Design and Analysis of Digital Counters for VLSI Applications

Raja Ram Gautam[1], Deepak Kumar[2]

2[1]M. Tech Scholar, Department of Electronics and Communication Engineering
[2]Assistant Professor, Department of Electronics and Communication Engineering
Vidyapeeth Institute of Science and Technology
Bhopal, India

---***---

Abstract - *Counters are widely used as essential building blocks for variety of circuit operations. Flip-flops can be connected together to perform counting operations. Such a group of flip-flops is a counter. The purposes of this review and survey simultaneously is to collate information on Digital Synchronous Counters. Particular emphasis was placed on the following areas Types of Synchronous Counters and How they work The number of flip-flops are used and the way in which they are connected determine the number of states called the modulus and also the sequence of states that the counter goes through during each complete cycle. Counters are classified into two broad categories according to the way they are clocked. In this paper various design methodologies are presented for designing digital counters.*

Key Words: Flip-flops, Counters, VLSI CAD, Simulation, Clock, Twisted Johnson counter, .

1.INTRODUCTION

The binary counter is a fundamental unit of computer or digital circuit operation. There are many situations when a low voltage operation of a counter would be beneficial mostly now in everyday chips basically. The goal of this paper is to observe and simulate the operation of a binary counter 4 bit basically at very low voltages, and evaluate the advantages and disadvantages associated with this operation. In digital logic and computing, a counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. The most common type is a sequential digital logic circuit with an input line called the "clock" and multiple output lines. The values on the output lines represent a number in the binary or BCD number system. Each pulse applied to the clock input increments or decrements the number in the counter. A counter circuit is usually constructed of a number of flip-flops connected in cascade. Counters are a very widely-used component in digital circuits, and are manufactured as separate integrated circuits and also incorporated as parts of larger integrated circuits. There are lots of types of counter. The one is asynchronous. (ripple) counter that changing state bits are used as clocks to subsequent state flip-flops. Synchronous counter is that all state bits change under control of a single clock. And there are also decade counter, up/down counter, ring counter, Johnson counter, cascaded counter and modulus counter [4]. For an asynchronous counter, a single d-type flip-flop with its J (data) input fed from its own inverted output. Synchronous counter uses the clock inputs of all the flip-flops connected together and triggered by the input pulses.

2. CIRCUIT DESIGN

There are a number of factors to consider when choosing an appropriate design for a low voltage counter. By using a high static leakage technology, a large number of transistors in the design can be a significant source of static power dissipation. It can therefore be beneficial to keep the size of the circuit as small as possible. However, if in reducing the size of the circuit, the delay is increased, then this can negatively affect the amount of energy consumed per cycle. The simplest counter with the fewest number of gates is the asynchronous (ripple) counter shown in Figure 1. This type of counter can be constructed using nothing but flip-flops connected in series. Each previous flip flop acts as the clock for the higher order flip flop. When the least significant bit makes a transition, the information is rippled through to each successive flip-flop changing values as necessary. The asynchronous counter can be useful in small counters, but as the number of bits of the counter grows, this ripple effect causes an increasing delay period in which the output of the counter is indeterminate.

3. SYNCHRONOUS COUNTERS

A. Binary Up Counters

A synchronous binary counter counts from 0 to 2N-1, where N is the number of bits/flip-flops in the counter. Each flip-flop is used to represent one bit. The flip-flop in the lowest-order position is complemented/toggled with every clock pulse and a flip-flop in any other position is complemented on the next clock pulse provided all the bits in the lower-order positions are equal to 1.

Take for example A4 A3 A2 A1 = 0011. On the next count, A4 A3 A2 A1 = 0100. A1, the lowest-order bit, is always complemented. A2 is complemented because all the lower-order positions (A1 only in this case) are 1's. A3 is also complemented because all the lower-order positions, A2 and A1 are 1's. But A4 is not complemented the lower-order positions, A3 A2 A1 = 011, do not give an all 1 condition.

To implement a synchronous counter, we need a flip-flop for every bit and an AND gate for every bit except the first and the last bit. The diagram below shows the implementation of a 4-bit synchronous up-counter.

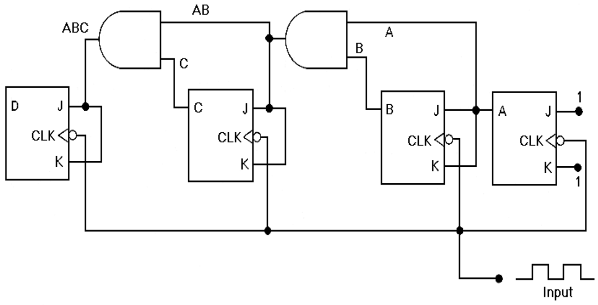

Figure 1 Four-bit Synchronous Binary up counter

From the diagram above, we can see that although the counter is synchronous and is supposed to change simultaneously, we have a propagation delay through the AND gates which add up to give an overall propagation delay which is proportional to the number of bits of the counter.

B. Binary Down Counters

In a binary up counter, a particular bit, except for the first bit, toggles if all the lower-order bits are 1's. The opposite is true for binary down counters. That is, a particular bit toggles if all the lower-order bits are 0's and the first bit toggles on every pulse.

Taking an example, A4 A3 A2 A1 = 0100. On the next count, A4 A3 A2 A1 = 0011. A1, the lowest-order bit, is always complemented. A2 is complemented because all the lower-order positions (A1 only in this case) are 0's. A3 is also complemented because all the lower-order positions, A2 and A1 are 0's. But A4 is not complemented the lower-order positions, A3 A2 A1 = 011, do not give an all 0 condition.

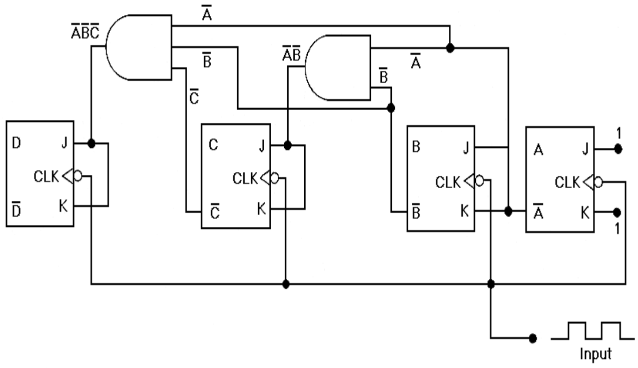

Figure 2 Four-bit Synchronous Binary down counter

The implementation of a synchronous binary down counter is exactly the same as that of a synchronous binary up counter except that the inverted output from each flip-flop is used. All the methods used improve a binary up counter can be similarly applied here.

C. Binary Up Down Counters

The similarities between the implementation of a binary up counter and a binary down counter leads to the possibility of a binary up/down counter, which is a binary up counter and a binary down counter combined into one. Since the difference is only in which output of the flip-flop to use, the normal output or the inverted one, we use two AND gates for each flip-flop to "choose" which of the output to use.

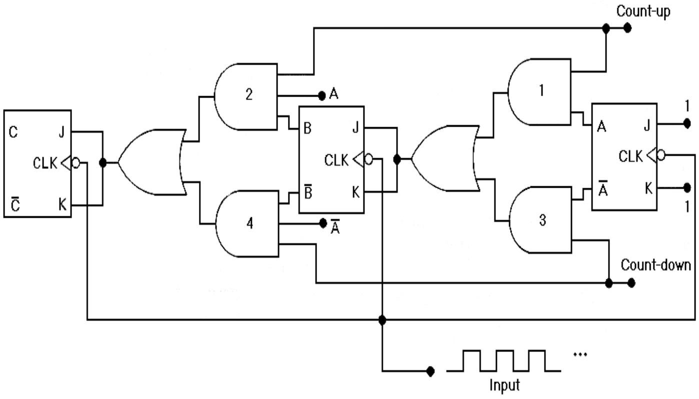

Figure 3 Three bit Synchronous Binary Up/Down Counter

From the diagram, we can see that COUNT-UP and COUNT-DOWN are used as control inputs to determine whether the normal flip-flop outputs or the inverted ones are fed into the J-K inputs of the following flip-flops. If neither is at logic level 1, the counter doesn't count and if both are at logic level 1, all the bits of the counter toggle at every clock pulse. The OR gate allows either of the two outputs which have been enabled to be fed into the next flip-flop. As with the binary up and binary down counter, the speed up techniques apply.

D. *Johnson/Twisted-Ring Counters*

The Johnson counter, also known as the twisted-ring counter, is exactly the same as the ring counter except that the inverted output of the last flip-flop is connected to the input of the first flip-flop. The Johnson counter works in the following way: Take the initial state of the counter to be 000. On the first clock pulse, the inverse of the last flip-flop will be fed into the first flip-flop, producing the state 100. On the second clock pulse, since the last flip-flop is still at level 0, another 1 will be fed into the first flip-flop, giving the state 110. On the third clock pulse, the state 111 is produced. On the fourth clock pulse, the inverse of the last flip-flop, now a 0, will be shifted to the first flip-flop, giving the state 011. On the fifth and sixth clock pulse, using the same reasoning, we will get the states 001 and 000, which is the initial state again. Hence, this Johnson counter has six distinct states: 000, 100, 110, 111, 011 and 001, and the sequence is repeated so long as there is input pulse. Thus this is a MOD-6 Johnson counter.

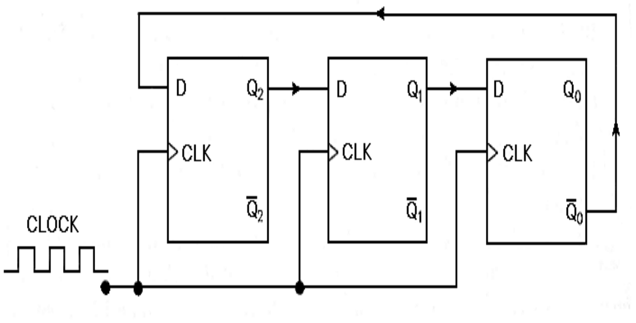

Figure 4 Four bit Synchronous Johnson Counter

The MOD number of a Johnson counter is twice the number of flip-flops. In the example above, three flip-flops were used to create the MOD-6 Johnson counter. So for a given MOD number, a Johnson counter requires only half the number of flip-flops needed for a ring counter. However, a Johnson counter requires decoding gates whereas a ring counter doesn't. As with the binary counter, one logic gate (AND gate) is required to decode each state, but with the Johnson counter, each gate requires only two inputs, regardless of the number of flip-flops in the counter. Note that we are comparing with the binary counter using the speed up technique discussed above. The reason for this is that for each state, two of the N flip-flops used will be in a unique combination of states. In the example above, the combination Q2 = Q1 = 0 occurs only once in the counting sequence, at the count of 0. The state 010 does not occur. Thus, an AND gate with inputs (not Q2) and (not Q2) can be used to decode for this state. The same characteristic is shared by all the other states in the sequence.

A Johnson counters represent a middle ground between ring counters and binary counters. A Johnson counter requires fewer flip-flops than a ring counter but generally more than a binary counter; it has more decoding circuitry than a ring counter but less than a binary counter. Thus, it sometimes represents a logical choice for certain applications.

4. COMPARISON BETWEEN SYNCHRONOUS AND ASYNCHRONOUS COUNTERS

Asynchronous counters, also known as ripple counters, are not clocked by a common pulse and hence every flip-flop in the counter changes at different times. The flip-flops in an asynchronous counter is usually clocked by the output pulse of the preceding flip-flop. The first flip-flop is clocked by an external event. A synchronous counter however, has an internal clock, and the external event is used to produce a pulse which is synchronized with this internal clock. The diagram of a ripple counter is as shown

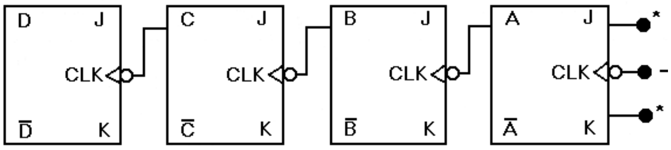

Figure 5 Four-bit Ripple counter

It can be seen that a ripple counter requires less circuitry than a synchronous counter. No logic gates are used at all in the example above. Although the asynchronous counter is easier to construct, it has some major disadvantages over the synchronous counter.

First of all, the asynchronous counter is slow. In a synchronous counter, all the flip-flops will change states simultaneously while for an asynchronous counter, the propagation delays of the flip-flops add together to produce the overall delay. Hence, the more bits or number of flip-flops in an asynchronous counter, the slower it will be. Secondly, there are certain "risks" when using an asynchronous counter. In a complex system, many state changes occur on each clock edge and some ICs respond faster than others. If an external event is allowed to affect a system whenever it occurs (unsynchronized), there is a small chance that it will occur near a clock transition, after some IC's have responded, but before others have. This intermingling of transitions often causes erroneous operations. And the worse this is that these problems are difficult to foresee and test for because of the random time difference between the events.

5. SYNCHRONOUS COUNTER DESIGN

A synchronous counter usually consists of two parts: the memory element and the combinational element. The memory element is implemented using flip-flops while the combinational element can be implemented in a number of ways. Using logic gates is the traditional method of implementing combinational logic and has been applied for decades. Since this method often results in minimum component cost for many combinational systems, it is still a popular approach. However, there are other methods of implementing combinational logic which offers other advantages. Some of the alternative methods which are discussed here are: multiplexers (MUX), read-only memory (ROM) and programmable logic array (PLA).

Following methods are used in Counter design methodologies

A.Multiplexer

The multiplexer, also called the data selector, it has n select inputs, 2n input lines and 1 output line (and usually also a complement of the output). The 2n possible combinations of the select inputs connects one of the input lines to the output. When used as a combinational logic device, the n select inputs represent n variables and the 2n input lines represent all the midterms of the n variables.

B.Read-Only Memory

The ROM is usually used as a storage unit for fixed programs in a computer. However, it can also be used to implement combinational logic. It is useful for systems requiring changeable functions. When a different function is required, a different ROM producing this function can be plugged into the circuit. No wiring change is necessary. The ROM has n input lines pointing to 2n locations within the ROM that store words of M bits. As with the MUX, each input line is used to represent a variable and the 2n locations represent the midterms.

C.Programmable Logic Array

The PLA is very similar to the ROM. It can be thought of as a ROM with a large percentage of its locations deleted. A ROM with 16 input address lines must have 216, or 65,536 storage locations, and all the words stored in these have to be decoded. The PLA only decodes a small percentage of the midterms. The PLA is sometimes used to produce a system with a small number of chips in a minimum time.

6. IMPLEMENTATION TILL NOW

A few different design tools were considered and tested for implementation of the counter design. The first tool used was Mentor Graphics' Modelsim to test the basic functionality of a VHDL coded version of the circuit. Once the counter operation was verified, we began testing a number of different tools to implement a netlist for spice simulation. The following netlist creation tools were tested.

The following synchronous sequential network is a module-4 Twisted Johnson counter. It is based on a two-stages Shift Register, with the last bit returned, inverted, to the first stage input. Click on the figure to open the schematic in the d-DcS: Firstly, the **count sequence** observed on the outputs (MSB) and
(LSB).

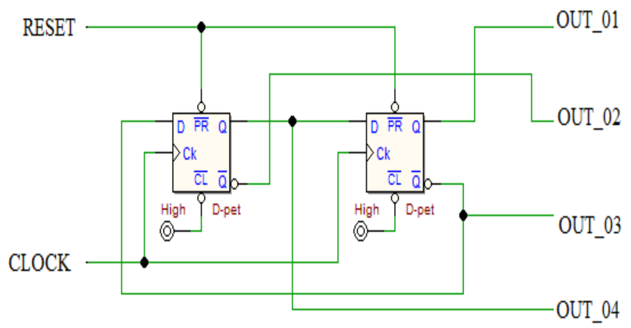

Figure 6 Basic Schematic for Module 4 Twisted Johnson counter.

Next, consider the **four waveform**s the outputs (**OUT_01**, **OUT_02 OUT_03 OUT_04**). The particulare **time relation** among the **four waveforms** permits to use this circuit to generate **polyphase clock signals**, i.e. shifted in a regular mode (in this case, by a 1/4 of the clock period).

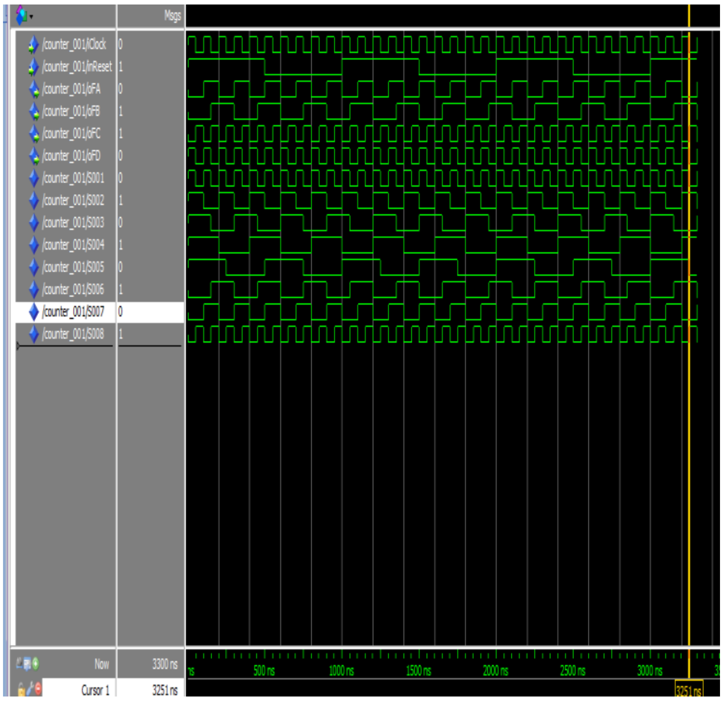

Figure 7 Simulation plot for Module 4 Twisted Johnson counter for 3000ns.

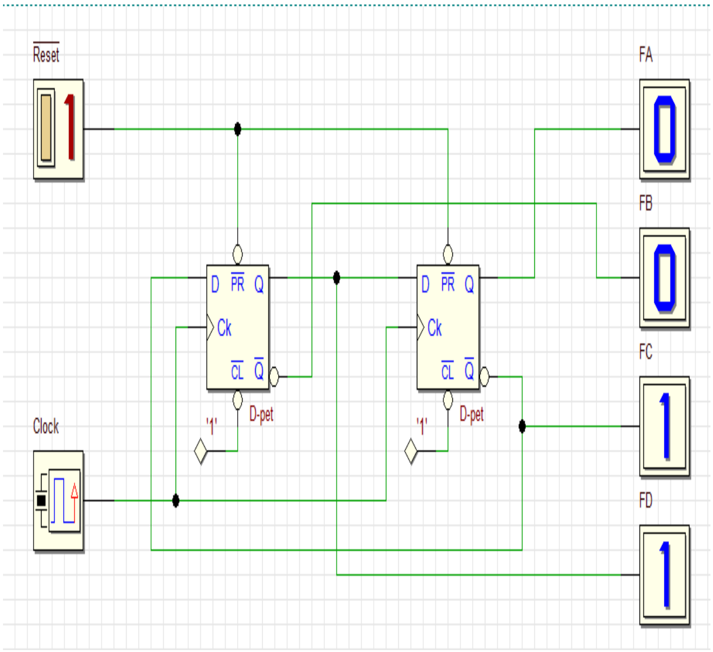

Figure 8 Schematic simulation for Module 4 Twisted Johnson counter.

7. CONCLUSION

The operation of the 04-bit counter can achieve dramatic design variations over numerous number of applications. These are other digital assets in terms of savings, however, are offset by the increase in delay as the supply voltage is lowered close to the threshold operation of the frontend modules. These delays can be offset by optimizing the circuit and utilizing other techniques.

REFERENCES

[1] Zhu, Yue. "Low Voltage Operation of a 16-bit Counter in 32 nm CMOS Technology."

[2] Sheafor, Stephen James. "Low Power Asynchronous Counters in a Synchronous System." U.S. Patent No. 20,160,109,901. 21 Apr. 2016.

[3] Van Huben, G.A., Meaney, P.J., Dodson, J.S., Rider, S.H., Gregerson, J.C., Retter, E.E., Baysah, I.G., Gilda, G.D., Curley, L.D. and Papazova, V.K., International Business Machines Corporation, 2016. *Dual asynchronous and synchronous memory system*. U.S. Patent 9,318,171.

[4] Pandey, Bishwajeet, et al. "Energy efficency of asynchronous and synchronous VLSI circuit on 40nm and 90nm FPGA." *Energy Efficient Technologies for Sustainability (ICEETS), 2013 International Conference on*. IEEE, 2013.

[5] Jiang, Yu, Hehua Zhang, Huafeng Zhang, Han Liu, Xiaoyu Song, Ming Gu, and Jiaguang Sun. "Design of mixed synchronous/asynchronous systems with multiple clocks." *IEEE Transactions on Parallel and Distributed Systems* 26, no. 8 (2015): 2220-2232.

[6] Swartzlander, E. E. (2004, February). A review of large parallel counter designs. In VLSI, 2004. Proceedings. IEEE Computer society Annual Symposium on (pp. 89-98). IEEE.

[7] Mehta, Mayur, Vijay Parmar, and E. Swartzlander. "High-speed multiplier design using multi-input counter and compressor circuits." Computer Arithmetic, 1991. Proceedings., 10th IEEE Symposium on. IEEE, 1991.

[8] Kulkarni, Vidya (nd). Logic Design Chapter – 5 [PowerPoint Slides]. Retrieved from: *f*orum.vtu.ac.in/ ~edusat/Prog5/logd/ vrk/Chapter-5.ppt

[9] Counter. (nd) .Retrieved April 15, 2009, from Wikipedia Website: http://en.wikipedia.org/wiki/Counter

[10] Palnitkar, Samir. Verilog HDL: a guide to digital design and synthesis. Vol. 1. Prentice Hall Professional, 2003.

[11] Brown, Stephen D. Fundamentals of digital logic with Verilog design. Tata McGraw-Hill Education, 2007.

[12] Katz, Randy H., and Gaetano Borriello. "Contemporary logic design." (2005).

[13] A. Chandrakasan, W. J. Bowhill and F. Fox, Design of High-Performance Microprocessor Circuits, New York: IEEE Press, 2001.

[14] Counter. (nd) .Retrieved April 15, 2009, from Wikipedia Website: http://en.wikipedia.org/wiki/Counter

[15] Kulkarni, Vidya (nd). Logic Design Chapter – 5 [PowerPoint Slides]. Retrieved from: forum.vtu.ac.in /~edusat/Prog5/ logd/vrk/Chapter-5.ppt

BIOGRAPHIES

Raja Ram Gautam is M.Tech Scholar in department of Electronics and Communication Engineering, Vidyapeeth Institute of Science and Technology, Bhopal His area of interest includes VLSI Design and Simulation.