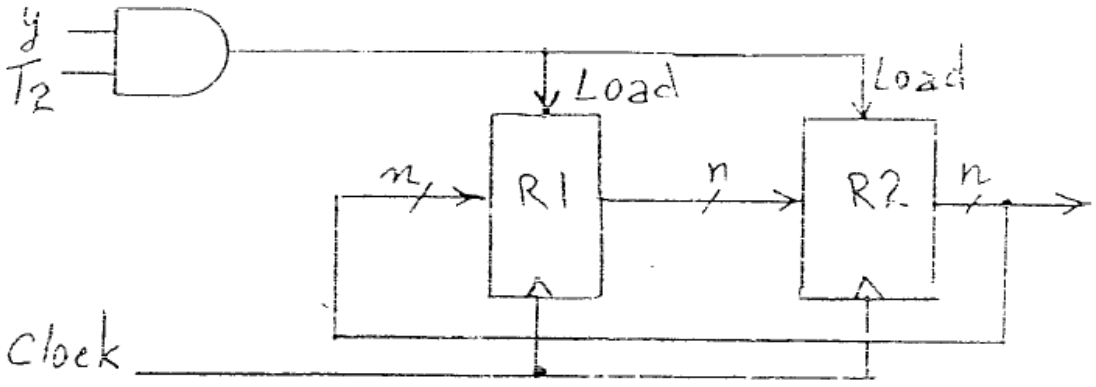
Unit 2 Numerical

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

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

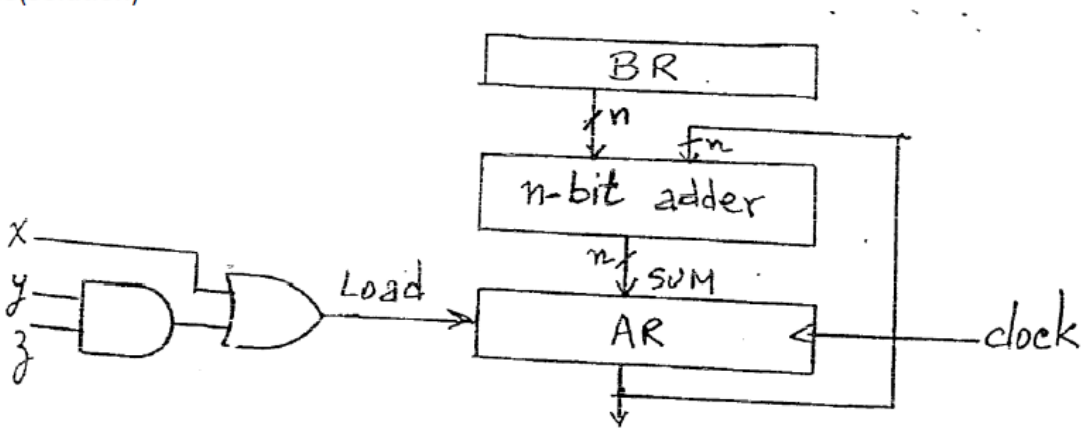


4-8. Draw the block diagram for the hardware that implements the following statements:

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

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.)

4.8(Solution)



4-12. The adder-subtractor circuit of Fig. 4-7 has the following values for input mode M and data inputs A and B. In each case, determine the values of the outputs: S_3, S_2, S_1, S_0, and C_4.

	M	A	B
a.	0	0111	0110
b.	0	1000	1001
c.	1	1100	1000
d.	1	0101	1010
e.	1	0000	0001

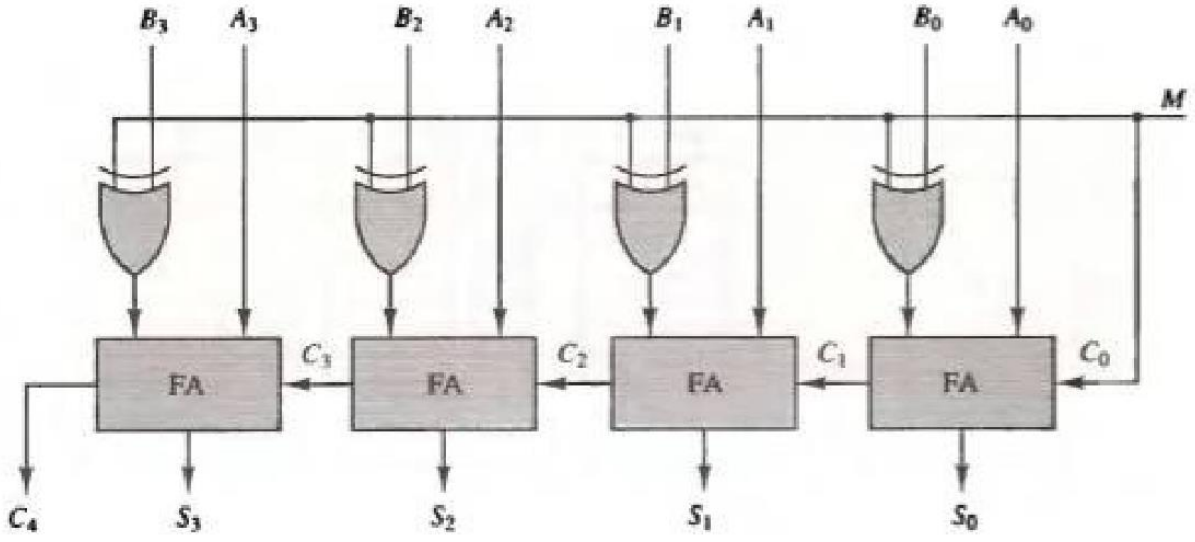


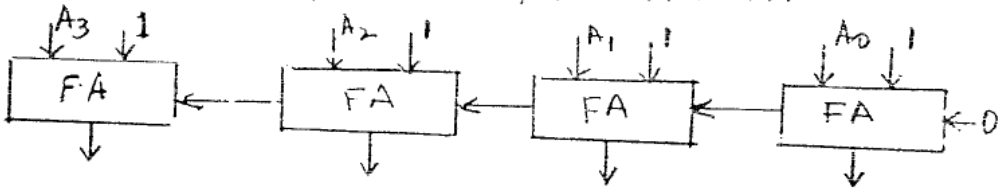
Figure 4-7 4-bit adder-subtractor.

4-12

M	A	B	Sum	C_4	
0	0111	+ 0110	1101	0	7+6 = 13
0	1000	+ 1001	0001	1	8+9 = 16+1
1	1100	− 1000	0100	1	12−8 = 4
1	0101	− 1010	1011	0	5−10 = −5 (in 2's comp.)
1	0000	− 0001	1111	0	0−1 = −1 (in 2's comp.)

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

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



4-14. Assume that the 4-bit arithmetic circuit of Fig. 4-9 is enclosed in one IC package. Show the connections among two such ICs to form an 8-bit arithmetic circuit.

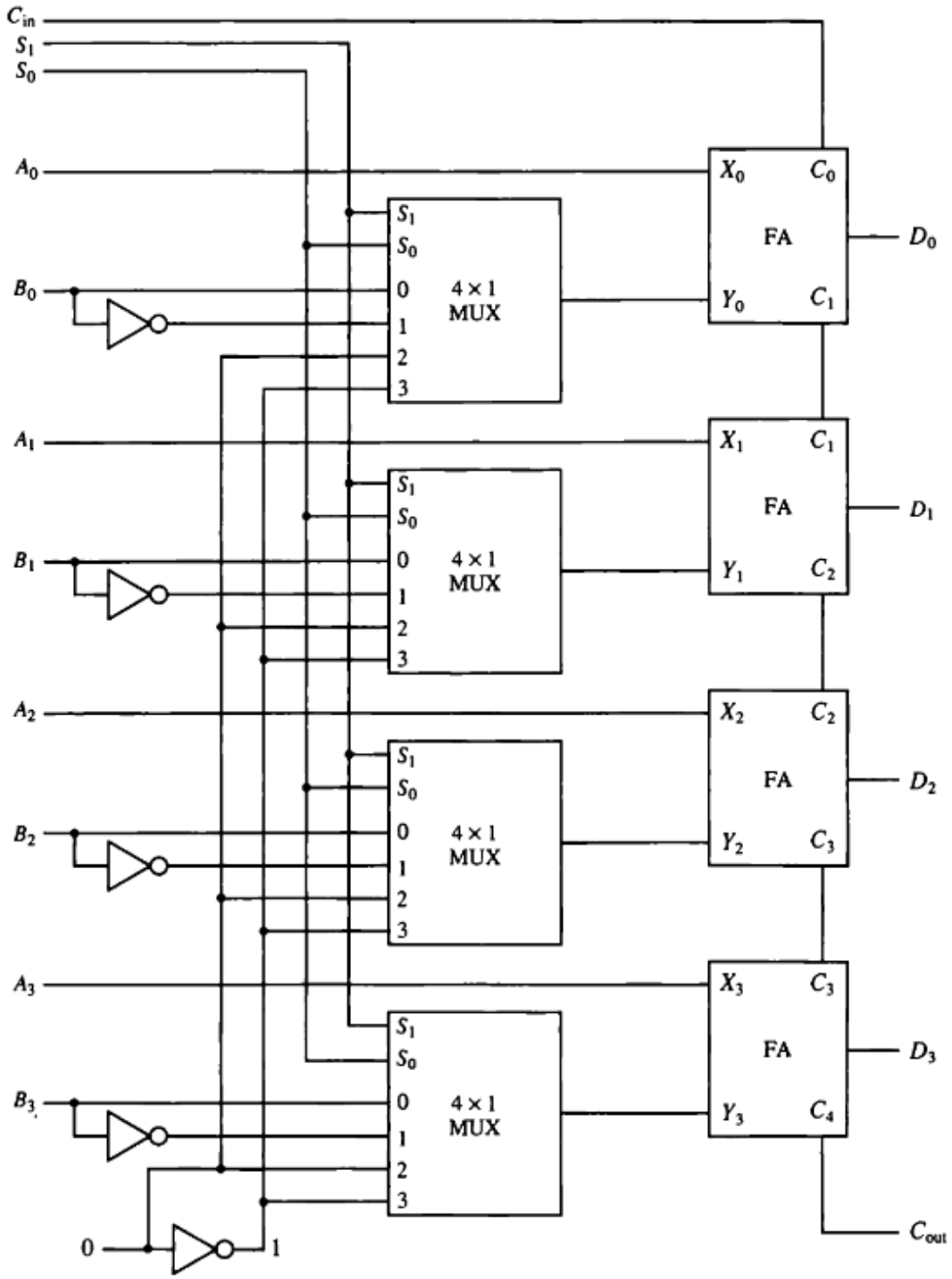
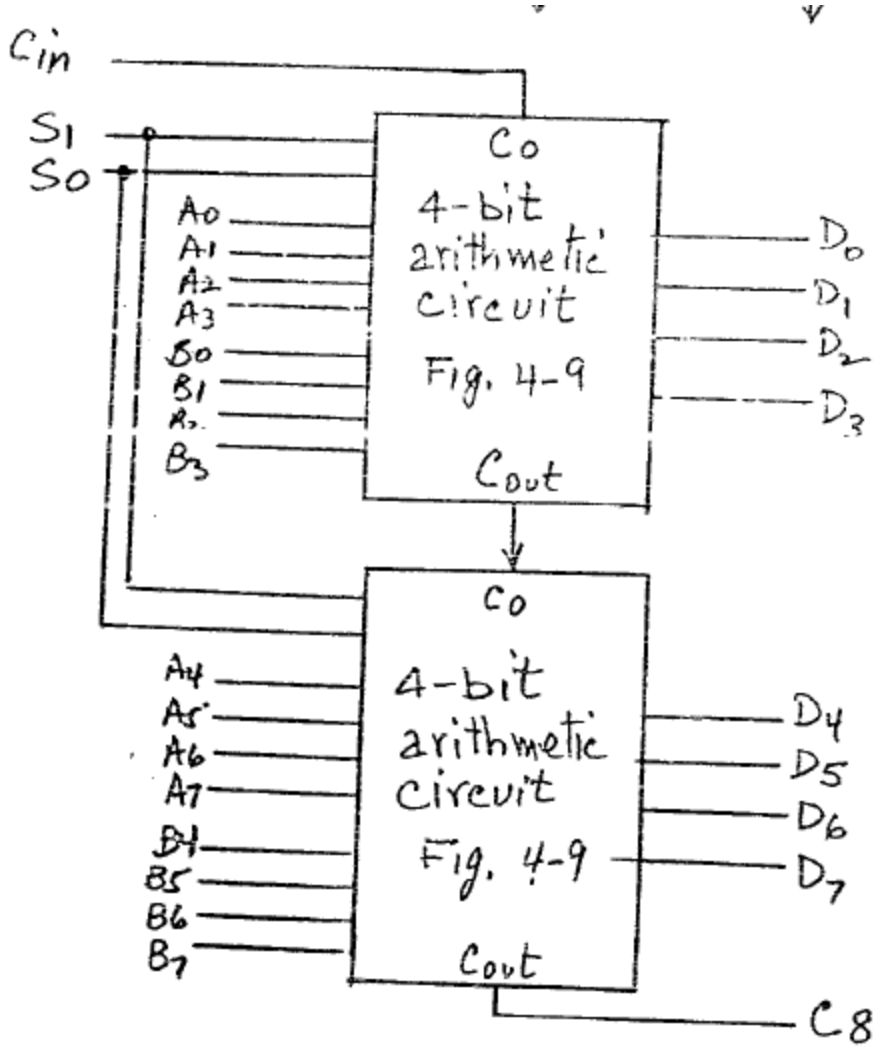


Figure 4-9 4-bit arithmetic circuit.

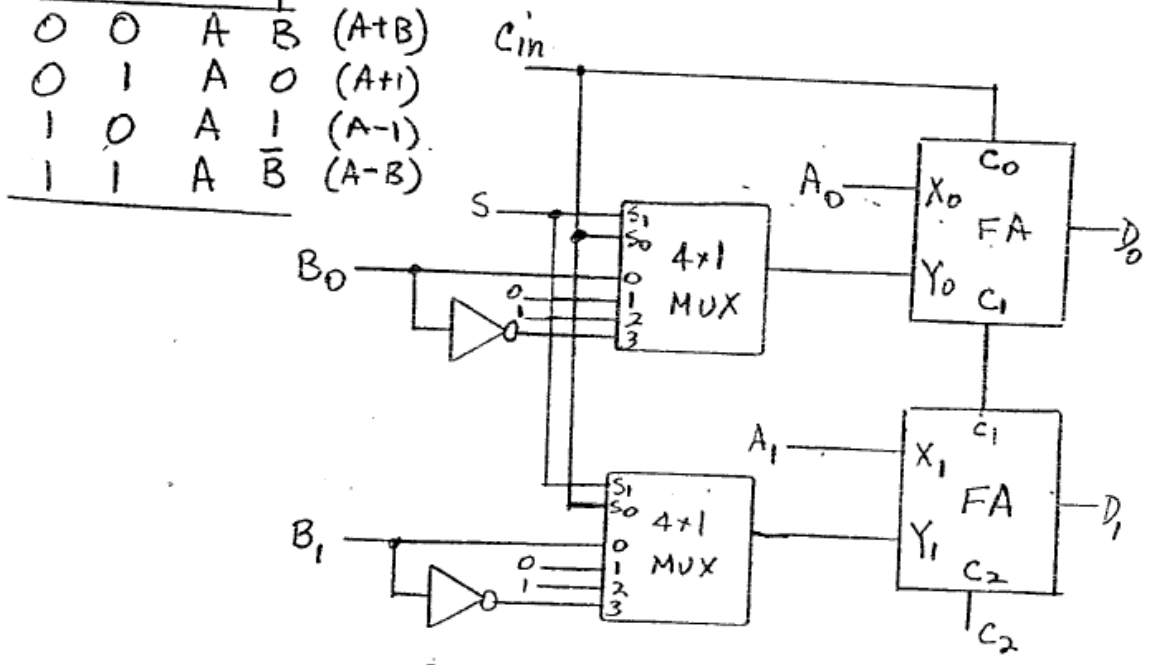


4-15. Design an arithmetic circuit with one selection variable S and two n-bit data inputs A and B. The circuit generates the following four arithmetic operations in conjunction with the input carry C_{in}. Draw the logic diagram for the first two stages.

S	$C_{in} = 0$	$C_{in} = 1$
0	$D = A + B$ (add)	$D = A + 1$ (increment)
1	$D = A - 1$ (decrement)	$D = A + \overline{B} + 1$ (subtract)

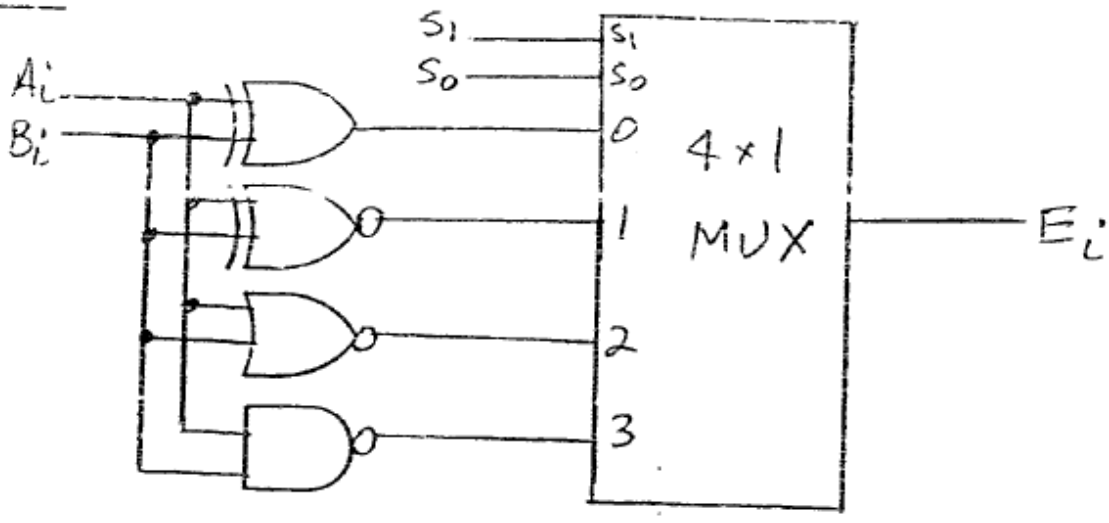
4-15

S	C_{in}	X	Y	
0	0	A	B	(A+B)
0	1	A	0	(A+1)
1	0	A	1	(A-1)
1	1	A	$\overline{B}$	(A-B)



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.

4-17



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

4-18

(a) A = 11011001
B = 10110100 ⊕
A ← A⊕B 01101101

A = 11011001
B = 11111101 (OR)
11111101 A ← 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*

4-19

(a) AR = 11110010
BR = 11111111 (+)
AR = 11110001 BR = 11111111 CR = 10111001 DR = 11101010

(b) CR = 10111001 BR = 11111111
DR = 11101010 (AND) + 1
CR = 10101000 BR = 00000000 AR = 11110001 DR = 11101010

(c) AR = 11110001
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.

4-20

R = 10011100

Arithmetic shift right : 11001110
Arithmetic shift left : 00111000 overflow because a negative number changed to positive

4-21

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.

R = 11011101

logical shift left: 10111010

Circular shift right: 01011101

logical shift right: 00101110

Circular shift left: 01011100

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 ($R1$) 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.