*SPH 336*

*COMPUTING LABORATORY II*

*GROUP D*

*REPORT ON FLIP FLOP 4-BIT COUNTER*

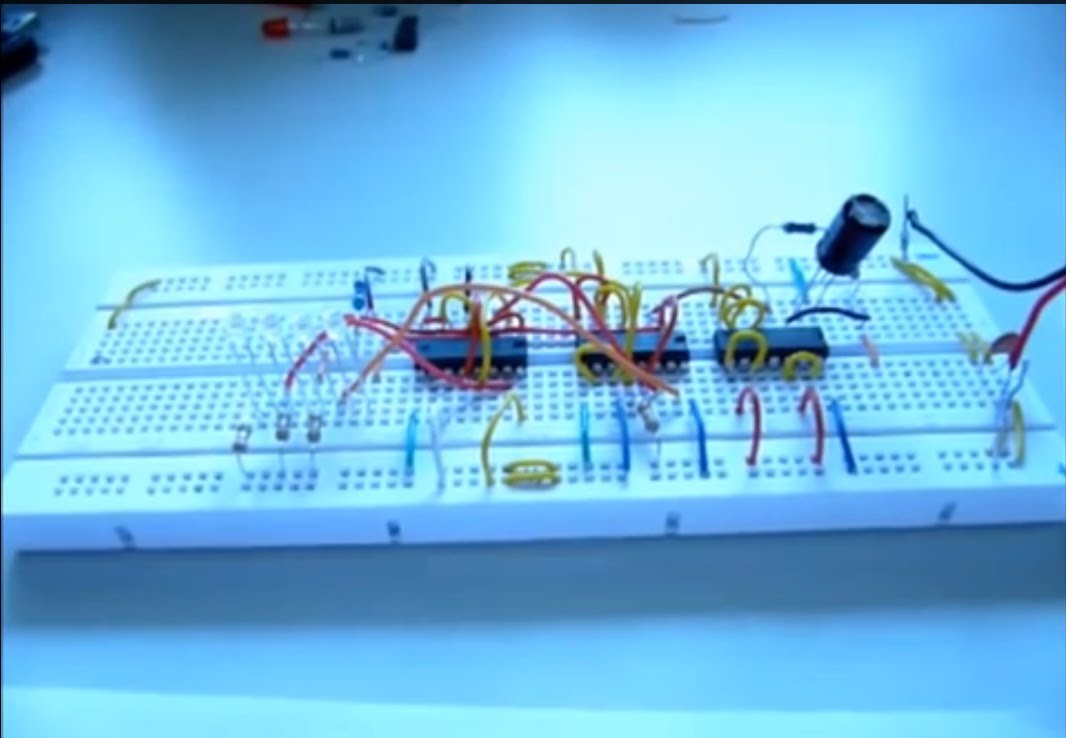
*MEMBERS*

1.PHINEAS MUTHAURA I39/2217/2013

2.SAID UMURU JILLO I39/2145/2012

3.ROBERT NJENGA I39/2209/2013

4.DERRICK KIPKIRUI I39/ /2013



**ABSTRACT**

The project involved the design of the schematic as well as the layout. In both cases, simulations were done. The counter is simulated under no-load condition.

. The rise time, fall time and delay are measured.

We designed an Asynchronous four bit counter using T flip flops.

INTRODUCTION

A Flip-flop is the name given to two-state devices which offer basic memory for sequential logic operations. Flip-flops are synchronous bistable devices that operate as memory elements. A flip-flop circuit contains two outputs, one is for the normal value and the other is for the complement value of the stored bit. Flip-flops are used for digital data storage and transfer and are commonly used in banks called "registers" for the storage of binary numerical data.

Counters are designed using flip-flops. Counters can be classified as synchronous and asynchronous counters based on the application of clock to the flip-flops. A synchronous counter is clocked by a single clock for all the stages and the output for each stage changes at the same time. In an asynchronous counter the output from the previous stage is given as the clock for the next stage so that the output ripples across each stage to reach the final count.

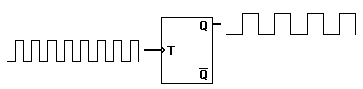
The following were the steps involved in designing this project:

1. Design a flip flop using nand gates and an inverter
2. Draw the schematic and layout using cadence
3. Design a four bit counter using the designed flip flop
4. Draw the schematic and layout of the counter using cadence
5. Measure the rise time and fall time of the various bits
6. Measure the propagation delay across various stages with different capacitive loads

We have designed a four bit asynchronous counter with the aid of T flip-flops.

INTRODUCTION TO T FLIP FLOP:

The T or "toggle" flip flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. T flip flop is useful for constructing binary counters, frequency dividers, and general binary addition devices

. 

A T flip flop can be designed from a J K flip flop by shorting the J and K inputs.

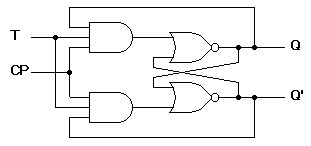


Diagram of a T flip-flop



Transition table of T Flip Flop

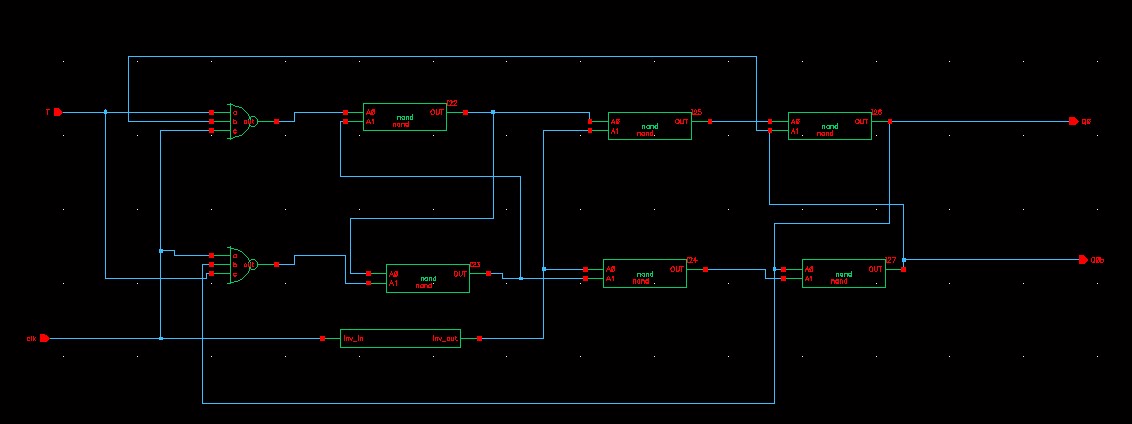
OBJECTIVE

The objective of this project is to design a 4-bit counter and implement it.

The chosen design for the 4-bit counter is a simple 4-bit synchronous counter with synchronous set and reset option and input and output carry option. The reasons behind choosing this design are

1. Synchronous counter is the most used and reliable counter design
2. Synchronous design ensures that all the output bits change simultaneously at the edge of a clock signal and holds that output until the next clock signal
3. Low propagation delay than asynchronous counter iv. The set reset options become effective with clock edge signal, so the outputs do not change suddenly in mid-clock-period.
4. Set/ reset options allows to clear the count data to start new counting session.

SCHEMATIC OF T FLIP FLOP:



***Truth tables and K-maps:***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Present state** | **Next state** | **J3** | **K3** | **J2** | **K2** | **J1** | **K1** | **J0** | **K0** |
| **0000** | 0001 | 0 | X | 0 | X | 0 | X | 1 | X |
| **0001** | 0010 | 0 | X | 0 | X | 1 | X | X | 1 |
| **0010** | 0011 | 0 | X | 0 | X | X | 0 | 1 | X |
| **0011** | 0100 | 0 | X | 1 | X | X | 1 | X | 1 |
| **0100** | 0101 | 0 | X | X | 0 | 0 | X | 1 | X |
| **0101** | 0110 | 0 | X | X | 0 | 1 | X | X | 1 |
| **0110** | 0111 | 0 | X | X | 0 | X | 0 | 1 | X |
| **0111** | 1000 | 1 | X | X | 1 | X | 1 | X | 1 |
| **1000** | 1001 | X | 0 | 0 | X | 0 | X | 1 | X |
| **1001** | 1010 | X | 0 | 0 | X | 1 | X | X | 1 |
| **1010** | 1011 | X | 0 | 0 | X | X | 0 | 1 | X |
| **1011** | 1100 | X | 0 | 1 | X | X | 1 | X | 1 |
| **1100** | 1101 | X | 0 | X | 0 | 0 | X | 1 | X |
| **1101** | 1110 | X | 0 | X | 0 | 1 | X | X | 1 |
| **1110** | 1111 | X | 0 | X | 0 | X | 0 | 1 | X |
| **1111** | 0000 | X | 1 | X | 1 | X | 1 | X | 1 |

***K-maps:***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00 | 01 | 11 | 10 | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 00 | x | x | x | X |
| 0 | 0 | 1 | 0 | 01 | x | X | x | x |
| x | x | x | X | 11 | 0 | 0 | 1 | 0 |
| x | x | x | x | 10 | 0 | 0 | 0 | 0 |

Q1 Q0

Q3 Q2

00 01 11

10

J3 = Q2Q1Q0 K3 = Q2Q1Q0

|  |  |  |  |
| --- | --- | --- | --- |
| 00 | 01 | 11 | 10 |

Q1 Q0

Q3 Q2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 00 | 0 | 0 | 1 | 0 | | 01 | X | X | X | X | | 11 | X | X | X | X | | 10 | 0 | 0 | 1 | 0 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 | | 00 | x | x | x | X | | 01 | 0 | 0 | 1 | 0 | | 11 | 0 | 0 | 1 | 0 | | 10 | x | x | x | X | |

J2 =Q1Q0 K2 = Q1Q0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 | | 00 | 0 | 1 | x | X | | 01 | 0 | 1 | x | X | | 11 | 0 | 1 | x | x | | 10 | 0 | 1 | x | x | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 | | 00 | x | x | 1 | 0 | | 01 | x | x | 1 | 0 | | 11 | x | x | 1 | 0 | | 10 | x | x | 1 | 0 | |

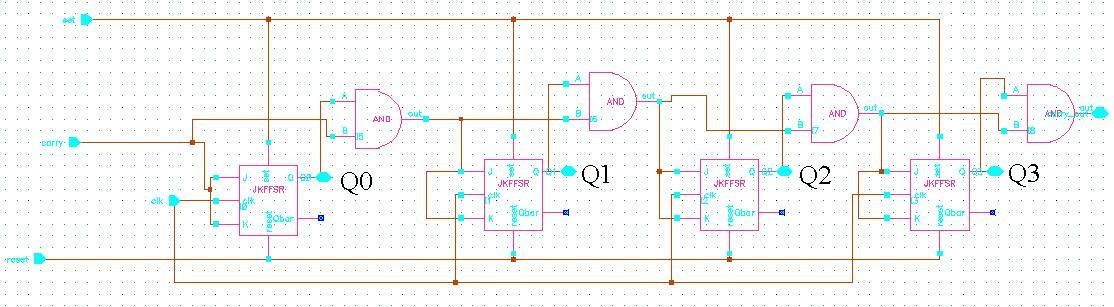
J1 =Q0 K1 =Q0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 | | 00 | 1 | 1 | 1 | 1 | | 01 | x | x | x | x | | 11 | x | x | x | X | | 10 | 1 | 1 | 1 | 1 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Q1 Q0  Q3 Q2 | 00 | 01 | 11 | 10 | | 00 | x | x | x | X | | 01 | 1 | 1 | 1 | 1 | | 11 | 1 | 1 | 1 | 1 | | 10 | x | x | x | x | |

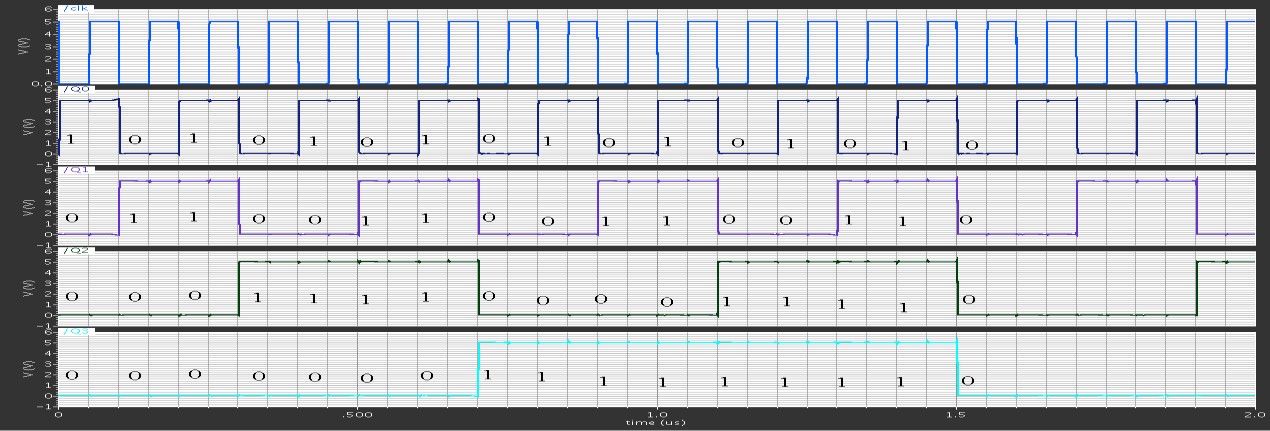
J0 = 1 K0 = 1

***Logic circuit:***

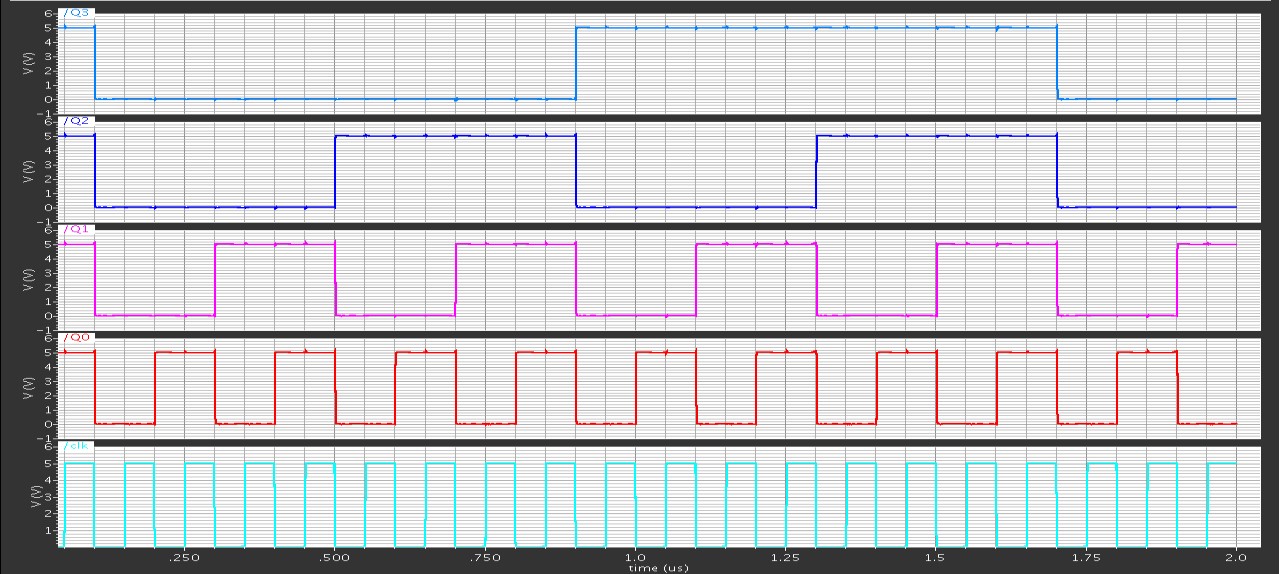
*Schematic of 4-bit counter:*



**Pre-layout simulation output:**

\

**simulation output: POST**



CONCLUSION

The 4-bit counter designed above consisted of the following files: counter.cpp, counter.h,driver.h,monitor.h. The codes were compiled and tested in system c and were found to run successfully .This is shown from the screenshot of the output as shown above.