**LAB-14**

**State Diagram and State Process**

**Equipment/Components**

**Hardware:** Explorer Board

IC type: 7474 D flip flop

IC Type: 7476 JK flip flop

**Software:** Circuit Maker, Waveform

**Description**

**Description State Diagrams**

The reduction in the number of flip-flops in a sequential circuit is referred to as the state-reduction problem. State-reduction algorithms are concerned with procedures for reducing the number of states in a state table, while keeping the external input–output requirements unchanged. Since m flip-flops produce 2m states, a reduction in the number of states may (or may not) result in a reduction in the number of flip-flops. An unpredictable effect in reducing the number of flip-flops is that sometimes the equivalent circuit (with fewer flip-flops) may require more combinational gates to realize its next state and output logic.

**Even counter**

Even counter is a counter with the ability to count even numbers. A 3 bit even counter follows the sequence i.e. 000, 010, 100, 110, 000 and so on. Odd counter is a counter with the ability to count odd numbers. A 3 bit odd counter follows the sequence i.e. 001, 011, 101, 111, 001 and so on.

**Objective**

* To understand the concept of serial and parallel registers.
* To understand the concept of state diagrams.

**Task # 1**

1. Fill in the state table for the following state diagram.

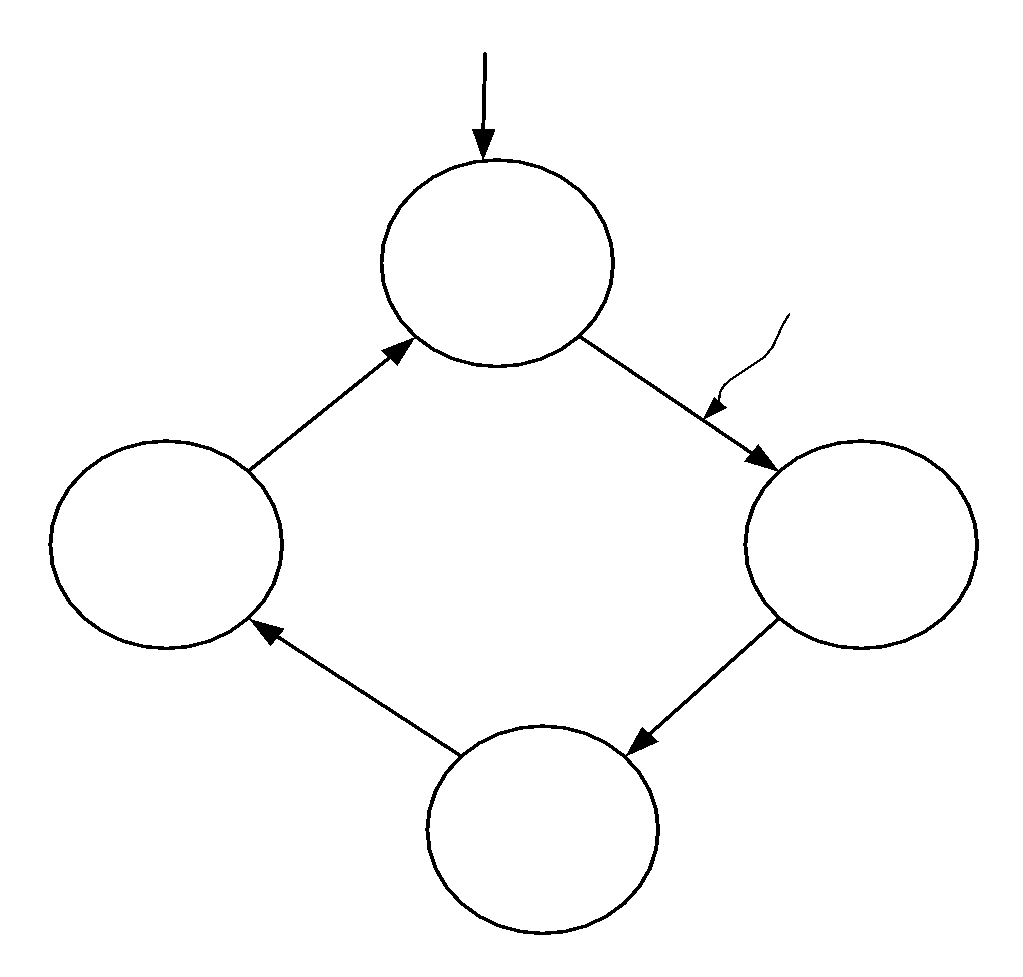


Figure 1: State diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1: State Table** | | | |
| Present State | | Next state | |
| Q0 | Q1 | Q0\_nxt | Q1\_nxt |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

1. Fill in the state table with D Flip-flop Excitations (See lecture notes for Help)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 2: State Table** | | | | | |
| Present State | | Next state | | Flip-flops Inputs | |
| Q0 | Q1 | Q0\_nxt | Q1\_nxt | D0 | D1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |

1. Use K-Map and generate simplest Boolean expressions for the inputs of the flip flops (D0, D1).
2. Draw and verify the logic diagram of the sequential circuit using the derived equations on **paper**.
3. Implement the sequential circuit on **Circuit Maker**.

|  |
| --- |
|  |

**Note:** Implement Task 1 Using All Other Flip-Flop.

**Task # 2**

Design a 3-Bit state machine with irregular sequence (001 – 011 – 101 – 111- 010 -100 - 110 - 000) using J-K Flip Flop. Draw a State Diagram and drive all the tables and draw Logic Circuit on Circuit Maker.

























**Table:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present state |  |  | Next state |  |  | Flip flop Inputs |  |  |  |  |  |
| Q0 | Q1 | Q2 | Q0+ | Q1+ | Q2+ | J0 | K0 | J1 | K1 | J2 | K2 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | X | 1 | X | X | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | X | X | 1 | 0 | X |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | X | X | 1 | X | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | X | 0 | 1 | X | 0 | X |
| 1 | 0 | 1 | 1 | 1 | 1 | X | 0 | 1 | X | X | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | X | 1 | X | 1 | 0 | X |
| 1 | 1 | 1 | 0 | 1 | 0 | X | 1 | X | 0 | X | 1 |

**BOOLEAN EXPRESSIONS:**

**JO= Q1**

**K0= Q1**

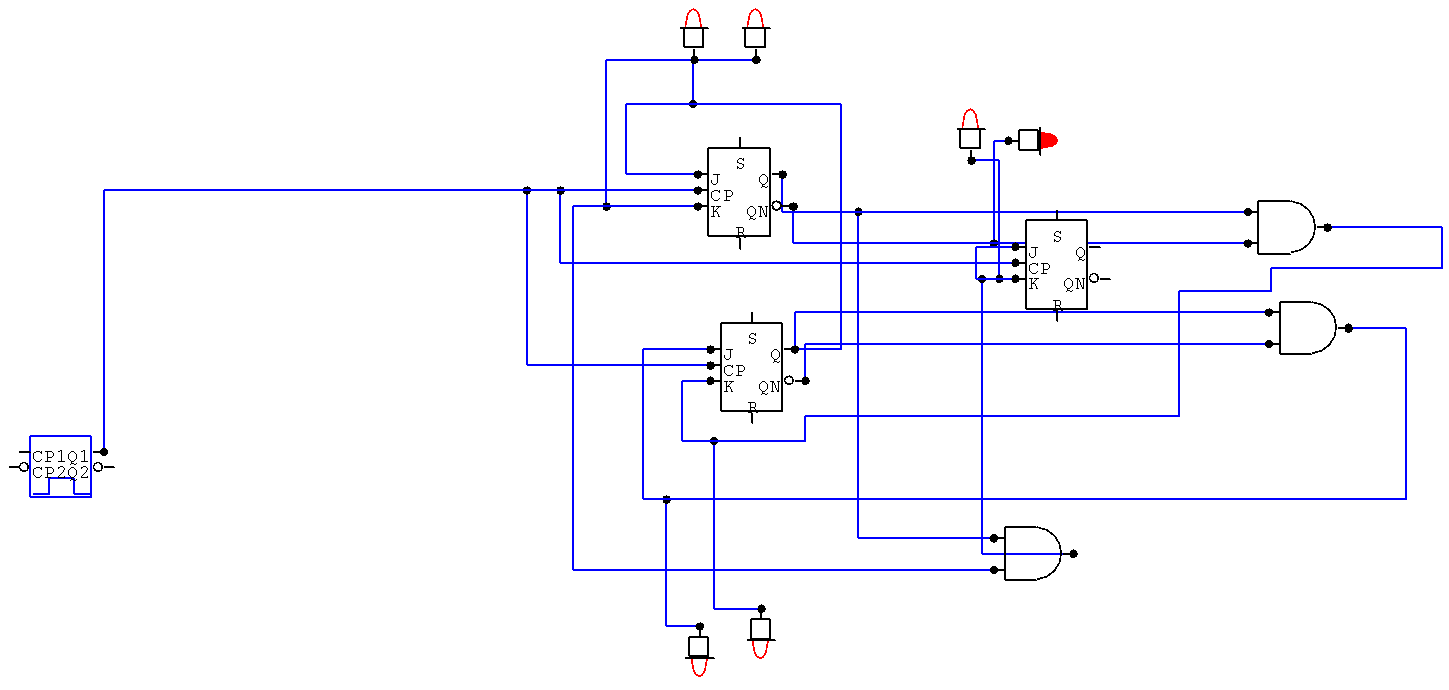
**J1= Q2+Q0**

**K1= QO’**

**J2=QO’. Q2’**

**K2= QIQ2**

**CIRCUIT DIAGRAM:**

****

**Task # 3**

1. Fill in the state table for the following state diagram.

**Note.** The state diagram is presenting 2 bit up-counter (i.e. 00 – 01 – 10 – 11) with enable input x.

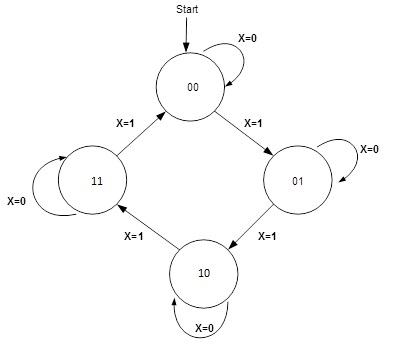


Figure 2: State diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1: State Table** | | | | |
| Present State | | Input | Next state | |
| Q0 | Q1 | X | Q0\_nxt | Q1\_nxt |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

1. Fill in the state table with D flip-flop Excitations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 2: State Table with D flip-flop Excitations** | | | | | | |
| Present State | | Input | Next state | | Flip-flops Inputs | |
| Q0 | Q1 | X | Q0\_nxt | Q1\_nxt | D0 | D1 |
| 0 | 0 | 0 |  |  |  |  |
| 0 | 1 | 1 |  |  |  |  |
| 1 | 0 | 1 |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |
| 0 | 1 | 1 |  |  |  |  |
| 0 | 0 | 0 |  |  |  |  |
| 1 | 1 | 1 |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |

1. Use K-Map and generate simplest Boolean expressions for the inputs of the flip flops (D0, D1).
2. Implement the sequential circuit on **Circuit Maker**.

|  |
| --- |
|  |

**Implement it on explorer board too.**

**Task # 3.** Implement the code given below using logic gates. You can use any flip-flop.

#include<iostream>

using namespace std;

void main()

{

int i;

for (i = 0; i < 8; i+=2)

{

cout << i;

}

}

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present State |  |  | Next state |  |  | Flip flop input |  |  |
| Qa | Qb | Qc | Qa | Qb | Qc | D(Qa) | D(Qb) | D(Qc) |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |