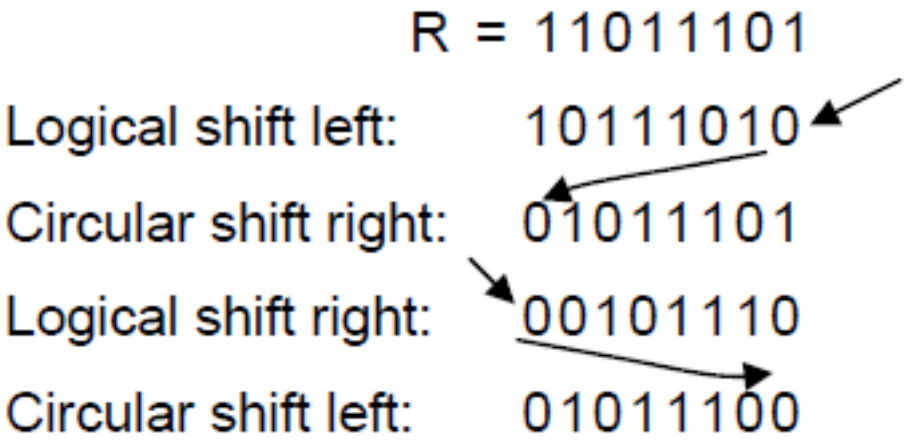
R = 10011100

Arithmetic shift right: 11001110

Arithmetic shift left: 00111000

overflow because a negative number changed to positive.

4-21. Starting from an initial value of $R = 11011101$, determine the sequence of binary values in R after a logical shift-left, followed by a circular shift-right, followed by a logical shift-right and a circular shift-left.



4-23. What is wrong with the following register transfer statements?

a. xT: $AR \leftarrow \overline{AR}$, $AR \leftarrow 0$

b. yT: $R1 \leftarrow R2$, $R1 \leftarrow R3$

c. zT: $PC \leftarrow AR$, $PC \leftarrow PC + 1$

(a) Cannot complement and increment the same register at the same time.

(b) Cannot transfer two different values (R_2 and R_3) to the same register (R_1) at the same time.

(c) Cannot transfer a new value into a register (PC) and increment the original value by one at the same time.