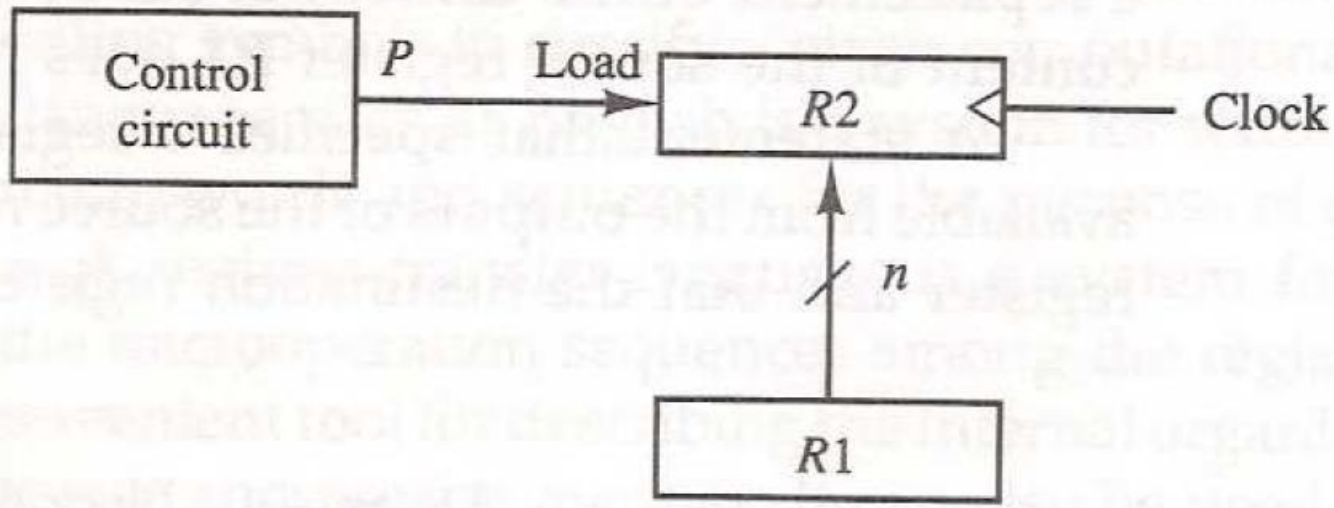
Numericals Chapter 4

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

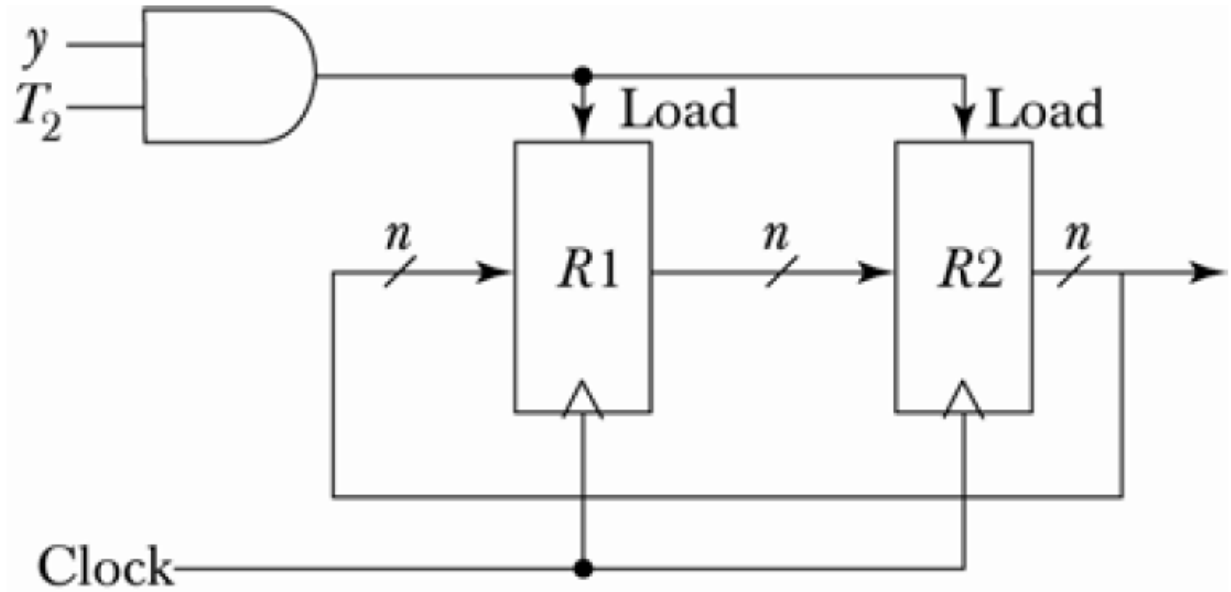
$$yT_2: \quad R2 \leftarrow R1, \quad R1 \leftarrow R2$$

Figure 4-2 Transfer from $R1$ to $R2$ when $P = 1$.



(a) Block diagram

4.1

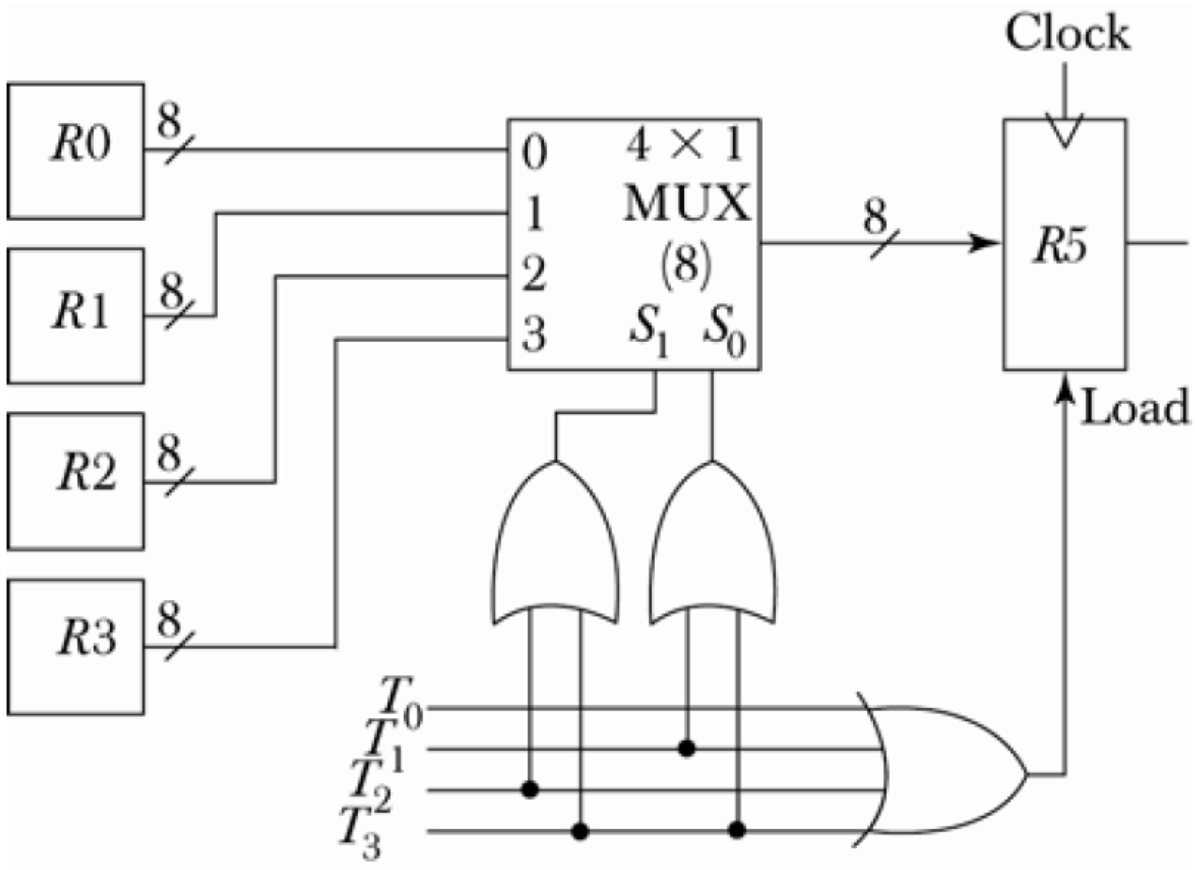


4-2. The outputs of four registers, $R0$, $R1$, $R2$, and $R3$, are connected through 4-to-1-line multiplexers to the inputs of a fifth register, $R5$. Each register is eight bits long. The required transfers are dictated by four timing variables T_0 through T_3 as follows:

$$
\begin{aligned}
T_0&: \quad R5 \leftarrow R0 \\
T_1&: \quad R5 \leftarrow R1 \\
T_2&: \quad R5 \leftarrow R2 \\
T_3&: \quad R5 \leftarrow R3
\end{aligned}
$$

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 $R5$.

4.2



$T_0T_1T_2T_3$	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

$S_1 = T_2 + T_3$

$S_0 = T_1 + T_3$

$load = T_0 + T_1 + T_2 + T_3$

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

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

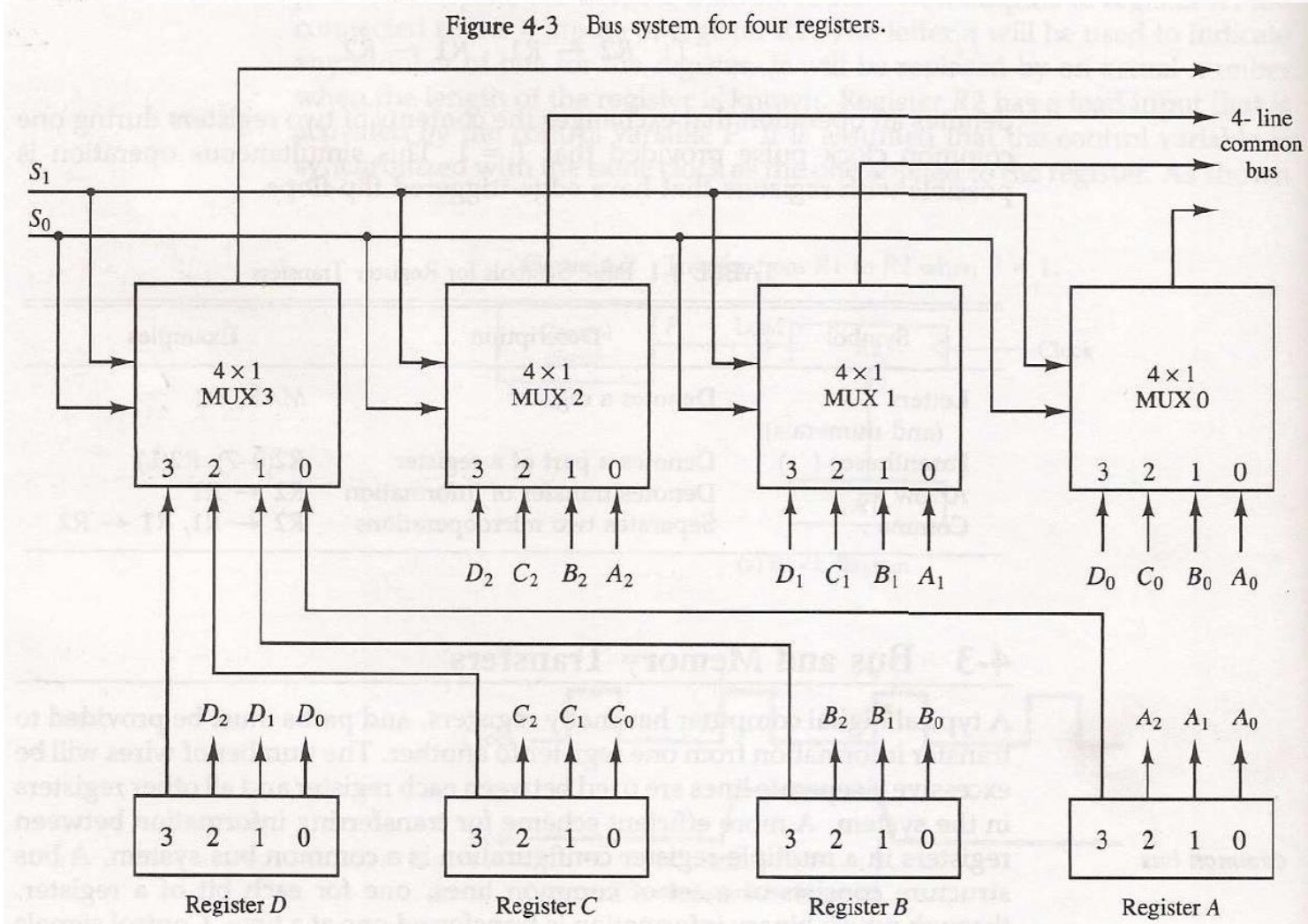
4.3

P: R1 ← R2

P'Q: R1 ← R3

4-4. What has to be done to the bus system of Fig. 4-3 to be able to transfer information from any register to any other register? Specifically, show the connections that must be included to provide a path from the outputs of register C to the inputs of register A.

Figure 4-3 Bus system for four registers.



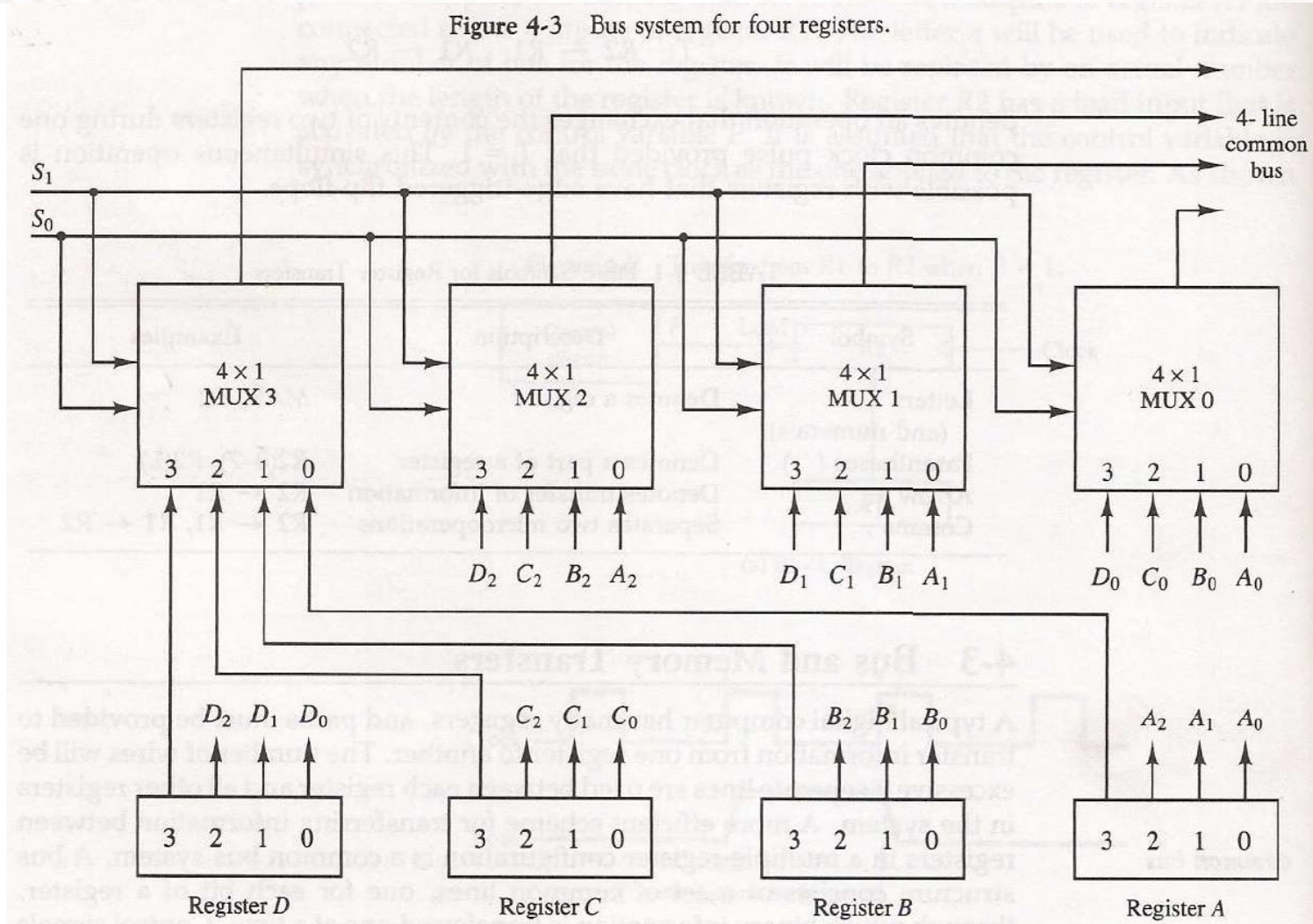
4.4

Connect the 4-line common bus to the four inputs of each register.
Provide a “load” control input in each register.
Provide a clock input for each register.

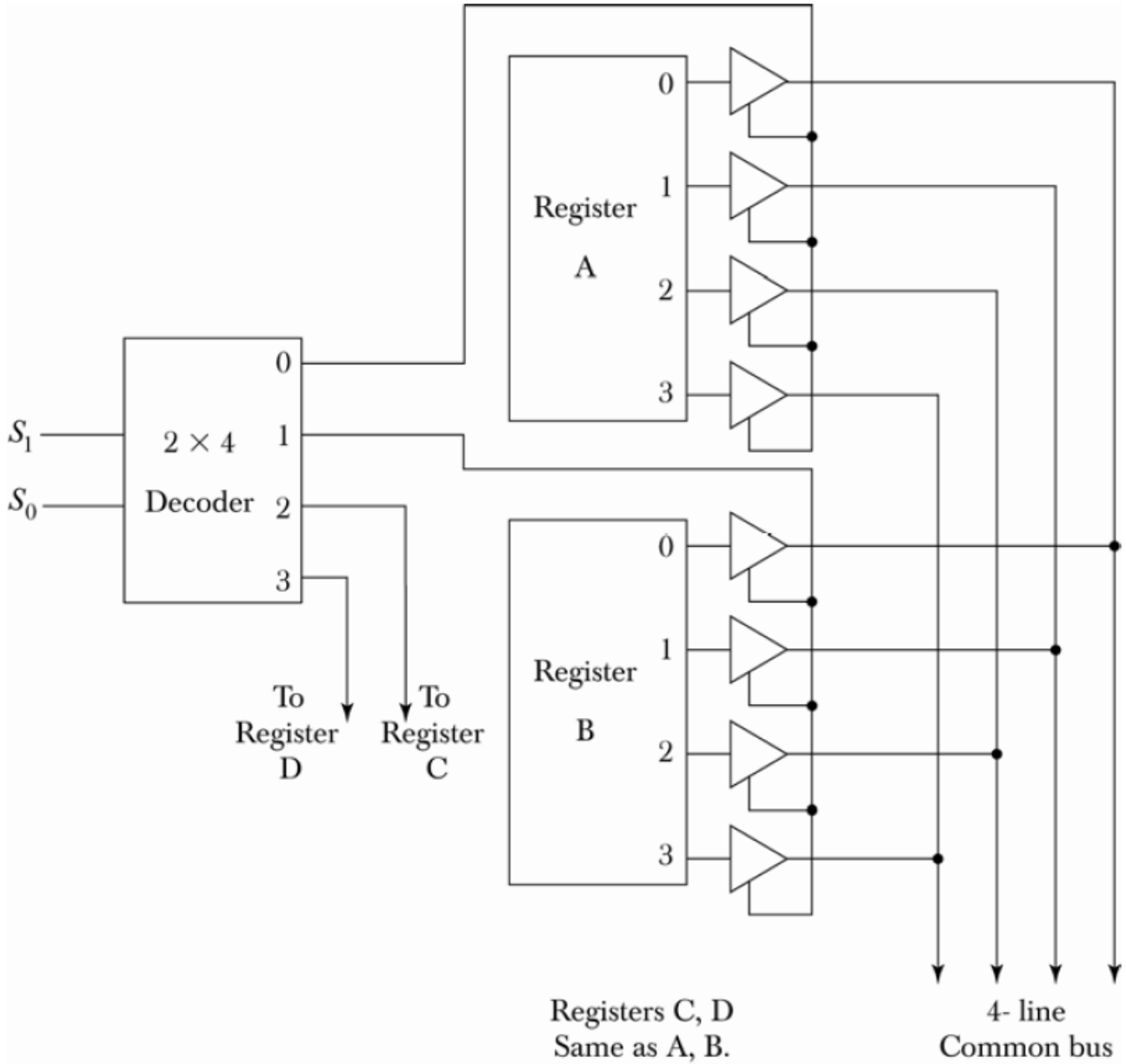
To transfer from register C to register A:
Apply $S_1S_0 = 10$ (to select C for the bus.)
Enable the load input of A
Apply a clock pulse.

4-5. Draw a diagram of a bus system similar to the one shown in Fig. 4-3, but use three-state buffers and a decoder instead of the multiplexers.

Figure 4-3 Bus system for four registers.



4.5


Register A
Register B
0
1
2
3
S_1
S_0
2 × 4 Decoder
To Register D
To Register C
Registers C, D Same as A, B.
4- line Common bus

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?

4.6

(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. $R5 \leftarrow M[R5]$

4.7

(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.

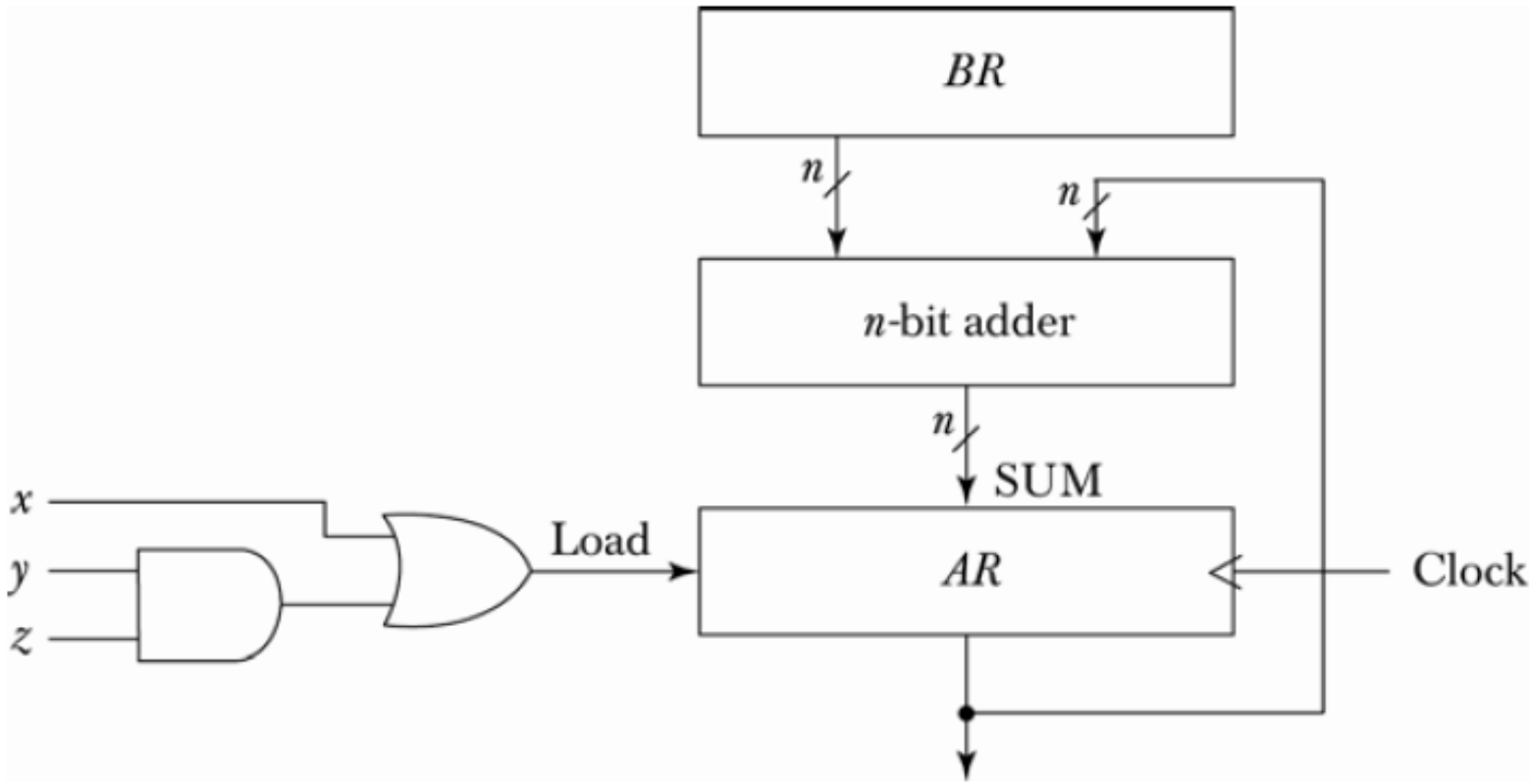
(c) Read memory word specified by the address in R5 and transfer content to R5 (destroys previous value)

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.)

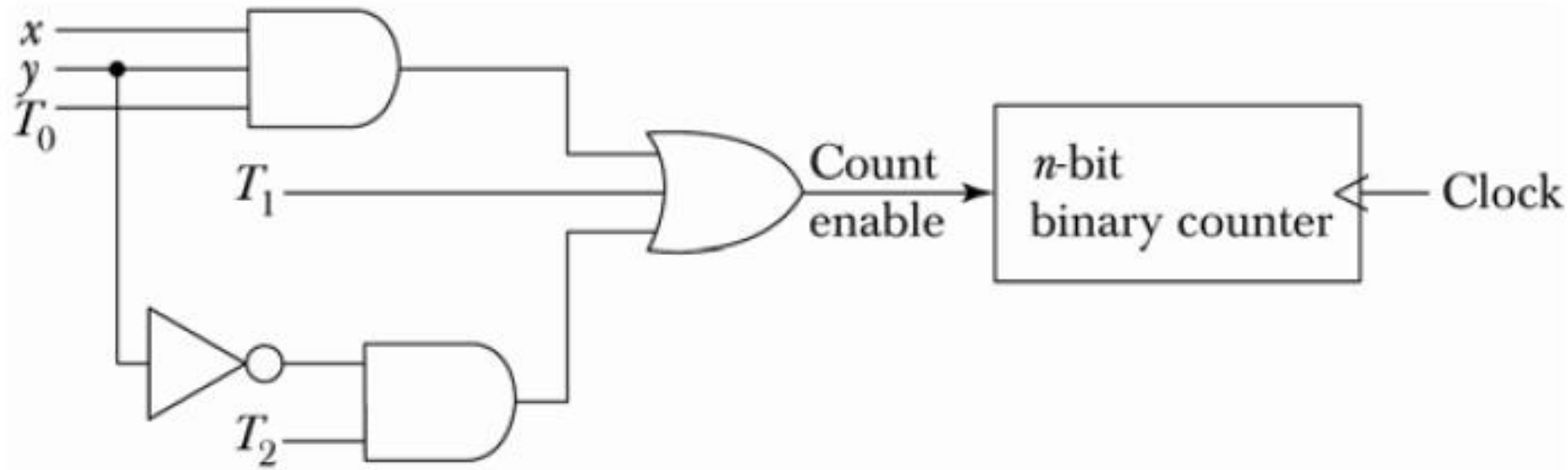
4.8



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.

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

4.9

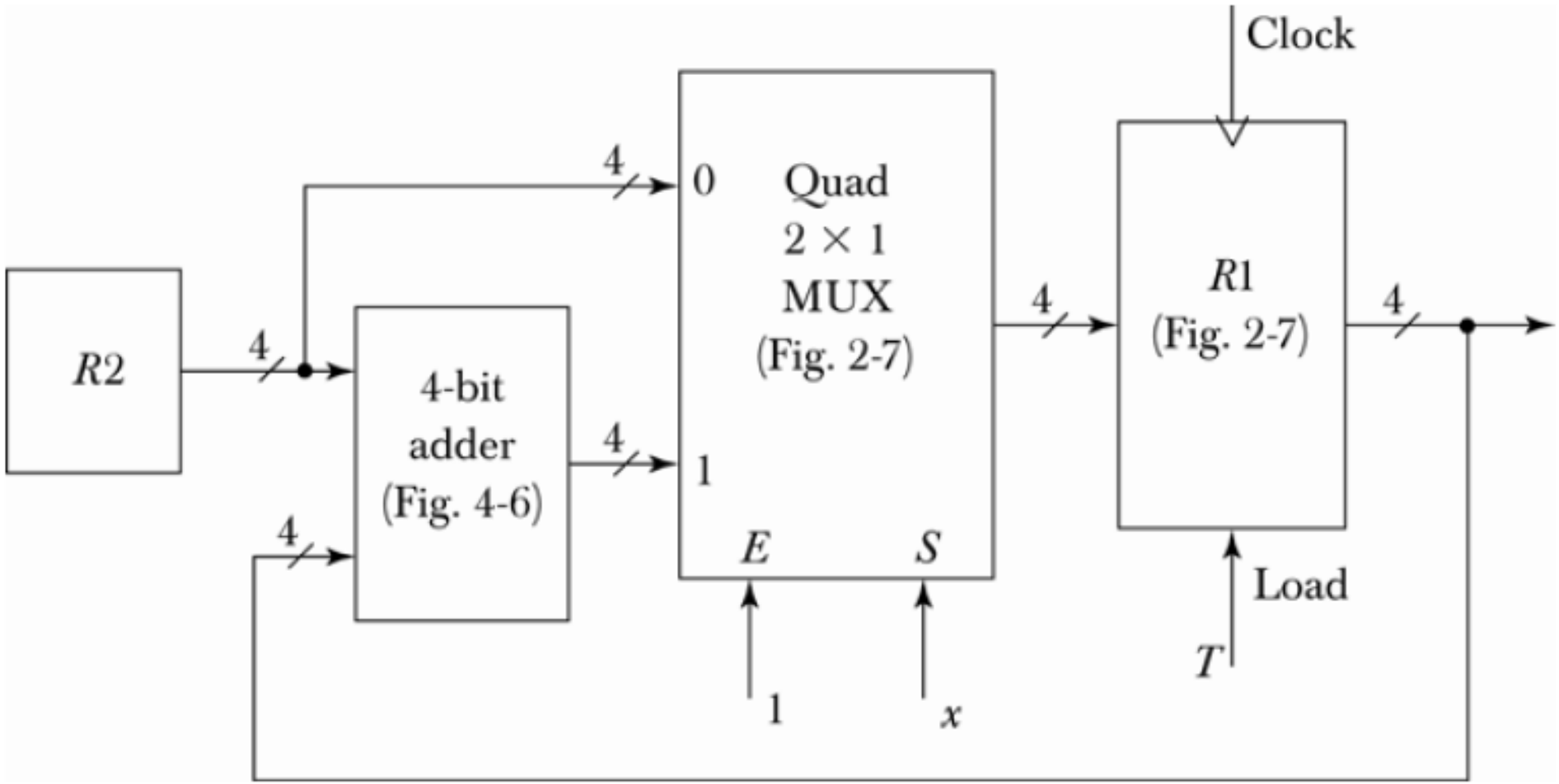


4-10. Consider the following register transfer statements for two 4-bit registers $R1$ and $R2$.

$$xT: \quad R1 \leftarrow R1 + R2$$
$$x'T: \quad R1 \leftarrow R2$$

Every time that variable $T = 1$, either the content of $R2$ is added to the content of $R1$ if $x = 1$, or the content of $R2$ is transferred to $R1$ if $x = 0$. Draw a diagram showing the hardware implementation of the two statements. Use block diagrams for the two 4-bit registers, a 4-bit adder, and a quadruple 2-to-1-line multiplexer that selects the inputs to $R1$. In the diagram, show how the control variables x and T select the inputs of the multiplexer and the load input of register $R1$.

4.10


Clock
4
0
Quad
2 × 1
MUX
(Fig. 2-7)
R2
4
4-bit
adder
(Fig. 4-6)
4
1
4
4
R1
(Fig. 2-7)
4
4
E
S
Load
T
1
x

4-11. Using a 4-bit counter with parallel load as in Fig. 2-11 and a 4-bit adder as in Fig. 4-6, draw a block diagram that shows how to implement the following statements:

x:	$R1 \leftarrow R1 + R2$	Add $R2$ to $R1$
$x'y$:	$R1 \leftarrow R1 + 1$	Increment $R1$

where $R1$ is a counter with parallel load and $R2$ is a 4-bit register.

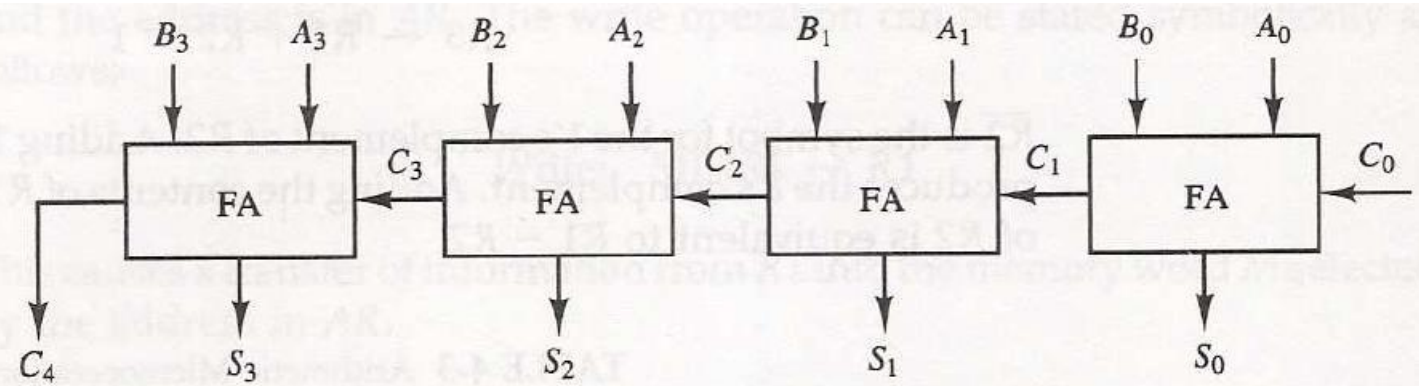
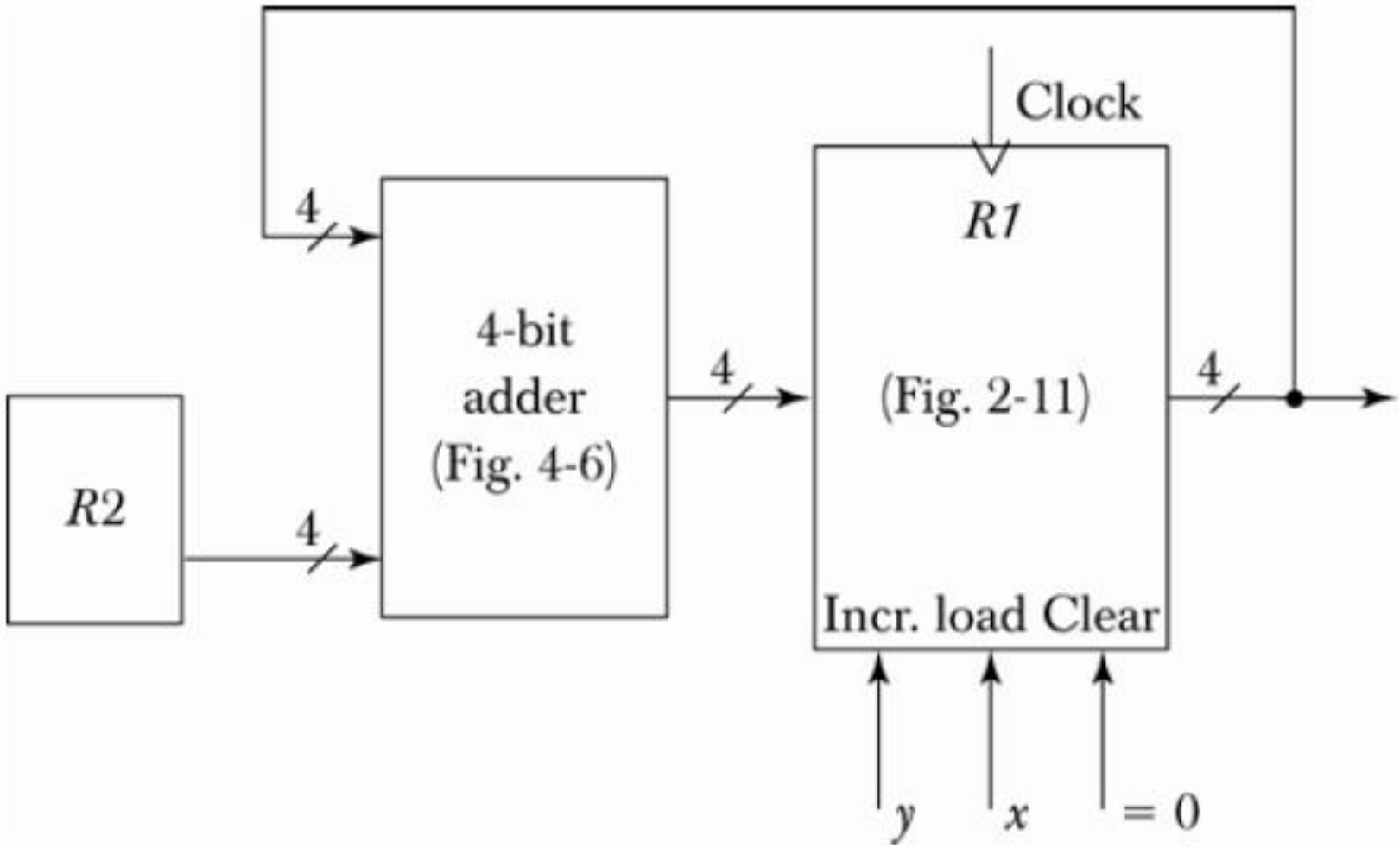


Figure 4-6 4-bit binary adder.

4.11



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

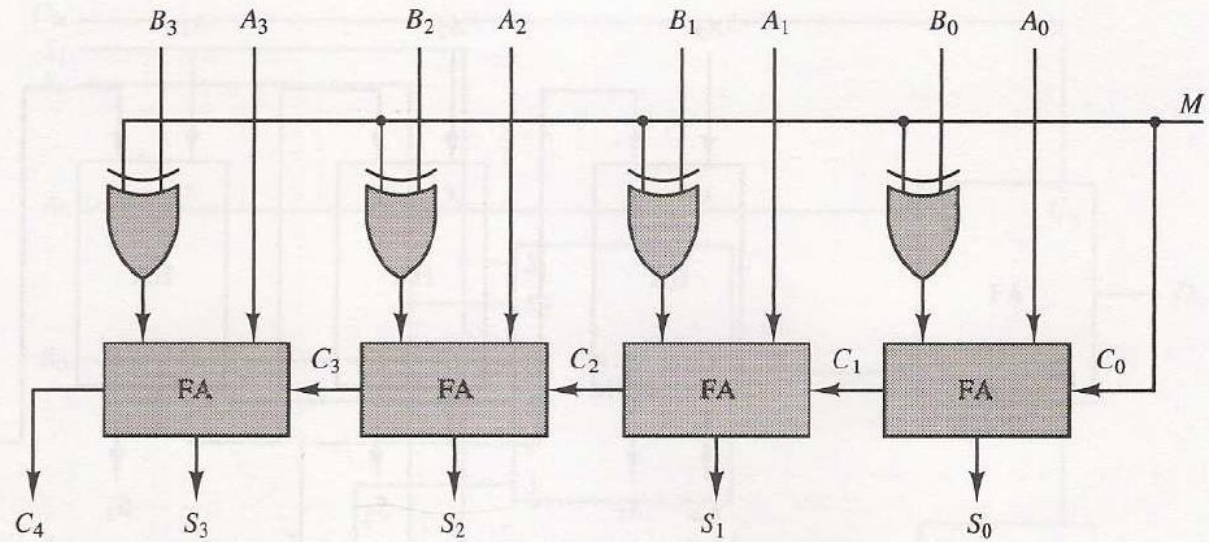


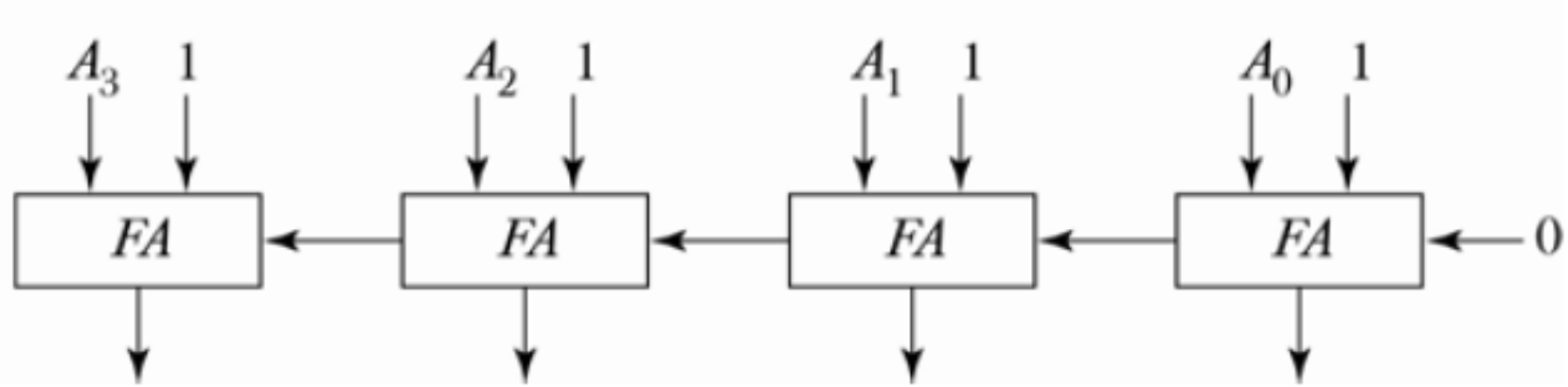
Figure 4-7 4-bit adder-subtractor.

4.12

M	A		B	Sum	Cu	
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.

4.13 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.

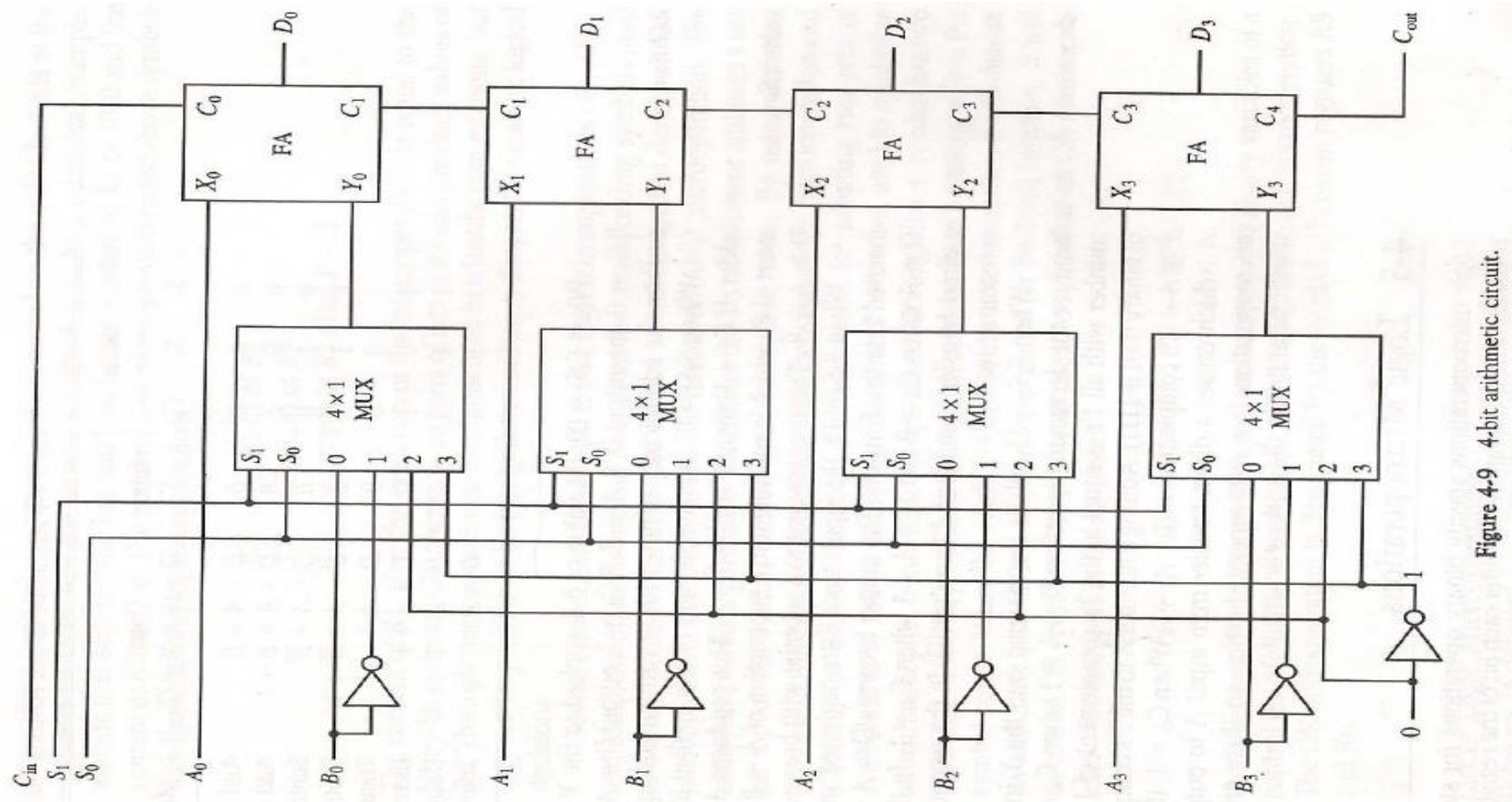
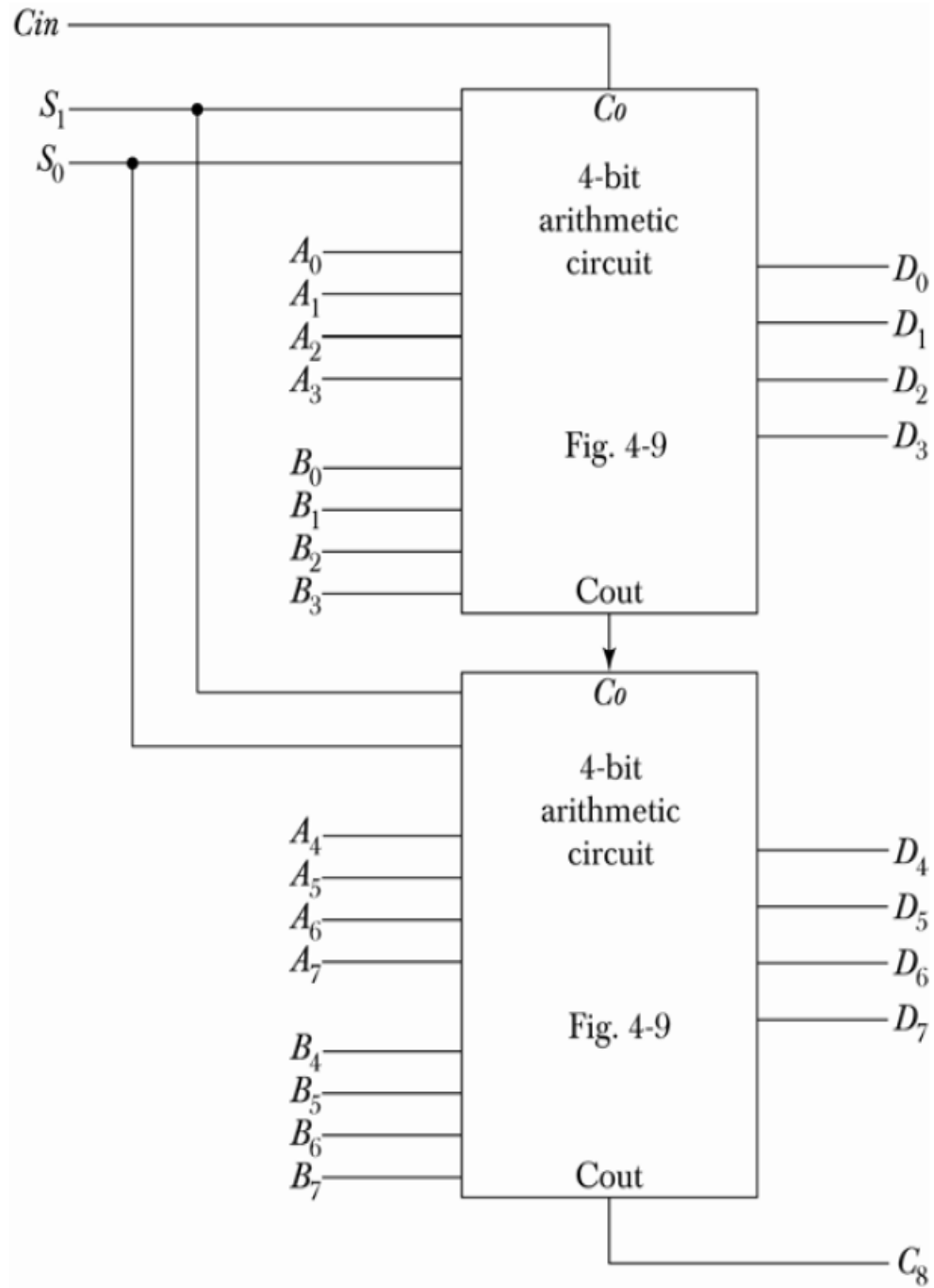


Figure 4-9 4-bit arithmetic circuit.

4.14

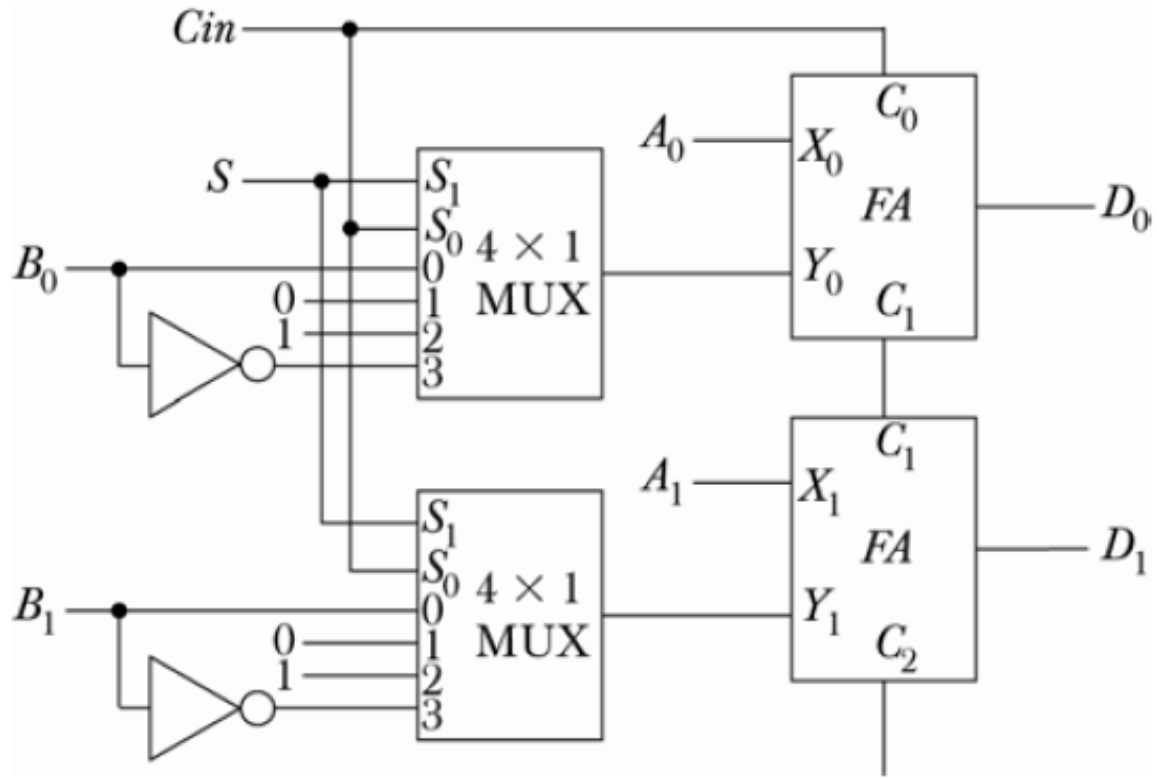

Cin
S1
S0
Co
4-bit
arithmetic
circuit
A0
A1
A2
A3
B0
B1
B2
B3
D0
D1
D2
D3
Fig. 4-9
Cout
Co
4-bit
arithmetic
circuit
A4
A5
A6
A7
B4
B5
B6
B7
D4
D5
D6
D7
Fig. 4-9
Cout
C8

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	Cin	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)



4-16. Derive a combinational circuit that selects and generates any of the 16 logic functions listed in Table 4-5.

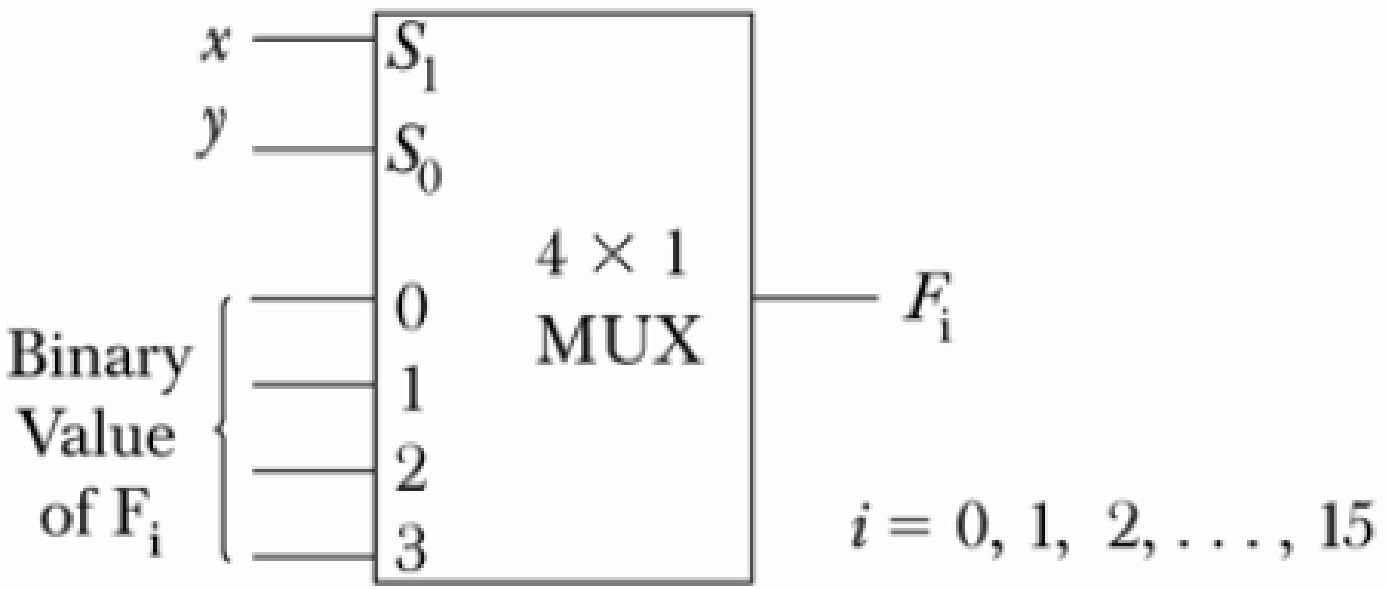
TABLE 4-5 Truth Tables for 16 Functions of Two Variables

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

TABLE 4-6 Sixteen Logic Microoperations

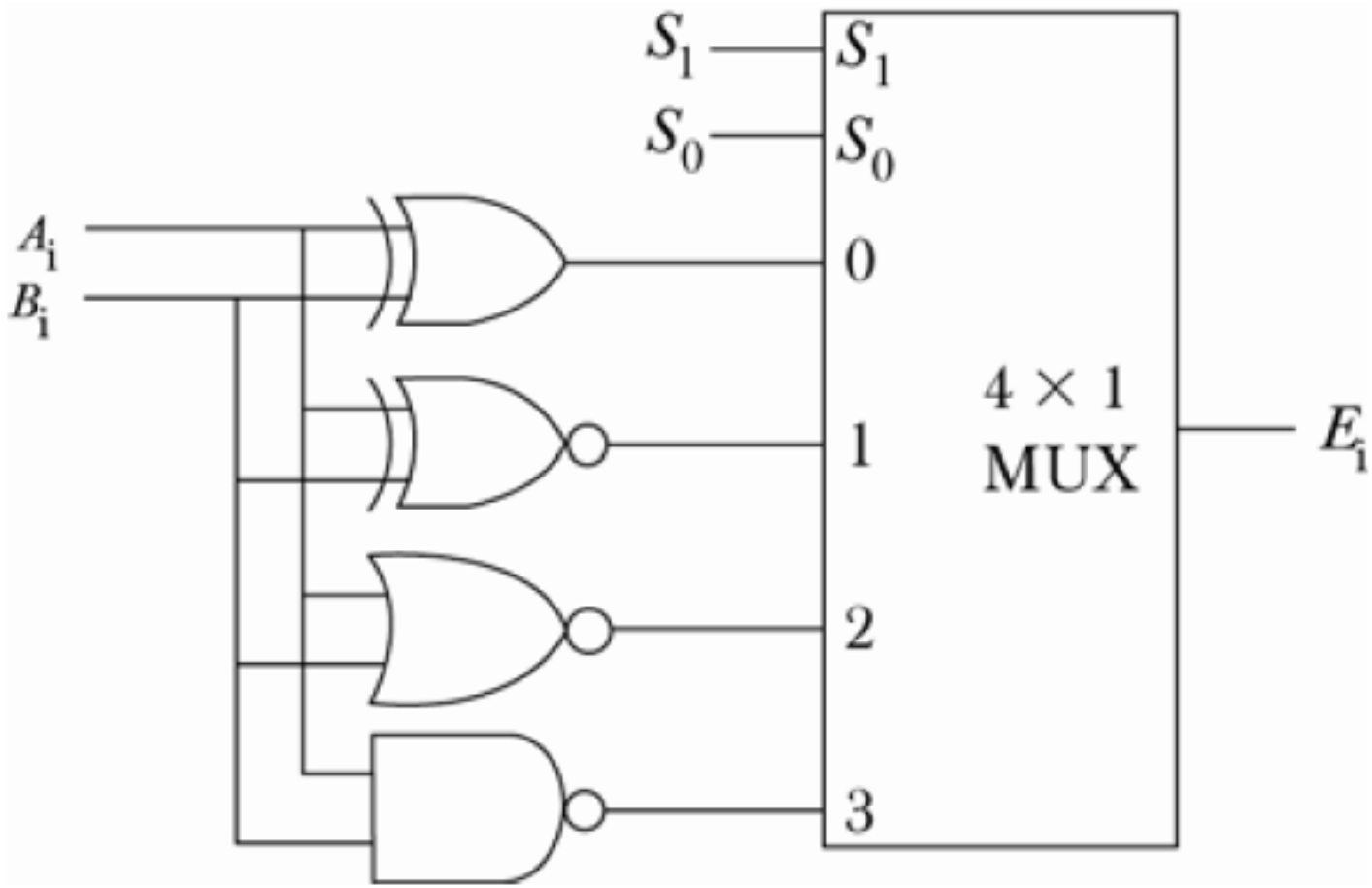
Boolean function	Microoperation	Name
$F_0 = 0$	$F \leftarrow 0$	Clear
$F_1 = xy$	$F \leftarrow A \wedge B$	AND
$F_2 = xy'$	$F \leftarrow A \wedge \overline{B}$	
$F_3 = x$	$F \leftarrow A$	Transfer A
$F_4 = x'y$	$F \leftarrow \overline{A} \wedge B$	
$F_5 = y$	$F \leftarrow B$	Transfer B
$F_6 = x \oplus y$	$F \leftarrow A \oplus B$	Exclusive-OR
$F_7 = x + y$	$F \leftarrow A \vee B$	OR
$F_8 = (x + y)'$	$F \leftarrow \overline{A \vee B}$	NOR
$F_9 = (x \oplus y)'$	$F \leftarrow \overline{A \oplus B}$	Exclusive-NOR
$F_{10} = y'$	$F \leftarrow \overline{B}$	Complement B
$F_{11} = x + y'$	$F \leftarrow A \vee \overline{B}$	
$F_{12} = x'$	$F \leftarrow \overline{A}$	Complement A
$F_{13} = x' + y$	$F \leftarrow \overline{A} \vee B$	
$F_{14} = (xy)'$	$F \leftarrow \overline{A \wedge B}$	NAND
$F_{15} = 1$	$F \leftarrow$ all 1's	Set to all 1's

4.16



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)

$$\begin{array}{ll} A = 11011001 & \\ \underline{B = 10110100} & \oplus \\ A \leftarrow A \oplus B\ 01101101 & \end{array}$$

$$\begin{array}{ll} A = 11011001 & \\ \underline{B = 11111101} & \text{(OR)} \\ 11111101 & A \leftarrow AVB \end{array}$$

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

$$\begin{aligned} AR &= 11110010 \\ BR &= 11111111 \\ CR &= 10111001 \\ DR &= 11101010 \end{aligned}$$

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= 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.

4.20

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.

4.21

R = 11011101

Logical shift left: 10111010

Circular shift right: 01011101

Logical shift right: 00101110

Circular shift left: 01011100