TSN1101 - COMPUTER ARCHITECTURE AND ORGANIZATION

TRIMESTER 2510

Assignment 1

TUTORIAL SECTION:
T21L

GROUP 3

Name	Student ID
EBA MOHAMED ABBAS AHMED (Leader)	242UC243BE
LAMA M. R. SIAM	242UC243B4
HAMSA HASHIM OMER	242UC241DB
SITI ZULAIKHA BINTI ABDUL RAZIF	242UC243TL

1. **Problem statement**

The goal of this project is to design and implement a 4-bit synchronous counter that operates under two different state sequences based on the value of an external input signal (I). Depending on the value of I, the counter must follow specific patterns.

- When I = 0, the counter must cycle through the sequences:
  7 →10 →12→9 and repeat.
  All undesired states must redirect to state 10.
- When I = 1, the counter must cycle through the sequence:
  9 →13 →11→5 and repeat
  All undesired states must redirect to state 13.

The design process involves creating a state transition diagram, developing a complete present-to-next state table, and deriving excitation equations for each flip-flop through K-map simplification. The final system should be simulated using CircuitVerse to ensure it operates correctly.

2. **Objectives**

The main objectives of this assignment are:

1. To design a 4-bit synchronous counter controlled by an external input (I) that determines two different state sequences.
2. To define and convert valid state sequences from decimal to binary, and redirect any invalid states.
3. To create state transition diagrams for both input cases (I = 0 and I = 1) to visualize the sequence behavior.
4. To build a complete state transition table listing current and next states and the required flip-flop inputs.
5. To identify the type of flip-flops (T, D, JK) to be used and find their excitation values.
6. To use K-maps to simplify Boolean expressions for flip-flop inputs.
7. To implement and simulate the final circuit using CircuitVerse, ensuring proper operation and correct handling of invalid states.

3. **State Transition Diagram:**

According to the assigned Question No. 75, the counter is required to operate based on two distinct state sequences, determined by the value of the external input signal **I**. These sequences are initially defined in decimal format as follows:

- For I = 0: 7 → 10 → 12 → 9 → repeat (undesired states go to 10)
- For I = 1: 9 → 13 → 11 → 5 → repeat (undesired states go to 13)

Since the counter is 4-bit, each decimal state is converted to its corresponding 4-bit binary representation:

- For I = 0: 0111 → 1010 → 1100 → 1001 (repeat)
- For I = 1: 1001 → 1101 → 1011 → 0101 (repeat)

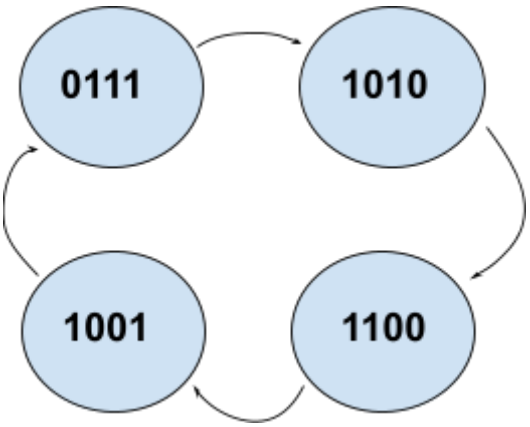
All undesired states are redirected to:

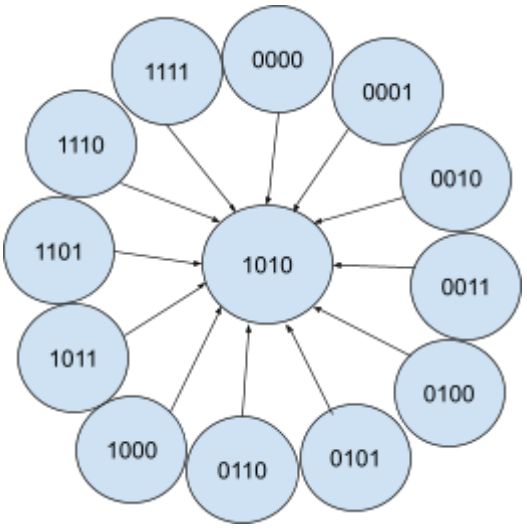
- 1010 (decimal 10) when I = 0
- 1101 (decimal 13) when I = 1

The following two figures illustrate the state transition sequences and transitions for both input cases I = 0 and I = 1:

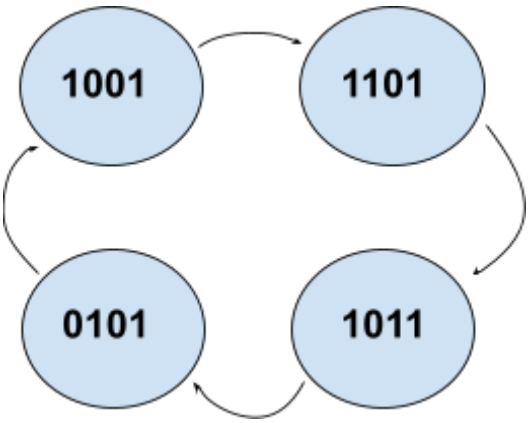
Input=0:

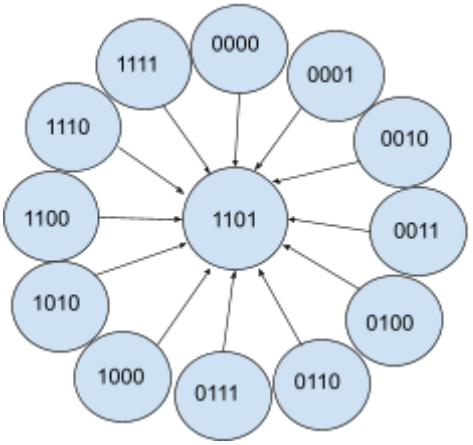
0111 → 1010 → 1100 → 1001(repeat), all undesired state go to 1010





input=1:
1001 → 1101 → 1011 → 0101(repeat),all undesired states go to 1101





4. **Excitation Table for Flip Flops:**

In sequential circuit design, excitation tables specify the necessary inputs for a flip-flop to transition from the present state (Q) to the next state (Q^+). These tables are essential for determining the correct flip-flop input conditions during state transitions. Below are the excitation rules for T, D and JK flip-flops:

Present State(Q)	Next State(Q+)	TD	DC	DB	JA	KA
0	0	0	0	0	0	X
0	1	1	1	1	1	X
1	0	1	0	0	X	1
1	1	0	1	1	X	0

5. **State Transition Table :**

The state transition table outlines the current state, the input value **I**, the corresponding next state, and the necessary flip-flop inputs for both valid and undesired states. This table serves as a comprehensive definition of the 4-bit synchronous counter's behavior.

There are 16 possible states in a 4-bit system:
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

For I = 0, the counter cycles through the sequence:
0111 → 1010 → 1100 → 1001 → repeat.
All undesired states are redirected to 1010.

For I = 1, the counter cycles through the sequence:
1001 → 1101 - 1011 - 0101 → repeat.
All undesired states are redirected to 1101.

X value	MSB Flip-Flop	LSB Flip-Flop
0	JK	T
1	T	JK
2	D	T
3	T	D

In this design, the flip-flops used are:
• TD(MSB) (T flip-flop for QD)
• DC (D flip-flop for QC)
• DB (D flip-flop for QB)
• JA and KA(LSB) (JK flip-flop for QA)

The following table represents the state transition and the required inputs for each flip-flop:

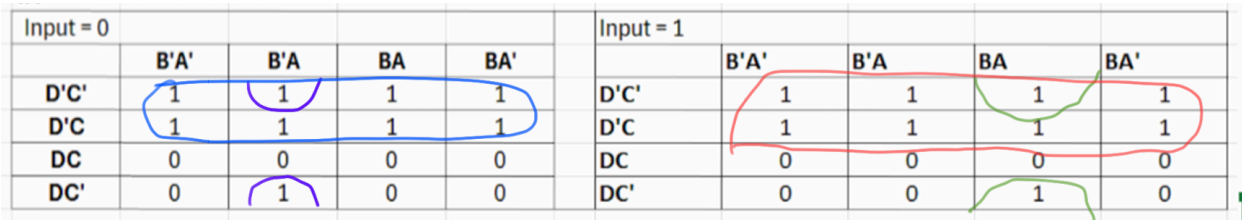
State	xternal Inp	Present state (Q)				Next state (Q+1)				Decimal Present State	Decimal Next State	INPUT to Flip Flops				
	I	QD (MSB)	QC	QB	QA (LSB)	QD+1 (MSB)	QC+1	QB+1	QA+1 (LSB)			TD(MSB)	DC	DB	JA(LSB)	KA(LSB)
S0	0	0	0	0	0	1	0	1	0	0	10	1	0	1	0	X
S1	0	0	0	0	1	1	0	1	0	1	10	1	0	1	X	1
S2	0	0	0	1	0	1	0	1	0	2	10	1	0	1	0	X
S3	0	0	0	1	1	1	0	1	0	3	10	1	0	1	X	1
S4	0	0	1	0	0	1	0	1	0	4	10	1	0	1	0	X
S5	0	0	1	0	1	1	0	1	0	5	10	1	0	1	X	1
S6	0	0	1	1	0	1	0	1	0	6	10	1	0	1	0	X
S7	0	0	1	1	1	1	0	1	0	7	10	1	0	1	X	1
S8	0	1	0	0	0	1	0	1	0	8	10	0	0	1	0	X
S9	0	1	0	0	1	0	1	1	1	9	7	1	1	1	X	0
S10	0	1	0	1	0	1	1	0	0	10	12	0	1	0	0	X
S11	0	1	0	1	1	1	0	1	0	11	10	0	0	1	X	1
S12	0	1	1	0	0	1	0	0	1	12	9	0	0	0	1	X
S13	0	1	1	0	1	1	0	1	0	13	10	0	0	1	X	1
S14	0	1	1	1	0	1	0	1	0	14	10	0	0	1	0	X
S15	0	1	1	1	1	1	0	1	0	15	10	0	0	1	X	1
S16	1	0	0	0	0	1	1	0	1	0	13	1	1	0	1	X
S17	1	0	0	0	1	1	1	0	1	1	13	1	1	0	X	0
S18	1	0	0	1	0	1	1	0	1	2	13	1	1	0	1	X
S19	1	0	0	1	1	1	1	0	1	3	13	1	1	0	X	0
S20	1	0	1	0	0	1	1	0	1	4	13	1	1	0	1	X
S21	1	0	1	0	1	1	0	0	1	5	9	1	0	0	X	0
S22	1	0	1	1	0	1	1	0	1	6	13	1	1	0	1	X
S23	1	0	1	1	1	1	1	0	1	7	13	1	1	0	X	0
S24	1	1	0	0	0	1	1	0	1	8	13	0	1	0	1	X
S25	1	1	0	0	1	1	1	0	1	9	13	0	1	0	X	0
S26	1	1	0	1	0	1	1	0	1	10	13	0	1	0	1	X
S27	1	1	0	1	1	0	1	0	1	11	5	1	1	0	X	0
S28	1	1	1	0	0	1	1	0	1	12	13	0	1	0	1	X
S29	1	1	1	0	1	1	0	1	1	13	11	0	0	1	X	0
S30	1	1	1	1	0	1	1	0	1	14	13	0	1	0	1	X
S31	1	1	1	1	1	1	1	0	1	15	13	0	1	0	X	0

6. **K-map Simplification:**

The following Boolean expressions were derived using K-map minimization for each flip - flop input based on the state transitions of the counter:

TD

TD=0,1,2,3,4,5,6,7,9,16,17,18,19,20,21,22,23,27



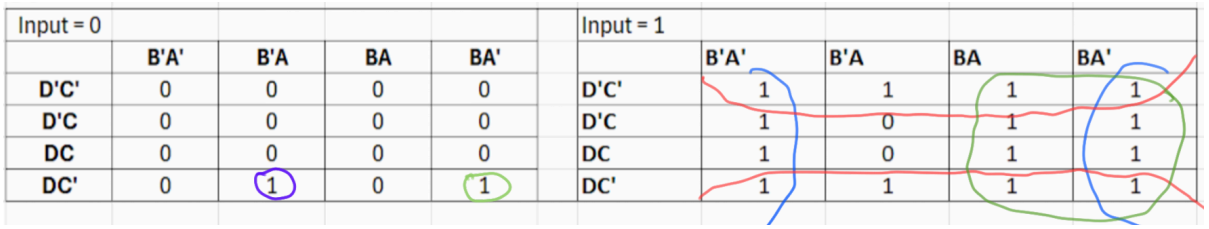
Input = 0	B'A'	B'A	BA	BA'
D'C'	1	1	1	1
D'C	1	1	1	1
DC	0	0	0	0
DC'	0	1	0	0

Input = 1	B'A'	B'A	BA	BA'
D'C'	1	1	1	1
D'C	1	1	1	1
DC	0	0	0	0
DC'	0	0	1	0

TD = ((C’ · A) · ((Input’ · B’) + (Input · B))) + (D’ · A’) + (D’ · A)

DC

DC=9,10,16,17,18,19,20,22,23,24,25,26,27,28,30,31



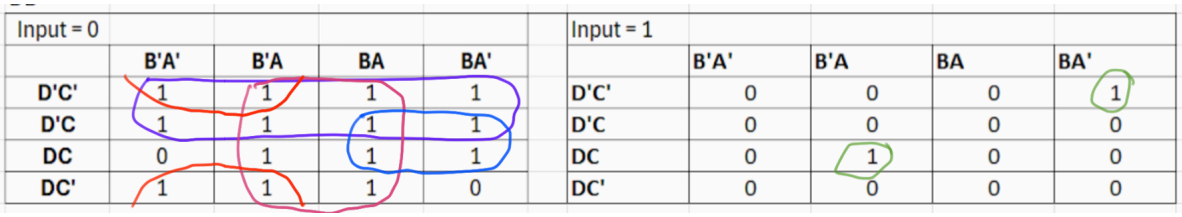
Input = 0	B'A'	B'A	BA	BA'
D'C'	0	0	0	0
D'C	0	0	0	0
DC	0	0	0	0
DC'	0	1	0	1

Input = 1	B'A'	B'A	BA	BA'
D'C'	1	1	1	1
D'C	1	0	1	1
DC	1	0	1	1
DC'	1	1	1	1

DC = (Input · (C’ + B + A’)) + ((D · C’) · ((B · A’) + (B’ · A)))

DB

DB= 0,1,3,2,4,5,6,7,8,9,11,13,14,15,18,29



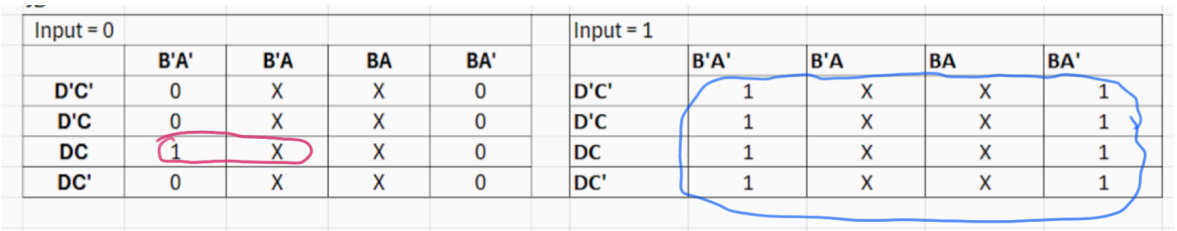
Input = 0	B'A'	B'A	BA	BA'
D'C'	1	1	1	1
D'C	1	1	1	1
DC	0	1	1	1
DC'	1	1	1	0

Input = 1	B'A'	B'A	BA	BA'
D'C'	0	0	0	1
D'C	0	0	0	0
DC	0	1	0	0
DC'	0	0	0	0

DB = (Input’ · (A + D’ + (C’ · B’) + (C · B))) + (D · C · B’ · A)

JA

JA=12,16,18,20,22,24,26,28,30



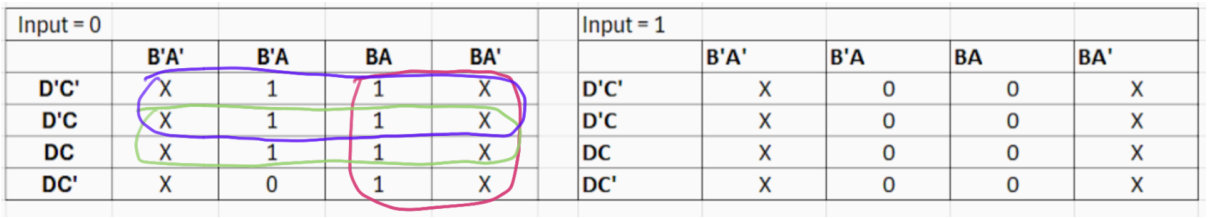
Input = 0	B'A'	B'A	BA	BA'
D'C'	0	X	X	0
D'C	0	X	X	0
DC	1	X	X	0
DC'	0	X	X	0

Input = 1	B'A'	B'A	BA	BA'
D'C'	1	X	X	1
D'C	1	X	X	1
DC	1	X	X	1
DC'	1	X	X	1

JA = A' · ((D · C · B') + Input)

KA

KA=1,3,5,7,11,13,15



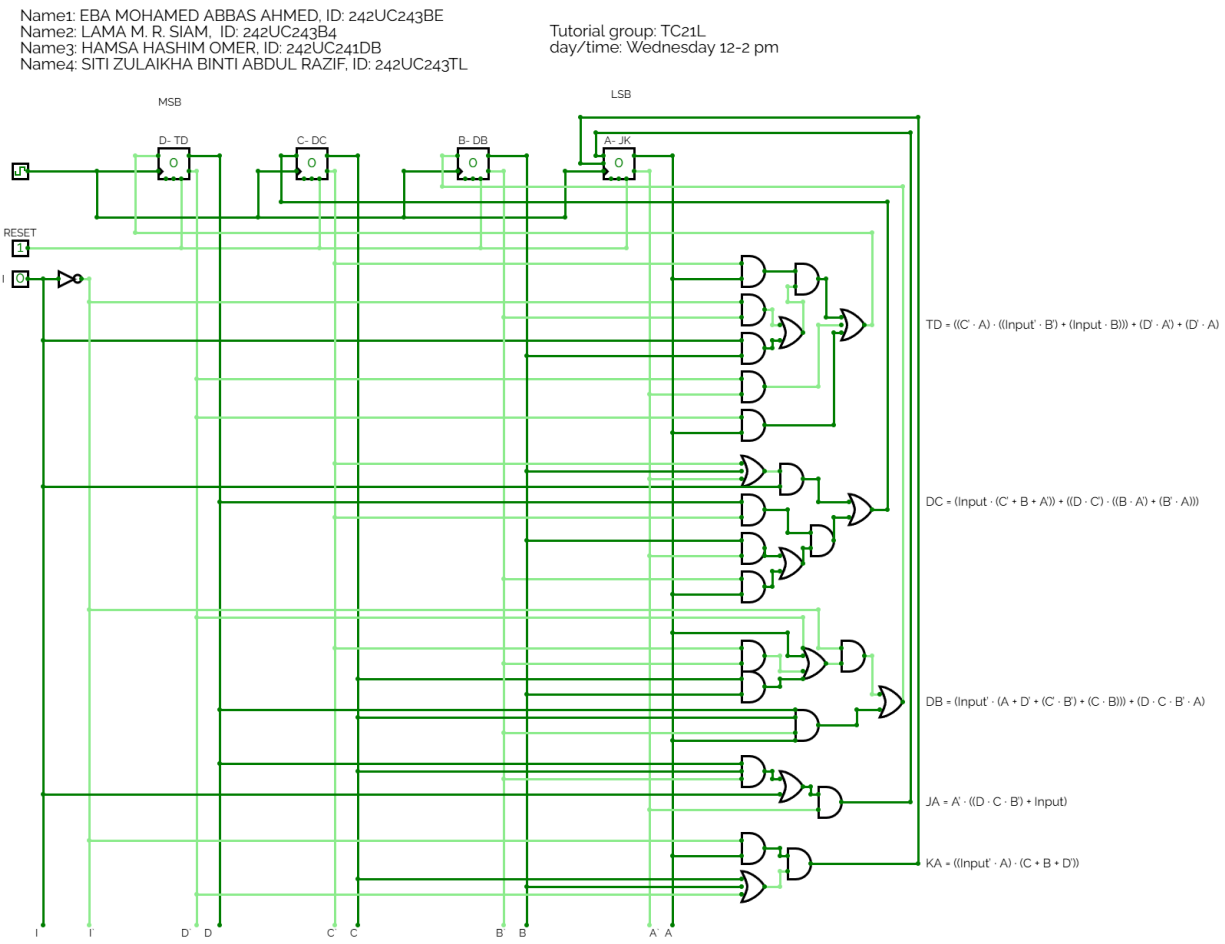
Input = 0	B'A'	B'A	BA	BA'
D'C'	X	1	1	X
D'C	X	1	1	X
DC	X	1	1	X
DC'	X	0	1	X

Input = 1	B'A'	B'A	BA	BA'
D'C'	X	0	0	X
D'C	X	0	0	X
DC	X	0	0	X
DC'	X	0	0	X

KA = ((Input' · A) · (C + B + D'))

7. **Circuit Implementation**

The final counter circuit was built and simulated using the CircuitVerse platform. The following figure shows the completed design with all outputs.



8. Conclusion:

This project focused on designing a 4-bit synchronous counter that changes its sequence based on an external input. We created and analyzed state transition diagrams, generated excitation tables, and simplified logic expressions using Karnaugh Maps. The use of T, D, and JK flip-flops ensured efficient design. A complete state transition table was built to define the counter's behavior, including handling of undesired states. Finally, the circuit was implemented and tested using CircuitVerse, and it successfully followed the intended sequences for both input conditions.