PROJECT REPORT

EEE-244

SRIHARSHA JADHAV
SUBMITTED TO:
DR PREETAM KUMAR

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING-CALIFORNIA STATE UNIVERSITY SACRAMENTO

TABLE OF CONTENTS:

1. INTRODUCTION:

Counters are devices that can generate a sequence of numbers at equal intervals of time and repeats the same sequence once the end of the previous sequence is reached. [2]

There are several ways this counting can be done. The most common counters are either mechanical counters or electrical/ digital counter.

Mechanical counters make use of a series of disks mounted on an axle, with digits from 0 to 9 marked on the disks. The right most disk moves in one step increments. When the disk completes one rotation it moves the disk to the left by one step. Thus, increasing the numbers on the second disk by 1. [2]

Electronic counter does the counting process using electric power and circuitry. The electrical counters use the same technique as mechanical counters. Instead of disks and axles, electronic counters are usually constructed using several flip-flops connected in series or parallel. Electronic counters are a commonly used component in digital circuits, and are manufactured as separate IC or incorporated as parts of larger ICs.[1]

Counter find its application in many everyday devices where counting or time keeping is involved. They are used in devices like oven timers, televisions, VCR/CD players, digital clocks, memory devices (mechanical and solid state).

The most common of the counter applications are as follows:[6]

1. A Digital Clock:

Counters finds its application is the timekeeping systems. Counters can be designed to measure time very accurately by using a very stable frequency (internal clock source like quartz). Digital clocks are used in many daily devices like the oven timers, washers, wall clocks, digital wrist watches, cellphones, and computers.

2. Memory devices:

Counter are used in memory systems like buffers, RAMs and other flash storage devices. They are used to keep the track of the read and write pointers. They can be either an up or down counter depending on the read/write operation.

3. Event counters:

Counters can be used in event counters i.e. to keep as track / count the number of times an event has occurred. They are usually used in places like parking lots to keep a track of vacant and occupied slots, ATMs, and currency counting machines.

4. Frequency dividers:

Since counters like the asynchronous counter use a single clock input, it can be used as an efficient and a simple frequency division circuit. By increasing the number of stages in the counter the division factor can be varied. This type of frequency dividers can be used in SoC where different components have different operating frequencies.

2. COUNTERS:

Digital counters are easy to implement using register-type circuits such as the flip-flop. As we all know a flip-flop or latch is a circuit that has two stable states i.e. 0’s and 1’s. It can be made to change state by applying signals to its control inputs. Based on how the counters operate and how these flipflops are arranged counters can be widely classified as:

1. Asynchronous (ripple) counter – changing state bits are used as clocks to subsequent flip-flops
2. Synchronous counter – all bits change under a single clock

Asynchronous (Ripple) Counters[4]

In this counter, the output is not dependent entirely on the input clock of the device. The input clock is connected to only the first flip flop. The output of the first flip flop controls the next and so on.

Below *fig 1* is a circuit diagram of a 4-bit asynchronous counter using JK flip-flops. The main clock is connected to clock input of the 1st flip-flop only. So, FF-0 changes state at the falling edge of each clock pulse, but FF-1 changes only when triggered by the falling edge of the A output of FF-0. Because of the internal propagation delay through a flip-flop, the transition of the input clock pulse and a transition of the A output of FF-0 can never occur at the same instance. Therefore, the flip-flops cannot be triggered simultaneously, thus producing an asynchronous operation.

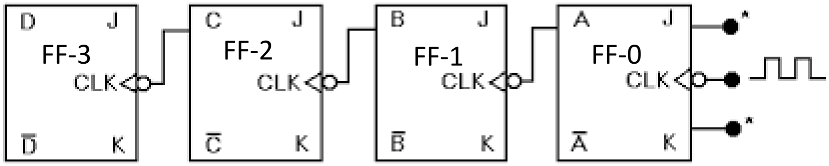

Fig 1: Asynchronous Counter

Synchronous counter[5]

In this type of counters, the input clock of all the flip-flops are connected and are triggered by a common input clock. Thus, all the flip-flops state change simultaneously.

The circuit below *fig 2* is a 4-bit synchronous UP counter using JK flip-flop. The J and K inputs of FF-A are connected to constant value 1. FF-B has its J and K inputs connected to the output of FF-A, and the J and K inputs of FF-C are connected to the output of an AND gate that is fed by the outputs of FF-A and FF-B. A way of implementing the logic is for each bit of an UP counter to toggle when all the less significant bits are at a logic high state. i.e. when the output of FF-A and FF-B go high the FF-C and FF-D are triggered thus making their outputs to go high.

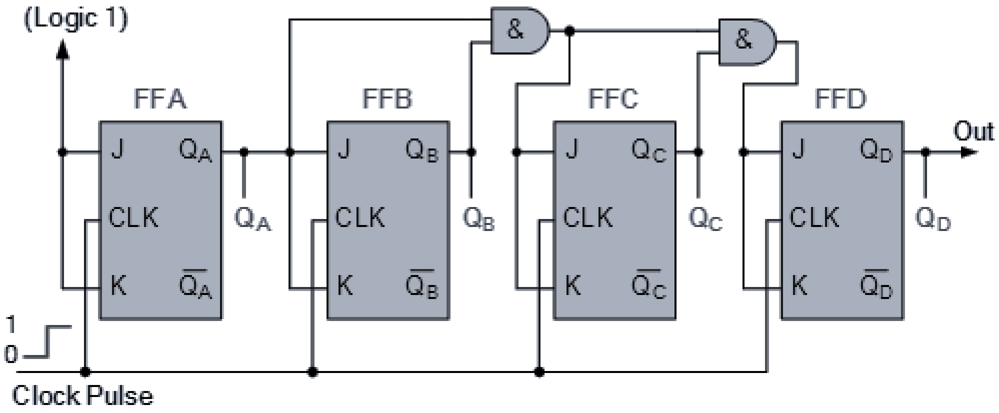

Fig 2: Synchronous UP Counter

3.MORE ON RIPPLE COUNTER

A ripple counter is an asynchronous counter where only the first flip-flop is connected and controlled by the external clock. All subsequent flip-flops are clocked by the output of the previous flip-flop.

Asynchronous counters are called ripple-counters because of the way the clock pulse ripples it way through the flip-flops.

These types of counters are slower than synchronous counters because of the delay in the transmission of the pulses from one flip-flop to another.

The number of outputs of the ripple counter or asynchronous counter is 2^n if n flip-flops are used. For a 4-bit counter, the output sequence will have 16 values and in the range of 0 to 15 (2^4-1).

A counter may count up (increment) or count down (decrement) depending on the input control. The count sequence repeats itself after the count reaches the last count i.e. 2^n-1. Like in case of a 4-bit counter the sequence repeats after the 15th term.

The complement of the count sequence counts in reverse direction that’s the down counter. If the uncomplemented output counts up, the complemented output counts down. i.e. if the up-count sequence of a 4-bit counter is 0000, 0001, 0010, 0011 … then it’s complement would be 1111, 1110, 1101, 1100 … this is output of a 4-bit down counter. If the uncomplemented output counts down, the complemented output counts up. i.e. if the down-count sequence of a 4-bit counter is 1111, 1110, 1101, 1100 … then it’s complement would be 0000, 0001, 0010, 0011…. this is output of a 4-bit down counter.

Ripple counter can be implemented in many ways, based on the characteristics of the flip flops used and the requirements of the count sequence.

- Clock Trigger: Positive edged or Negative edged
- JK or D flip-flops
- Count Direction: Up, Down.

a. D FLIP-FLOP:

D flip-flop is a commonly used flip-flop, it is also known as a "data" or "delay" flip-flop it’s based on JK latch. It has one input signal (D) and two output signals (Q and Qbar) and two control signals clock and clear/reset. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge or falling edge of the clock). That captured value becomes the Q output and the invert of Q i.e. ~Q (Q bar). At other times, the output Q does not change. The D flip-flop can be viewed as a memory cell, a zero-order hold, or a delay line. *Fig 3* is a general representation of the D flip flop.

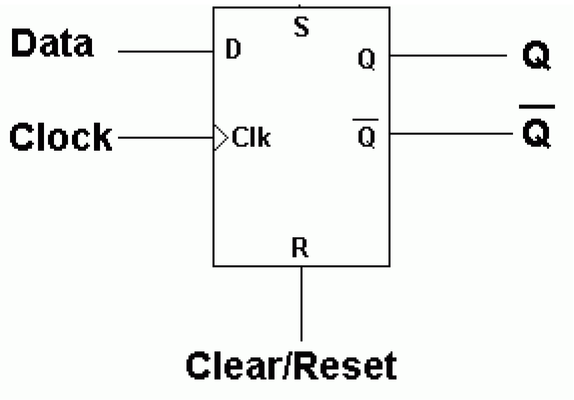

Fig 3: D-flip flop (using JK)

Truth table below *table1* shows the possible inputs and the corresponding outputs at the clock events.

clock	**D**	**Q**	**Qbar**
Rising edge	0	0	1
Rising edge	1	1	0
Non-Rising	X	Q	~Q

Table 1: truth table of D-flip flop

Below *fig 4* is the waveform of the D-flip flop.

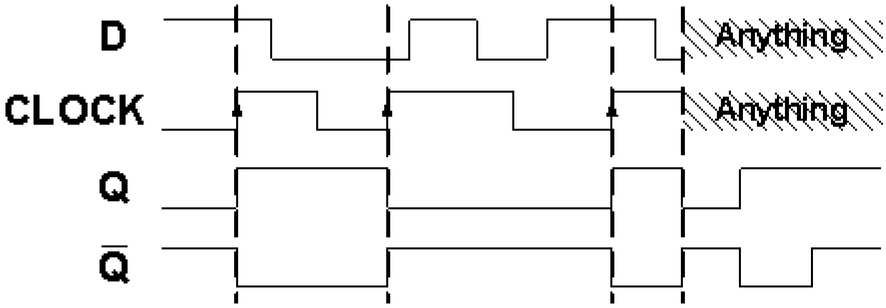

Fig 4: D-flip flop waveform

4. MATLAB SIMULINK SIMULATION MODEL:

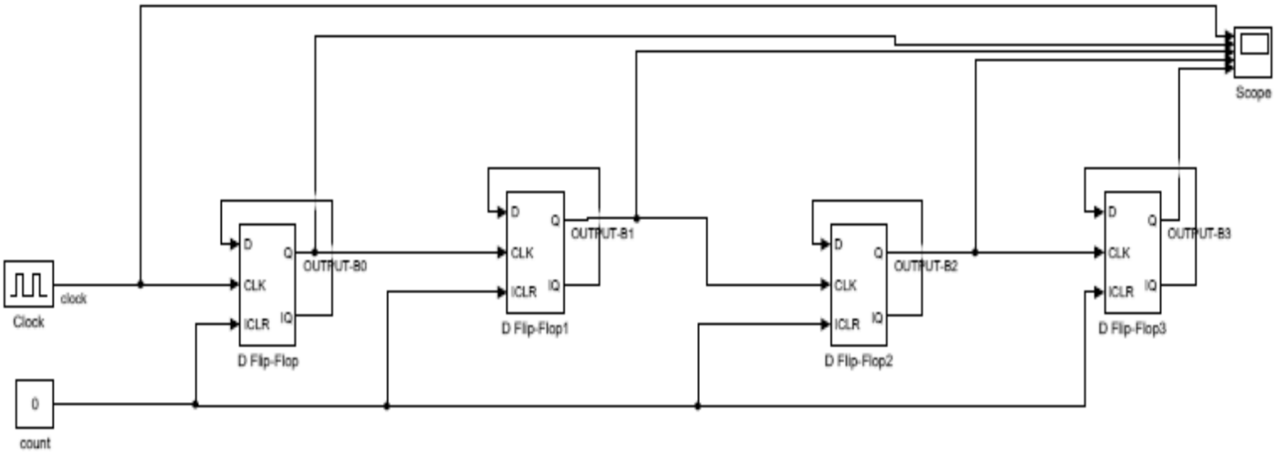

Fig 5: block diagram of 4-bit Ripple counter.

Above is the circuit diagram *fig 5* of a 4-bit counter that counts-up. This is simulated using the MATLAB Simulink. The blocks used are taken from the standard library that's available in the MATLAB Simulink.

For this implementation of the 4-bit counter uses 4 D-FF, a clock generator, constant value generator and a 5-input scope.

The D-flip-flop block has 5 pins 3 input D, CLK and !CLR and 2 output pins Q and Q~ (non-inverting and inverting output).

CLOCK: This block generates a square wave of a desired period/ frequency. Its connected to the CLK pin of the FF-0. For this simulation, its period has been set of 0.05s or a frequency of 20hz.

The clock that is generated in this block is only applicable to the FF-0. The reset of the flip-flops doesn't get the clock input from the clock generator as it is an asynchronous counter. The effect of the clock is rippled through the counter from the first flip-flop.

Higher the frequency the faster the counter generates the numbers. Longer the period longer the counter retains its current value.

COUNT (CONSTANT SIGNAL): block generates a constant signal of value 1. This constant value signal is connected to the! CLR pin of the D-FF. We know that the D-FF used has a reset/clear which is an active low in nature, meaning when the reset/clear signal is low the device is reset i.e. the output becomes 0000.

SCOPE: this block is a 5-input scope. Its function is to display the signal connected to it in the form of a wave. The number of inputs to the scope module can be changed depending on the number of signals to be read.

The Q output lines from the 4 flip-flops and the input clock is connected to the inputs of the scope. This block displays the outputs in the form of waveforms. It displays the waveforms of all the 5 inputs simultaneously.

5. RESULTS:

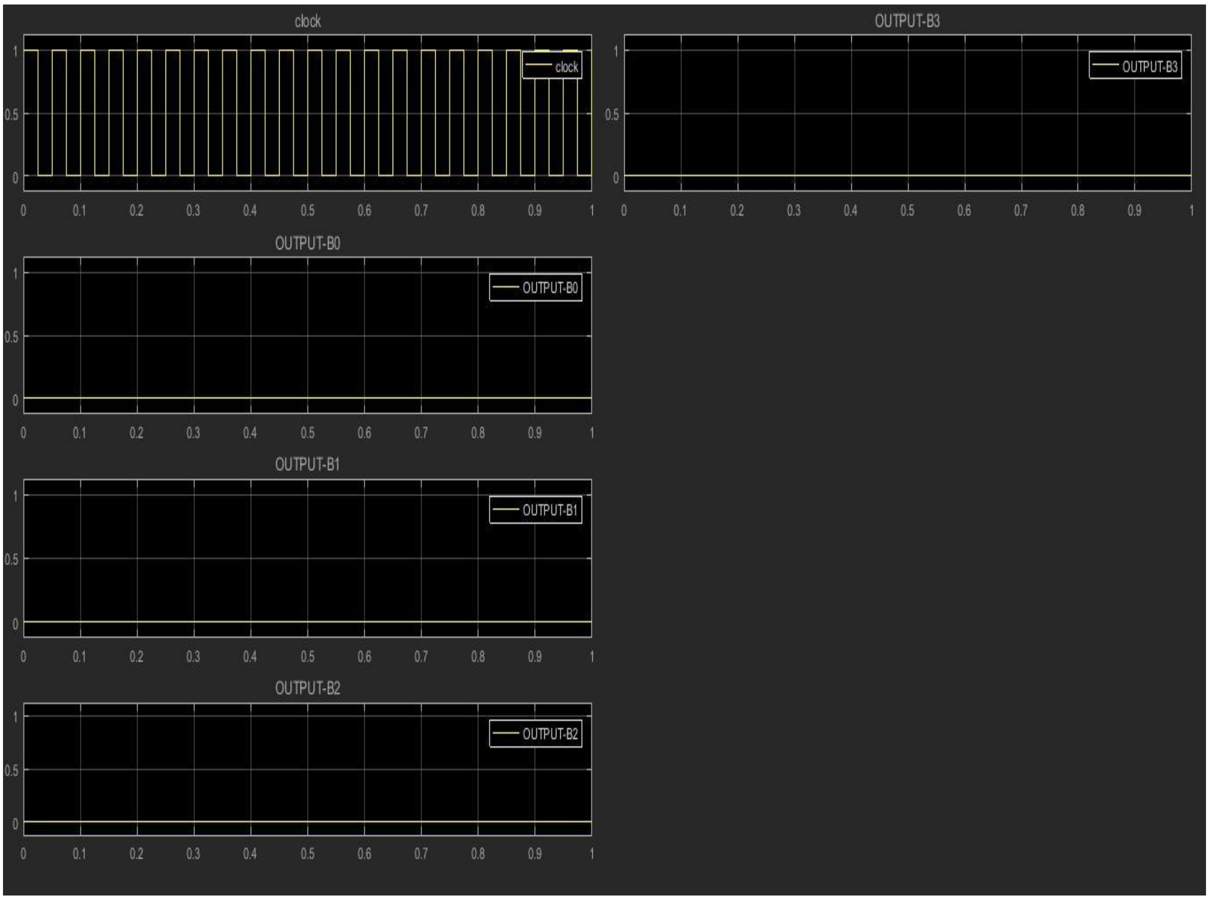

Fig 6: 4-bit counter output when count=0.

This is the output waveform of the 4-bit counter that's displayed by the scope. As we know that it has a count input signal which controls the output of the counter. The above graph *fig 6* shows the output of the 4-bit counter when the count signal is low or 0. We can observe that all the 4 output bit lines are 0, so the value is 0000.

This can be used as a reset pin for the counter, as it resets all the output lines to low or 0. This reset option can be used in devices where its required to reset the device once it's used like a digital stop clock where it needs to be reset to use it again.

The figure below shows the output of the 4-bit counter when the count=1, i.e. when the count is enabled. We know that when the count is enabled the counter starts counting.

Here we can observe that the period of every output wave is increasing as it ripples through the next flip flops. The output waveform from the first flip-flop has a period equal to twice the clock period and frequency equal to half of the input clock frequency.

In the second waveform *fig 7*, we can see that the period is twice the 1st waveform and the frequency is half of the 1st. This is due to the asynchronous nature of the counter. The 2nd flip-flop gets a clock which is equal to the output of the 1st flip-flop. This happens with every passing flip-flop the period doubles and frequency reduces by half.

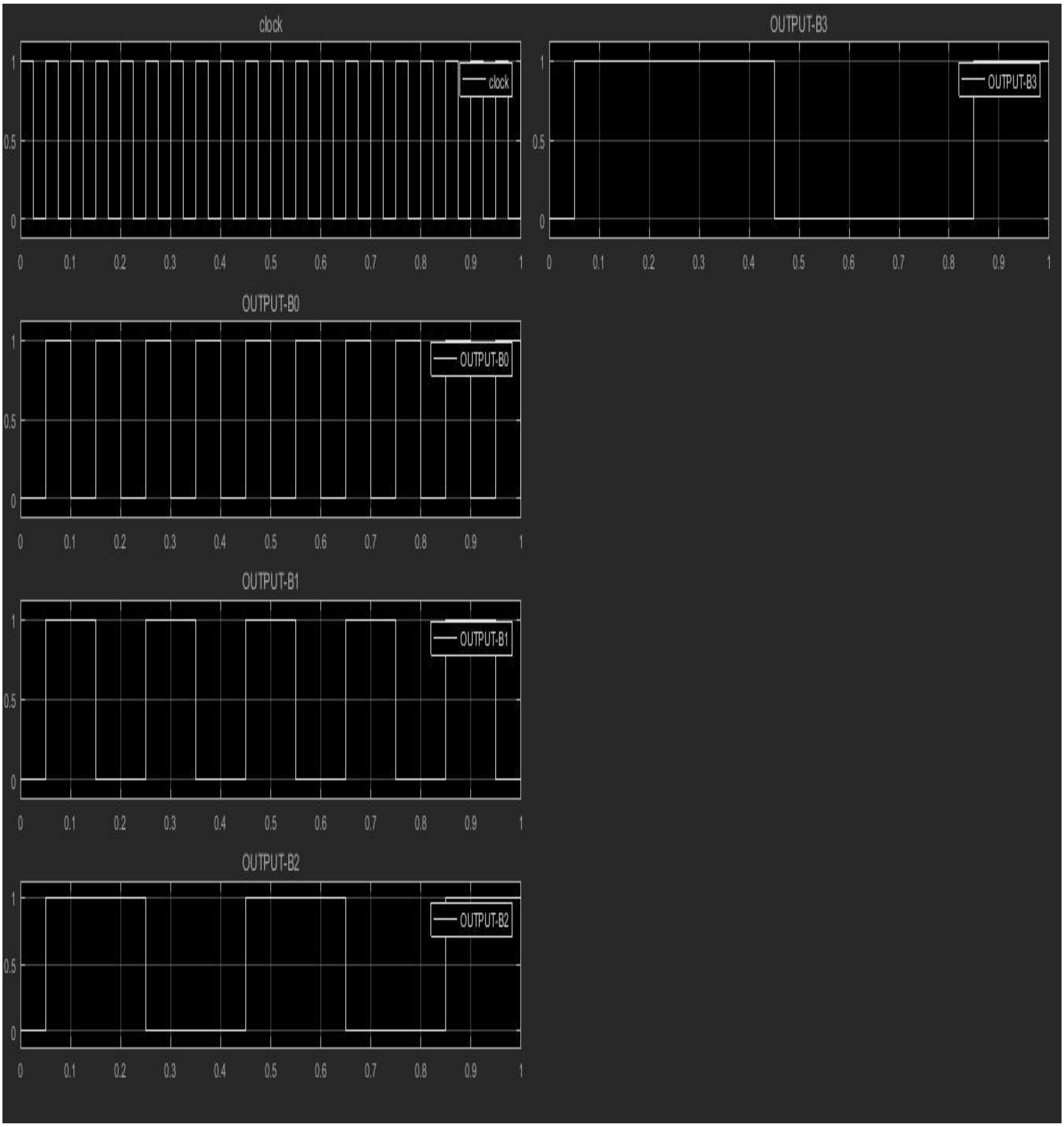

Fig 7: 4-bit counter output when Count=1.

In digital circuits, we consider a high voltage/signal an output ‘1’ and a low voltage/signal as output ‘0’. Now we can now apply the same to the counters. So, we can see that we have a sequence of ‘0’ and ‘1’ in a specific pattern.

COUNT	CLOCK	B4	B3	B2	B1	Decimal output
1	1	1	1	1	1	15
1	0	1	1	1	0	14
1	1	1	1	0	1	13
1	0	1	1	0	0	12
1	1	1	0	1	1	11
1	0	1	0	1	0	10
1	1	1	0	0	1	9
1	0	1	0	0	0	8
1	1	0	1	1	1	7
1	0	0	1	1	0	6
1	1	0	1	0	1	5
1	0	0	1	0	0	4
1	1	0	0	1	1	3
1	0	0	0	1	0	2
1	1	0	0	0	1	1
1	0	0	0	0	0	0
0	-	0	0	0	0	0

Table 2: truth table for a 4-bit counter

The table above *table 2* shows the signal's binary equivalent B1, B2, B3, and B4 are the outputs of the 1st , 2nd , 3rd and 4th flipflops and the last column is the decimal equivalent of the 4-bit binary number formed by combining them in "B4-B3-B2-B1" form.

We can observe that the counter is counting from 15-0 i.e. from 2^4-1 to 0.

Thus we have successfully designed, simulated and verified a 4-bit asynchronous counter using MATLAB Simulink.

6. REFERENCES:

1. Floyd, Thomas l. Digital Fundamentals. USA: Pearson education international, 2006
2. Counter(digital): https://en.wikipedia.org
3. New Jersey Institute of Technology. ECE394 – Digital Systems Laboratory. Experiment 7: Counters. [http://web.njit.edu/~gilhc/ECE394/ECE394-VII.htm]. (08/15/2017)
4. All About Circuits. Asynchronous counters. [http://www.allaboutcircuits.com/vol_4/chpt_11/2.html]. (08/15/2017)
5. All About Circuits. Synchronous counters. [http://www.allaboutcircuits.com/vol_4/chpt_11/3.html]. (08/15/2017)
6. Digital Electronics Basics - Chapter 4: Counters [http://www.ni.com/example/14496/en/#toc2]Publish Date: Mar 27, 2013
7. DAEnotes Counters [http://www.daenotes.com/electronics/digital-electronics/counters-types-of-counters] (08/15/2017)