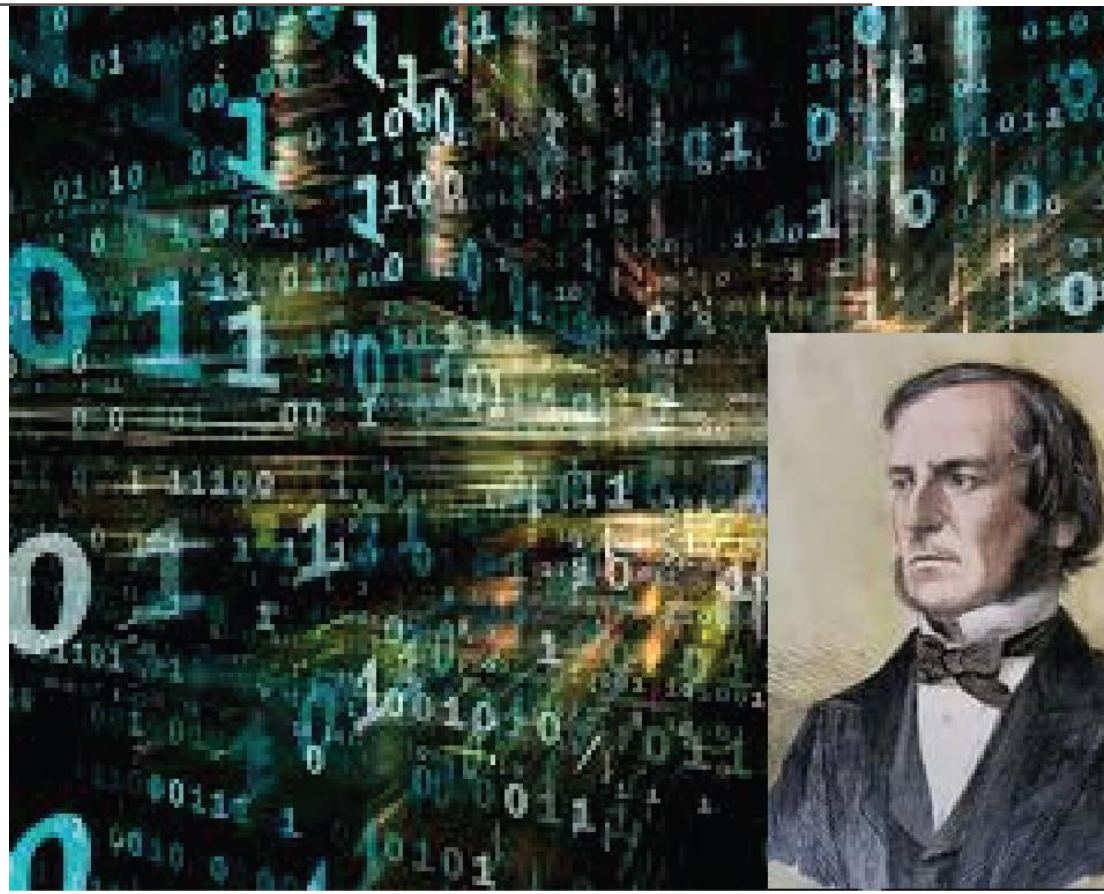
DIGITAL CIRCUITS

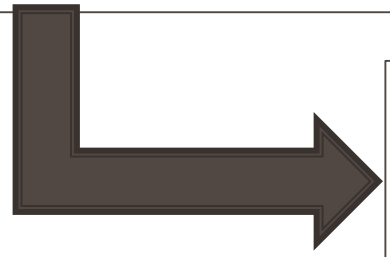
Week-13, Lecture-1
Sequential Circuits



Sneh Saurabh
6th November, 2018

Digital Circuits: Announcements/Revision

Sequential Circuits



Design

Sequential Circuit Design: Excitation Table

Characteristics Table/Function:

- Given a current state and the inputs, what would be the next state
- Useful for Analysis

Excitation Table/Function:

- Given a current state and the next state, what are the required inputs
- Useful for Synthesis

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) JK Flip-Flop

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) T Flip-Flop

Sequential Circuit Design: JK Flip-Flop

Problem 1:

Design a circuit using JK flip-flops that implements the state table shown below.

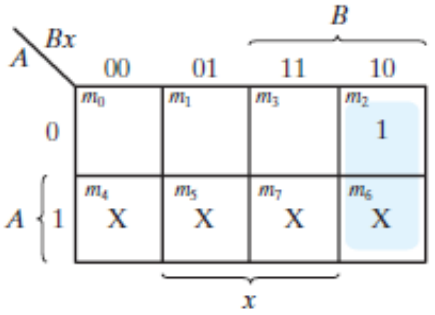
Present State		Input	Next State		Flip-Flop Inputs			
A	**B**	**x**	**A**	**B**	J_A	K_A	J_B	K_B
0	0	0	0	0				
0	0	1	0	1				
0	1	0	1	0				
0	1	1	0	1				
1	0	0	1	0				
1	0	1	1	1				
1	1	0	1	1				
1	1	1	0	0				

Q(t)	Q(t = 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

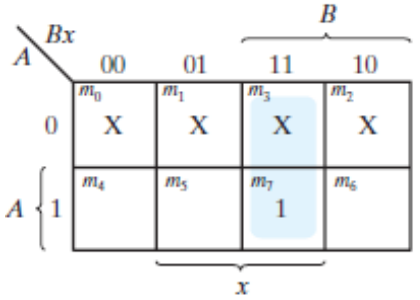
(a) *JK* Flip-Flop

Sequential Circuit Design: JK Flip-Flop (Input Equation)

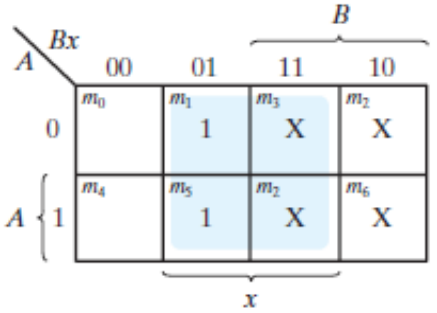
Present State		Input	Next State		Flip-Flop Inputs			
A	**B**	**x**	**A**	**B**	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



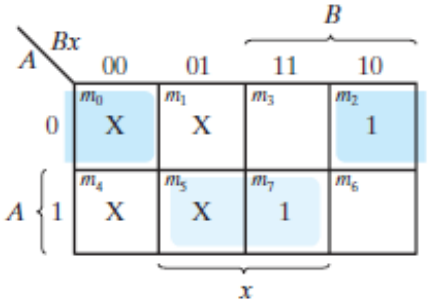
$J_A = Bx'$



$K_A = Bx$



$J_B = x$



$K_B = (A \oplus x)'$

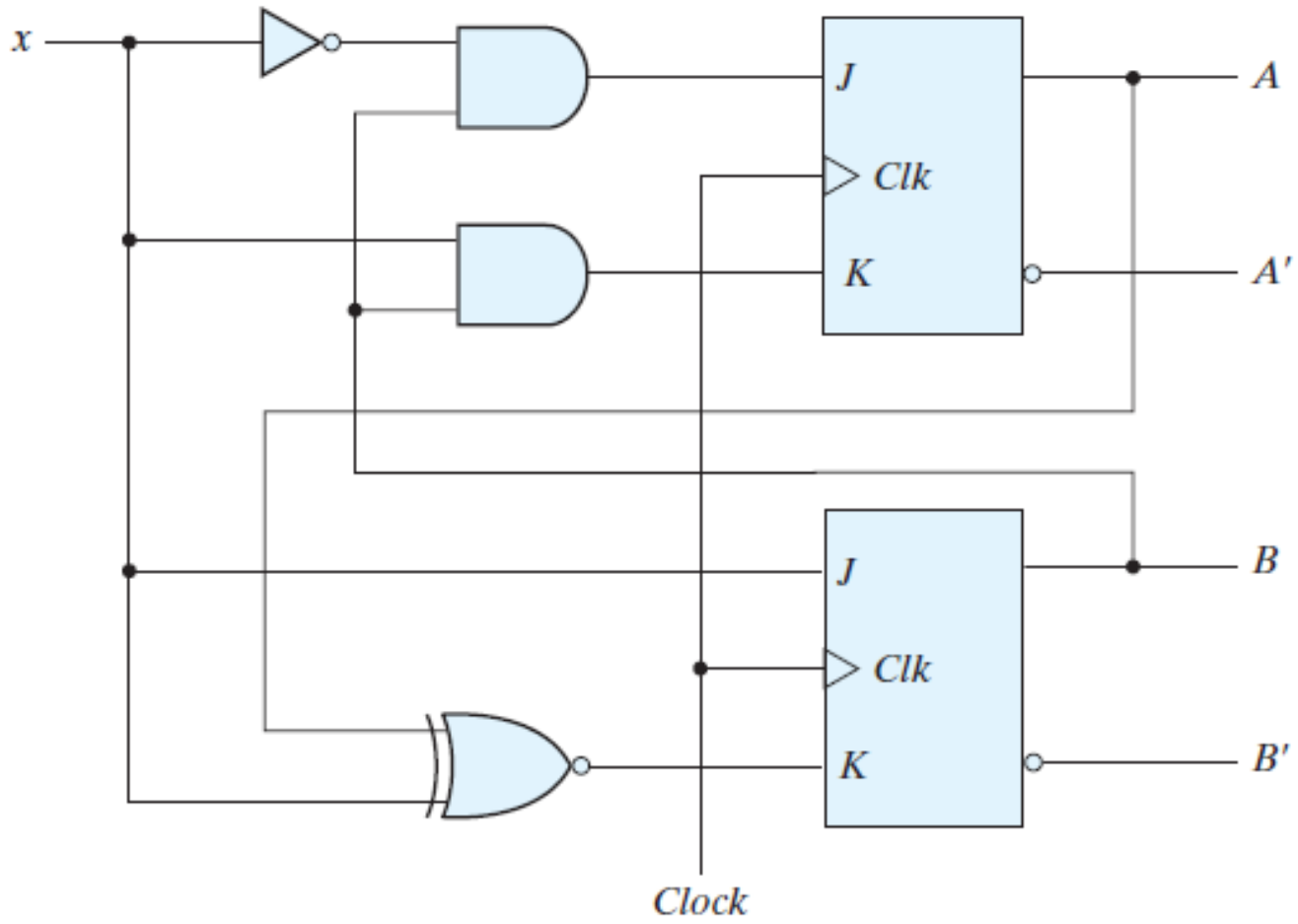
Sequential Circuit Design: Logic Diagram

$$J_A(A, B, x) = Bx'$$

$$K_A(A, B, x) = Bx$$

$$J_B(A, B, x) = x$$

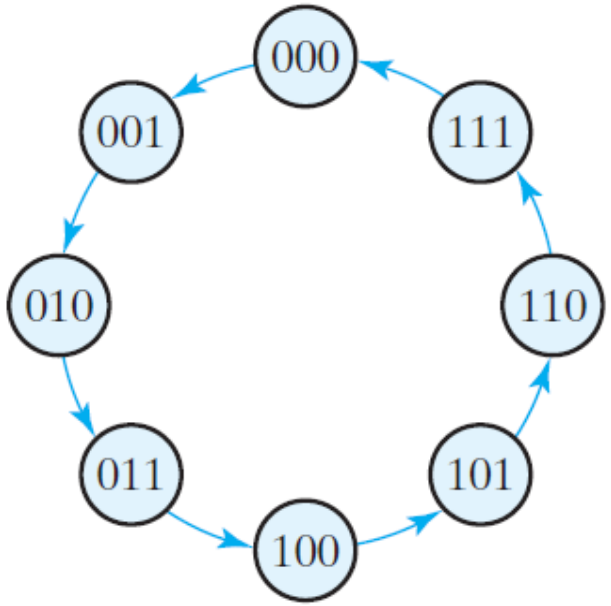
$$K_B(A, B, x) = (A \oplus x)'$$



Sequential Circuit Design: T flip-flop

Problem 2:

Design a 3-bit binary counter using T flip-flops.



- Transition happens at each clock edge.
- Input (Clock Edge) is implicit in this state diagram.

- Assume that the output of the flip-flops are A_2, A_1 and A_0
- Draw the state table using the state diagram

Present State			Next State		
A_2	A_1	A_0	A_2	A_1	A_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Sequential Circuit Design: Input Equation for T flip-flop

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1			
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

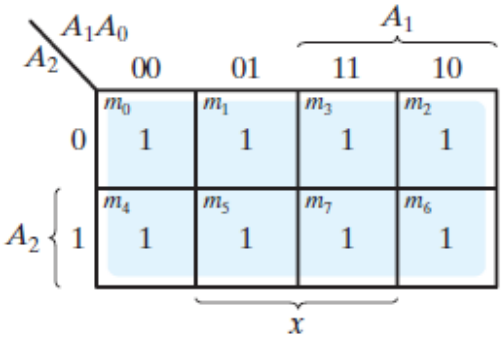
$Q(t)$	$Q(t = 1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) T Flip-Flop

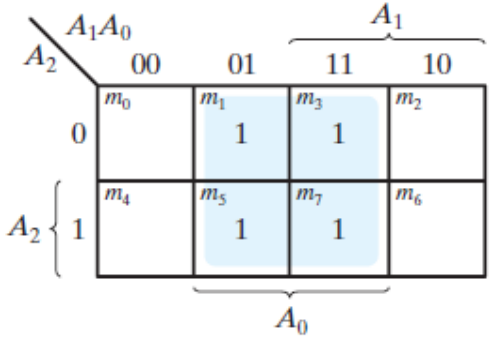
- Using the state table and the excitation table of T flip-flop, find the input values required at the T flip-flop.

Sequential Circuit Design: Simplified Input Equation

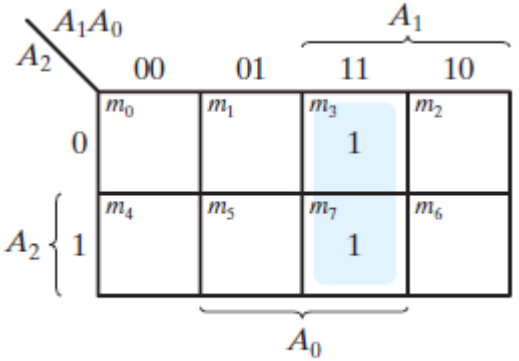
Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	1	1	1



$T_{A0} = 1$



$T_{A1} = A_0$



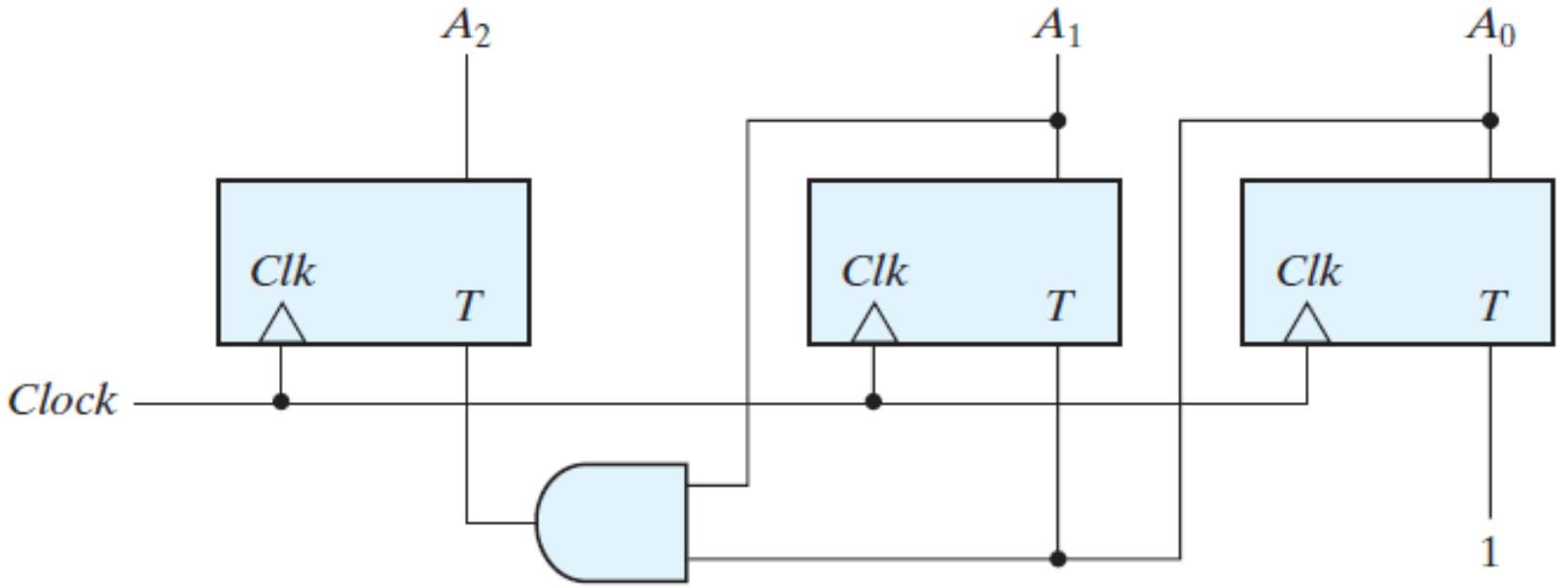
$T_{A2} = A_1A_0$

Sequential Circuit Design: Logic Diagram

$T_{A0} = 1$

$T_{A1} = A_0$

$T_{A2} = A_0 A_1$

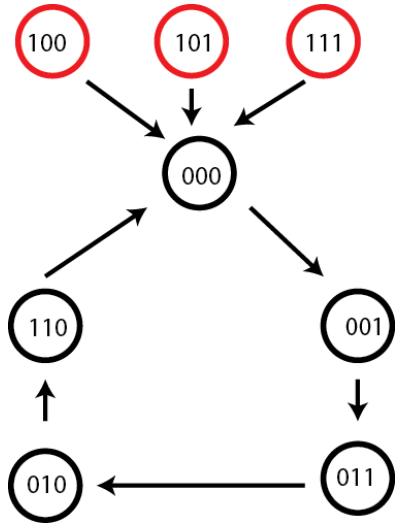


Sequential Circuit Design: Invalid states

Problem 3:

Design a 3-bit binary counter using T flip-flops that counts as: 0 → 1 → 3 → 2 → 6 → 0 ….

Assume that the next state from the invalid state is the 0 state.



- First draw the state diagram

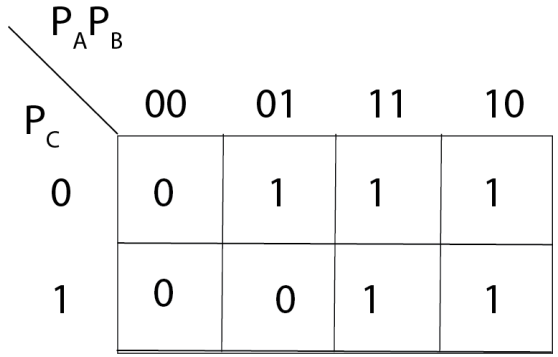
- Draw the state table

Present State			Next State			Flip-flop inputs		
P_A	P_B	P_C	N_A	N_B	N_C	T_A	T_B	T_C
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	**0**	**0**						
1	**0**	**1**						
1	1	0						
1	**1**	**1**						

Sequential Circuit Design: Optimized Implementation

Present State			Next State			Flip-flop inputs		
P_A	P_B	P_C	N_A	N_B	N_C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	0	1
1	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
1	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**
1	1	0	0	0	0	1	1	0
1	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**

- Find T_A, T_B and T_C in terms of P_A, P_B and P_C



$$T_A = P_A + P_C{}'P_B$$

$$T_B = P_AP_B + P_A{}'P_B{}'P_C$$

$$T_C = P_BP_C + P_AP_C + P_A{}'P_B{}'P_C{}'$$

Resources used: NOT=3, AND=8, OR=4

Can it be improved?

Sequential Circuit Design: When Invalid states are don't cares

Problem 4:

Design a 3-bit binary counter using T flip-flops that counts as: 0 → 1 → 3 → 2 → 6 → 0 ….

Assume that the next state from the invalid state is a don't care state. (From an invalid state, it is allowed to go to any state)

- Draw the state table again with don't cares

Present State			Next State			Flip-flop inputs		
P_A	P_B	P_C	N_A	N_B	N_C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	0	1
1	**0**	**0**	***X***	***X***	***X***	***X***	***X***	***X***
1	**0**	**1**	***X***	***X***	***X***	***X***	***X***	***X***
1	1	0	0	0	0	1	1	0
1	**1**	**1**	***X***	***X***	***X***	***X***	***X***	***X***

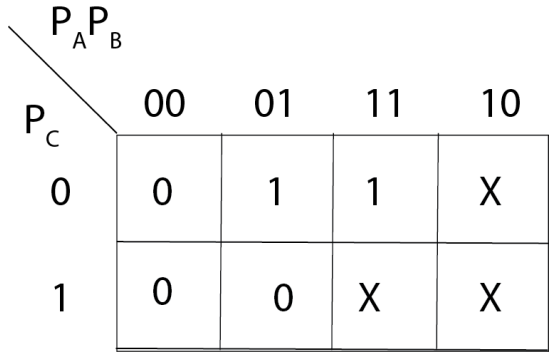
Sequential Circuit Design: Optimized Implementation

Present State			Next State			Flip-flop inputs		
P_A	P_B	P_C	N_A	N_B	N_C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	0	1
1	**0**	**0**	***X***	***X***	***X***	***X***	***X***	***X***
1	**0**	**1**	***X***	***X***	***X***	***X***	***X***	***X***
1	1	0	0	0	0	1	1	0
1	**1**	**1**	***X***	***X***	***X***	***X***	***X***	***X***

Resources used: NOT=2, AND=4, OR=2

Without don't care it was: NOT=3, AND=8, OR=4

- Find T_A, T_B and T_C in terms of P_A, P_B and P_C



$$T_A = P_C{}'P_B$$

$$T_B = P_A + P_B{}'P_C$$

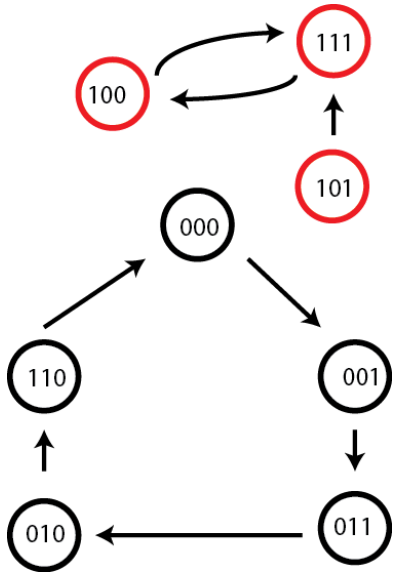
$$T_C = P_BP_C + P_B{}'P_C{}'$$

Is there any problem?

Sequential Circuit Design: Optimized Implementation

- With the following implementation, what would be the next state for the invalid states:
  - $T_A = P_C{}'P_B$
  - $T_B = P_A + P_B{}'P_C$
  - $T_C = P_BP_C + P_B{}'P_C{}'$

First find T_A, T_B and T_C and then N_A, N_B and N_C



If the counter enters an invalid state, it remains in the invalid state for ever!

How to avoid this?

Present State			Next State			Flip-flop inputs		
P_A	P_B	P_C	N_A	N_B	N_C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0	0
0	1	1	0	1	0	0	0	1
1	**0**	**0**						
1	**0**	**1**						
1	1	0	0	0	0	1	1	0
1	**1**	**1**						

Digital Circuits: Practice Problems

Problems 5.1-5.4

from “Digital Design”– M. Morris Mano & Michael D. Ciletti, Ed-5, Pearson (Prentice-Hall).

Problems 7.1-7.9

from Fundamentals of Digital Logic with Verilog Design - S. Brown, Z. Vranesic