INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

LOW PROPAGATION DELAY DESIGN OF 3-BIT RIPPLE COUNTER ON 0.12 MICRON TECHNOLOGY

[1]Rachit Manchanda, [2]Rajesh Mehra
[1]ME STUDENT (ECE) ,[2]ASSISTANT PROFESSOR (ECE)
[1]rachit.tech2@gmail.com, [2]rajeshmehra@yahoo.com
NATIONAL INSTITUTE OF TECHNICAL TEACHERS TRAINING AND RESEARCH, CHANDIGARH

Abstract

The evolution of integrated circuit (IC) fabrication techniques is a unique fact in the history of modern industry. The improvements in terms of low propagation delay, density and cost have kept constant for more than 30 years. The main improvements in terms of feature size reduction for MOS devices is increased number of metal interconnects to link MOS together within the chip. Consequently, the clock frequency of the chip has never stopped increasing, with an expected 800MHz. This paper compares counter design on the basis of parameters like speed (low propagation delay) ,power consumption ,layout area using two different CMOS technology using microwind CMOS layout tool. Thus it provides solution to a high speed implementation of counter in CMOS VLSI.

Index Terms— Microwind, micron Technology, layout,
Ripple counter, DSCH2, low propagation delay.

1. INTRODUCTION

Counter is a digital circuit which is able to count from a specific number to another specific number. A simple application of a counter is a digital clock having 3 counters, one for counting the seconds, another for counting the minute and the last one for counting the hour. An M-bit counter can be created by cascading M stages of flip-flops. This creates a divide-by-N counter where N = 2M. Such a counter is called a ripple counter since there is a ripple or domino effect as one flip-flop overflows into the next stage. The propagation delay between the input clock and when the last stage settles is the accumulated delay of each stage. Thus, all stages do not change at the same time. The simplest counter circuit is a single D-type flip flop, with its D (data) input fed from its own inverted output. This circuit can store one bit, and hence can count from zero to one before it overflows starts over from 0. This counter will increment once for every clock cycle and takes two clock cycles to overflow, so every cycle it will alternate between a transition from 0 to 1 and a transition from 1 to 0. Notice that this creates a new clock with a 50% duty cycle at exactly half the frequency of the input clock. If this output is then used as the clock signal for a similarly arranged D flip flop you will get another 1 bit counter that counts half as fast. Putting them together yields a two bit counter. You can continue to add additional flip flops, always inverting the output to its own input, and using the output from the previous flip flop as the clock signal. The result is called a ripple counter, which can count to 2n-1

where n is the number of bits in the counter. Ripple counters suffer from unstable outputs as the overflows "ripple" from stage to stage, but they do find frequent application as dividers for clock signals.

I. DESIGN STEPS FOR RIPPLE-COUNTER

1. *Selection of type up counter-*

An up counter is simply a digital counter which counts up at some predefined increment. A Binary up Counter with 'n' stages can count up to 2n states. If we are implementing Up Counter with flip flops, this 'n' stages becomes the number of flip flops. Here 3 bit Up Counter can count from binary 000 to 111, i.e. 8 states.

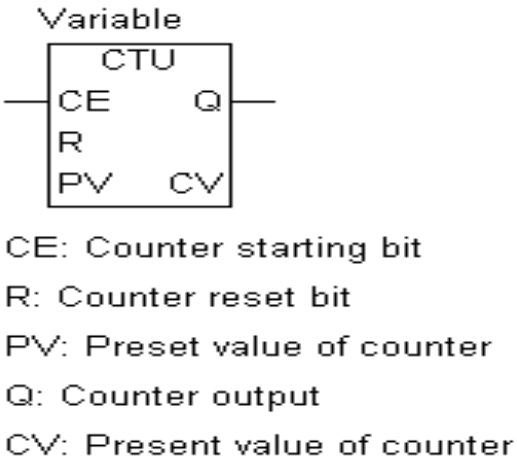


Fig.1 Basic Diagram of Up- Counter

When the counter trigger bit CE passes power, the current value CV is decremented by one. When the current value becomes equal to or less than zero after decrementing, the counter output bit Q is turned ON, and the instruction passes power.

Modules

A Counter Module (also known as a Counter) is a cog's configurable state machine for generating or sensing repetitive signals. Each cog has two independent counter modules, Counter "A" and Counter "B."The counter modules provide simple, flexible subsystems for each cog to perform repetitive tasks on potentially every clock cycle; they can often take the place of dedicated peripheral driver hardware, reducing component count in an application.

Counters can provide a cog with a variety of services. These can be used to:

1. Generate numerically-controlled oscillator (NCO) signals, like square waves
2. Measure pulse and decay durations for variable-resistance or variable-capacitance sensors
3. Count signal cycles and measure its frequency
4. Detect signal edge or level
5. Perform Digital to Analog (D/A) conversion
6. Perform Analog to Digital (A/D) conversion
7. Provide internal signals for timing and video generation

2. Select flip- flop type-

D Flip Flop-

In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information. A flip-flop is a Bi-stable Multi-vibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite-state machine, the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing

signal. Flip-flops can be either simple (transparent or opaque) or clocked (synchronous or edge-triggered); the simple ones are commonly called latches. The word latch is mainly used for storage elements, while clocked devices are described as flip-flops.

The D flip-flop is widely used. It is also known as a data or delay flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock). That captured value becomes the Q output. 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.

These flip-flops are very useful, as they form the basis for shift registers, which are an essential part of many electronic devices. The advantage of the D flip-flop over the D-type "transparent latch" is that the signal on the D input pin is captured the moment the flip-flop is clocked and subsequent changes

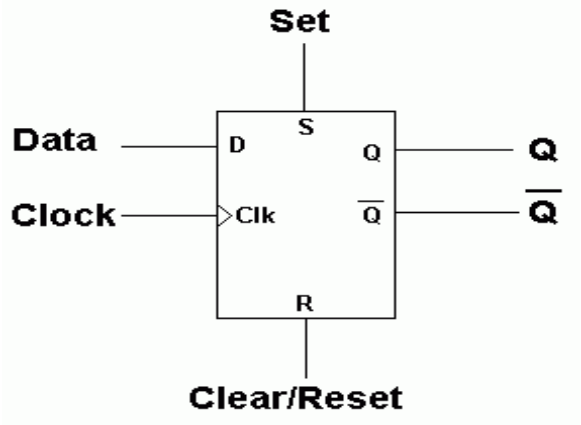
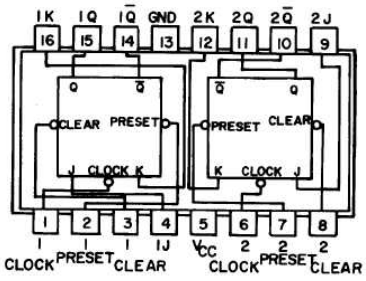


Fig.2 Symbol of D Flip-Flop

Table1 Truth Table of D Flip Flop

Before Clock Pulse	After clock Pulse	
INPUT D	OUTPUT Q	OUTPUT Q
0	0	1
1	1	0



On the D input will be ignored until the next clock event. An exception is that some flip-flops have a "reset" signal input, which will reset Q (to zero), and may be either asynchronous or synchronous with the clock.

3. Edge Triggering of a flip-flop-

This circuit consists of two stages implemented by SR NAND latches. The input stage (the two latches on the left) processes the clock and data signals to ensure correct input signals for the output stage (the single latch on the right). If the clock is low, both the output signals of the input stage are high regardless of the data input; the output latch is unaffected and it stores the previous state. When the clock signal changes from low to high, only one of the output voltages (depending on the data signal) goes low and sets/resets the output latch: if D = 0, the lower output becomes low; if D = 1, the upper output becomes low. If the clock signal continues staying high, the outputs keep their states regardless of the data input and force the output latch to stay in the corresponding state as the input logical zero

remains active while the clock is high. Hence the role of the output latch is to store the data only while the clock is low.

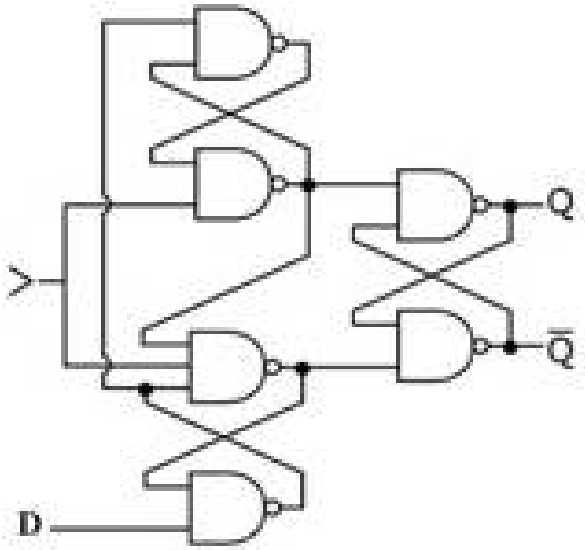


Fig.3 Edge Trigger D Flip-Flop

The circuit is closely related to the gated D latch as both the circuits convert the two D input states (0 and 1) to two input combinations (01 and 10) for the output SR latch by inverting the data input signal (both the circuits split the single D signal in two complementary S and R signals). The difference is that in the gated D latch simple NAND logical gates are used while in the positive-edge-triggered D flip-flop SR NAND latches are used for this purpose. The role of these latches is to "lock" the active output producing low voltage. Thus the edge-triggered D flip-flop can be thought of as a gated D latch with latched input gates. An efficient functional alternative to a D flip-flop can be made with dynamic circuits as long as it is clocked often enough; while not a true flip-flop, it is still called a flip-flop for its functional role. While the master–slave D element is triggered on the edge of a clock, its components are each triggered by clock levels. The "edge-triggered D flip-flop", as it is called even though it is not a true flip-flop, does not have the master–slave properties. Edge-triggered D flip-flops are often implemented in integrated high-speed operations using dynamic logic. This means that the digital output is stored on parasitic device capacitance while the device is not transitioning. This design of dynamic flip flops also enables simple resetting since the reset operation can be performed by simply discharging one or more internal nodes. A common dynamic flip-flop variety is the true single-phase clock (TSPC) type which performs the flip-flop operation with little power and at high speeds. However, dynamic flip-flops will typically not work at static or low clock speeds: given enough time, leakage paths may discharge the parasitic capacitance enough to cause the flip-flop to enter invalid states.

II. OPERATION OF COUNTER

A good first thought for making counters that can count higher is to chain Divide-by-2 counters together. We can feed the Q out of one flop into the CLK of the next stage. The result looks something like this:

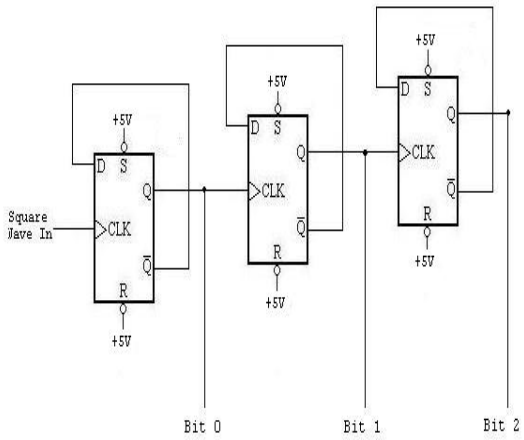


Fig.4 3 Bit Ripple Counter using D flip-flop

The ripple counter is easy to understand. Each stage acts as a Divide-by-2 counter on the previous stage's signal. The Q out of each stage acts as both an output bit, and as the clock signal for the next stage.

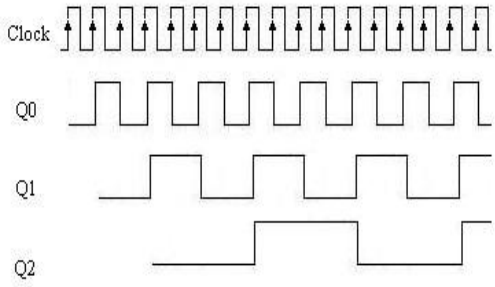


Fig.5 Timing Diagram of 3 Bit Ripple Counter

We can chain as many ripple counters together as we like. A three bit ripple counter will count 8 numbers, and an n-bit ripple counter will count 2n numbers. Now we can see that the propagation delay does not only slow down the counter. The problem with ripple counters is that each new stage put on the counter adds a delay. This propagation delay is seen when we look at a less idealized timing diagram

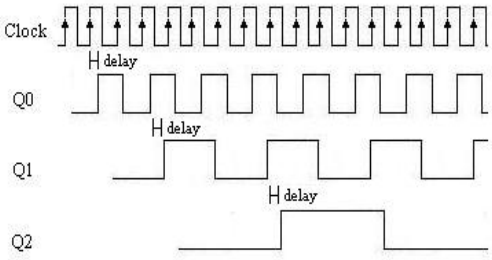


Fig.6 Timing Diagram of 3 Bit Ripple Counter showing Propagation Delay

Now we can see that the propagation delay does not only slow down the counter, but it actually introduces errors into the system. These errors increase as we add additional stages to the ripple counter. Firstly counter is designed by using .18 micron technology and simulate with microwind tools.

The following is a 3-BIT Ripple counter and its timing diagram for one cycle .It is having 8 states due to 3 Flip flop. This ripple counter displays from 000 to 111 binary number constructed using master slave arrangement of D flip flops. Here firstly the counter design implemented in DSCH 2 tool in fig 7 below:-

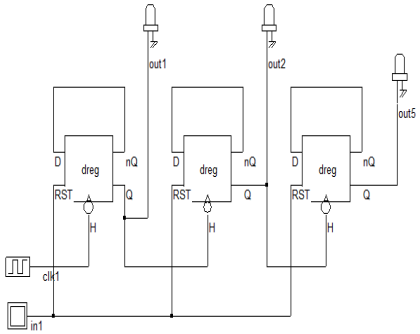


Fig.7 Design of 3 Bit Ripple Counter in DSCH2

Here in above fig reset is connected to all flip flops when reset is high all flip flop set to their initial state i.e. 000 and when reset is low flip flop makes transition from 000 to 111.

Table 2- Three Bit Ripple Binary Counter

FF2	FF1	FF0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The layout of counter is design with Microwind software using .18 micron technology shown in Fig8 :

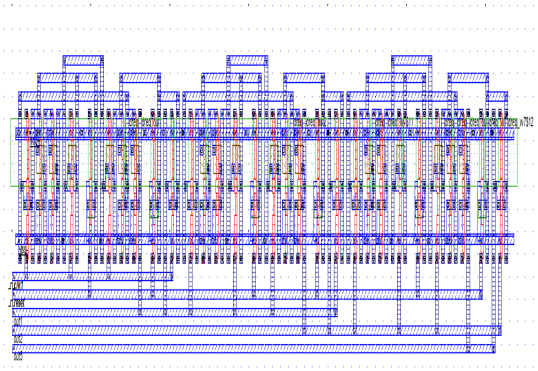


Fig.8 -Layout of Counter using .18micron technology

On the following simulation, we see that the truth-table of the counter is verified. It counts from 000 to 111.The output not fall to "0" when reset starts. This is the output of Ripple counter using .18 micron technology. From this output we get the knowledge of power dissipation and speed of counter.

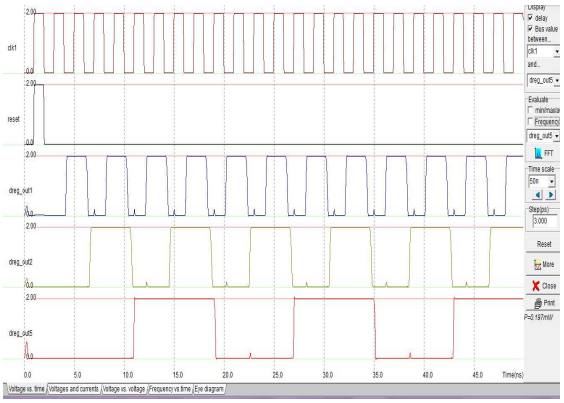


Fig.9- Simulation of Counter using .18 micron technology

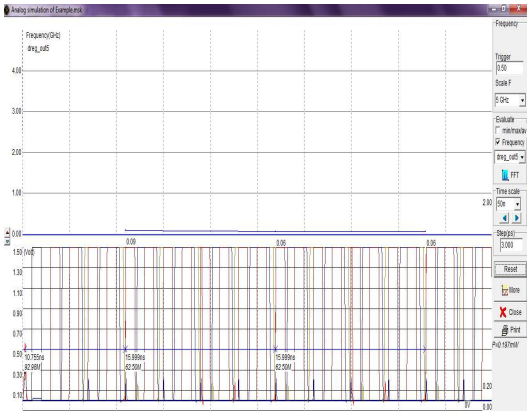


Fig.10 Frequency v/s Time waveform of Counter with reset using .18µm technology

Now this layout of the counter is designed by using 0.12 micron technology and simulate with microwind tools.

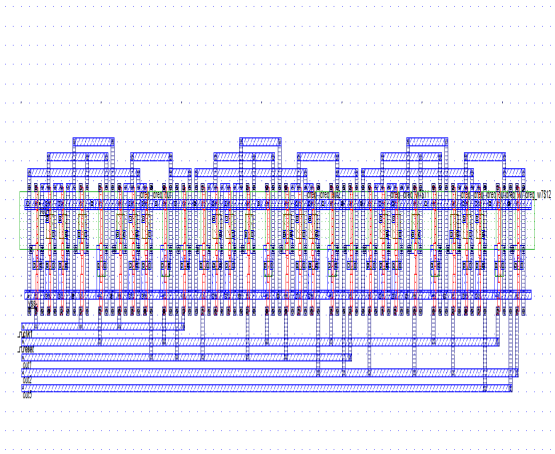


Fig11 -Layout of counter using .12micron technology

This is the output waveform of counter using 0.12 micron technology. This is the timing diagram of 3 bit counter. Initial state of counter is 000 and the last state is 111.

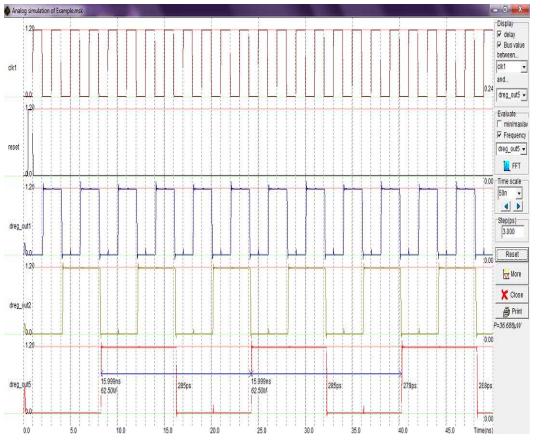


Fig.12- Simulation of Counter using 0.12 micron technology

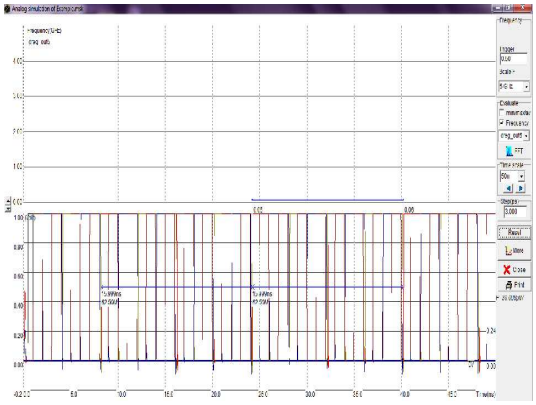


Fig.13- Frequency v/s Time waveform of Counter with Reset using 0.12µm technology

III. CONCLUSIONS

Comparison of 3 bit Ripple counter using 0.18µm using Technology and 0.12µm Technology

S. No.	PARAMETERS	0.18µm Technology	0.12µm Technology
1.	Propagation Delay	10.934 ns	8.2 ns
2.	Power Consumption	0.197mW	0.036mW
3.	No. of transistors used	42	42
4.	Frequency Content	92.98 MHz	122 MHz
5.	Layout Area	1404 μm^2	305.1 μm^2
6.	Capacitance	0.712fF	0.354fF
7.	Pad Size	1176 x 1176 μm^2	1151 x 1151 μm^2

8.	Low Leakage Current	0.327mA	0.203mA
9.	Low Leakage Threshold Voltage	0.50 V	0.40 V

This paper concludes that 3 bit ripple counter is best implemented using the 0.12 micron technology. In this the required propagation delay is minimum i.e. 8.2ns, power consumption is 0.036mwatt , Max operating frequency is 122MHZ, layout size area is 305.1micro sq. meter. Thus that 3 bit ripple counter is best implemented using the 0.12micron technology is preferable over 0.18 micron technologies in maintaining the logic density in fabrication process, power optimization, reducing the propagation delay & surface area. Thus this counter implemented in CMOS chip technology, is the best illustration of VLSI.

REFERENCES

[1]Simmy Hirkaney, Sandip Nemade, Vikash Gupta"POWER EFFICIENT DESIGN OF COUNTER ON .12 MICRON TECHNOLOGY" IJSCE ISSN: 2231-2307, Volume-1, Issue-1, March 2011.
[2] M. Quirk and J. Serda, “Semiconductor manufacturing Technology,”
New Jersey: Prentice Hall, pp.388-434, 2001, IEEE.
[3] Neil H. E. Weste, Principal of CMOS VLSI Design, Pearson
Education, 2003
[4] Prof. Yusuf Leblebici, CMOS Digital Integrated Circuits, TMH, 2003.
[5] W. Wolf, Modern VLSI Design- Systems on Silicon, Prentice Hall,
1998.
[6] Y. Tsividis, Operation and Modeling of The MOS Transistor, Mc
Graw-Hill, 1999.
[7]http://wiki.answers.com/Q/What_is_UP_Counter_in_Digital_Electronics
[8]http://www.hmisource.com/otasuke/qa/first/glc/timer_043.htm
[9]http://www.parallax.com/portals/0/help/P8X32A/QnaMobile/Advanced/Content/QnaTopics/QnaCounters.htm
[10]http://en.wikipedia.org/wiki/Flip-flop_%28electronics%29
[11]http://www.eecs.tufts.edu/~dsculley/tutorial/flopsandcounters/flops6.html