J.L.V Ramana Kumari , Bekkam Satheesh / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, June-July 2012, pp.851-855

FPGA Implementation of Counter by Using State Look Ahead Logic

J.L.V Ramana Kumari*,Bekkam Satheesh*
* (Department of ECE, VNR VJIET, Hyderabad, India
*(Department of ECE, VNR VJIET, Hyderabad, India

ABSTRACT

The main objective of this paper consists of the state look-ahead path and the counting path. The proposed counter is a single mode counter, which sequences through a fixed set of pre assigned count states, of which each next count state represents the next counter value in sequence. The counter is partitioned into uniform 2-bit synchronous up counting modules. Next state transitions in counting modules of higher significance are enabled on the clock cycle preceding the state transition using stimulus from the state look-ahead path. Therefore, all counting modules concurrently transition to their next states at the rising clock edge.

Keywords **–** FPGA, State Look Ahead Logic, Interrupt Controller, Rotating Priority Mode, Special Mask Mode.

INTRODUCTION

Counters are widely considered as essential building blocks for a variety of circuit operations such as programmable frequency dividers, shifters, code generators, memory select management, and various arithmetic operations. Since many applications are comprised of these fundamental operations, much research focuses on efficient counter architecture design. Counter architecture design methodologies explore tradeoffs between operating frequency, power consumption, area requirements and target application specialization.

Early design methodologies improved counter operating frequency by partitioning large counters into multiple smaller counting modules, such that modules of higher significance (containing higher significant bits) were enabled when all bits in all modules of lower significance (containing lower significant bits) saturate. Initializations and propagation delays such as register load time, AND logic chain decoding, and the half incrementer component delays in half adders dictated operating frequency. Subsequent methodologies improved counter operating frequency using half adders in the parallel counting modules that enabled carry signals generated at counting modules of lower significance to serve as the count enable for counting modules of higher significance, essentially implementing a carry chain from modules of lower significance to modules of higher significance. The carry chain cascaded synchronously through intermediate D-type flip-flops (DFFs). The maximum operating frequency was limited by the half adder module delay, DFF access time, and the detector logic delay. Since the module outputs did not directly represent count state, the detector logic further decoded the module outputs to the output count state value. Further enhancements improved operating frequency using multiple parallel counting modules separated by DFFs in a pipelined structure. The counting modules were composed of an incrementer that was based on a carry-ripple adder with one input hardcoded to "1". In this design, counting modules of higher significance contained more cascaded carry-ripple adders than counting modules of lower significance. Each counting module's count enable signal was the logical AND of the carry signals from all the previous counting modules (all counting modules of lower significance), thus pre scaling clocked modules of higher significance using a low frequency signal derived from modules of lower significance. Due to this pre scaling architecture, the maximum operating frequency was limited by the incrementer, DFF access time, and the AND gate delay. The AND gate delay could potentially be large for large sized counters due to large fan-in and fan-out parasitic components. Design modifications enhanced AND gate delay, and subsequently operating frequency, by redistributing the AND gates to a smaller fan-in and fan-out layout separated by latches. However, the drawback of this redistribution was increased count latency (number of clock cycles required before the output of the first count value). In addition, due to the design structure, this counter architecture inherited an irregular VLSI layout structure and resulted in a large area overhead.

PARALLEL COUNTER ARCHITECTURE

The figure 1 depicts our proposed parallel counter architecture for a sample 8-bit counter. The main structure consists of the state look-ahead path (all logic encompassed by the dashed box) and the counting path (all logic not encompassed by the dashed box). We construct our counter as a single mode counter, which sequences through a fixed set of pre assigned count states, of which each next count state represents the next counter value in sequence.

The counter is partitioned into uniform 2-bit synchronous up counting modules.

Next state transitions in counting modules of higher significance are enabled on the clock cycle preceding the state transition using stimulus from the state look-ahead path. Therefore, all counting modules concurrently transition to their next states at the rising clock edge (CLKIN).

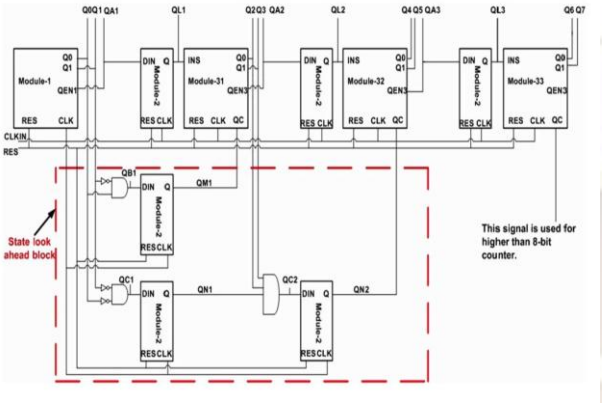

Figure 1: Functional block diagram of the proposed 8-bit parallel counter with state look-ahead logic and counting logic.

The state look-ahead logic consists of all logic encompassed by the dashed box and the counting logic consists of all other logic (not encompassed by the dashed box).

2.1 Architecture Functionality

The counting path's counting logic controls counting operations and the state look-ahead path's state look-ahead logic anticipates future states and thus prepares the counting path for these future states. Figure shows the three module types, (module-1, module-2, and module-3S, where S= (1, 2, 3 increasing from left to right) and represents the position of module-3) used to construct both paths. Module-1 and module-3 are exclusively to counting path and each module represents two counter bits. Module-2 is a conventional positive edge triggered DFF and is present in both paths. In the counting path, each module-3S serve two main purposes. Their first purpose is to generate all counter bits associated with their ordered position and the second purpose is to enable (in conjunction with stimulus from the state look–ahead path) future states in subsequent module-3S's (higher S values) in conjunction with stimulus from the state look-ahead path.

2.2 Counting Path

Module-1 is a standard parallel synchronous binary 2-bit counter, which is responsible for lower order bit counting and generating future states for all module-3S's in the counting path by pipelining the enable for these future states through the state look ahead path.

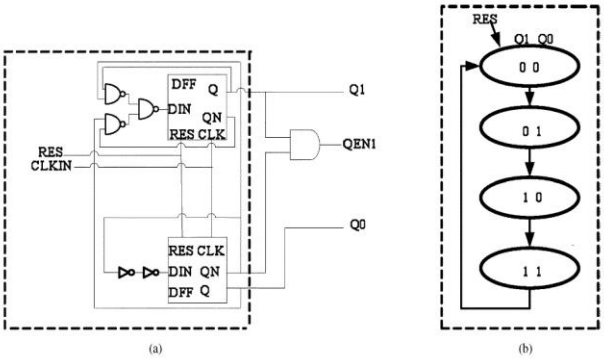

Figure 2: Module-1 (a) Hardware schematic and (b) State diagram. Note that the 1 in QEN1 denotes that this is the QEN for module-1.

The figure 2 depicts:

a) Hardware schematic.
b) State diagram for module-1.

Module -1 outputs Q1Q0 (the counter's two lower order bits) and QEN1 for module-1. QEN1 connects to module-2's DIN input.The placement of module-2s in the counting path is critical to the novelty of our counter structure. Module-2s in the counting path act as a pipeline between the module-1 and module-3 and between subsequent module-3S s(see in figure). Module-2 placement (coupled with state look ahead logic described in section II-A2) increases counter operating frequency by eliminating the lengthy AND-gate rippling and large AND gate fan in and fan-out typically present in large width parallel counters. Thus, instead of the modules of higher significance requiring the ANDing of all enable signals from modules of lower significance, modules of higher significance (module-3 s in our design) are simply enabled by the module-3's pre-ceding module-2 and state look-ahead logic. Since the coupling of module-2 with module-3 1 introduces an extra cycle delay before module-3 with module-3 introduces an extra cycle delay before module-3 1 is enabled, module-2's DIN is triggered when the module-1's count Q1Q0 =10 (note that this the only case for the left most module-2s require state look ahead logic as well). Thus, the module-2s in the counting path in figure, as subsequent module-2s in the counting path provide a 1- cycle look ahead mechanism for triggering the module-3S's enabling the module-2s to maintain a constant delay for all stages and all module-3S's to count in parallel at the rising clock edge instead of waiting for the overflow rippling in a standard ripple counter.

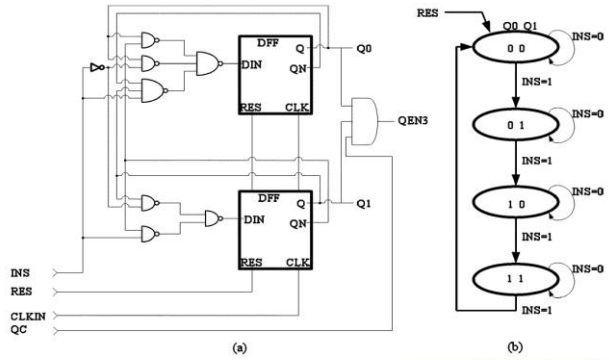

Figure 3: Module-3S (a) hardware schematic and (b) state diagram. Note that the 3 in QEN3 denotes that this is the QEN for module-3

Figure 3 depicts the (a) hardware schematic (b) State diagram for module-3S.Module-3S is a parallel synchronous binary 2 bit counter is enabled by INS. INS connects to the Q output of the preceding module-2. Module-3S outputs Q1Q0(which connect to the appropriate count output bits QX and Q(X-1) as shown in figure) and QEN3 denotes that this is the QEN for module 3S. The state look-ahead logic provides the QC input. QEN3 connects to the subsequent module-2's DIN input and provides the one cycle look-ahead mechanism.

2.3 State Look-Ahead Path

The state look-ahead path operates similarly to a carry look-ahead adder in that it decodes the low-order count states and carries this decoding over several clock cycles in order to trigger high-order count states. The state look-ahead logic is principally equivalent to the one-cycle look-ahead mechanism in the counting path. For example, in a 4-bit counter constructed of two 2-bit counting modules, the counting path's module-2 decodes the low- order state Q1Q0=10 and carries this decoding across one clock cycle and enables Q3Q2=01 at module-3 1on the next rising clock edge. This operation is equivalent to decoding Q1Q0=11 and enabling Q3Q2=01 on the next immediate rising clock edge. The state look-ahead logic expands this principle to an X cycle look-ahead mechanism. For example In a traditional 6-bit ripple counter constructed of three 2-bit counting modules, the enabling of bits Q5Q4 happens only after decoding the overflow at Q1Q0 to enable Q3Q2 and decoding the overflow at Q3Q2 to enable Q5Q4. However, combining the one cycle look-ahead mechanism in the counting path for Q3Q2=10 and a two-cycle look ahead mechanism for Q1Q0=01 from can enable Q5Q4 Q1Q0=01 is pipelined across one cycle, thus enabling Q5Q4 at the next rising clock edge (further details will be discussed in section II-C). Thus, enabling the next state's high order bits depends on early overflow pipelining across clock cycles through the module-2s in the state look-ahead path. This state look-ahead logic organization and operation avoids the use of an overhead delay detector circuit that decodes the low order modules to generate the enable signals for higher order modules, and enables all modules to be triggered concurrently on the clock edge, thus avoiding rippling and long frequency delay.

RESULTS AND DISCUSSIONS

Digital CMOS Parallel Counter Architecture Based on State Look-Ahead Logic is designed using the proposed pipelining and efficient architecture. The Simulation results are obtained using Xilinx ISE 9.1i. The synthesized results are single realizations obtained using Xilinx ISE 10.1.The simulation results related to positive edge triggered D-Flip Flop(Module-2) with asynchronous Reset and Clock, 2-bit Module-1,2-bit Module-3S,and 8-bit Counter module, which is realized by instantiating the positive edge triggered D-Flip Flop(Module-2) with asynchronous Reset and Clock, Module-1 and Module-3S.

The behavioral simulation for the D-Flip Flop with asynchronous Reset and Clock is shown in Figure 5.1. which is obtained using Xilinx ISE 9.1i.In the Figure 4,the inputs are din, clk, reset and outputs are of 1 bit each and they are "**q**", "**qb**". The reset is given High at the start of the simulation and when the reset is High, the output "**q**" is low and "**qb**" is High .The din is given after the reset is low, this causes the din input to be propagated to "**q**" in the same state as the din is and the inverted din is propagated to the "**qb**" output.

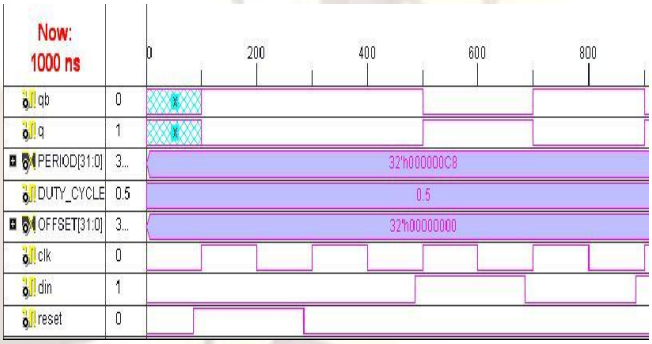

Figure 4: Behavioral Simulation waveform for D-Flip Flop

The Behavioral Simulation waveform for 2-bit Module-1 is shown in Figure 5.Module-1 is responsible for the low-order bit counting and generating future states for all module 3-S's in the counting path bye pipelining the enable for these future states through the state look ahead path.In Figure 5, the inputs are reset, clk, QEN1 and outputs are 2-bit "Q" and "QN", a 1-bit "**QEN1**". The r,s,t signals are the intermediate signals of Module-1. The reset is given high at the start of the simulation and when the reset is high, the output "**Q (00)**" is low and "**QN (11)**" is high. In case of reset is low, the 2-bit output Q changes from state 00 to 01 to 10 to 11 and

back to 00 and repeats these states till reset is low and once the reset is high it goes back to the 00 state.

The same is with the 2-bit output QN but is not of the Q output, with the initial state being the 11 state. The output QEN1 is the enable signal which is used for the purpose of enabling the future states in subsequent Module-3S's,the QEN1 is the AND operation of not(Q0) and Q1.

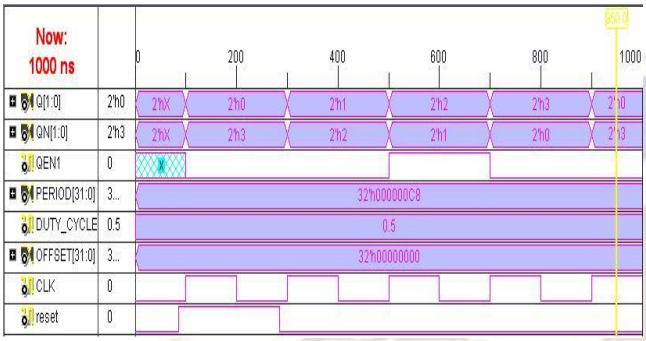

Figure 5: Behavioral Simulation waveform for Module-1

The Behavioral Simulation waveform for 2-bit Module-3-S is shown in Figure 6.Module-3S is a parallel synchronous binary 2-bit counter whose count is enabled by INS.INS connects to the Q output of the preceding module-2.In Figure 5 the inputs are reset, clk, INS, QC and the outputs are 2-bit "**Q**" and "**QEN3".** When the reset is high, the output "**Q (00)**" is low. In case of reset is low, the 2-bit output "**Q"** changes from state 00 to 01 to 10 to 11 and back to 00 and repeats these states till reset is low only when the INS signal is high, if INS signal is low the output stays at the same state where it was during the previous clock and goes to the next state only when the INS is high at this point. The "**QEN3"** denotes that this is the QEN for module-3S, it connects to the subsequent module-2's din input and provides the one-cycle look ahead mechanism.

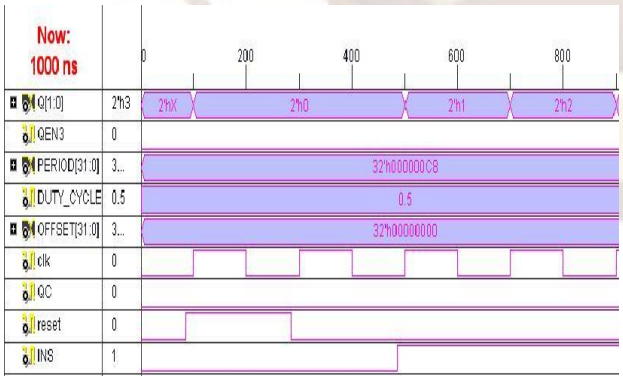

Figure 6: Behavioral Simulation waveform for Module-3S

The behavioral simulation waveform of the 8-bit Parallel Counter Architecture Based on State Look-Ahead Logic by instantiating all the above modules is shown in Figure 7.In Figure 1 the inputs are CLK, reset and the output is 8-bit Q . When the reset is high, the output "**Q**" is all zeroes (00000000).If case the reset is low, the counting starts and the output **Q** will change with the positive of the clock pulse, the total number of states for an 8-bit counter are 2^8 (256), which from 0 to 255.

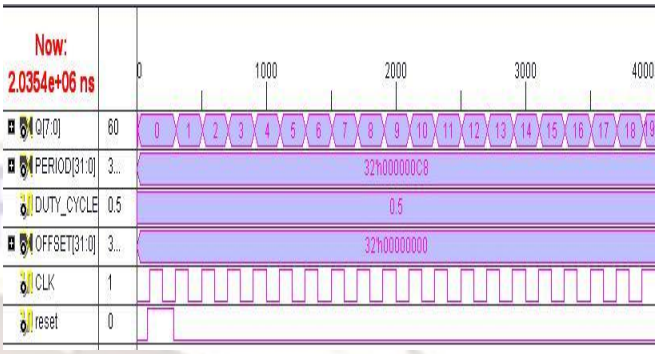

Figure 7: Behavioral Simulation waveform for Parallel Counter Architecture Based on State Look-Ahead Logic

4. FPGA IMPLIMENTATION

Figure 8: FPGA Implementation of Counter

CONCLUSION

In this paper, the counter design logic is comprised of only 2-bit counting modules and three-input AND gates. The counter structure's main features are a pipelined paradigm and state look-ahead path logic whose interpolation activates all modules concurrently at the system's clock edge, thus providing all counter state values at the exact same time without rippling affects. In addition, this structure avoids using a long chain detector circuit typically required for large counter widths. An initial m-bit counting module pre-scales the counter size and this initial module is responsible for generating all early overflow states for modules of higher significance. In addition, this structure uses a regular VLSI topology, which is attractive for continued technology scaling due to repeated module types (module-2S and module-3S) forming a pattern paradigm and no increase in fan-in or fan-out as the counter width increases, resulting in a uniform frequency delay that is attractive for parallel designs. Consequently, the counter frequency is greatly

improved by reducing the gate count on all timing paths to two gates using advanced circuit design techniques However, extra precautions must be considered during synthesis or layout implementations in order to align all modules in vertical columns with the system clock.

REFERENCES

[1] S. Abdel-Hafeez, S. Harb, and W. Eisenstadt, "High speed digital CMOS divide- by-N frequency divider," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2008, pp. 592–595.

[2] M. Alioto, R. Mita, and G. Palumbo, "Design of high-speed power-efficientMOS current-mode logic frequency dividers," IEEE Trans. CircuitsSyst. II, Expr. Briefs, vol. 53, no. 11, pp. 1165–1169, Nov. 2006.

[3] Altera Corp., Santa Clara, CA, "FLEX8000, field programmable gate array logic device," 2008.

[4] J. Ousterhout, Berkeley, 1980, "Berkley magic layout tools,"

[5] B. Chang, J. Park, and W. Kim, "A 1.2 GHz CMOS dual-modulus prescalar using new dynamic D-type flip-flops," IEEE J. Solid-State Circuits, vol. 31, no. 5, pp. 749–752, May 1996.

[6] M. Ercegovac and T. Lang, "Binary counters with counting period of one half adder independent of Counter size," IEEE Trans. Circuits Syst., vol. 36, no. 6, pp. 924–926, Jun. 1989.

[7] M. D. Ercegovac and T. Lang, Digital Arithmetic. San Mateo, CA:Mogan Kaufmann, 2004.

[8] N. Homma, J. Sakiyama, T. Wakamatsu, T. Aoki, and T. Higuchi, "A systematic approach for analyzing fast addition algorithms using counter tree diagrams," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2004, pp. V-197–V-200.

[9] B. Hoppe, C. Kroh, H. Meuth, and M. Stohr, "A 440MHz16 bit counter in CMOS standard cells," in Proc. IEEE Int. ASIC Conf., Sep. 1998, vol. 244, pp. 241–244

[10] Y. Ito, K. Nakano, and Y. Yamagishi, "Efficient hardware algorithms for n choose K counters," in Proc. 20th Int. Parallel Distrib. Process. Symp., Apr. 2006, pp. 1–8.

[11] Y. I. Ismail and E. G. Friedman, On-Chip Inductance in High Speed Integrated Circuits. Norwell, MA: Kluwer, 2001.

Ms. J.L.V Ramana Kumari has obtained her B.Tech degree From VRSEC, Vijayawada in 1991 and M.Tech in VNR VJIET, in 2009 in "VLSI System Design". She is working as Asst. Professor in ECE Dept at VNR Vignana Jyothi Institute of Engineering & Technology, Bachupally, and Hyderabad, India. . Her research interests include VLSI Testing, Image Processing and Digital Design.

Satheesh Bekkam received the B.Tech degree in electronics and communication engineering from Anurag Engineering College, affiliated to Jawaharlal Nehru Technological University Hyderabad, AP, India, in 2010, he is pursuing the M.Tech in VLSI System Design at VNR Vignana Jyothi Institute of Engineering & Technology, Bachupally, Hyderabad, India. His research interests include VLSI Chip Design (ASIC), Digital Design (FPGA).