Journal of Advances in Science and Technology
Vol. 12, Issue No. 25, (Special Issue) December-2016, ISSN 2230-9659

Design and Implementation of 8 Bit Barrel Shifter Using 2:1 Multiplexer in Verilog

S. B. Jondhale[1]*, T. S. Mulla[2], S. S. Patil[3]

[1]ECE Dept., HIT, Nidasoshi, India

[2]ECE Dept., HIT, Nidasoshi, India

[3]Assistant Professor, ECE Dept, HIT, Nidasoshi, India

Abstract – Barrel shifter is one of the most important data path elements and widely used in many key computer operations from address decoding to computer arithmetic, using basic operations like data shifting or rotation. In this paper multiplexer based barrel shifter circuit is implemented using the hardware description language "Verilog". The proposed barrel shifter architecture implementation shows large reduction in propagation delay, while keeping the almost similar average power consumption.

----------♦----------

I. INTRODUCTION

1. **Verilog Hdl *:*** Verilog language is used in design and implementation of the electronic circuits. The circuit description in Verilog language can be done using 4 description methods namely Data Flow, Behavioral, Structural, Mixed Mode and Mixed language. The barrel shifter implementation here is done by using the structural type of style.

2. **Barrel shifter using 2:1 MUX:** A barrel shifter is a digital circuit that can shift a data word by a specified number of bits in one clock cycle. It can be implemented as a sequence of multiplexers, and in such an implementation the output of one multiplexer is connected to the input of the next multiplexer in a way that depends on the shift distance. A barrel shifter is often implemented as a cascade of parallel 2×1 multiplexers. For example, take 8-bit barrel shifter, with inputs A,B,C,D,E,F,G,H. The barrel shifter can cycle order of bits ABCDEFGH as HABCDEFG, GHABCDEF etc…in this case, no bits are lost.

A barrel shifter is simply a bit-rotating shift register. The bits shifted out of LSB end of the register are shifted back into the MSB end of the register. The number of multiplexers required is n*log2(n), for an n bit word. Four common word sizes and the number of multiplexers needed are listed below:

64_bit -64*log2(64)=64*6=364

32_bit -32*log2(32)=32*5=160

16_bit -16*log2(16)=16*4=64

8_bit -8*log2(8)=64*3=24

II. BLOCK DIAGRAM:

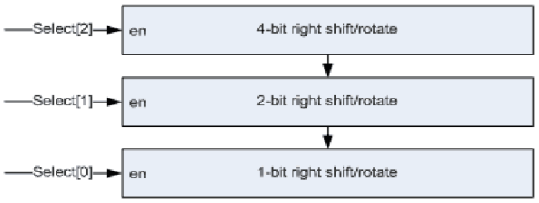


Figure 1: Block diagram of barrel shifter

III. CIRCUIT DIAGRAM:

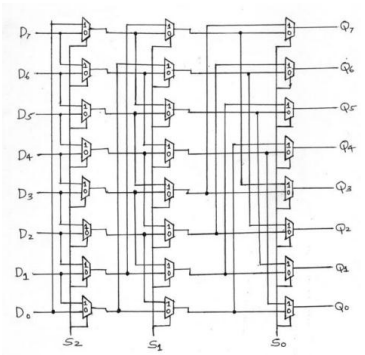


Figure 2: Circuit diagram of barrel shifter

IV. PORT SPECIFICATION:

D7 - D0	8 – bit input
Q7 - Q0	8 - bit output
S2 – S0	3 - bit select lines If S0 is high - input shifts by 4 bits If S1 is high - input shifts by 2 bits If S2 is high - input shifts by 1 bit
	per select lines

V. WORKING:

The above circuit shows a multiplexer based 8-bit barrel-shifter. The circuit allows rotating the input data word right, where the amount of rotating is selected by the control inputs. Several microprocessors include barrel-shifters as part of their ALUs to provide fast shift or rotate open.

The circuit shown above consists of three stages of 2:1 multiplexers. When all multiplexer select inputs are active (low), the input data passes straight through the cascade of the multiplexers and the output data (q7..q0) is equal to the input data (d7. d0). When *S2*control signal is selected, the first stage of multiplexers performs a rotate-right by one bit operation, due to their interconnection to the next-lower input.

Similarly, the second stage of multiplexers performs a rotate-right by two bits when*S1* control signal is selected. Here the corresponding multiplexer inputs are connected to their second next-lower input.

Finally, the third stage of multiplexers performs a rotate-right by four bits, when S0 control signal is selected.

Due to the cascade of three stages, all three rotate operations (by one bit, by two bits, by four bits and so on) can be activated independently from each other. For example, when both S2 and S0 are activated, the shifter performs a rotate-right by five bits.

VI. ROTATE OPERATION:

The rotate operation is a shift where the bit which is shifted out of the vector LSB is inserted at its MSB.

Operation	Q
3-bit shift right logical	000d0d1d2d3d4
3-bit shift right arithmetic	d0d0d0d0d1d2d3d4
3-bit rotate right	d5d6d7d0d1d2d3d4
3-bit shift left logical	d3d4d5d6d7000
3-bit shift left arithmetic	d0d4d5d6d7000
3-bit rotate left	d3d4d5d6d7d0d1d2

Table 1: shift and rotate examples for D=d0d1d2d3d4d5d6d7 and S = 3.

3-bit Opcode			Operation
left	Rotate	arithmetic	
0	0	0	shift right logical
0	0	1	shift right arithmetic
0	1	X	rotate right
1	0	0	shift left logical
1	0	1	Shift left arithmetic
1	1	X	rotate left

Table 2: operation of control bits

VII. 2:1 MUX:

In electronics, a mux is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A mux of 2*n* inputs has *n* select lines, which are used to select which input line to send to the output. Mux are mainly use increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A mux is also called a data selector. A straightforward realization of this 2-to-1 mux would need 2 AND gates, an OR gate, and a NOT gate.

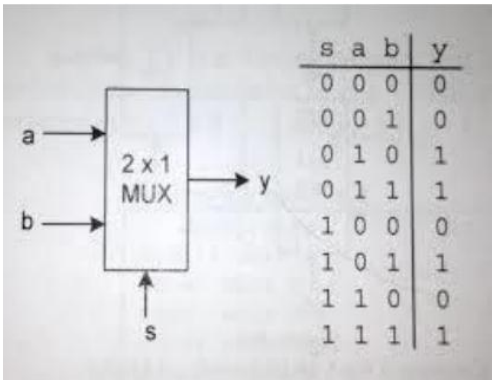


Figure 3: symbol and truth table of 2:1 mux

S. B. Jondhale[1]*, T. S. Mulla[2], S. S. Patil[3]

VIII. TRUTH TABLE OF BARREL SHIFTER IMPLEMENTATION:

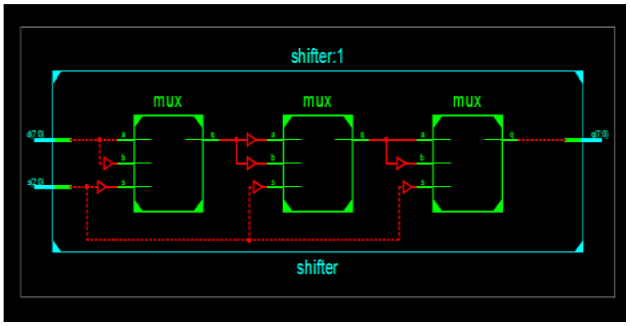
Select line			Input (d[7:0])	Output (q[7:0])
S0	S1	S2		
0	0	0	00001111	00001111
0	0	1	00001111	10000111
0	1	0	00001111	11000011
0	1	1	00001111	11100001
1	0	0	00001111	11110000
1	0	1	00001111	01111000
1	1	0	00001111	00111100
1	1	1	00001111	00011110

Table3: Truth table of barrel shiftr implementation

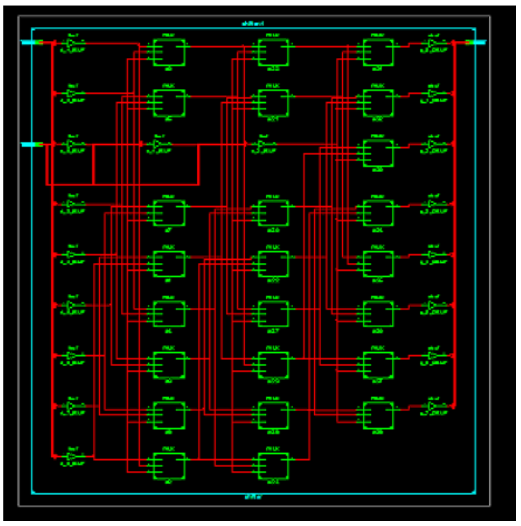
IX. RESULT:

The design and implementation of Barrel shifter using 2:1 multiplexer was tested and verified on Xilinx 13.1 ISE. The following is the RTL view generated:

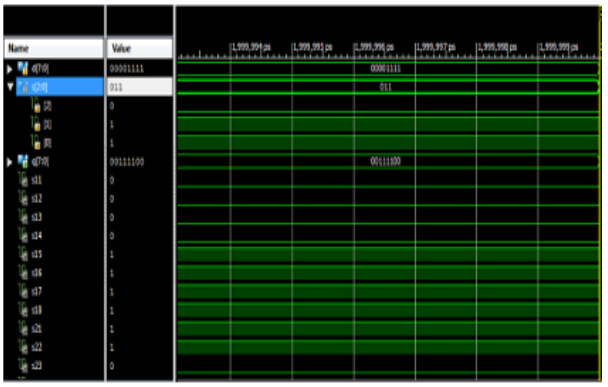
RTL View:



Technological View:



Simulation Results:



X. ADVANTAGES:

1. Using Barrel shifter input word is either rotated left or right or remains unchanged as per selected line at a time in one clock cycle.
2. The propagation delay of barrel shifter is theoretically constant and independent of the shift value or shifter size.
3. Barrel shifter is approximately for smaller shifts for larger shifts values, logarithmic shifter becomes more effective.

XI. APPLICATIONS:

1. Several microprocessors incorporate it as a part of their ALU to provide fast shift operations.
2. Barrel shifter circuits are essential elements in the design of data paths for DSP applications.
3. Barrel shifter are often are required for performing data shifting or rotation in many key computer operations from address decoding to computer arithmetic.
4. It is used extensively in floating point units, scalars and multiplications by constant numbers.

XII. CONCLUSION:

Design of 8-bit barrel shifter using 2:1 multiplexer is successfully implemented and tested using Xilinx13.1 ISE. This design can be expanded to larger value of bits by using the same design by introducing slight changes. A barrel shifter can become stepping stone to improving computer organization and memory.

XIII. REFERENCE:

Implimentation of barrel shifter using multiplexers by Paul Gigliotti.

Design alternatives for barrel shifter by Michael J. Schulte.

Combinational design examples by John F. Warkely.

Corresponding Author

S. B. Jondhale*

ECE Dept., HIT, Nidasoshi, India

E-Mail – 197sneha.j@gmail.com