现态下标n省略

Problems 4.6

4-6. A sequential circuit with two D flip-flops A and B, two inputs X and Y, and one output Z is specified by the following input equations:

$$D_A = XA + \bar{X}\bar{Y}, \qquad D_B = XB + \bar{X}A, \qquad Z = \bar{X}B$$

(a) Draw the logic diagram of the circuit.

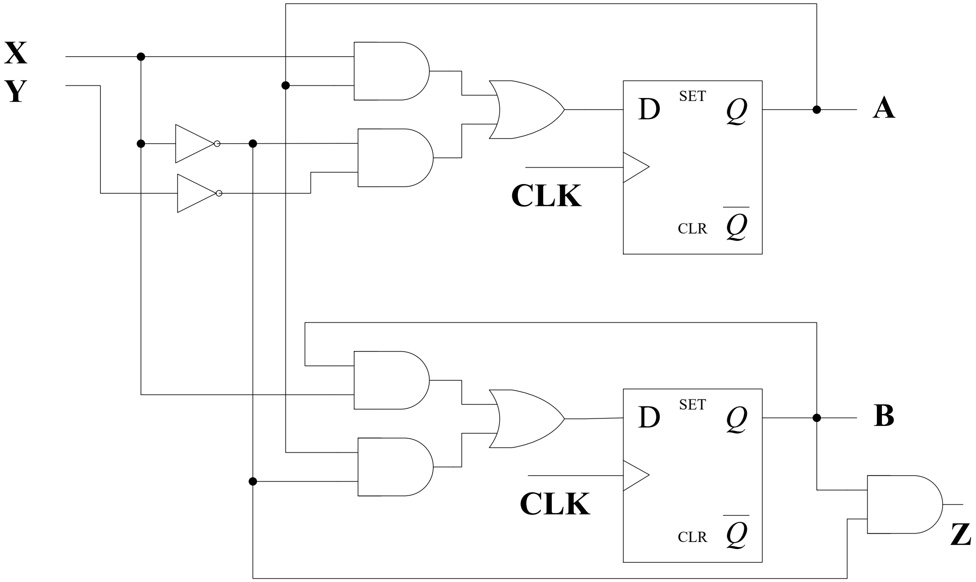
(b) Derive the state table.

(c) Derive the state diagram.

(d) Is this a Mealy or a Moore machine?

Solution 4.6：

a) According to the input equations and output equation, the logic diagram is:



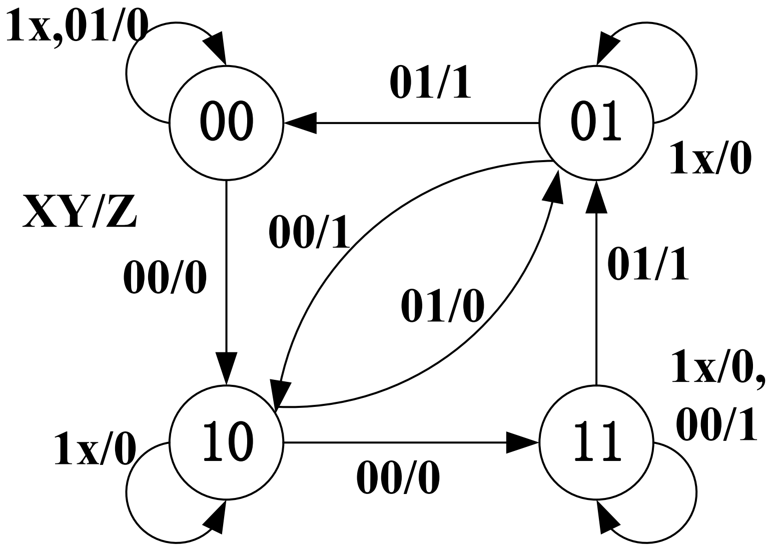
现态下标n省略

Problems 4.6-Continue before：

b) According to the input equations and output equation, the state table is:

Present state		Inputs		Next state		Output
A	B	X	Y	A	B	Z
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	1	0	0
1	1	0	0	1	1	1
1	1	0	1	0	1	1
1	1	1	0	1	1	0
1	1	1	1	1	1	0

c) According to the state table, the state diagram is:



现态下标n省略

Problems 4.7

4-7. *A sequential circuit has three D flip-flops A, B, and C, and one input X. The circuit is described by the following input equations:

$$D_A = (B\overline{C} + \overline{B}C)X + (BC + \overline{B}\,\overline{C})\,\overline{X} \qquad D_B = A \qquad D_C = B$$

(a) Derive the state table for the circuit.

(b) Draw two state diagrams, one for and the other for X = 1.

Solution 4.7：

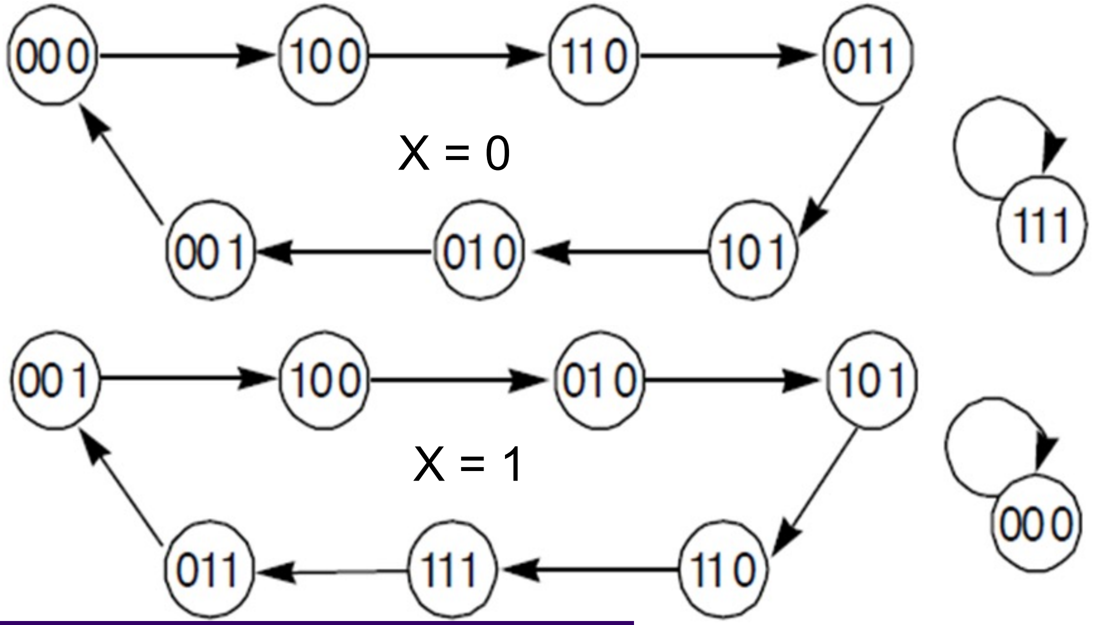
(a) State equations：$A_{n+1} = (B\overline{C} + \overline{B}C)X + (BC + \overline{B}\,\overline{C})\,\overline{X}$， $B_{n+1} = A$， $C_{n+1} = B$

现态下标*n*省略

State table ：

Present state			Input	Next state		
A	*B*	*C*	*X*	*A*	*B*	*C*
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1

(b) State diagrams：

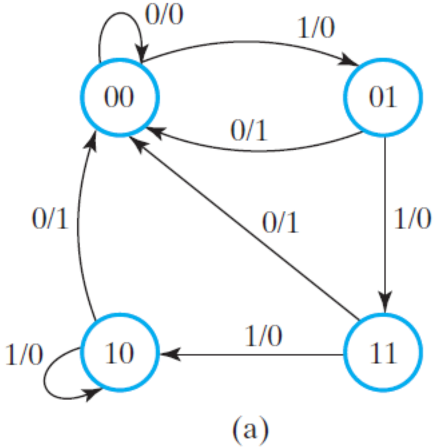


Problems 4.9

4.9 Starting from state 00 in the state diagram of Figure 4-15(a), determine the state transitions and output sequence that will be generated when an input sequence of 10011011110 is applied.

Solution 4.9 :

Figure 4-15(a)



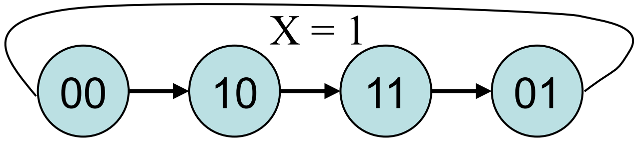
Present State	00	01	00	00	01	11	00	01	11	10	10
Input	1	0	0	1	1	0	1	1	1	1	0
Output	0	1	0	0	0	1	0	0	0	0	1
Next State	01	00	00	01	11	00	01	11	10	10	00

Problems 4.13

4-13. *Design a sequential circuit with two D flip-flops A and B and one input X. When X = 0, the state of the circuit remains the same. When X = 1, the circuit goes through the state transitions from 00 to 10 to 11 to 01, back to 00, and then repeats.

Solution 4.13 :

(a) State diagrams ：



(b) State table ：

Present state		Input	Next state	
A	*B*	*X*	*A*	*B*
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1

(c) State/Input equations :

D_A

			B	B
		1		
A	1	1		1
		X	X	

$$D_A = A\bar{X} + \bar{B}X$$

D_B

			B	B
				1
A		1	1	1
		X	X	

$$D_B = AX + B\bar{X}$$

(d) Logic diagram :

Skip， Draw your own.

Problems 4.21

4-20. A Universal Serial Bus (USB) communication link requires a circuit that produces the sequence 00000001. You are to design a synchronous sequential circuit that starts producing this sequence for input E = 1. Once the sequence starts, it completes. If E = 1, during the last output in the sequence, the sequence repeats. Otherwise, if E = 0, the output remains constant at 1.

(a) Draw the Moore state diagram for the circuit.

(b) Find the state table and make a state assignment.

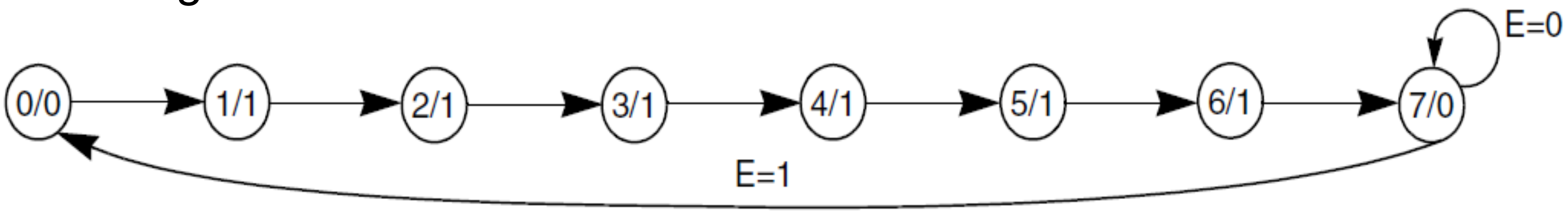
(c) Design the circuit using D flip-flops and logic gates.

A reset should be included to place the circuit in the appropriate initial state at which E is examined to determine if the sequence of constant 1s is to be produced.

4-21. Repeat Problem 4-20 for the sequence **01111110** that is used in a different communication network protocol.

Solution 4.21 :

(a) State diagram

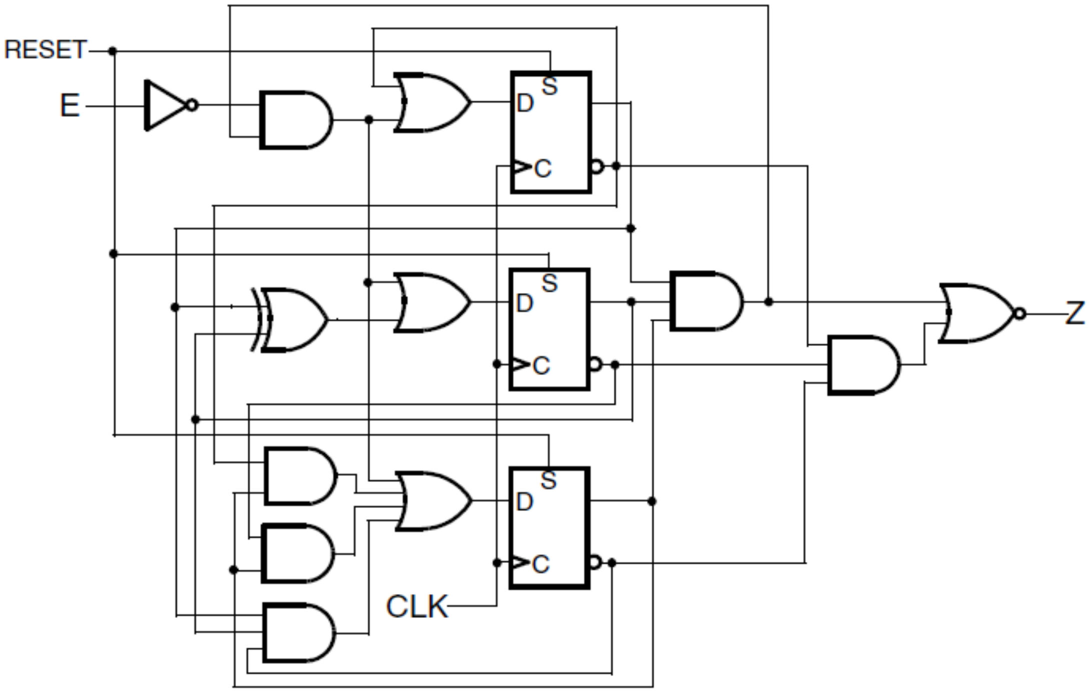


Solution 4.21-Continue before :

(b) State :

Present state	Next State For Input		Output
$D_2D_1D_0$	*E=0*	*E=1*	*Z*
000	001	001	0
001	010	010	1
010	011	011	1
011	100	100	1
100	101	101	1
101	110	110	1
110	111	111	1
RESET 111	111	000	0

(d) Logic diagram



(c) State Equation:

$$D_2(t+1) = D_2\overline{D_1} + D_2\overline{D_0} + \overline{D_2}D_1D_0 + D_2\overline{E} \quad (D_2D_1D_0\overline{E})$$

$$D_1(t+1) = D_1\overline{D_0} + \overline{D_1}D_0 + D_2D_0\overline{E} \quad (D_2D_1\overline{E}, \quad D_2D_1D_0\overline{E})$$

$$D_0(t+1) = \overline{D_0} + D_2D_1\overline{E} \quad (D_2D_1D_0\overline{E})$$

$$Z = \overline{D_2D_1D_0 + \overline{D_2}\,\overline{D_1}\,\overline{D_0}} = D_1\overline{D_0} + D_2\overline{D_1} + \overline{D_2}D_0 = \overline{D_1}D_0 + \overline{D_2}D_1 + D_2\overline{D_0}$$

Problems 4.22

4-22. +The sequence in Problem 4-21 is a flag used in a communication network that represents the beginning of a message. This flag must be unique. As a consequence, at most five 1s in sequence may appear anywhere else in the message. Since this is unrealistic for normal message content, a trick called zero insertion is used. The normal message, which can contain strings of 1s longer than 5, enters input X of a sequential zero-insertion circuit. The circuit has two outputs, Z and S. When a fifth 1 in sequence appears on X, a 0 is inserted in the stream of outputs appearing on Z and the output S = 1, indicating to the circuit supplying the zero-insertion circuit with inputs that it must stall and not apply a new in one clock cycle. This is necessary because the insertion o output sequence causes it to be longer than the input se ithout the stall. Zero insertion is illustrated by the following examp sequences:

Problem: Flage Sequence Generator

Sequence on X without any stalls:	011111 0011111 100001011110101
Sequence on X with stalls:	011111 1 0011111 1 100001011110101
Sequence on Z:	011111 | 0 0011111 | 0 100001011110101
Sequence on S:	000000 | 1 0000000 | 1 000001011110101

(a) Find the state diagram for the circuit.

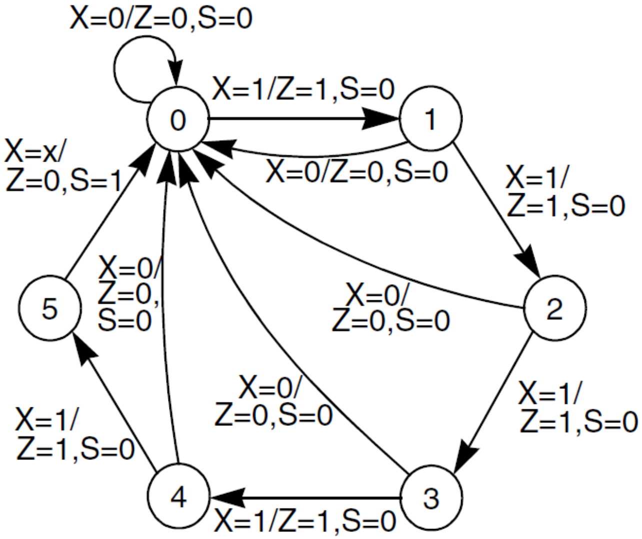
(b) Find the state table for the circuit and make a state assignment.

(c) Find an implementation of the circuit using *D* flip-flops and logic gates.

Solution 4.22-Continue before：

(a) State diagram

电路需要能够接受带有停顿的输入序列，出现连续5个“1”以后指示输入停止1位并插入1位“0”(接收时将忽略插入的“0”)，此时输出Z = 0，S = 1。因此状态图为：



(b) State table

Present state			Input	Next state			Output	
A	*B*	*C*	*X*	*A*	*B*	*C*	*Z*	*S*
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	1	0	1	0
0	1	0	0	0	0	0	0	0
0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	0	0
0	1	1	1	1	0	0	1	0
1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	1	0
1	0	1	0	0	0	0	0	1
1	0	1	1	0	0	0	0	1

(c) State Equation

$$D_A = A\overline{C}X + BCX$$

$$D_B = B\overline{C}X + \overline{A}\overline{B}CX$$

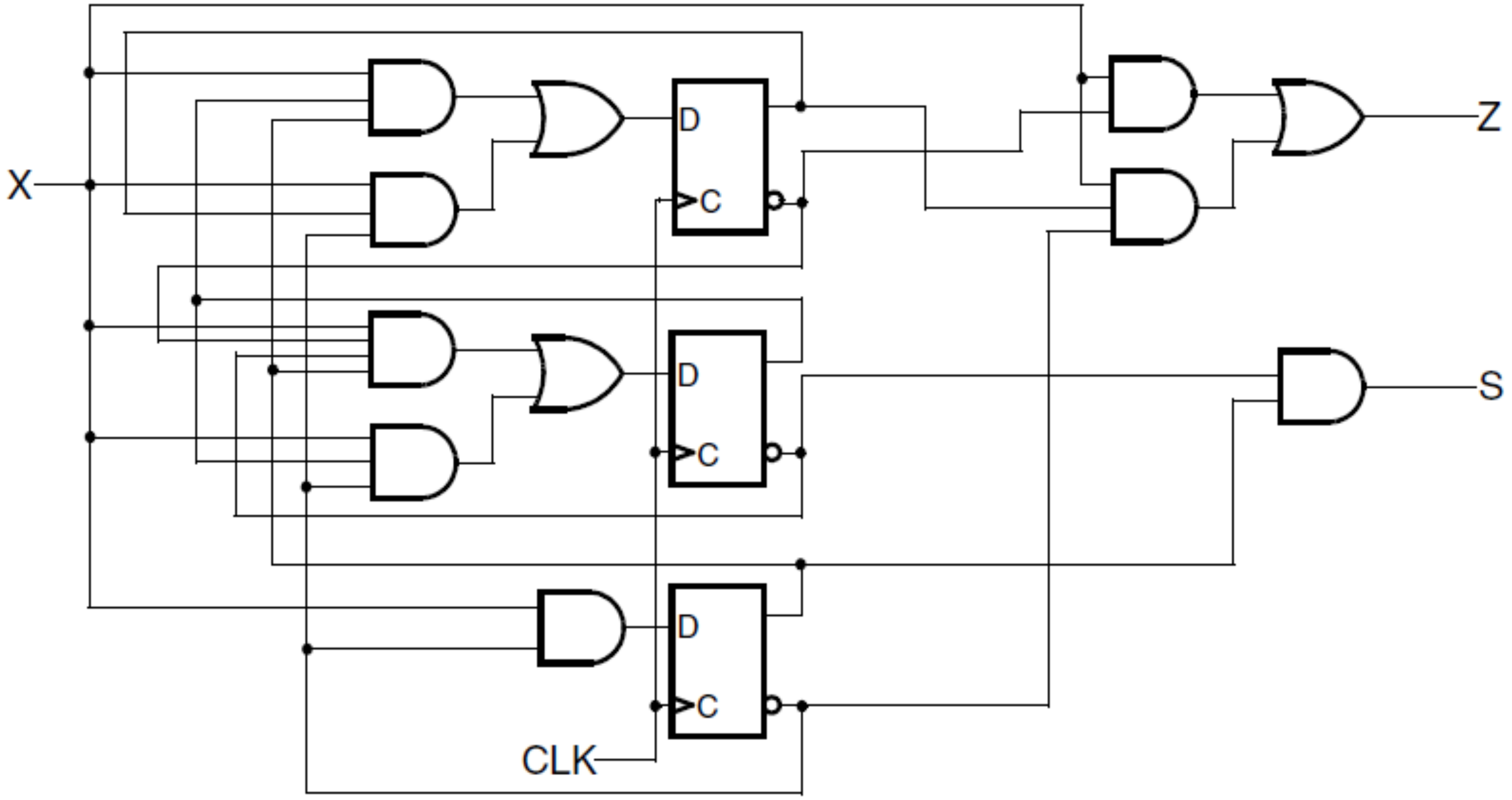
$$D_C = \overline{C}X$$

$$Z = \overline{A}X + \overline{C}X = (\overline{A} + \overline{C})X$$

$$S = AC$$

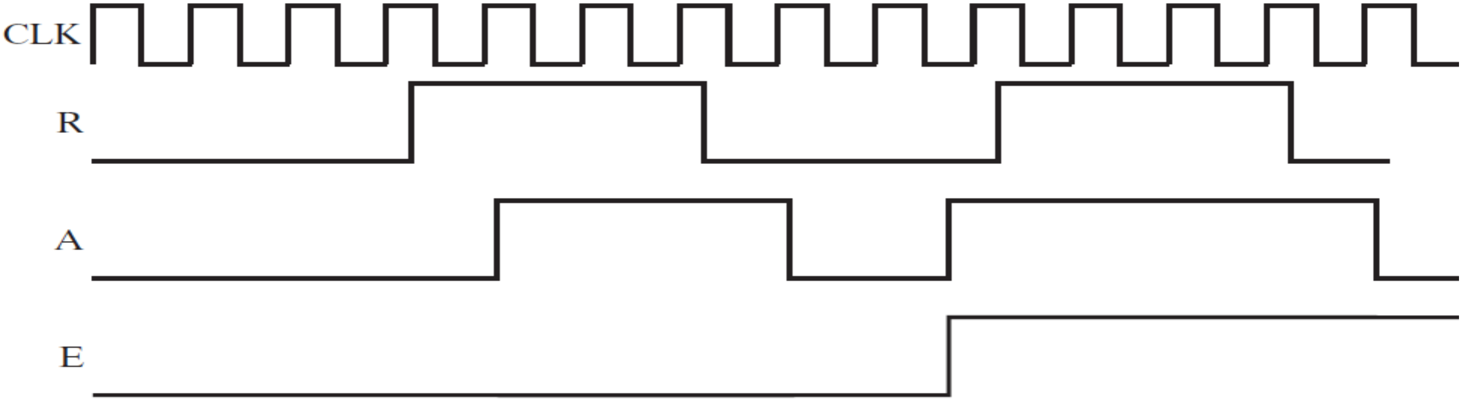
Solution 4.22-Continue before :

(d) Logic diagram



Problems 4.25

4-25. A pair of signals Request (*R*) and Acknowledge (*A*) is used to coordinate transactions between a CPU and its I/O system. The interaction of these signals is often referred to as a “handshake.” These signals are synchronous with the clock and, for a transaction, are to have their transitions always appear in the order shown in Figure 4-53. A handshake checker is to be designed that will verify the transition order. The checker has inputs, *R* and *A*, asynchronous reset signal, RESET, and output, Error (*E*). If the transitions in a handshake are in order, *E* = 0. If the transitions are out of order, then *E* becomes 1 and remains at 1 until the asynchronous reset signal (RESET = 1) is applied to the CPU.

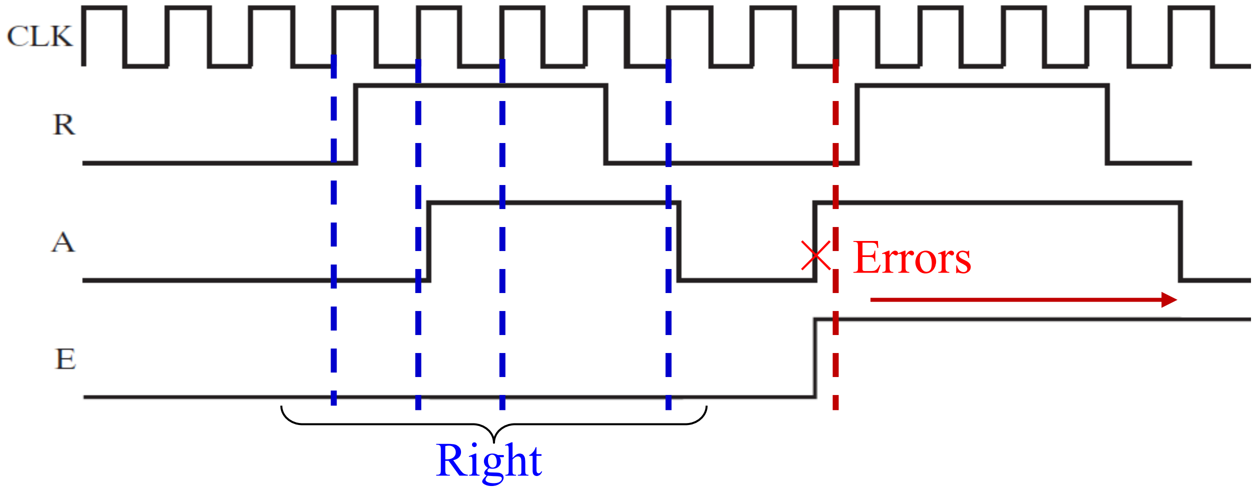


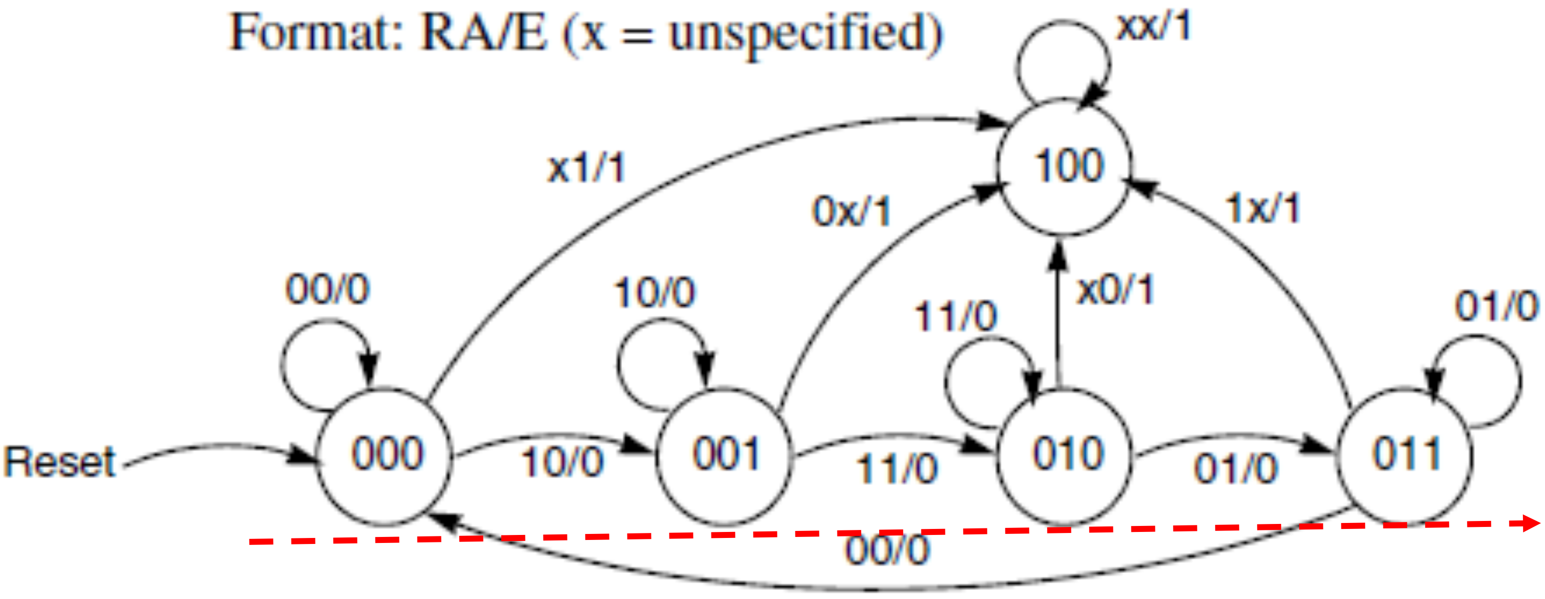
(a) Find the state diagram for the handshake checker.
(b) Find the state table for the handshake checker.

Solution 4.25-Continue before :

(a) State diagram

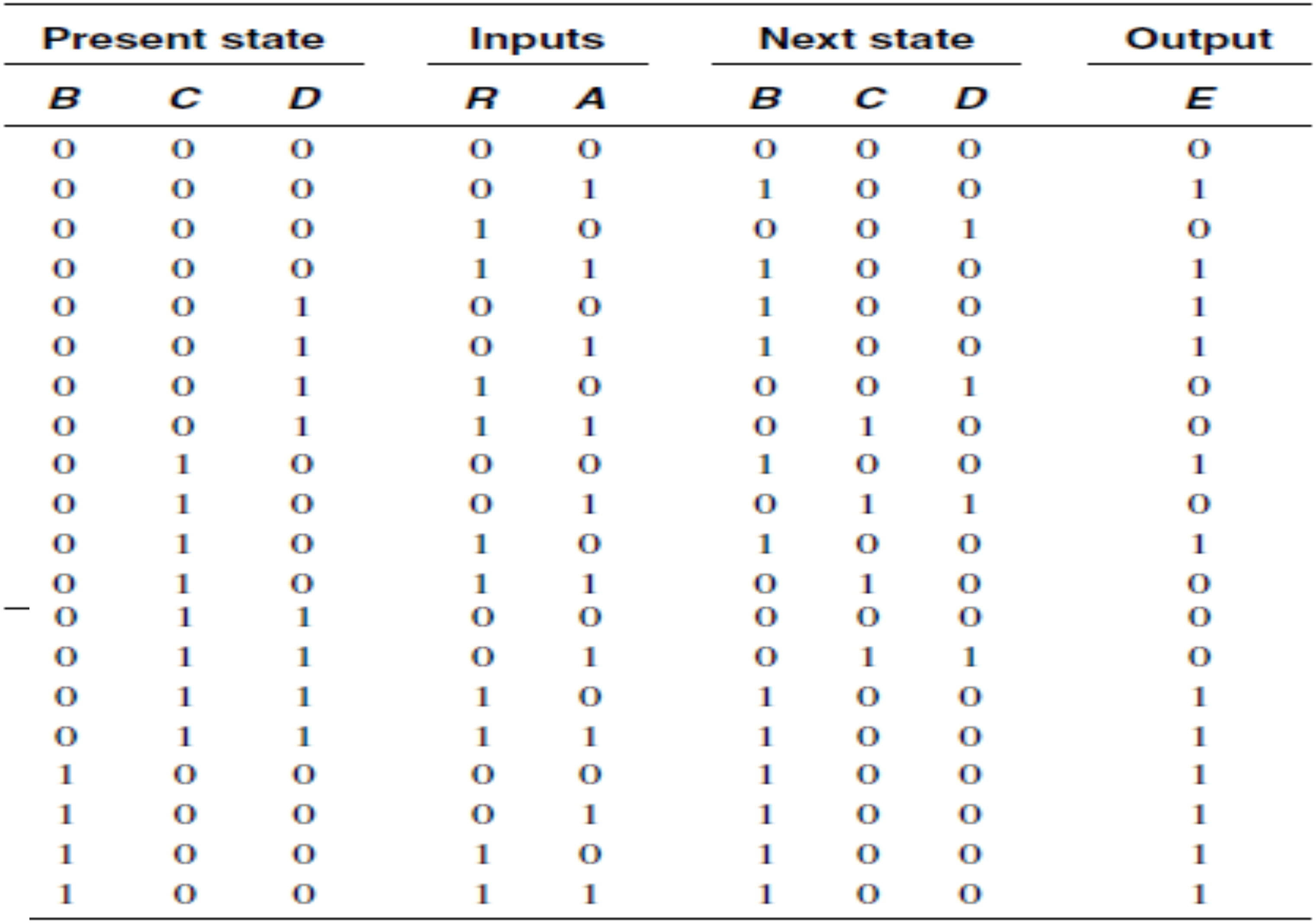
= 00 10 11 01 is right





Solution 4.25-Continue before :

(b) State table



Present state			Inputs		Next state			Output
B	*C*	*D*	*R*	*A*	*B*	*C*	*D*	*E*
0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1
0	0	1	0	0	1	0	0	1
0	0	1	0	1	1	0	0	1
0	0	1	1	0	0	0	1	0
0	0	1	1	1	0	1	0	0
0	1	0	0	0	1	0	0	1
0	1	0	0	1	0	1	1	0
0	1	0	1	0	1	0	0	1
0	1	0	1	1	0	1	0	0
0	1	1	0	0	0	0	0	0
0	1	1	0	1	0	1	1	0
0	1	1	1	0	1	0	0	1
0	1	1	1	1	1	0	0	1
1	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	1
1	0	0	1	0	1	0	0	1
1	0	0	1	1	1	0	0	1

Problems 4.29

4-29. +The state table for a 3-bit twisted ring counter is given in Table 4-15. This circuit has no inputs, and its outputs are the uncomplemented outputs of the flip-flops. Since it has no inputs, it simply goes from state to state whenever a clock pulse occurs. It has an asynchronous reset that initializes it to state 000.

(a) Design the circuit using D flip-flops and assuming the unspecified next states are don’t-care conditions.

(b) Add the necessary logic to the circuit to initialize it to state 000 on power-up master reset.

(c) In the subsection “Designing with Unused States” of Section 4-5, three techniques for dealing with situations in which a circuit accidentally enters an unused state are discussed. If the circuit you designed in parts (a) and (b) is used in a child’s toy, which of the three techniques given would you apply? Justify your decision.

Present State	Next State
ABC	ABC
000	100
100	110
110	111
111	011
011	001
001	000

010 ?

101 ?

(d) Based on your decision in part (c), redesign the circuit if necessary.

(e) Repeat part (c) for the case in which the circuit is used to control engines on a commercial airliner. Justify your decision.

(f) Repeat part (d) based on your decision in part (e).

Solution 4.29-Continue before：

(a) -State table with don't-care conditions

现态			次态			输出		
A	B	C	A	B	C	X	Y	Z
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1	0
0	1	1	0	0	1	0	1	1
1	0	0	1	1	0	1	0	0
1	0	1	0	1	0	1	0	1
1	1	0	1	1	1	1	1	0
1	1	1	0	1	1	1	1	1

(a)-State/input Equation &output Equation

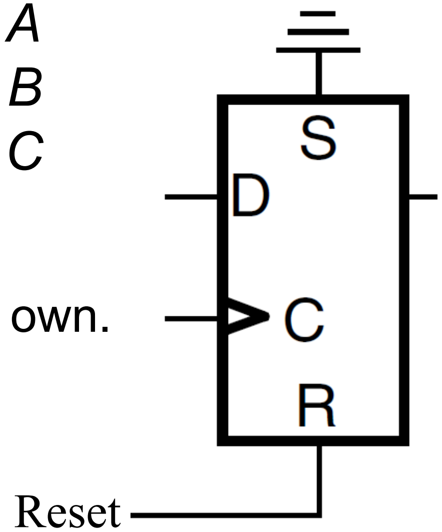
$D_A = \bar{C}$ $\qquad X = A$

$D_B = A$ $\qquad Y = B$

$D_C = B$ $\qquad Z = C$

(a)-Logic diagram

Skip， Draw your own.



(b) 将各触发器异步复位端(信号)配置为SR = 0 Reset。加电复位用异步更方便。

Clear A = Clear B = Clear C = Reset

(c~d) This circuit using the design in (a), does return from the invalid states to a valid state automatically after one or two clock periods. The circuit is suitable for child's toy, but not for life critical applications. In the case of the child's toy, it is the cheapest implementation. If an error occurs the child just needs to reset it.

Solution 4.29-Continue before :

(e~f) In life critical applications, the immediate detection of errors is critical. The circuit above enters invalid states for some errors.

For a life critical application, additional circuitry is needed for immediate detection of the error :

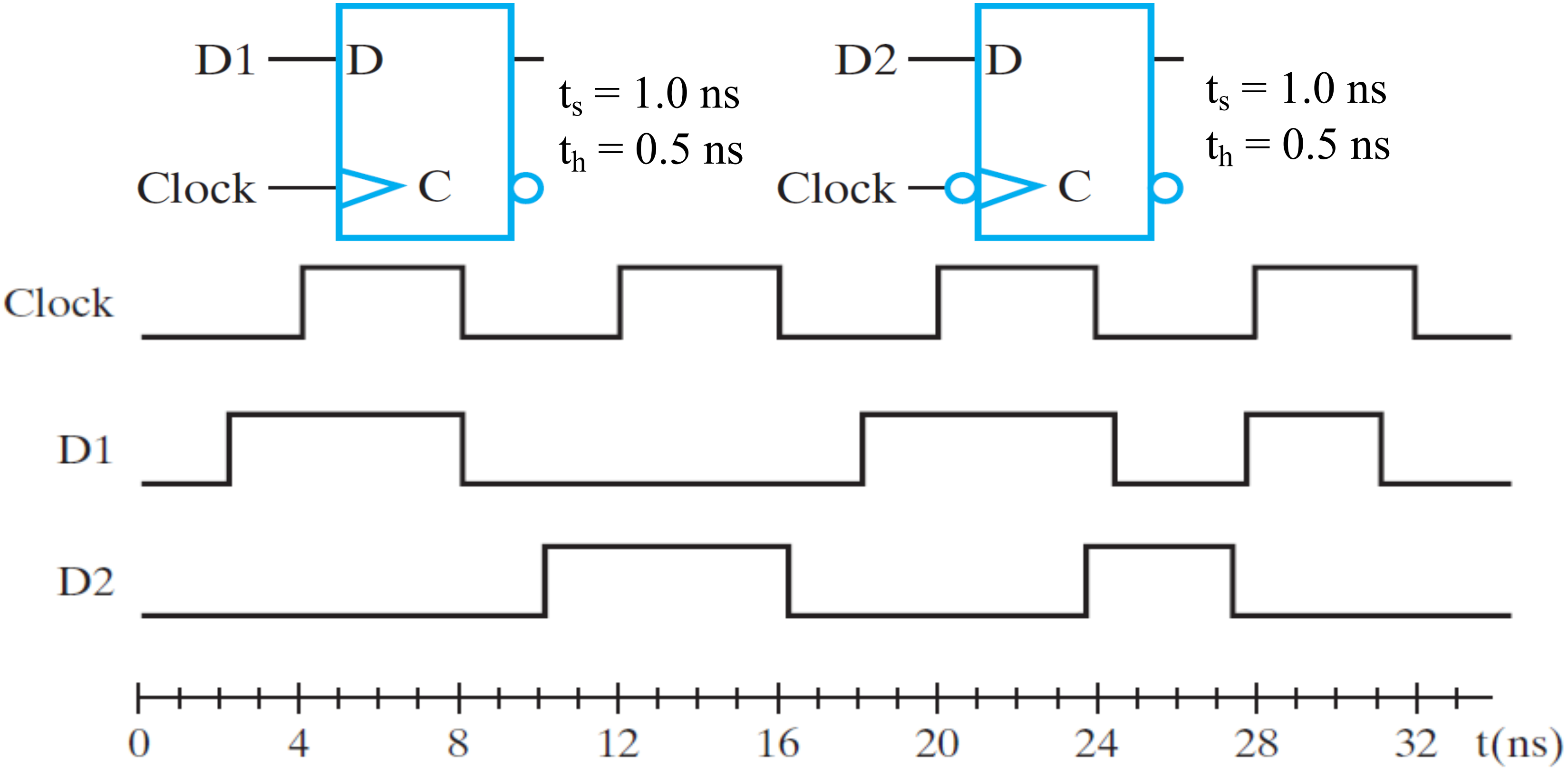
$$E = A\overline{B}C + \overline{A}B\overline{C}$$

Problems 4.58

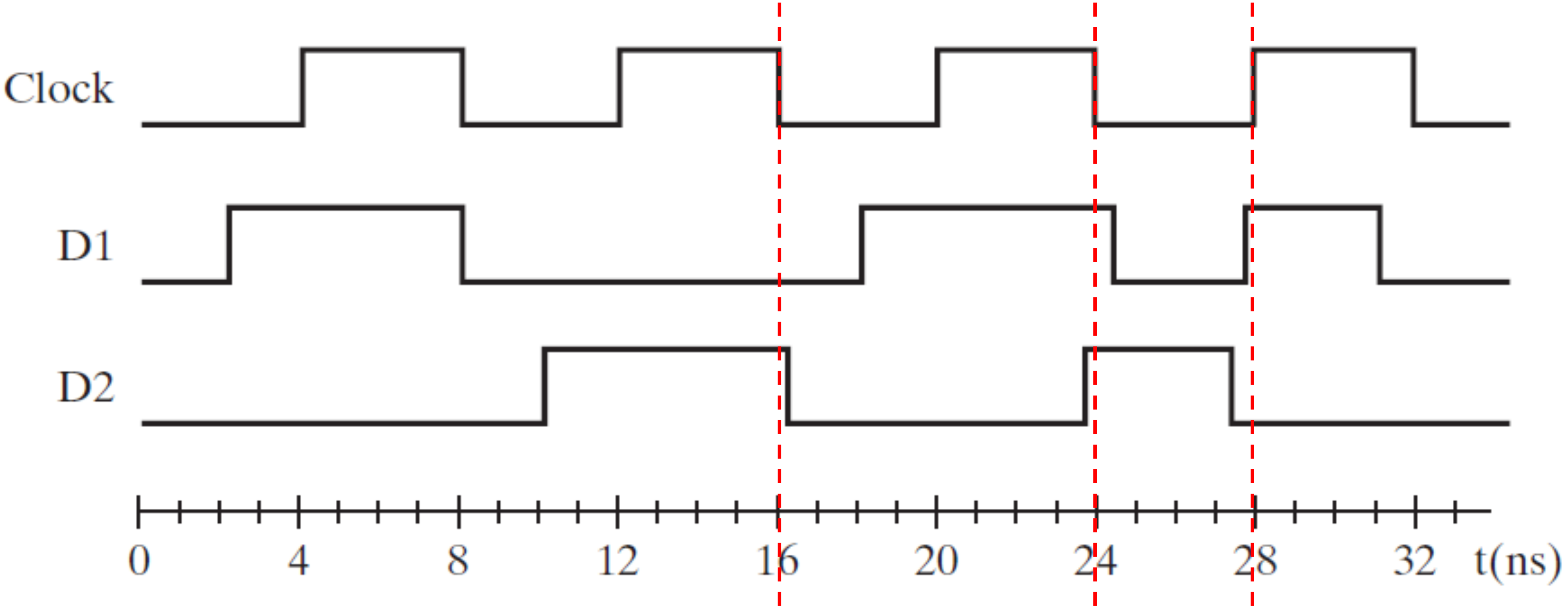
4-58. A set of waveforms applied to two D flip-flops is shown in Figure 4-56. These waveforms are applied to the flip-flops shown along with the values of their timing parameters.

(a) List the time(s) at which there are timing violations in signal D1 for flip-flop 2.

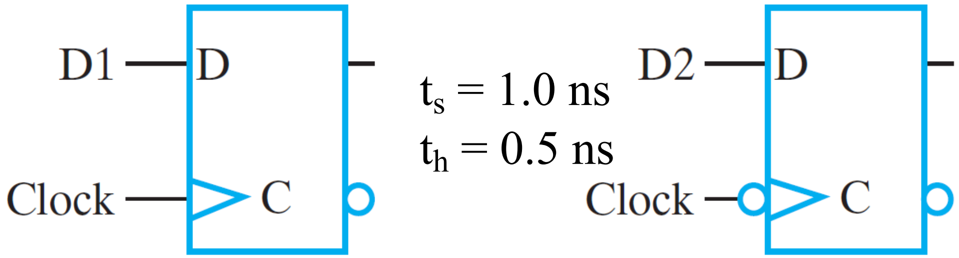
(b) List the time(s) at which there are timing violations in signal D2 for flip-flop 2.



Solution 4.58-Continue before :



(a)There is a setup time violation at 28ns near clock rising edge about D1.
(b)There is a hold time violation at 16ns and a setup time violation at 24ns for D2.



Problems 4.59

4-59. *A sequential circuit is shown in Figure 4-49. The timing parameters for the gates and flip-flops are as follows:

Inverter： $t_{pd} = 0.01$ ns; XOR gate： $t_{pd} = 0.04$ ns

Flip-flop： $t_{pd} = 0\ .08$ ns, $t_s = 0\ .02$ ns, and $t_h = 0\ .01$ ns

(a)Find the longest path delay from an external circuit input passing through gates only to an external circuit output.

(b) Find the longest path delay in the circuit from an external input to positive clock edge.

(c) Find the longest path delay from positive clock edge to output.

(d) Find the longest path delay from positive clock edge to positive clock edge.

(e) Determine the maximum frequency of operation of the circuit in megahertz (MHz).

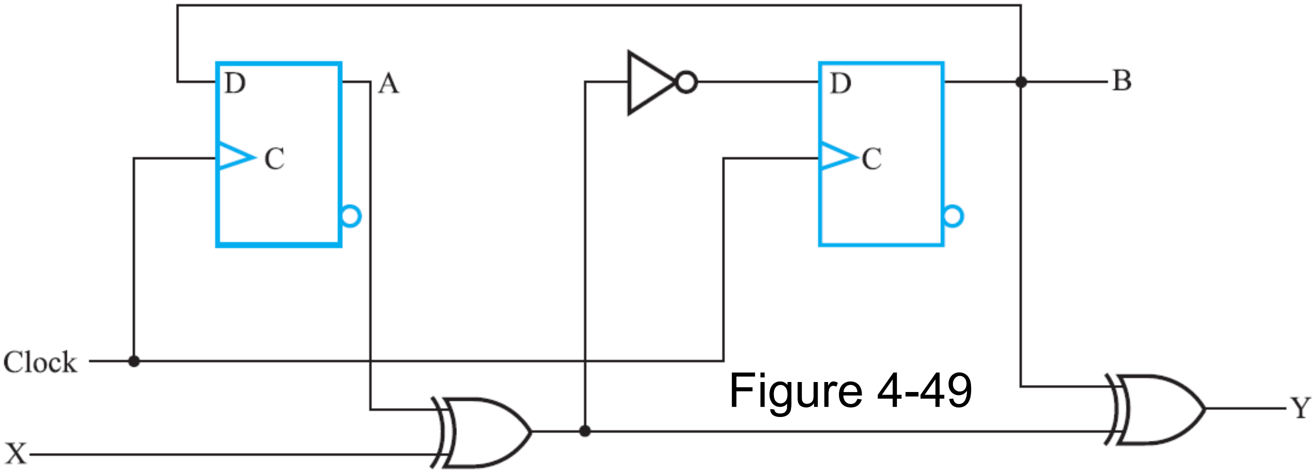
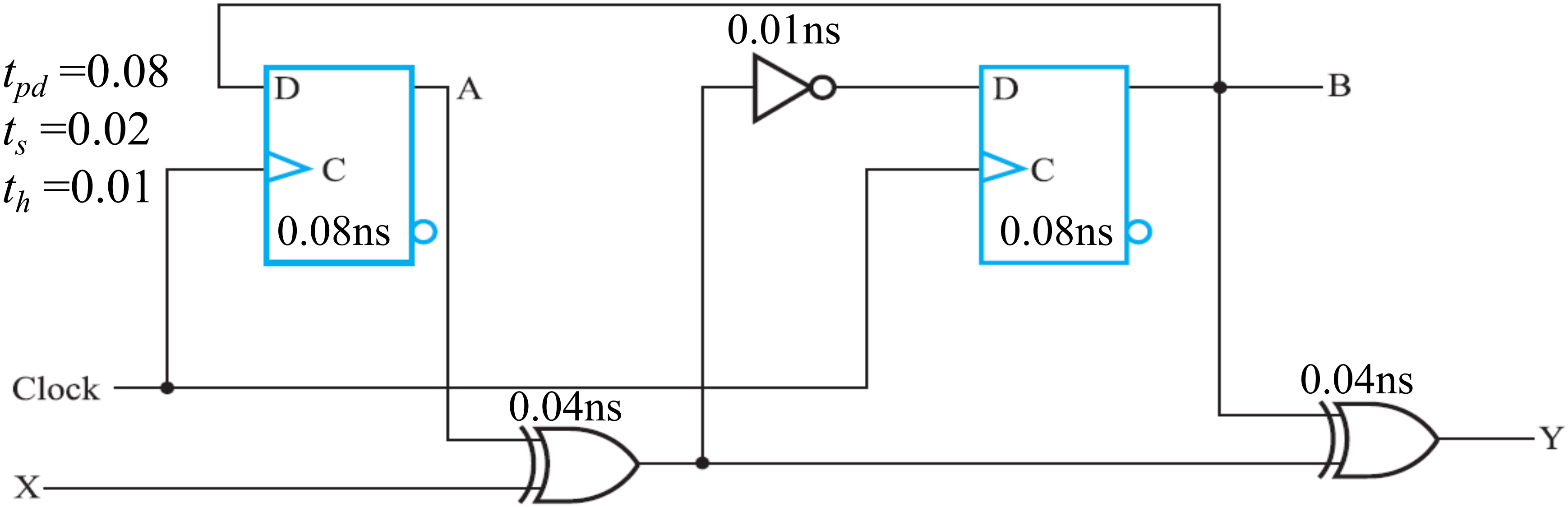


Figure 4-49

Solution 4.59



(a) $t_{delay} = t_{pdXOR} + t_{pdXOR} = 0.04 + 0.04 = 0.08$ ns

(b) $t_{delay} = t_{pdXOR} + t_{pdINV} + t_{sFF} = 0.04 + 0.01 + 0.02 = 0.07$ ns

(c) $t_{delay} = t_{pdFF} + 2\ t_{pdXOR} = 0.08 + 2(0.04) = 0.16$ ns

(d) $t_{delay\text{-clock edge to clock edge}} = t_{pdFF} + t_{pdXOR} + t_{pdINV} + t_{sFF}$
$= 0.08 + 0.04 + 0.01 + 0.02 = 0.15$ ns

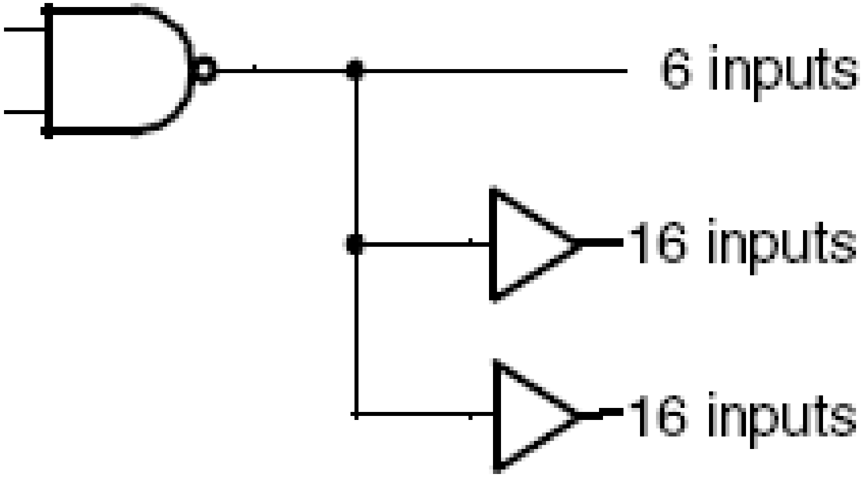
(e) $f_{max} = 1/t_{delay\text{- clock edge to clock edge}} = 1/0.15$ ns $= 6.67$ GHz

Comment: The clock frequency may need to be lower due to other delay paths that pass outside of the circuit into its environment. Calculation of this frequency cannot be performed in this case since data for paths through the environment is not provided.

Problems 5.3

5-3. An integrated circuit logic family has NAND gates with a fan-out of eight standard loads and buffers with a fan-out of 16 standard loads. Sketch a schematic showing how the output signal of a single NAND gate can be applied to 38 other gate inputs, using as few buffers as possible. Assume that each input is one standard load.

Solution 5.3:

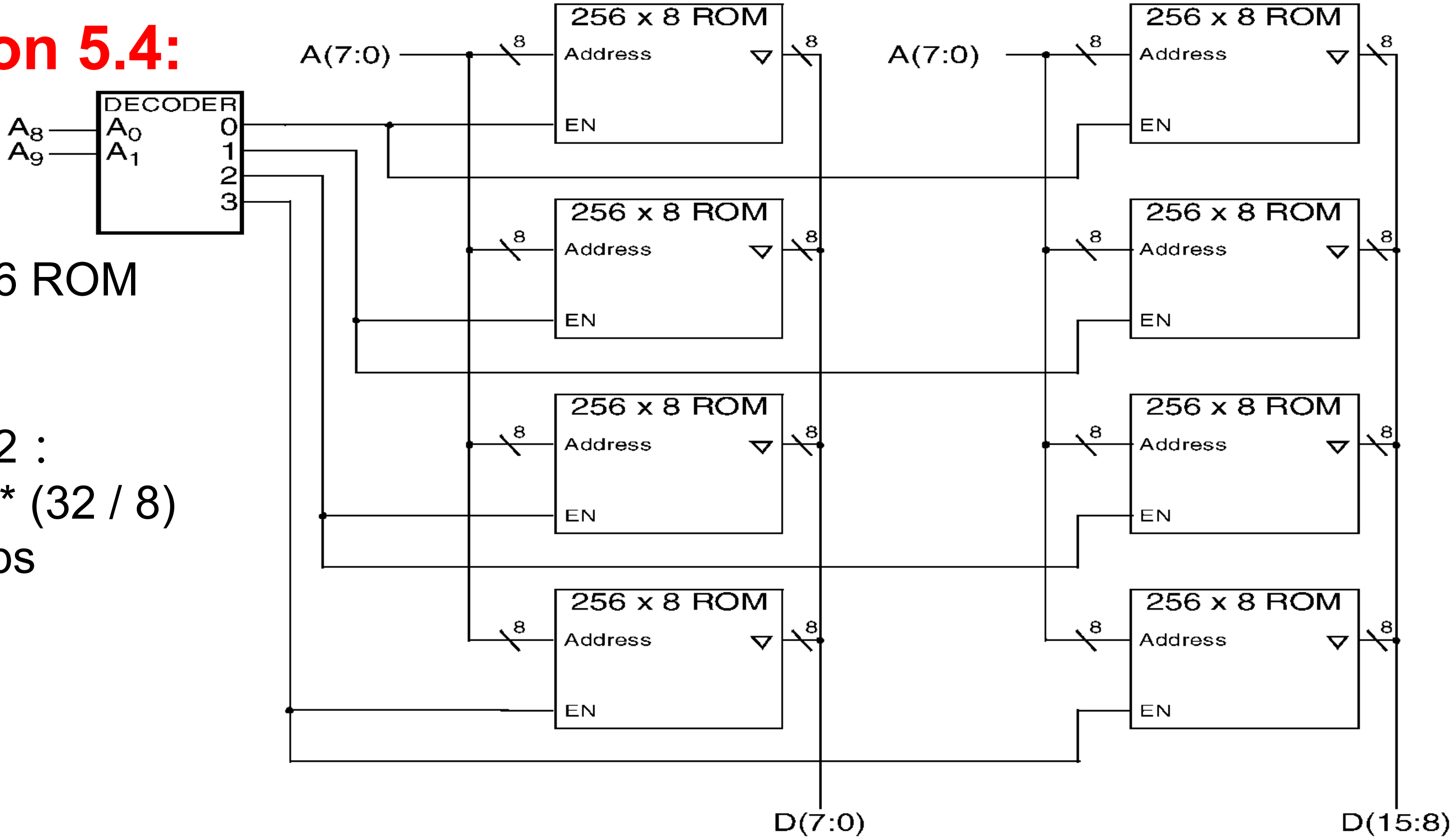


Problems 5.4

5-4. (a) Given a 256×8 ROM chip with an enable input, show the external connections necessary to construct a 1K×16 ROM with eight chips and a decoder.

(b) How many 256×8 ROM chips would be required to construct a 4K×32 ROM?

Solution 5.4:

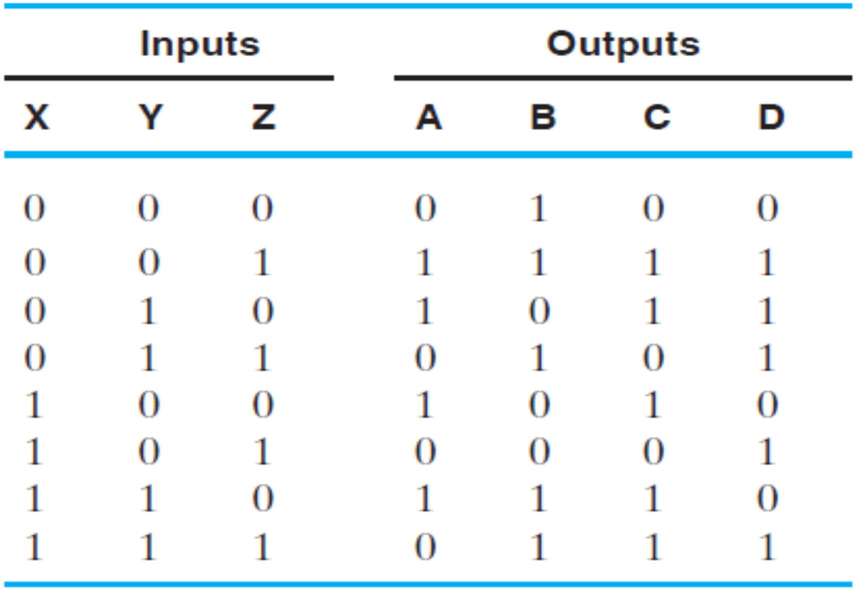


(a) 1K×16 ROM

(b) 4K×32 :
4K / 256 * (32 / 8)
= 64 Chips

Problems 5.12

5-12. The following is the truth table of a three-input, four-output combinational circuit. Obtain the equations for programming the PAL device shown in Figure 5-10.

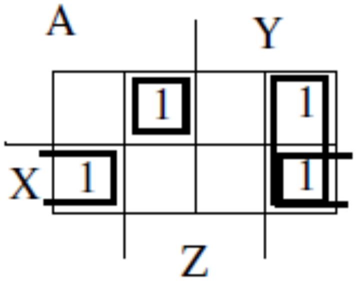


Inputs			Outputs			
X	Y	Z	A	B	C	D
0	0	0	0	1	0	0
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1

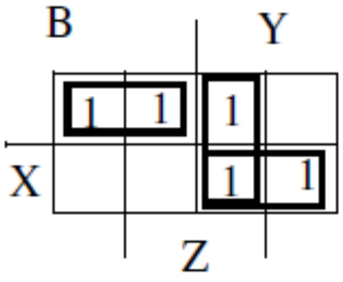
Solution 5.12:

Figure 5-10 uses 3-input OR gates :

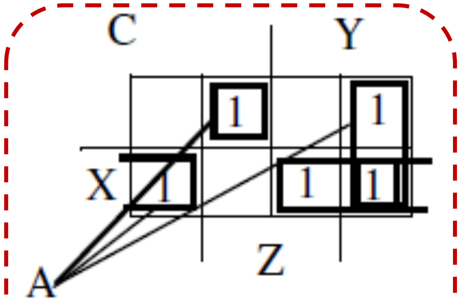
A, B, and D each require three or fewer product terms so can be implemented with 3-input OR gates.



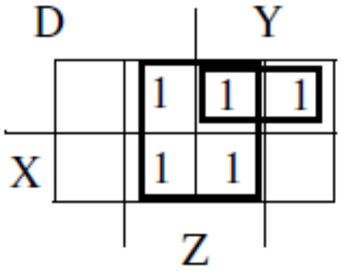
$A = X\bar{Z} + Y\bar{Z} + \bar{X}\,\bar{Y}Z$



$B = XY + YZ + \bar{X}\,\bar{Y}$



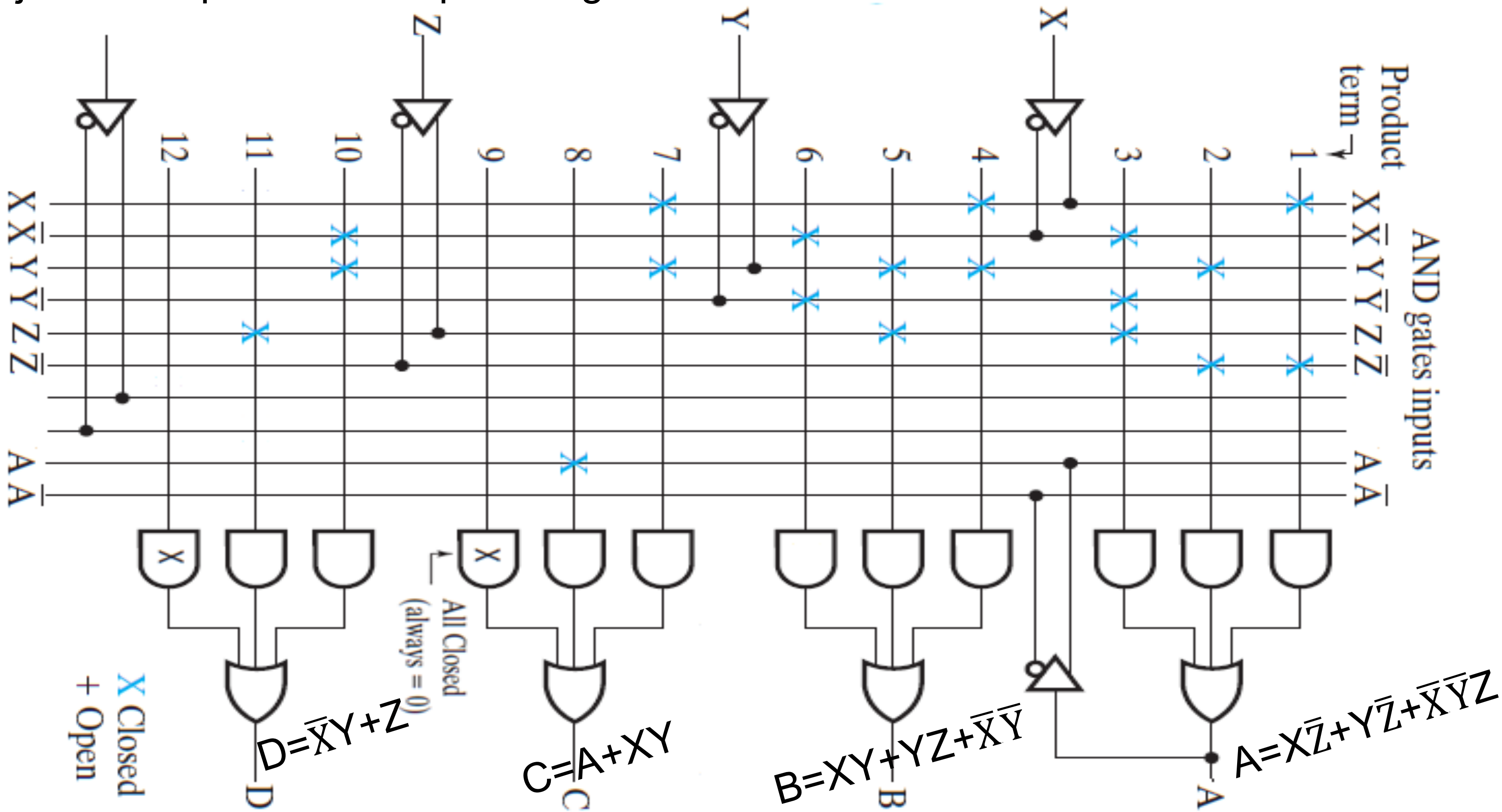
$C = A + XY$



$D = \bar{X}Y + Z$

Solution 5.12-Continue before :

C requires four terms so cannot be implemented with a 3-input OR gate. But because the first PAL device output can used as an input to implement other functions it can be assigned to A and A can then be used to implement C using just two inputs of a 3-input OR gate.


Z
Y
X
Product term
12
11
10
9
8
7
6
5
4
3
2
1
AND gates inputs
$X \; \overline{X} \; Y \; \overline{Y} \; Z \; \overline{Z}$
$A \; \overline{A}$
All Closed (always = 0)
X Closed
+ Open
$D=\overline{X}Y+Z$
$C=A+XY$
$B=XY+YZ+\overline{X}\,\overline{Y}$
$A=X\overline{Z}+Y\overline{Z}+\overline{X}\,\overline{Y}Z$
D
C
B
A

Problems 6.6

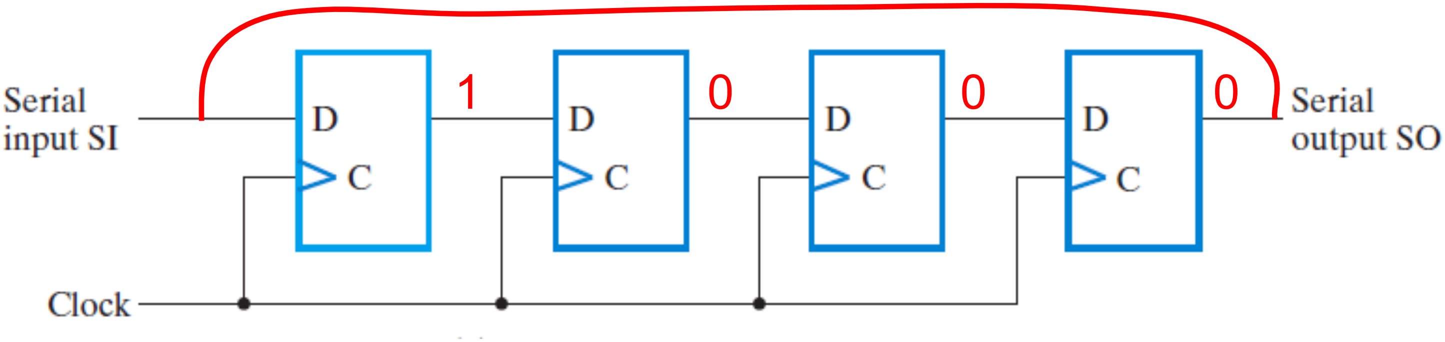
6-6. *A ring counter is a shift register, as in Figure 6-9, with the serial output connected to the serial input.

(a) Starting from an initial state of , list the sequence of states of the four flip-flops after each shift.

(b) Beginning in state 10…0, how many states are there in the count sequence of an *n*-bit ring counter?

Solution 6.6:

The circuit diagram of the ring counter is as follows :



(a) 1000 0100 0010 0001 1000 ……

(b) States number = n

Problems 6.13

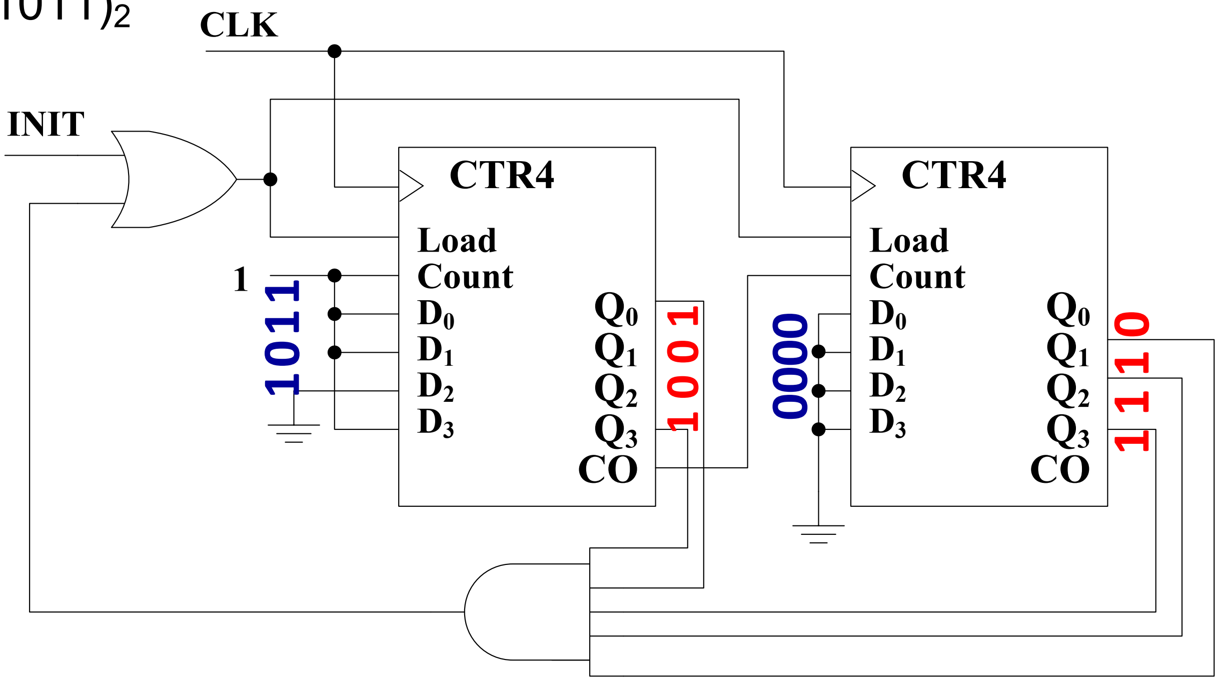
6-13. Using two binary counters of the type shown in Figure 6-14 and logic gates, construct a binary counter that counts from decimal 11 through decimal 233. Also, add an additional input and logic to the counter to initialize it synchronously to 11 when the signal INIT is 1.

Solution 6.13:

Counts from decimal : 11→ $(233)_{10}$ ==> $(1011)_2$ → $(1110\ 1001)_2 = (E9)_{16}$

INIT =1： Counter = $(1011)_2$

Decimal To Carry 9



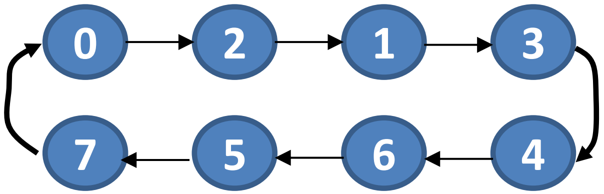
Counter = E9, Load **0000 1011**

Problems 6.16

6-16. Use *D*-type flip-flops and gates to design a counter with the following repeated binary sequence: 0, 2, 1, 3, 4, 6, 5, 7.

Solution 6.16:

(a) State diagram：



(b) State Table：

Present state	Nest state
A B C	A B C
0 0 0	0 1 0
0 0 1	0 1 1
0 1 0	0 0 1
0 1 1	1 0 0
1 0 0	1 1 0
1 0 1	1 1 1
1 1 0	1 0 1
1 1 1	0 0 0

(c) State equation and input equation

$$D_A = A_{t+1} = A\overline{B} + A\overline{C} + \bar{A}BC$$

$$D_B = B_{t+1} = \overline{B}$$

$$D_C = C_{t+1} = \overline{B}C + B\overline{C} = B \oplus C$$

(d) The circuit diagram

Skip， Do it by yourself!

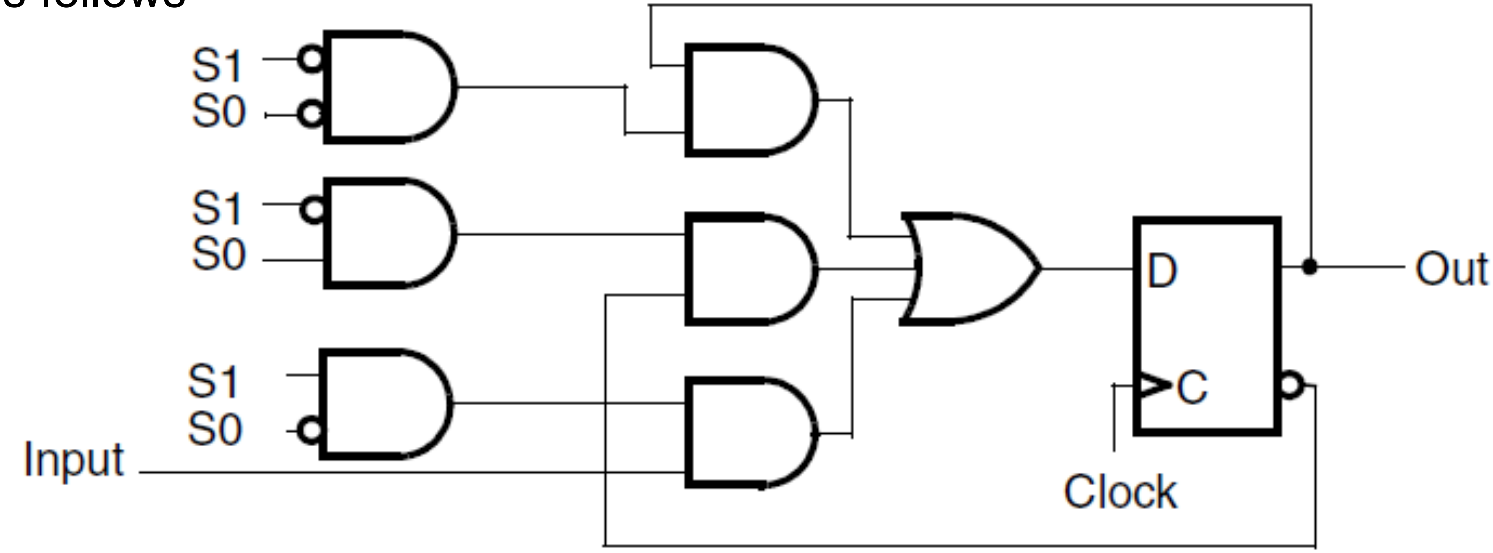
Problems 6.17

6-17. Draw the logic diagram of a 4-bit register with mode selection inputs *S*1 and *S*0. The register is to be operated according to the function table below.

S_1	S_0	Register Operation
0	0	No change
0	1	Complement output
1	0	Load parallel data
1	1	Clear register to 0

Solution 6.17:

"与"门使能直接构成多路选择器，选择控制是S_1S_0： The basic cell of the register is as follows



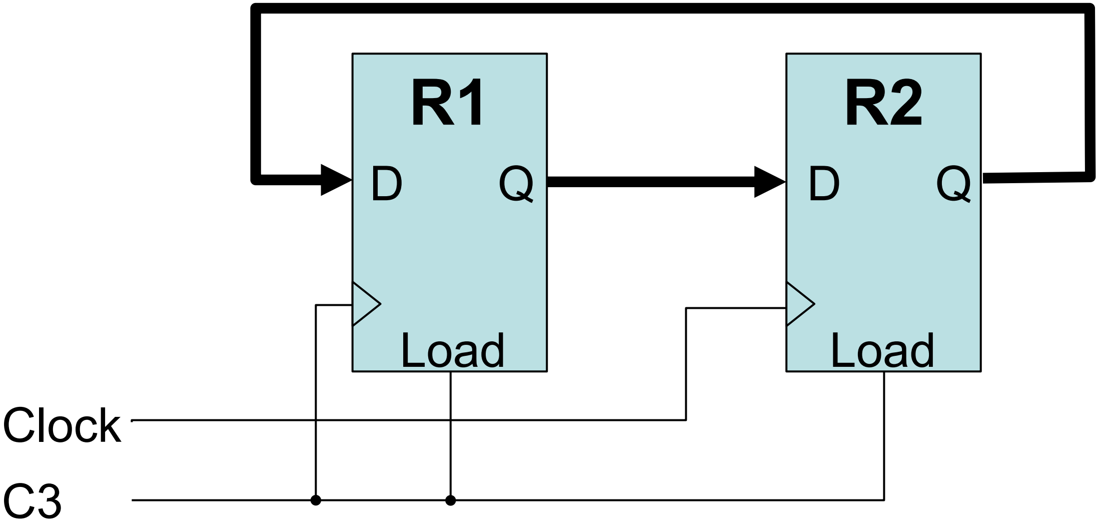
Problems 6.19

6-19. *Show the diagram of the hardware that implements the register transfer statement：

$C3\text{: } R2 \leftarrow \mathrm{R1},\ R1 \leftarrow R2$

Solution 6.19:

典型的寄存器传输，C3同时控制寄存器Load加载信号：



Problems 6.23

6-23. A register cell is to be designed for an 8-bit register *A* that has the following register transfer functions:

$$C0: A \leftarrow A \cap B,\ C1: A \leftarrow A \cup \overline{B}$$

Find optimum logic using AND, OR, and NOT gates for the *D* input to the *D* flip-flop in the cell.

Solution 6.23:

典型寄存器传输操作控制：

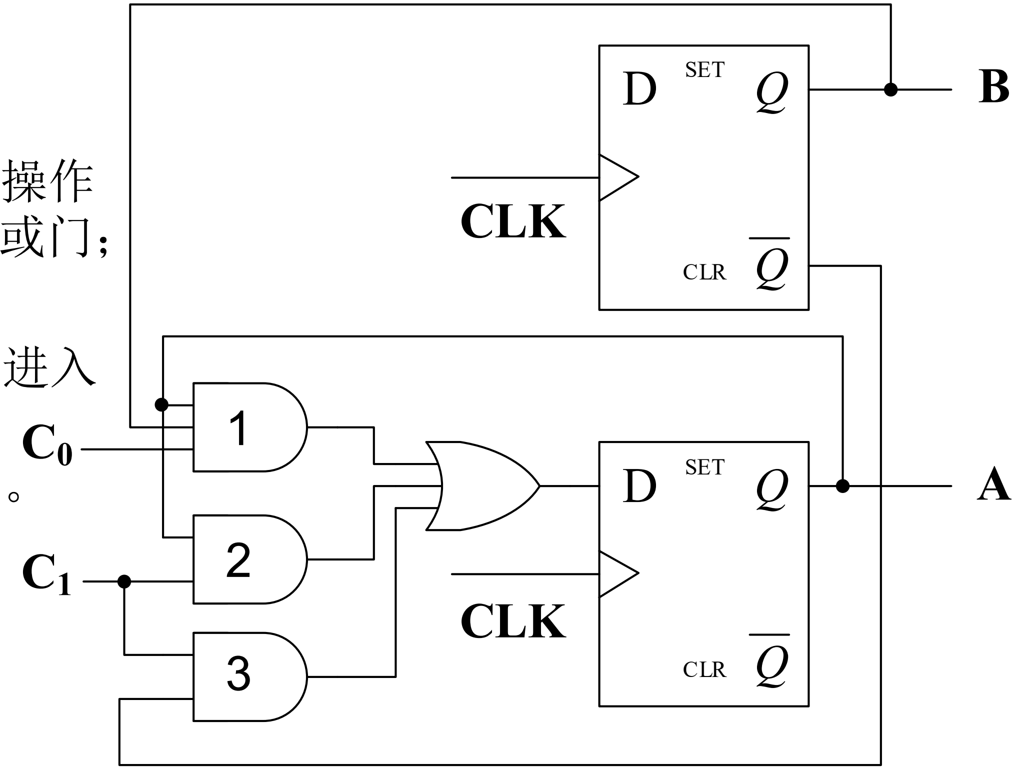
寄存器传输 + 微操作

C_0 =1时，与门1使能$A \cap B$ 传输进或门；

C_0 =0时， $A \cap B$ 传输禁止传输。

C_1 =1时，与门2~3使能 A、$\overline{B}$传输进入或门运算$A \cup \overline{B}$ ；

C_1 =0时，与门2~3禁止 A、$\overline{B}$传输。



问：此电路设计合理吗？

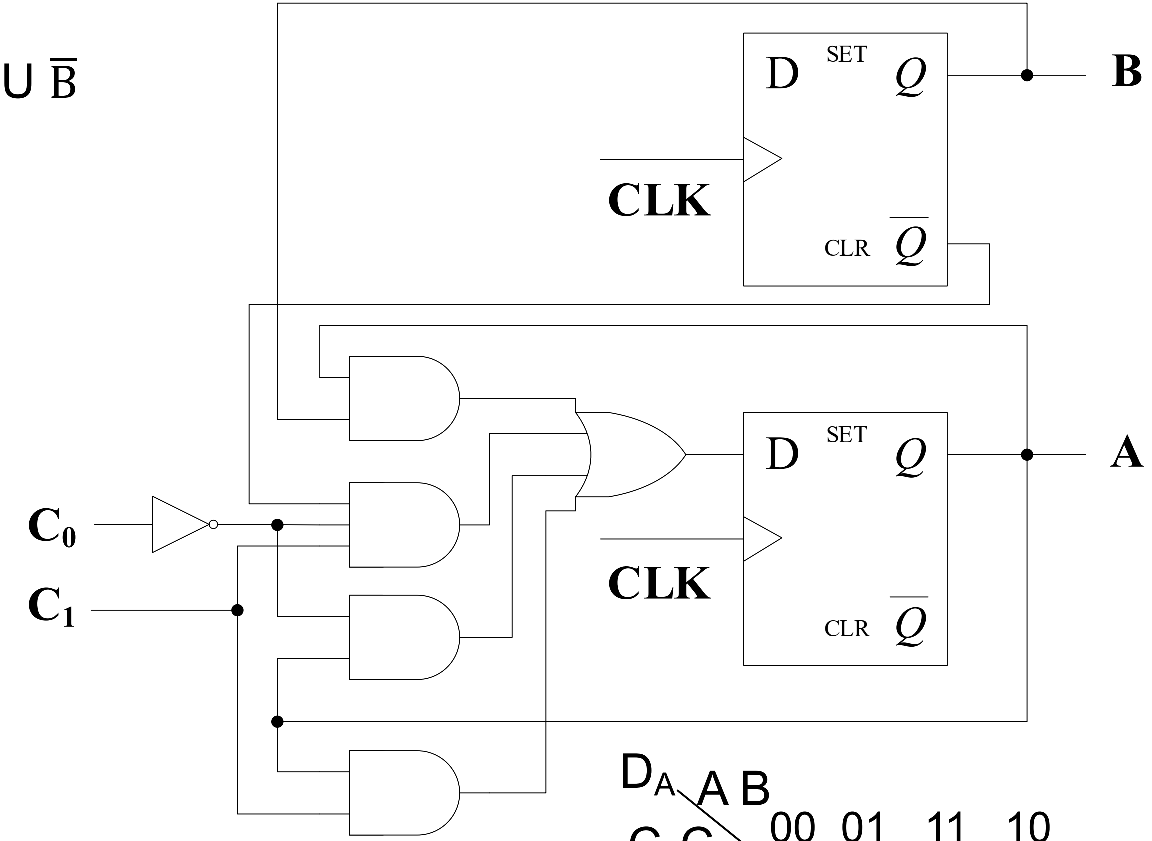
提示：C_0C_1 = 11，？？

C_0C_1 = 00，？？

改进

$C0: A \leftarrow A \cap B,\ C1: A \leftarrow A \cup \overline{B}$

C_0 C_1	A B	$A_{(t+1)}$
0 0	0 0	0
0 0	0 1	0
0 0	1 0	1
0 0	1 1	1
0 1	0 0	1
0 1	0 1	0
0 1	1 0	1
0 1	1 1	1
1 0	0 0	0
1 0	0 1	0
1 0	1 0	0
1 0	1 1	1
1 1	0 0	0
1 1	0 1	0
1 1	1 0	1
1 1	1 1	1



D_A / C_0C_1 \ AB	00	01	11	10	
00	0	0	1	1	$A\,\overline{C}_0$
$\overline{C}_0C_1\overline{B}$ 01	1	0	1	1	
11	0	0	1	1	$A\,C_1$
10	0	0	1	0	

A B

$D_A = AB + \overline{C}_0C_1\overline{B} + C_1A + \overline{C}_0A$

Problems 6.27

6-27. *Two register transfer statements are given (otherwise, R1 is unchanged):

$C1: R1 \leftarrow R1 + R2$ Add R2 to R1

$\overline{C1}\, C2: R1 \leftarrow R1 + 1$ Increment R1

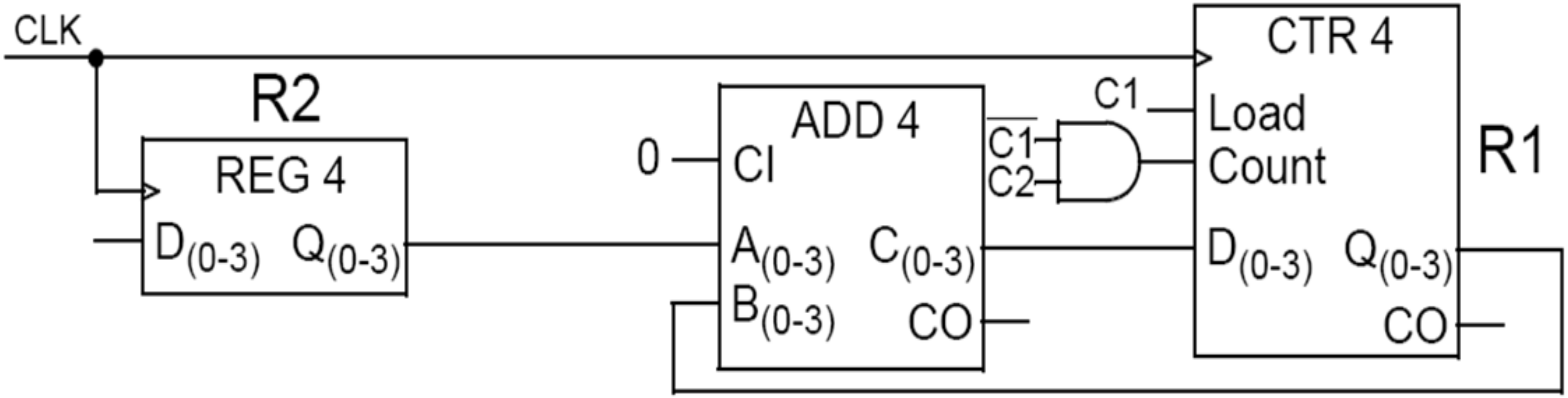
(a) Using a 4-bit counter with parallel load as in Figure 6-14 and a 4-bit adder as in Figure 4-5, draw the logic diagram that implements these register transfers.

(b) Repeat part (a) using a 4-bit adder as in Figure 3-43 plus external gates as needed. Compare with the implementation in part (a).

Solution 6.27:

(a) $\overline{C1}$ C2=11 时，Load=C1=0，禁止加载，CRT4计数，实现Increment R1；

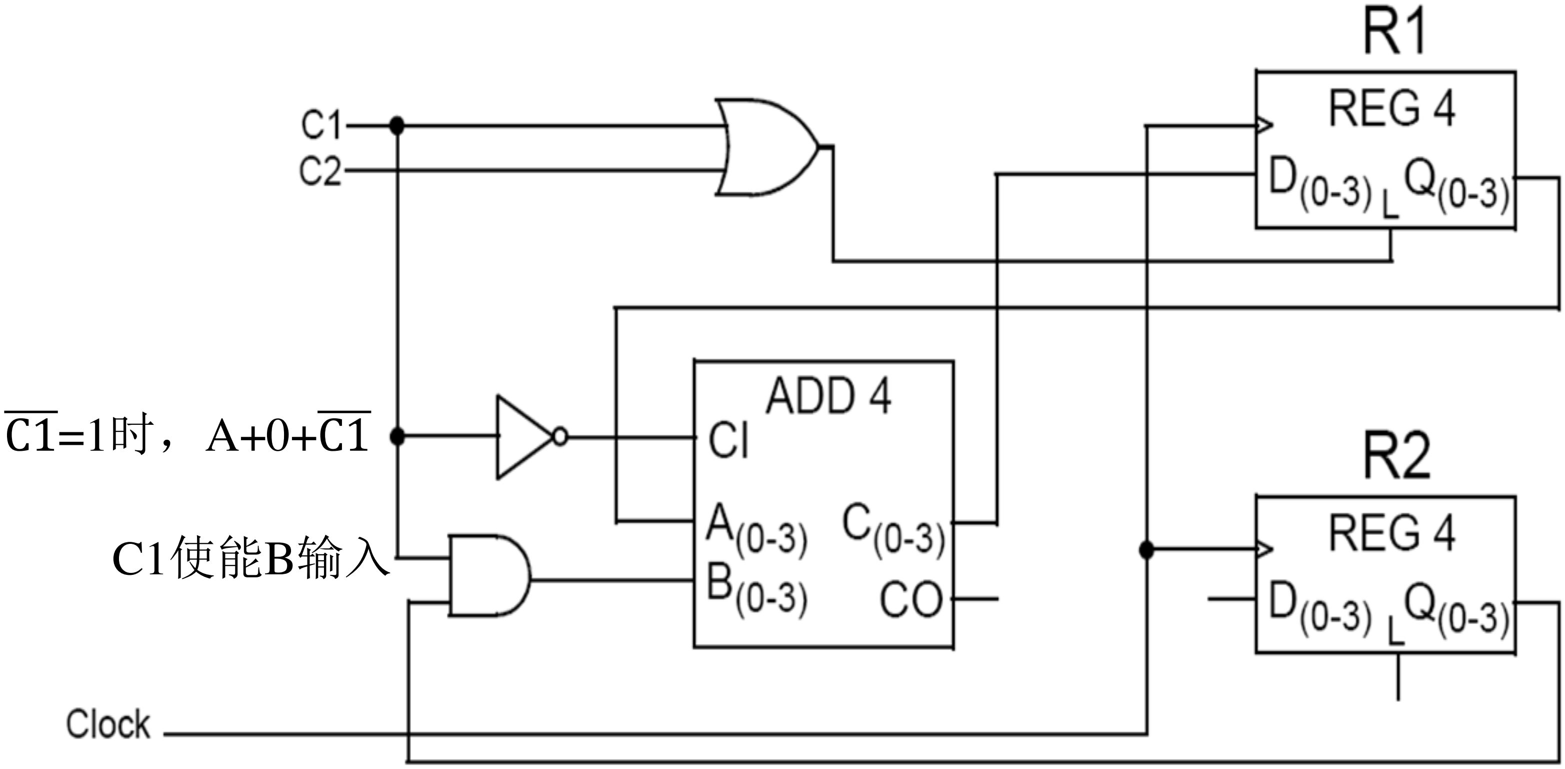
C1C2 =1x时，CRT4计数禁止，Load=C1=1，加载ADD4输出，实现Add R2 to R1。



Solution 6.27-Continue before：

(b) Repeat part (a) using a 4-bit adder as in Figure 3-43 plus external gates as needed. Compare with the implementation in part (a).

$\overline{C1}$ C2=11 时，ADD4输入：CI=$\overline{C1}$=1，B被与门禁止输入，此时实现 R1 ← R1+1；
C1C2 =1x时， ADD4输入：CI=$\overline{C1}$=0，与门使能B正常输入，实现R1 ← R1+1R2。



$\overline{C1}$=1时，A+0+$\overline{C1}$

C1使能B输入

Problems 6.34

6-34. *The content of a 4-bit register is initially 0101. The register is shifted eight times to the right, with the sequence 10110001 as the serial input. The leftmost bit of the sequence is applied first. What is the content of the register after each shift?

Solution 6.34:

The leftmost bit of the sequence is applied first： 10110001 → 10001101

10001101-0101 → 1000110-1010 → 100011-0101 →

10001-1010 → 1000-1101 →

100-0110 → 10-0011 →

1-0001 → 1000

Problems 7.1

7-1. *The following memories are specified by the number of words times the number of bits per word. How many address lines and input–output data lines are needed in each case? (a) 48K×8, (b) 512K×32, (c) 64M×64, and (d) 2G×1.

Solution 7.1:

No	Problem	Address lines	I-O data lines
(a)	48 k×8	$2^{5+10}=32K < 2^{6+10}=64K \rightarrow 15$	8
(b)	512 k×32	$2^{9+10}=512K \rightarrow 29$	32
(c)	64M k×64	$2^{6+20}=64M \rightarrow 26$	64
(d)	2G×1	$2^{1+30}=2G \rightarrow 31$	1

Problems 7.4

7-4. Assume that the largest decoder that can be used in an m × 1 RAM chip has 14 address inputs and that coincident decoding is employed. In order to construct RAM chips that contain more one-bit words than m, multiple RAM cell arrays, each with decoders and read/write circuits, are included in the chip.

(a) With the decoder restrictions given, how many RAM cell arrays are required to construct a 2G ×1 RAM chip?

(b) Show the decoder required to select from among the different RAM arrays in the chip and its connections to address bits and cell array select (CS) bits.

Solution 7.4:

(a) Due to the row and column decodes, and each decoder has 14 address inputs, Basic cell RAM has 28bit address range.

$$m = 2^{14} \times 2^{14} = 2^{28} = 256\text{M}$$

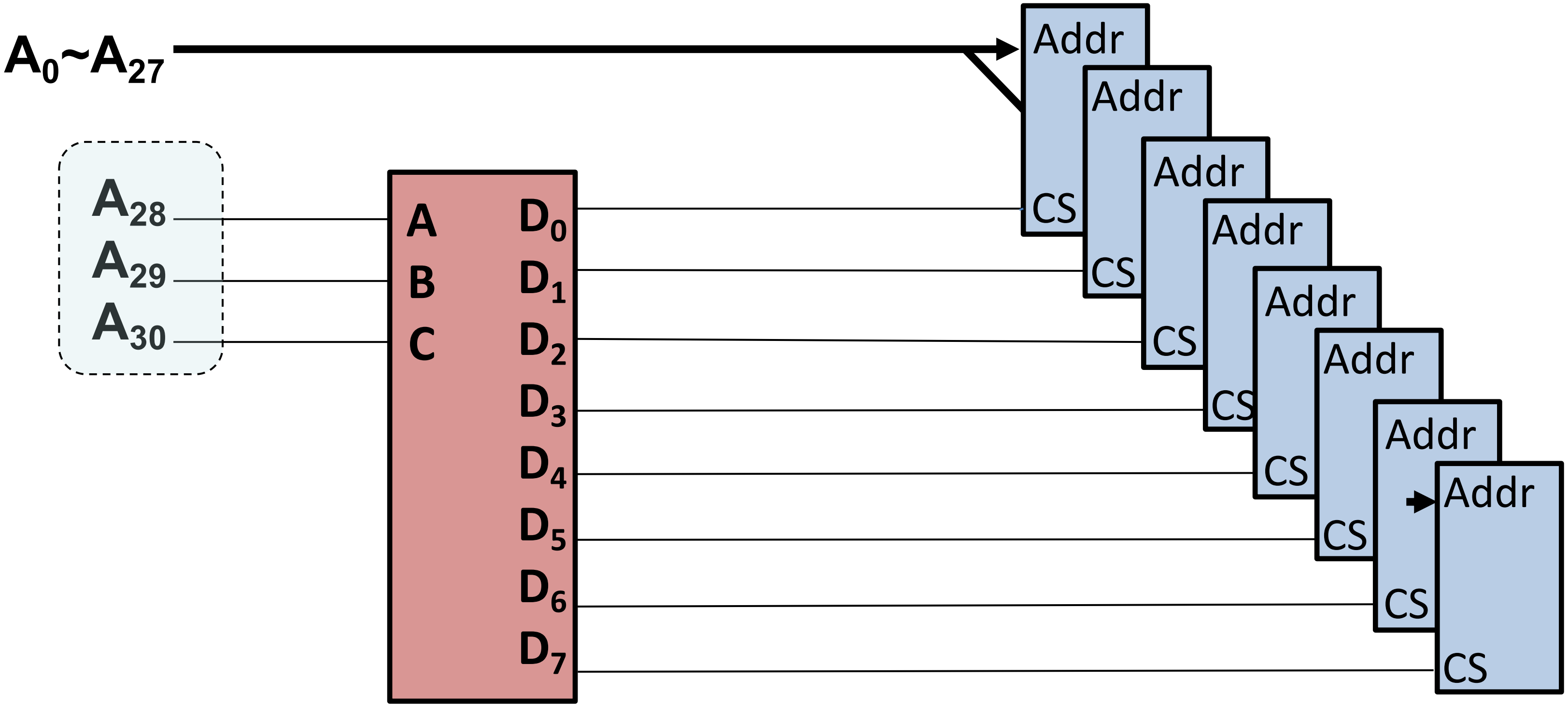
$$\text{RAM chip} = 2\text{G}(2^{31}) \div 256\text{M}(2^{28}) = 2^3 = 8$$

Solution 7.4-Continue before :

(b) Show the decoder required to select from among the different RAM arrays in the chip and its connections to address bits and cell array select (CS) bits.

For construct a 2G×1 RAM, there are 8 chips, to be required a 3-8 decoder

Logic connecting Diagram is as follows :



Problems 7.5

7-5. A DRAM has 15 address pins and its row address is 1 bit longer than its column address. How many addresses, total, does the DRAM have?

Solution 7.5:

- ● DRAM has 15 address pins
- ● ROW address = Column address + 1

⊙ To maximize design, Column address is given 15-1=14 bit

Then may be：ROW address = 15 bit

DRAM have 15+14= 29 bit addresses

DRAM addresses range = 2^{29} =512M

Problems 7.8

7-8. *(a) How many 128K × 16 RAM chips are needed to provide a memory capacity of 2 MB?

(b) How many address lines are required to access 2 MB? How many of these lines are connected to the address inputs of all chips?

(c) How many lines must be decoded to produce the chip select inputs? Specify the size of the decoder.

Solution 7.8:

(a) Chips for 2 MB = 2MB÷(128K×2)=2^{21}÷2^{17+1}= 2^3=8

(b) Address lines to access 2 MB = ($\log_2 2^{21}$) = 21

lines for basic chips = ($\log_2$ 128K×2)=$\log_2 2^{18}$= 18 A_0~A_{17}

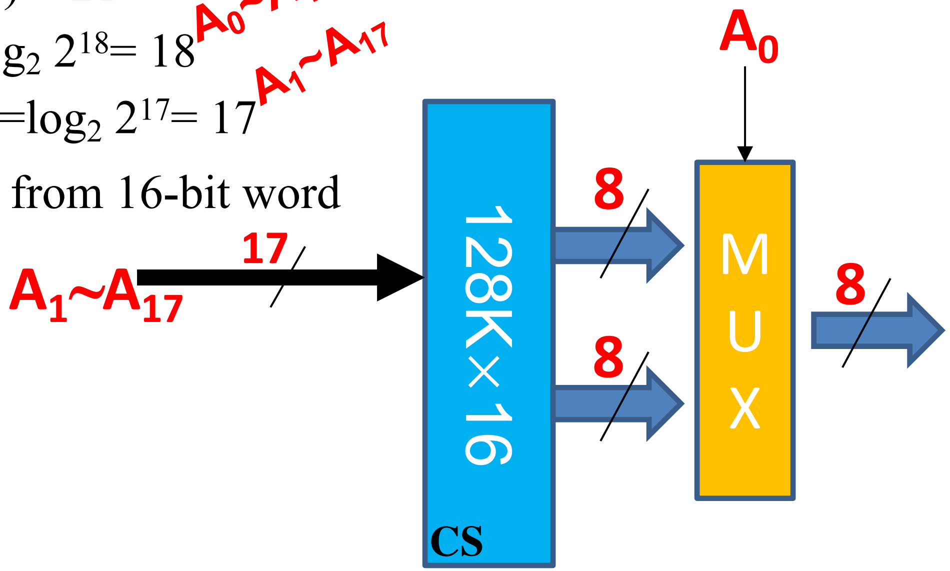
lines are input to all chips = ($\log_2$ 128K)=$\log_2 2^{17}$= 17 A_1~A_{17}

another one address line A_0 is byte select from 16-bit word

(c) Address lines for CS = 21 - 18 = 3:

CS: A_{20}~A_{18}

size of the decoder = 3 to 8 lines


A0
A1~A17
17
128K×16
CS
8
8
M
U
X
8