3 BIT SYNCHRONOUS UP COUNTER USING JK FLIP FLOP

Student: Butaev Temur

Id:U1910035

Section:001

Introduction

During this term in the course of digital logic circuits we studied two types of counters. First type of counter is called asynchronous and the second called synchronous. The word asynchronous says doing anything one by one and the word synchronous means doing anything simultaneously. In our case these words are used with the same meaning. In asynchronous counters output(Q) of one flip is used as input for clock in the next flip flop. Therefore the outputs of flip flops are derived one after another. Asynchronous counters are very easy to build and does not need logic gates and n-bit asynchronous counter is build in the same principle as n-1-bit asynchronous counter. But synchronous counters totally different from asynchronous counters.

Definition of the problem

Synchronous counters use one clock for whole circuit and input for every flip flop clock is got from this clock. In synchronous counters every flip flop does not need to wait for previous flip flop for clock input and n-bit synchronous counter is constructed differently from n-1-bit.Therefore it is much complex to build synchronous counters than asynchronous. My task to build 3 -bit synchronous counter using JK flip flops and logic gates.

Methodology

For designing 3-bit up counter we definitely need 3 JK flip flop. But we do not know what type of logic gates we are going to use. To find out this we should pay attention to the behavior of JK flip flop by looking to the excitation table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Present State** | | | **Input for Flip-flop** | | | | | |
| **QC** | **QB** | **QA** | **JC** | **KC** | **JB** | **KB** | **JA** | **KA** |
| 0 | 0 | 0 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | X | 1 | X | X | 1 |
| 0 | 1 | 0 | 0 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 1 | X | X | 1 | X | 1 |
| 1 | 0 | 0 | X | 0 | 0 | X | 1 | X |
| 1 | 0 | 1 | X | 0 | 1 | X | X | 1 |
| 1 | 1 | 0 | X | 0 | X | 0 | 1 | X |
| 1 | 1 | 1 | X | 1 | X | 1 | X | 1 |
| 0 | 0 | 0 | 1 | X | 1 | X | X | 1 |

Now we have to create logic function for every input. To do this we have to create Karnaugh map for every input.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | x | x | x | x |

JC=QBQA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | x | x | x | x |
| 1 | 0 | 0 | 1 | 0 |

KC=QBQA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | x | x |
| 1 | 0 | 1 | x | x |

JB=QA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | x | x | 1 | 0 |
| 1 | x | x | 1 | 0 |

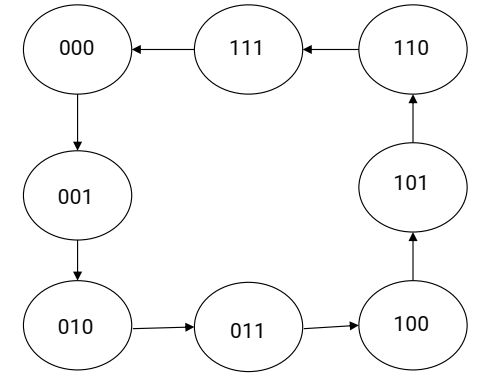
KB=QA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | 1 | x | x | 1 |
| 1 | 1 | x | x | 1 |

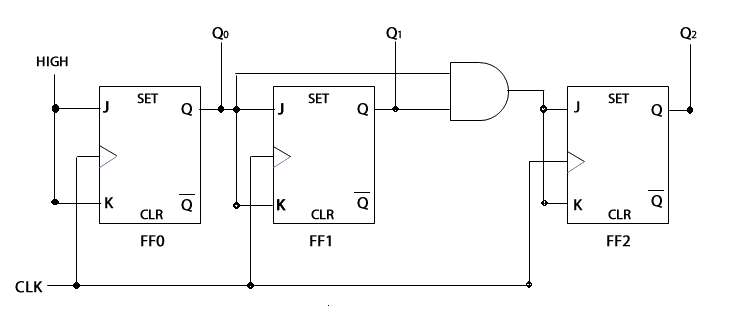
JA=1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| QCQBQA | 00 | 01 | 11 | 10 |
| 0 | x | 1 | 1 | x |
| 1 | x | 1 | 1 | x |

KA=1

 State Diagram

Implementation



Timing Diagram

