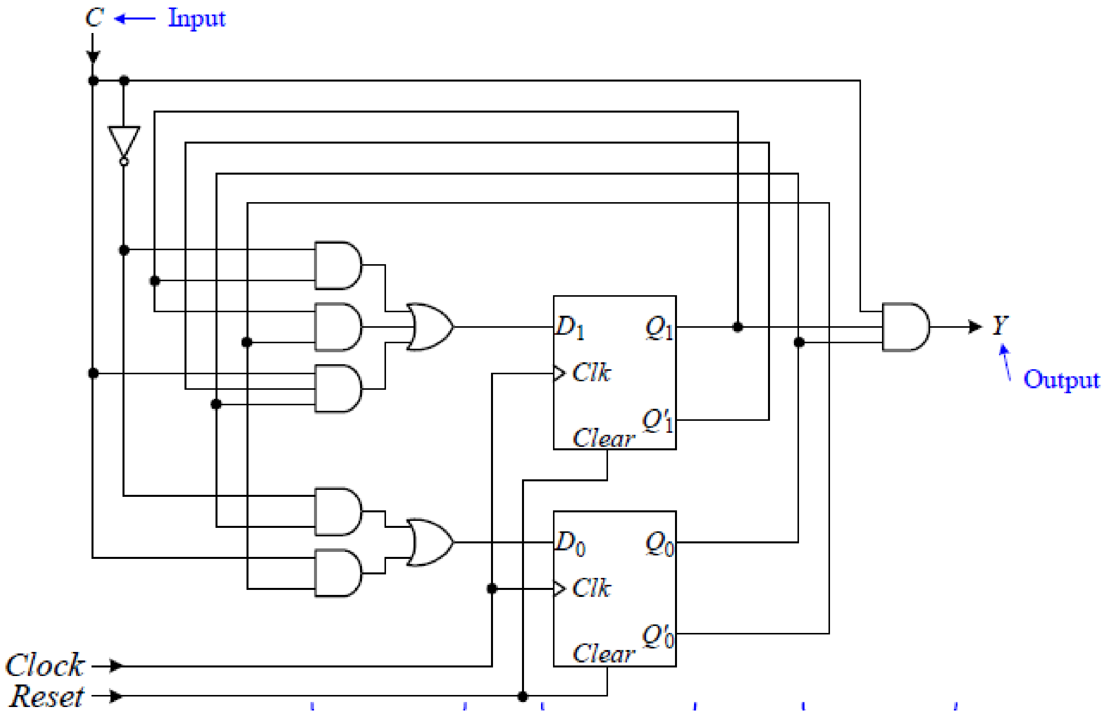
數位系統第 5 章練習

練習 1：Sequential Circuit 分析



(1)It is a Moore or Mealy FSM? (2)To derive the next-state equations; (3)To derive the next-state table; (4)To derive the output equations; (5)To derive the output table;

Ans：

(1) Mealy Machine

(2)

$$Q_{1next} = D_1 = C'Q_1 + Q_1Q_0' + CQ_1'Q_0$$
$$Q_{0next} = D_0 = C'Q_0 + CQ_0'$$

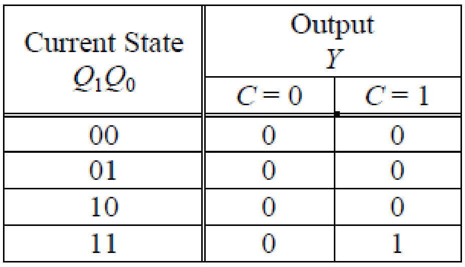
(3)

Current State Q_1Q_0	Next State $Q_{1next}Q_{0next}$	
	$C = 0$	$C = 1$
00	00	01
01	01	10
10	10	11
11	11	00

(4)

$$Y = CQ_1Q_0$$

(5)



Current State Q_1Q_0	Output Y	
	$C=0$	$C=1$
00	0	0
01	0	0
10	0	0
11	0	1

練習 2：Sequence Detector 設計

Design of the 11011 Sequence Detector A sequence detector accepts as input a string of bits: either 0 or 1. Its output goes to 1 when a target sequence has been detected. There are two basic types: overlap and non-overlap. In an sequence detector that allows overlap, the final bits of one sequence can be the start of another sequence.

11011 detector with overlap X = 11011011011

Z = 00001001001

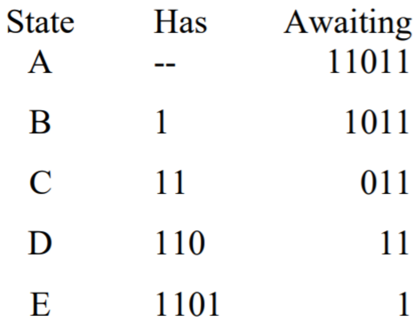
11011 detector with no overlap X = 11011011011

Z = 00001000001

Step 1 – Derive the State Diagram and State Table for the Problem

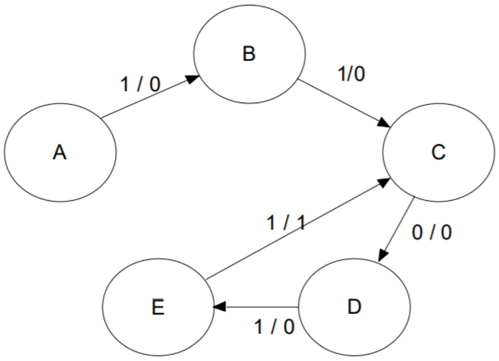
Step 1a – Determine the Number of States We are designing a sequence detector for a 5-bit sequence, so we need 5 states. We label these states A, B, C, D, and E. State A is the initial state.

Step 1b – Characterize Each State by What has been Input and What is Expected

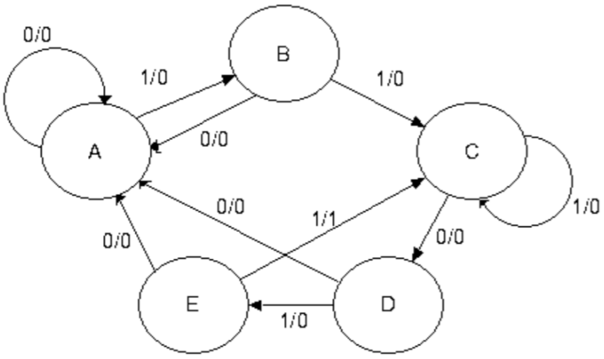


State	Has	Awaiting
A	--	11011
B	1	1011
C	11	011
D	110	11
E	1101	1

Step 1c – Do the Transitions for the Expected Sequence Here is a partial drawing of the state diagram. It has only the sequence expected. Note that the diagram returns to state C after a successful detection; the final 11 are used again.



Step 1d – Insert the Inputs That Break the Sequence, we obtain the state diagram in Mealy machine.



Each state has two lines out of it – one line for a 1 and another line for a 0. The notes below explain how to handle the bits that break the sequence.

Step 1e – Generate the State Table with Output

Present State	**Next State / Output**	
	X = 0	**X = 1**
A	A / 0	B / 0
B	A / 0	C / 0
C	D / 0	C / 0
D	A / 0	E / 0
E	A / 0	C / **1**

Step 2 – Determine the number of Flip-Flops required

We have 5 states, so N = 5. We solve the equation $2^{P-1} < 5 < 2^{P}$ by P = 3. So we need three flip-flops.

Step 3 – State assignment : assign a unique binary code to each state

The simplest way is to make the following assignments

A = 000

B = 001

C = 010

D = 011

E = 100

We can also use gray code, or others…

Step 4 – Derive state equations from next state table

Step 5 – Derive output equations

Step 6 – K-map Optimization

Step 7 – Draw the circuit