Int. J. Adv. Res. Sci. Technol. Volume 2, Issue3, 2013, pp 127-129.

IJARST

International Journal of Advanced Research in Science and Technology

journal homepage: www.ijarst.com

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

Implementation of Area Efficient Barrel Shifter

A.V.S Naidu* and B. Vijayabhaskar

Department of Electronics and Communications Engineering, St. Theresa Institute of Engineering College, Vijayanagaram, Andhra Pradesh, INDIA.

****Corresponding Author's Email: naidu_avs@yahoo.com***

A R T I C L E I N F O

Article history:

Received 10 Oct. 2013

Accepted 26 Oct. 2013

Available online 30 Oct. 2013

Keywords:

8x1 mux barrel shifter,
2x1 mux barrel shifter,
area efficient barrel shifter,
high speed barrel shifter.

A B S T R A C T

Barrel shifters are often utilized by embedded digital signal processors and general purpose processors to manipulate data. Barrel shifters are widely used where more than one bit shift operation is required in arithmetic and logical operations. The design of barrel shifter is a most challenging task towards its speed, area and power consumption. Here we are going to implement an 8-bit barrel shifter in the form of two approaches and we are going to analyze the circuits in terms of speed area and power consumption. Here the circuits are implemented and analyzed by using the most popular semi-custom design tool VHDL and will be synthesized by using SPARTAN-3 FPGA.

Introduction:

Barrel Shifter is a combinational circuit with n data inputs, n data outputs and a set of control inputs that specify shift between input and output. It is widely used in Arithmetic and logic units of processors. In n-bit data processing system n-bit barrel shifter should be designed. But n-bit barrel shifter requires n number of n-bit multiplexers. If n is increased the circuit complexity also increases i.e., circuit overhead, it leads to occupy more area and high power consumption and also shows the effect on speed of the operation. A common usage of a barrel shifter is in the hardware implementation of floating-point arithmetic. For a floating-point add or subtract operation, the significant of the two numbers must be aligned, which requires shifting the smaller number to the right, increasing its exponent, until it matches the exponent of the larger number. This is done by subtracting the exponents, and using the barrel shifter to shift the smaller number to the right by the difference, in one cycle.

Shifting and rotating data is required in several applications including arithmetic operations, variable-length coding, and bit-indexing. Consequently, barrel shifters, which are capable of shifting or rotating data in single cycle, are commonly found in both digital signal processors and general purpose processors.

A VHDL approach is used to especially for larger designs that will be realized in a single CPLD, FPGA or ASICs. These examples do not target a specific CPLD or FPGA, this is one of the advantages of HDL-based designs, most or all of the designs are effort is portable and can be targeted to any of a variety of technologies.

8-Bit Barrel Shifter with 8-Bit Mux:

The straight forward design of an 8-bit barrel shifter is consisting eight number of eight bit multiplexers as we discussed in the introduction. Each multiplexer consists eight inputs, three select lines and one output. Here all the select lines are having common connection, these select lines are decides the shifting operation in barrel shifter. The three select lines are able to generate eight different combinations; each combination represents the number of shifts that are being performed. Usually if the select lines are all zeroes, zero bit shift operation takes place and if the select lines combination is 001 then the output will be shifted 1-bit right or left (rotation) depending upon the direction decided by the designer. It is a

single stage design because all the multiplexers are placed vertically.

The design is implemented, simulated and synthesized. The schematic is shown in fig.1, and followed by the simulation results in fig.2. And the results are compared with same design but it is implemented by using 2-bit multiplexers for the purpose of finding the performance of the two designs. Here we have another approach i.e., an algorithm which is used to develop the same application by using 2-bit multiplexer.

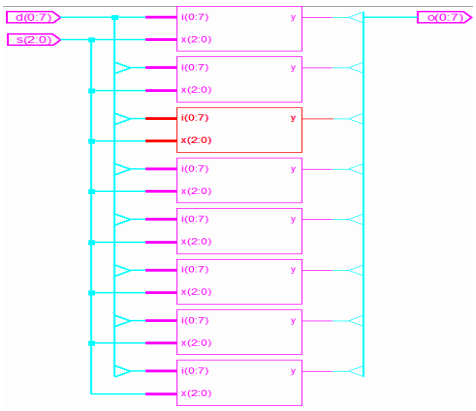

Fig: 1. Schematic of 8-bit barrel shifter

The simulation results show the actual delay of shifting operation. In this all the timing delays are extracted. These delays are varies from technology to technology. Here we are using SPARTAN-3 FPGA technology both for simulation as well as for syntheses. From this we can find out the area, speed and power. But here we are concentrating on only two issues that is area and speed. For this many number of designs are implemented, simulated and synthesized. Finally the results are concluded in between two designs. In the next part we are going explore 8-bit barrel shifter by using 2-bit multiplexers.

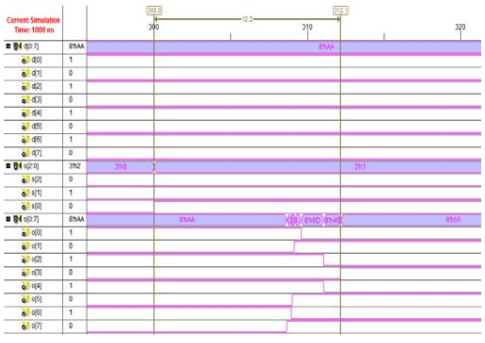

Fig: 2. simulation results by using 8-bit mux

8-Bit Barrel Shifter with 2-Bit Mux:

Here the 8-bit barrel shifter is implemented by using two input multiplexers. The difference between 8-bit barrel shifter with 8-bit mux and 8-bit barrel shifter with 2-bit mux is the number of stages that the design consists and the hardware utilization that is area occupied by the design. With 2-bit mux, the design consist three stages. Each stage consist eight 2-bit multiplexers, so total 24 multiplexers are used in this design. Fig.3 shows schematic diagram of 8-bit barrel shifter with 2-bit multiplexer.

Figure 3 shows the fast, compact barrel shifter. Depending on the encoded shift control lines, S0-2, the data inputs D0-7 are shifted when they reach the outputs Q0-7. If S0 is asserted and S1 and S2 are unasserted, the value at D0-7 is passed to the next most significant Q output. For example, D0 is passed to Q1, and D1 passed to Q2, while D7 wraps around and is passed to Q0. If none of the shift controls are asserted, the data inputs D0-7 are passed to the corresponding outputs Q0-7 without being shifted.

Here the hardware utilization is much reduced; because of this reason area occupied by the design is reduced comparatively to the straight design. The delay is little bit increased because of the increased stages. This may not be a big issue because the time difference is very less. As we discussed earlier this is area efficient barrel shifter. The following figure shows schematic of the design of 8-bit barrel shifter with 2-bit multiplexer, and followed by the simulation results of this corresponding design.

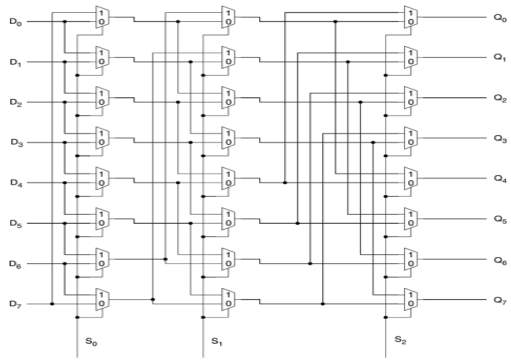

Fig: 3. 8-bit barrel shifter with 2-bit mux.

Fig: 4. simulation results by using 2-bit mux

The number of FPGA resources utilization is reduced. So we can optimize the design to get higher speed that is it consists less delay. The following table shows the area utilization of the each design.

Table: 1. FPGA resources utilization

Designed logic	No.of Slices utilized	No.of LUTs utilized	Delay (ns)
With 8-bit mux	16	32	12.1
With 2-bit mux	14	24	2.8

From the above table it is very easy to understand that, implementing 8-bit barrel shifter with 8-bit multiplexer is utilized more resources than 8-bit barrel shifter with 2-bit multiplexer. Even though the second design takes three stages for implementation, it occupies less area and be achieved the same speed through optimization.

Conclusion:

The design of 8-bit barrel shifter with 2-bit multiplexer is implemented and compared with another one. Even though the delay is a little bit more (about 0.7ns of 3-stages) the remaining resources are best optimized to the previous design. For larger design implementation, the designer may concentrate on area utilization. The proposed design has taken only 24 LUTs and 14 slices for 8-bit barrel shifter. Finally the design is occupied less area and achieved the same speed.

Acknowledgment:

This material is based upon work supported by the students of Avanthi Institute of Engineering and Technology. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily react the views of AIET.

References:

1. S. R. Nassif, “Process Variability at the 65nm Node and Beyond,” in *IEEE Custom Integrated Circuits Conference*, Sept. 2008.
2. S. Borkar, “Design Perspectives on 22nm CMOS and Beyond,” in *ACM/IEEE Design Automation Conference*, July 2009.
3. E. Axelsson, “Functional Programming Enabling Flexible Hardware Design at Low Levels of Abstraction,” Ph.D. dissertation, CSE Dept.,Chalmers University of Technology, 2008.
4. K. P. Subramaniyan, E. Axelsson, M. Sheeran, and P. Larsson-Edefors, “Layout Exploration of Geometrically Accurate Arithmetic Circuits,” in *IEEE International Conference on Electronics, Circuits, and Systems*, Dec. 2009.
5. H. Zhu, Y. Zhu, C. Cheng, and D. Harris, “An Interconnect-Centric Approach to Cyclic Shifter Design Using Fanout Splitting and Cell OrderOptimization,” in *Asia and South Pacific Design Automation Conference*,Jan. 2007.
6. S. Huntzicker, M. Dayringer, J. Soprano, A. Weerasignhe, D. Harris, and D. Patil, “Energy-Delay Tradeoffs in 32-bit Static Shifter Designs,” in *IEEE International Conference on Computer Design*, Oct. 2008.
7. R. Pereira, J. A. Mitchell, and J. M. Solana, \Fully Pipelined TSPC Barrel Shifter for High-speed Applications," IEEE Journal of Solid State Circuits, vol. 30, pp. 686{690, June 1995.
8. P. A. Beerel, S. Kim, P.-C. Yeh, and K. Kim, \Statistically Optimized Asynchronous Barrel Shifters for Variable Length Codecs," in Proceedings of the International Symposium on Low Power Electronics and Design, pp. 261{263, 1999.