CS2100 Computer Organisation

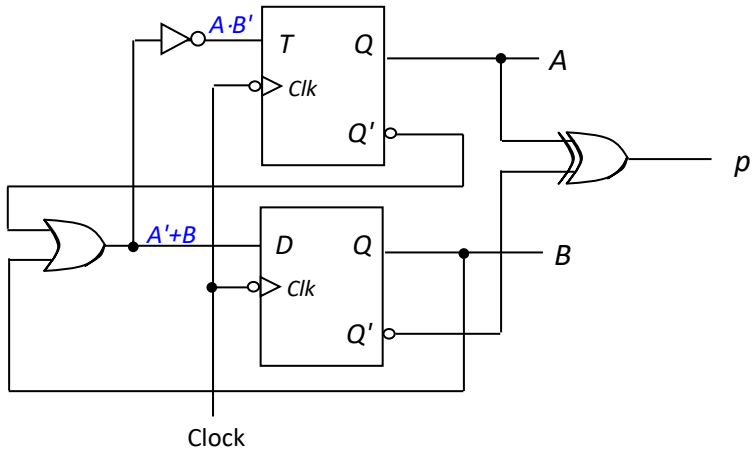
Tutorial #9: Sequential Circuits

(Week 11: 28 October – 1 November 2024)

Answers

Tutorial Questions

1. A four-state sequential circuit below consists of a ***T* flip-flop** and a ***D* flip-flop**. Analyze the circuit.



(a) Complete the state table and hence draw the state diagram.

(b) Assuming that the circuit is initially at state 0, what is the final state and the outputs generated after 3 clock cycles?

A state is called a ***sink*** if once the circuit enters this state, it never moves out of that state.

(c) How many sinks are there for this circuit?

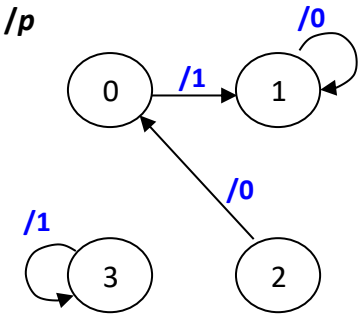
(d) Which is likely to be an unused state in this circuit?

Answers:

(a)

$p = A \cdot B + A' \cdot B'$
$TA = A \cdot B'$
$DB = A' + B$

Present state		Output	Flip-flop inputs		Next state	
A	***B***	***p***	***TA***	***DB***	***A+***	***B+***
0	0	**1**	**0**	**1**	**0**	**1**
0	1	**0**	**0**	**1**	**0**	**1**
1	0	**0**	**1**	**0**	**0**	**0**
1	1	**1**	**0**	**1**	**1**	**1**

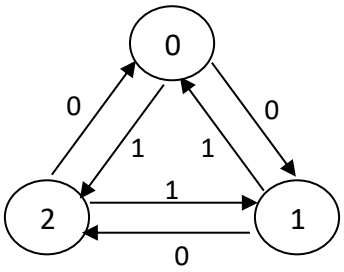


(b) After 3 clock cycles, the circuit is in state 1, and it generated 100 as output.

(c) There are 2 sinks: states 1 and 3.

(d) State 3 is likely to be an unused state.

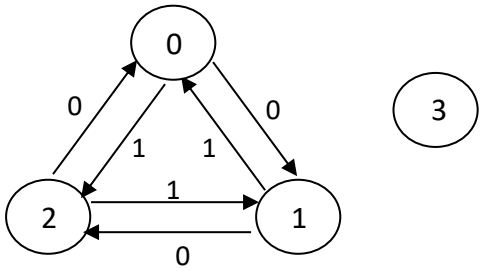
2. Given the state transition diagram on the right with states *AB* and input *x*, implement the circuit using ***JK* flip-flop**s and the <u>fewest number of logic gates</u>.



Fill in the state table below and draw the circuit. You do not need to follow the simplest SOP expression in your implementation as that might not give you a circuit with the fewest logic gates.

Present state		Input	Next state		Flip-flop *A*		Flip-flop *B*	
A	***B***	***x***	A^+	B^+	***JA***	***KA***	***JB***	***KB***
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

State 3 is unused. Can you complete the following state diagram with the unused state?



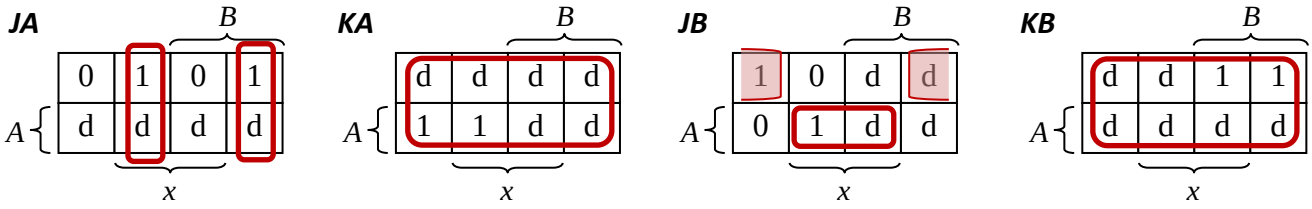
A circuit is **self-correcting** if for some reason the circuit enters into any unused (invalid) state, it is able to transit to a valid state after a finite number of transitions. Is your circuit self-correcting, and why?

Answers:

Using K-maps to find simplified expressions for flip-flop inputs.

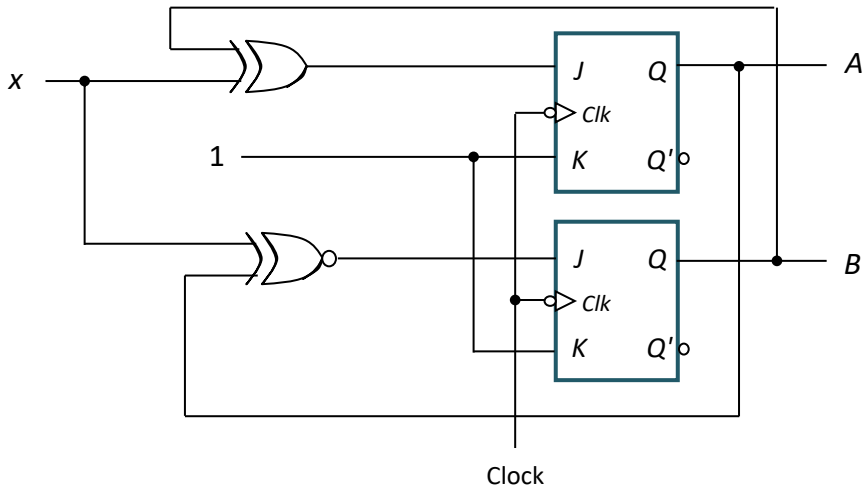
Present state		Input	Next state		Flip-flop *A*		Flip-flop *B*	
A	***B***	***x***	A^+	B^+	***JA***	***KA***	***JB***	***KB***
0	0	0	**0**	**1**	**0**	**d**	**1**	**d**
0	0	1	**1**	**0**	**1**	**d**	**0**	**d**
0	1	0	**1**	**0**	**1**	**d**	**d**	**1**
0	1	1	**0**	**0**	**0**	**d**	**d**	**1**
1	0	0	**0**	**0**	**d**	**1**	**0**	**d**
1	0	1	**0**	**1**	**d**	**1**	**1**	**d**
1	1	0	**d**	**d**	**d**	**d**	**d**	**d**
1	1	1	**d**	**d**	**d**	**d**	**d**	**d**

d = don't care



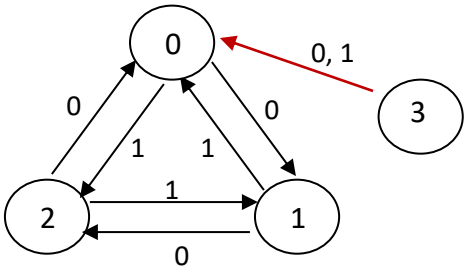
JA = B·x′ + B′·x = B ⊕ x ***KA = 1*** ***JB = A′·x′ + A·x = A ⊙ x*** ***KB = 1***

(Note: we are NOT finding the simplest SOP expressions here.)

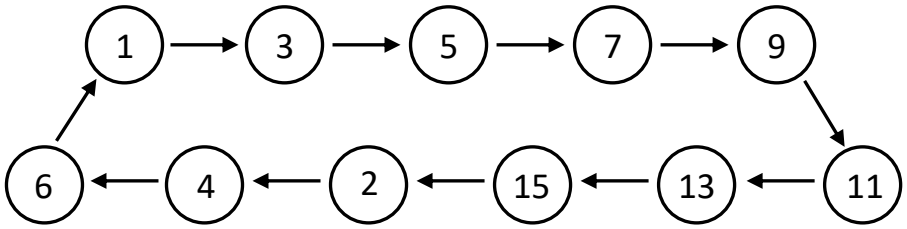


After committing the expressions for the flip-flop inputs, the don't-care values below are replaced with actual values (in parentheses). The state diagram with the unused state 3 is shown below. It is a self-correcting circuit, since there is an arrow out from state 3 to a used state.

Present state		Input	Next state		Flip-flop *A*		Flip-flop *B*	
A	***B***	***x***	A^+	B^+	***JA***	***KA***	***JB***	***KB***
0	0	0	**0**	**1**	**0**	**d(1)**	**1**	**d(1)**
0	0	1	**1**	**0**	**1**	**d(1)**	**0**	**d(1)**
0	1	0	**1**	**0**	**1**	**d(1)**	**d(1)**	**1**
0	1	1	**0**	**0**	**0**	**d(1)**	**d(0)**	**1**
1	0	0	**0**	**0**	**d(0)**	**1**	**0**	**d(1)**
1	0	1	**0**	**1**	**d(1)**	**1**	**1**	**d(1)**
1	1	0	**d(0)**	**d(0)**	**d(1)**	**d(1)**	**d(0)**	**d(1)**
1	1	1	**d(0)**	**d(0)**	**d(0)**	**d(1)**	**d(1)**	**d(1)**



3. [AY2018/19 Semester 2 exam]
A sequential circuit goes through the following states, whose state values are shown in decimal:



The states are represented by 4-bit values *ABCD*. Implement the sequential circuit using a *D* flip-flop for *A*, *T* flip-flops for *B* and *C*, and a *JK* flip-flop for *D*.

(a) Write out the **simplified SOP expressions** for all the flip-flop inputs.

(b) Implement your circuit according to your simplified SOP expressions obtained in part (a). Complete the given state diagram, by indicating the next state for each of the five unused states.

(c) Is your circuit self-correcting? Why?

Answers:

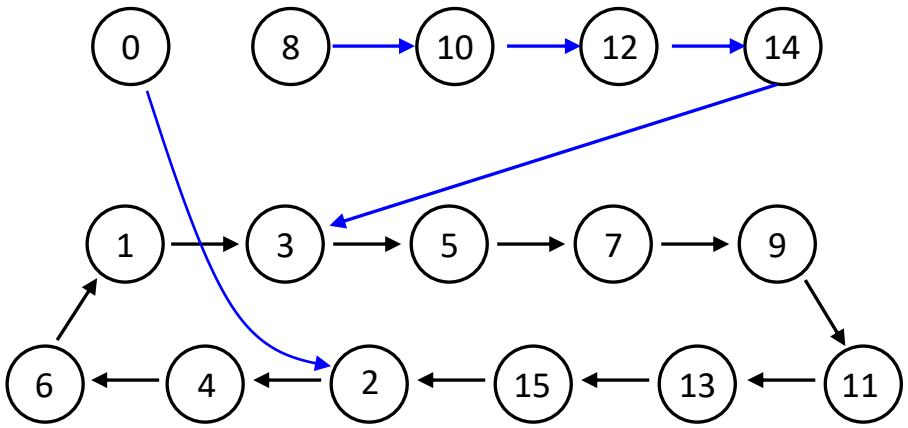
DA = ***A·B' + A·C' + A'·B·C·D***

TB = ***C***

TC = ***A' + B' + C'***

JD = ***B·C***

KD = ***A·B·C***



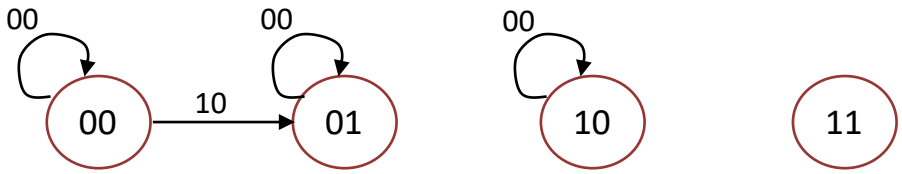
The circuit is self-correcting as any unused state can transit to a used state after a finite number of cycles.

4. Pokemone Theme Park offers locker rental to its visitors. Visitors may purchase two types of token: Pokemoney $1 (P$1) and Pokemoney $2 (P$2). A locker's rental costs P$3. When a visitor deposits P$3 into the locker's token slot, its door will open.

   Design a sequential circuit with states *AB* for the locker's door using *D* flip-flops. The circuit consists of 4 states representing the amount a visitor has deposited: 0, 1, 2 and 3 (or, in binary, *AB* = 00, 01, 10 and 11). The visitor can deposit only one token at a time. When the circuit reaches the final state 3, it remains in state 3 even if the visitor continues to put tokens into the slot. When the circuit is in state 2 and the visitor deposits a P$2 token, the circuit goes into state 3.

   The partial state diagram is shown below. The inputs *x* and *y* represent the P$1 and P$2 tokens respectively. The label on each arrow represents *xy*.

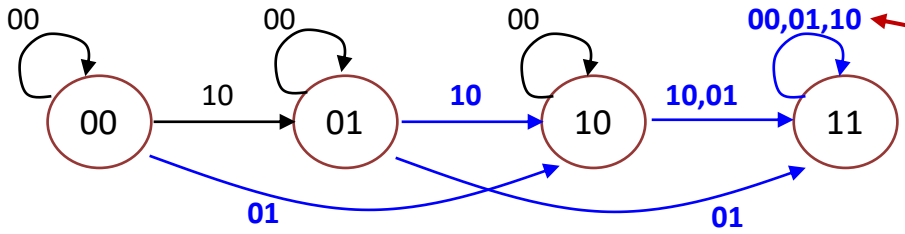
   (a) Draw and write the missing arrows and labels.



   (b) Write the **simplified SOP expressions** for the flip-flop inputs *DA* and *DB*.

Answers:

(a)



Common mistake: Some students take the input to be 2-bit binary value, i.e. 00=$0, 01=$1, 10=$2.

Common mistake: It is not correct to write dd (don't care) here as that would cover 00, 01, 10 and 11.

(b)

$DA = \boldsymbol{A + y + B \cdot x}$

$DB = \boldsymbol{B \cdot x' + B' \cdot x + A \cdot y + A \cdot x}$ or $DB = \boldsymbol{B \cdot x' + B' \cdot x + A \cdot y + A \cdot B}$