Implementation and Analysis of BCD Counter with Synchronous and Asynchronous Reset

Anjali M. K.

Abstract - A binary coded decimal (BCD) counter is a serial digital counter that counts ten digits and it resets for every new clock input. As it can go through 10 unique combinations of output, it is also called as "Decade counter". A BCD counter can count 0000, 0001, 0010, 0111, 0100, 0101 ,0110, 0111, 1000, 1001, 0000, 0001 and so on. A BCD counter with synchronous and asynchronous reset have been designed using VHDL Xilinx Synthesis tool ISE 14.2 and implemented on Spartan 3 FPGA. Analysis of the BCD counter with synchronous and asynchronous reset has also been carried out.

Keywords – BCD Counter, synchronous reset, asynchronous reset, VHDL, SPARTAN 3 FPGA implementation.

I. INTRODUCTION

The integrated circuit designers are facing new challenges due to the exponential growth in electronic devices and equipments in the past few years. Addition of more functionality along with real time applications demands revolutionary changes in the design process of a chip. Increase in computing requirements onto a single chip demands development of sophisticated tools that can perform complex operations which further results in increase in the processing power. Thus, development of high speed computational hardware is a prime concern in today's scenario.

In computing and electronic systems, binary-coded decimal (BCD) (sometimes called natural binary-coded decimal, NBCD) or, in its most common modern implementation, packed decimal, is an encoding for decimal numbers in which each digit is represented by its own binary sequence. Its main virtue is that it allows easy conversion to decimal digits for printing or display, and allows faster decimal calculations. Its drawbacks are a small increase in the complexity of circuits needed to implement mathematical operations. Uncompressed BCD is also a relatively inefficient encoding—it occupies more space than a purely binary representation. In BCD, a digit is usually represented by four bits which, in general, represent the decimal digits 0 through 9. Other bit combinations are sometimes used for a sign or for other indications (e.g., error or overflow).Although uncompressed BCD is not as widely used as it once was, decimal fixed-point and floating-point are still important and continue to be used in financial, commercial, and industrial computing. IBM used the terms binary-coded decimal and BCD for 6-bit alphamerics codes that represented numbers, upper-case letters and special characters. Some variation of BCD alphamerics was used in most early IBM computers, including the IBM 1620, IBM 1400 series, and non-Decimal Architecture members of the IBM 700/7000 series. Today, BCD data is still heavily used in IBM processors and databases, such as IBM DB2, mainframes, and Power6. In these products, the BCD is usually zoned BCD (as in EBCDIC or ASCII), Packed BCD (two decimal digits per byte), or "pure" BCD encoding (one decimal digit stored as BCD in the low four bits of each byte). All of these are used within hardware registers and processing units, and in software.

Digital counters count upwards from zero to some pre-determined count value on the application of a clock signal. Once the count value is reached, resetting them returns the counter back to zero to start again. A decade or BCD counter counts in a sequence of ten and then returns back to zero after the count of nine. In the design, in order to initialize all your signals to a predetermined state a reset signal is applied. A reset signal can change the system in two ways: Synchronous and asynchronous.

II. SYSTEM MODEL

In digital logic and computing, a counter is a device which stores (and sometimes displays) the number of times particular event or process has occurred, often in relationship to a clock signal. A BCD counter state diagram is shown in figure 1.

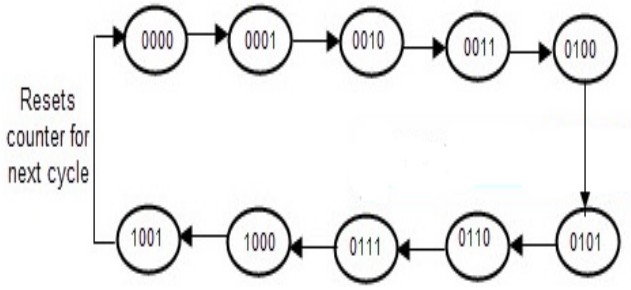

Figure 1: BCD Counter State Diagram

Binary-coded-decimal code is an 8421 code consisting of four binary digits. The 8421 designation refers to the binary weight of the four digits or bits used. The main advantage of BCD code is that it allows for the easy conversion between decimal and binary forms of numbers. The truth table is shown in Table 1.

Q3, Q2, Q1, Q0 represents the output BCD and CLK represents the clock signal based on which the output state changes.

Table-1: Truth table for BCD counter

CLK	Q3	Q2	Q1	Q0
↑	0	0	0	0
↑	0	0	0	1
↑	0	0	1	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	0	1
↑	0	1	1	0
↑	0	1	1	1
↑	1	0	0	0
↑	1	0	0	1
↑	0	0	0	0
↑	0	0	0	1
↑	0	0	1	0
↑	0	0	1	1
↑	0	1	0	0

A reset is required to initialize a hardware design for system operation and to force an ASIC into a known state for simulation. A reset simply changes the state of the device/design/ASIC to a user/designer defined state. There are two types of reset, they are Synchronous reset and Asynchronous reset. One of the decisions that designers need to finalize is to choose synchronous vs asynchronous reset strategy. Each of these reset strategies is capable of achieving the purpose of a reset. A design may also have a mixed approach in which a part of the device is driven by synchronous reset and another part has an asynchronous approach to reset.

Synchronous Reset

A synchronous reset signal will only affect or reset the state of the flip-flop on the active edge of the clock. The reset signal is applied as is any other input to the state machine.

Advantages:

- The advantage to this type of topology is that the reset presented to all functional flip-flops is fully synchronous to the clock and will always meet the reset recovery time.
- Synchronous reset logic will synthesize to smaller flip-flops, particularly if the reset is gated with the logic generating the d-input. But in such a case, the combinational logic gate count grows, so the overall gate count savings may not be that significant.
- Synchronous resets provide some filtering for the reset signal such that it is not effected by glitches, unless they occur right at the clock edge. A synchronous reset is recommended for some types of designs where the reset is generated by a set of internal conditions.

As the clock will filter the logic equation glitches between clock edges.Synchronous reset has to set up to the active clock edge in order to be effective. This provides for protection against accidental glitches as long these glitches don't happen near the active clock edges. In that sense it is not 100% protection as random glitch could happen near the active clock edge and meet both setup and hold requirements and can cause flops to reset, when they are not expected to be reset. This type of random glitches are more likely to happen if reset is generated by some internal conditions, which most of the time means reset travels through some combinational logic before it finally gets distributed throughout the system. As shown in the figure 2, x1 and x2 generate (reset)bar. Because of the way x1 and x2 transition during the first clock cycle we get a glitch on reset signal, but because reset is synchronous and because glitch did not happen near the active clock edge, it got filtered and we only saw reset take effect later during the beginning of 4th clock cycle, where it was expected.

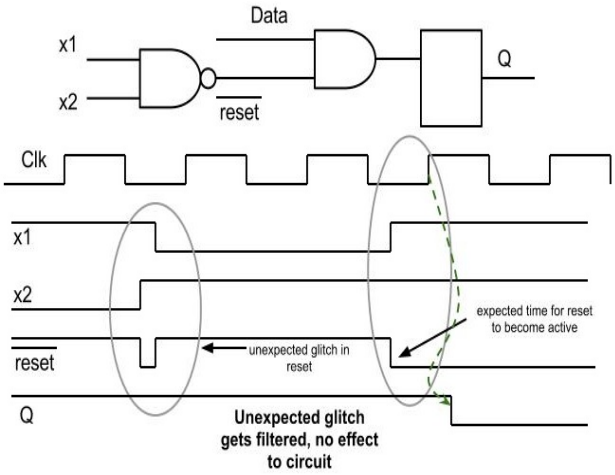

Figure 2 : Glitch with synchronous reset

Disadvantages:

- The problem in this topology is with reset assertion. If the reset signal is not long enough to be captured at active clock edge (or the clock may be slow to capture the reset signal), it will result in failure of assertion. In such case the design needs a pulse stretcher to guarantee that a reset pulse is wide enough to be present during the active clock edge.
- Another problem with synchronous resets is that the logic synthesis cannot easily distinguish the reset signal from any other data signal. So proper care has to be taken with logic synthesis, else the reset signal may take the fastest path to the flip-

flop input there by making worst case timing hard to meet.

- In some power saving designs the clocked is gated. In such designed only asynchronous reset will work.
- Faster designs that are demanding low data path timing, cannot afford to have extra gates and additional net delays in the data path due to logic inserted to handle synchronous resets.

Asynchronous Reset

An asynchronous reset will affect or reset the state of the flip-flop asynchronously i.e. no matter what the clock signal is. This is considered as high priority signal and system reset happens as soon as the reset assertion is detected.

Advantages:

- High speeds can be achieved, as the data path is independent of reset signal.
- Another advantage favoring asynchronous resets is that the circuit can be reset with or without a clock present.
- As in synchronous reset, no work around is required for logic synthesis.

Disadvantages:

- The problem with this type of reset occurs at logic de-assertion rather than at assertion like in synchronous circuits. If the asynchronous reset is released (reset release or reset removal) at or near the active clock edge of a flip-flop, the output of the flip-flop could go metastable.
- Spurious resets can happen due to reset signal glitches.

III. SIMULATION RESULTS

A BCD counter with synchronous and asynchronous reset has been designed using VHDL Xilinx Synthesis ISE 14.***2 tool***. The RTL schematic is shown in figure 3.The system has two inputs: clock and reset and one output which is the 4 bit BCD number.

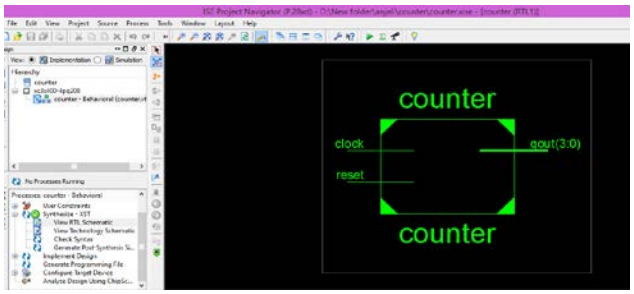

Figure 3 : RTL schematic

The simulation results for the BCD counter is shown in figure 4.It is clearly depicted in the simulation results that the counter counts from 0 to 9 and then resets.

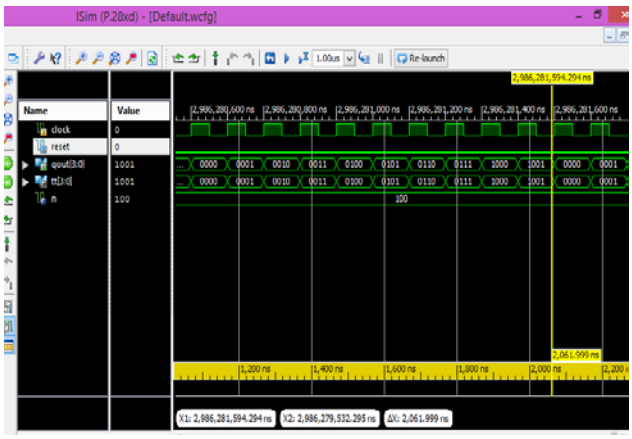

Figure 4 : Simulation results for the BCD counter

The schematic design of the counter with Synchronous reset and asynchronous reset is shown in figure 5 and figure 6.

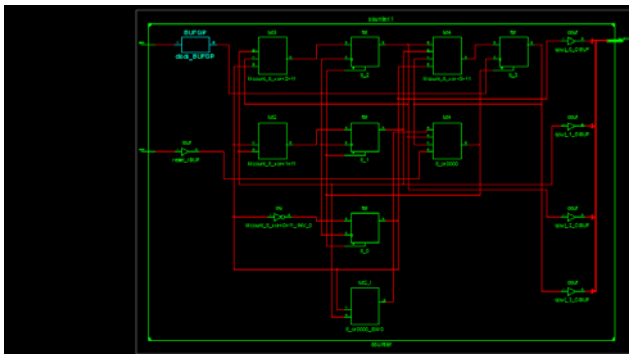

Figure 5 : Schematic design of counter with Synchronous reset

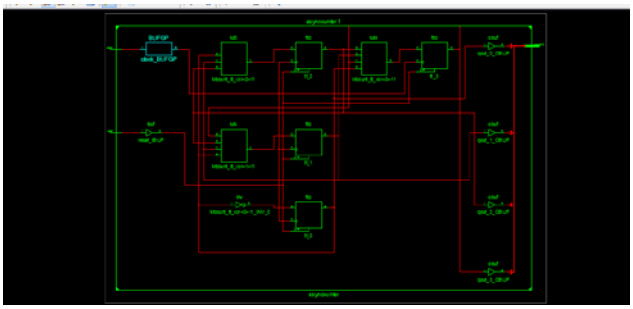

Figure 6: Schematic design of the counter with asynchronous reset

The synthesis report analysis of the counter with synchronous reset and asynchronous reset is shown in figure 7 and figure 8.This gives an idea about the devices utilised in each architecture.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	4	7,168	1%
Number of 4 input LUTs	5	7,168	1%
Number of occupied Slices	3	3,584	1%
Number of Slices containing only related logic	3	3	100%
Number of Slices containing unrelated logic	0	3	0%
Total Number of 4 input LUTs	5	7,168	1%
Number of bonded IOBs	6	141	4%
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	3.43		

Figure 7: Synthesis report analysis of the counter with synchronous reset

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	4	7,168	1%
Number of 4 input LUTs	3	7,168	1%
Number of occupied Slices	2	3,584	1%
Number of Slices containing only related logic	2	2	100%
Number of Slices containing unrelated logic	0	2	0%
Total Number of 4 input LUTs	3	7,168	1%
Number of bonded IOBs	6	141	4%
Number of BUFGMUXs	1	8	12%
Average Fanout of Non-Clock Nets	4.00		

Figure 8: Synthesis report analysis of the counter with synchronous reset.

IV. IMPLEMENTATION

The implementation is done in FPGA3S400 using UJTX software.The pin allocation in the FPGA3S400 kit is shown in figure 9.Once pins are alloted ,the program is run and the bit files are generated.These bit files are downloaded into the FPGA3S400 kit using UJTX software as shown in figure 10.The FPGA3S400 kit after downloading is shown in figure 11.The FPGA Kit displays the output of BCD counter as per the truth table after the implementation.

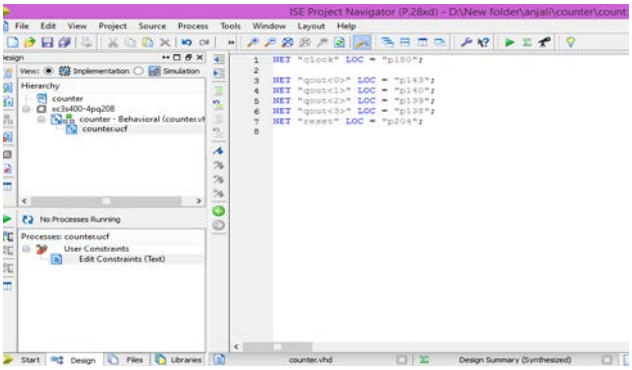

Figure 9: Pin allocation in the FPGA3S400 kit

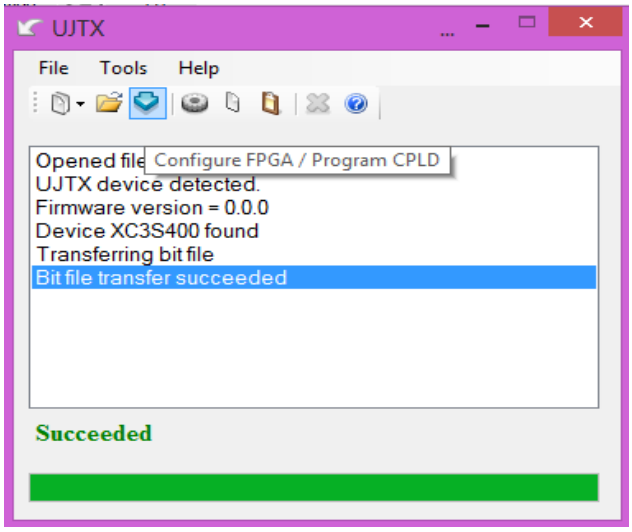

Figure 10 : Bit files download

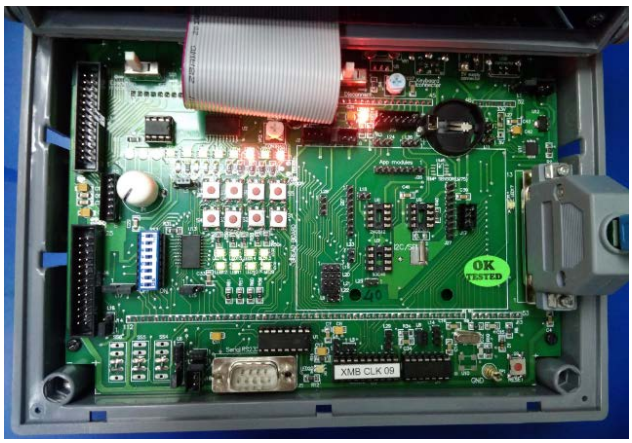

Figure 11 :FPGA Implementation

V. CONCLUSION

A binary coded decimal (BCD) counter with synchronous and asynchronous reset have been designed using VHDL Xilinx Synthesis tool ISE 14.2 and implemented on Spartan 3 FPGA. The devices utilised in the architectures of the BCD counter with synchronous and asynchronous reset have also been analysed.The use of BCD the counter with synchronous reset and asynchronous reset depends upon the applications. Both types of resets have positives and negatives and none of them assure fail-proof design. These would be used as per the design needs. For example if chip has to be powered up prior to clock, asynchronous reset has to be used. Similarly if a design which is completely synchronous circuit with no metastability issue related to reset is needed, go with synchronous reset. One of the decisions that designers need to finalize is to choose synchronous vs asynchronous reset strategy. Each of these reset strategies is capable of achieving the purpose of a reset. A design may also have a mixed approach in which a part of the device is driven by synchronous reset and another part has an asynchronous approach to reset.

REFERENCES

[1] David A. Hodges, Horace G. Jackson, Resve A. Saleh, "Analysis and Design of Digital Integrated Circuits "- Third Edition,2003.

[2] Himal Pokhrel, Deepak Kumar, Anjali Sharma, "Design and Analysis of 4- Bit Binary Synchronous Counter by Leakage Reduction Techniques", IJCTT Journal, Volume-50 Number-2, 2017.

[3] Harish K and Chinmaye R, "Design and Optimization of Asynchronous counter using Reversible Logic", IJERT, Volume. 4 - Issue. 06 , June – 2015.

[4] R. Arunya, A. Ramya, P. Umarani, V. Balamugaran, "Design Of 3 bit synchronous Counter using DLDFF" ,IJERA, ISSN: 2248-9622 Vol. 3, Issue 2, , pp.1258-1262 ,March -April 2013.

[5] Gopalakrishnan M. and C. Karthick, "Design And Performance Analysis Of Asynchronous Counter Using Feynman Gate Based T Flip Flop ", ARPN Journal of Engineering and Applied Sciences, Vol. 11, No. 17, September 2016.

[6] Ms. Manisha Rahamatkar, Ms. MittalPisalkar, Mr. YogeshThakare, "Design & Analysis Of 4-Bit Counter Using Sub-Micron Technology," International Journal of Scientific & Engineering Research, Volume 7, Issue 2, ISSN 2229-5518, February-2016.

[7] Md. Mizanur Rahman , Indrani Mandal , Md. Selim Al Mamun, "An Optimized Design of Counter Using Reversible Logic",International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 8, August 2015.

[8] Raju Odela,Sangani Rahul Gowtham, Hazrathiah ," A Novel Power Efficient Reversible Asynchronous Counter" International Journal for Scientific Research & Development| Vol. 3, Issue 10, ISSN (online): 2321-0613, 2015.

[9] Shuli Gao, Dhamin Al-Khalili,J.M. Pierre Langlois,and Noureddine Chabini ," Efficient Realization of BCD Multipliers Using FPGAs, Article ID 2410408, 12 pages, Volume 2017 (2017).

[10] Anshul Sharma, Pankaj Soni, " Low Power Dissipation Binary to BCD Converter for Multi-Operand B/D Adder ", International Journal of Computer Applications ,Volume 166 – No.9, May 2017.

[11] Emilliano , Chandan Kumar Chakrabarty , Ahmad Basri A. Ghani , Agileswari K. Ramasamy, " VHDL simulation of peak detector, 64 bit BCD counter and reset automatic block for PD detection system using FPGA", IEEE *X*plore August 2010.

[12] Sreehari Veeramachaneni , M.B. Srinivas, " Novel High-Speed Architecture for 32-Bit Binary Coded Decimal (BCD) Multiplier", IEEE Xplore, December 2008.

AUTHOR'S PROFILE

Anjali M.K received B.Tech degree in electronics and communication engineering from College of engineering, Thalassery affiliated to Cochin university of Science and Technology. She has completed M.E degree in VLSI Design at K.S.Rangasamy College of technology, Namakkal, India. She has presented and published papers in international and national conferences. She also has a vast teaching experience. Her research interests include VLSI design, low power applications.