ISSN No. 0976-5697

Volume 4, No. 3, March 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Designing of low power high speed 32-bit barrel shifter for ARM7 Processor

Latif khan
M. Tech Scholar,VLSI Design
Mewar university, Chittorgarah ,India
latifnizami@gmail.com

Abstract: - The main concern of this paper is to design and study about barrel shifter of 32 bit that is used in (ARM7) RISC processor using VHDL. The instructions include a set of 16- data processing instructions, two- 32-bit multiply instructions, branch instructions and a store instruction. Shifting instructions will be done by a barrel shifter unit.First barrel shifter is designed in VHDL and simulated on Modelsim and after it to get a synthesizable net list, and to generate RTL schematic VHDL code is run on XILINX ISE 12.1. After getting RTL schematic we calculate power and no. of transistor and layout through Tanner EDA 14.1 tool and MENTOR GRAPHICS(IC-Station) using 16 nm technology.

Keywords: ARM**,**Barrelshifter,RISC,ARM,Pipeline,crossbar switchVHDL,Microcontrol,Microprogram,Tanner

I. INTRODUCTION

The barrel shifter is simply a bit-rotating shift register. The bits shifted out the MSB end of the register are shifted back into the LSB end of the register. In a barrel shifter, the bits are shifted the desired number of bit positions in a single clock cycle. For example, an eight-bit barrel shifter could shift the data by two positions in a single clock cycle. If the original data was 10101011, one clock cycle later the result will be 10101110.

A. RISC PROCESSOR(ARM7):

ARM is short for Advanced RISC Machines Ltd. Founded 1990, owned by Acorn, Apple and VLSI Known before becoming ARM as computer manufacturer Acorn which developed a 32-bit RISC processor for its own use (used in Acorn Archimedes). ARM is one of the most licensed and thus widespread processor cores in the world.

a. Used especially in portable devices due to low power consumption and reasonable performance (MIPS / watt)
b. Several interesting extensions available or in development like Thumb instruction set and Jazelle Java machine. (ARM7TDMI)

Block diagram of typical RISC processor (ARM7) is given below

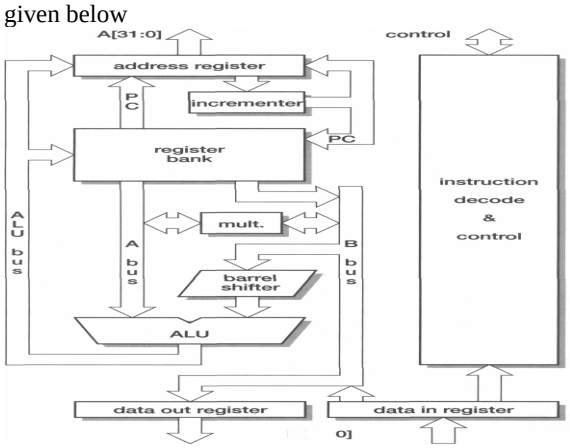

Figure 1: Block diagram of RISC processor (ARM7)

II. BARREL SHIFTER OVERVIEW

A. Four bit barrel shifter

The ARM architecture supports instructions which perform a shift operation in series with an ALU operation.The shifter performance is therefore critical since the shift time contributes directly to the data path cycle time as shown in the later diagram. (Other processor architectures tend to have the shifter in parallel with the ALU, so as long as the shifter is no slower than the ALU it does not affect the data path cycle time.)

In order to minimize the delay through the shifter, a cross-bar switch matrix is used to steer each input to the appropriate output. The principle of the cross-bar switch is illustrated in Figure 3.1 where a 4 x 4 matrix is shown. (The ARM processors use a 32 x 32 matrix.)

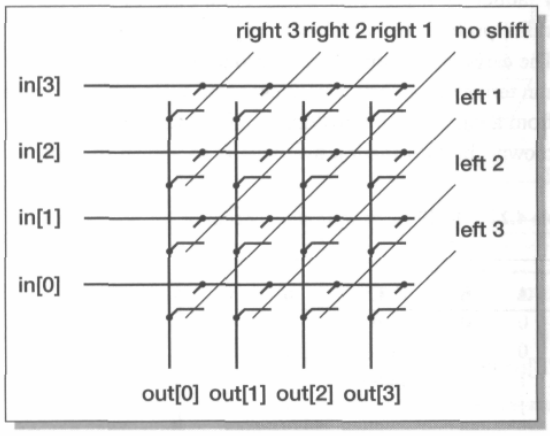

Figure 2 : The cross-bar switch barrel shifter principle (Steve Ferber, "ARM SoC Architecture", 2nd Edition)

Each input is connected to each output through a switch. If pre-charged dynamic logic is used, as it is on the ARM data paths, each switch can be implemented as a single NMOS transistor.

The shifting functions are implemented by wiring switches along diagonals to a common control Input:

a. For a left or right shift function, one diagonal is turned on. This connects all the input bits to their respective outputs where they are used. (Not all are used, since some bits 'fall off the end'.) In the ARM the barrel shifter operates in negative logic where a ' 1' is represented as a potential near ground and a '0' by a potential near the supply. Recharging sets all the outputs to a logic '0', so those outputs that are not connected to any input during a particular switching operation remain at '0' giving the zero filling required by the shift semantics.

b. For a rotate right function, the right shift diagonal is enabled together with the complementary left shift diagonal. For example, on the 4-bit matrix rotate right one bit is implemented using the 'right 1' and the 'left 3' (3 = 4 - 1) diagonals.

c. Arithmetic shift right uses sign-extension rather than zero-fill for the unconnected output bits. Separate logic is used to decode the shift amount and discharge those outputs appropriately.

B. Eight bit barrel shifter:

Functionally, since any bit can end up in any bit position, multiplexers are used to place the bits correctly for proper storage. Thus, a barrel shifter is implemented by feeding an N-bit data word into N, N-bit-wide multiplexers. An eight-bit barrel shifter is built out of eight flip-flops and eight 8-to-1 multiplexers; a 32-bit barrel shifter requires 32 registers and thirty-two, 32-to-1 multiplexers, and so on. A schematic representation of an 8-bit barrel shifter is shown in Figure 3.2.

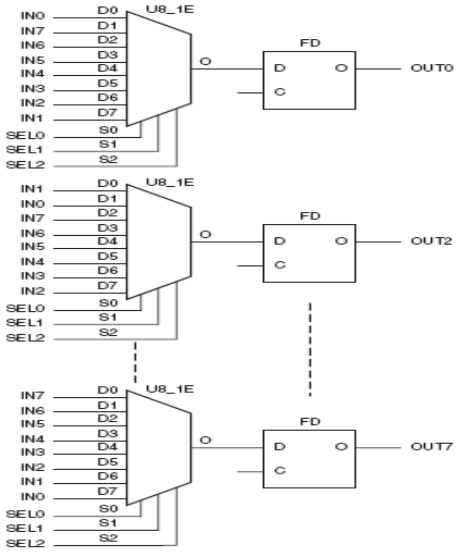

Figure 3: Eight bit barrel shifter

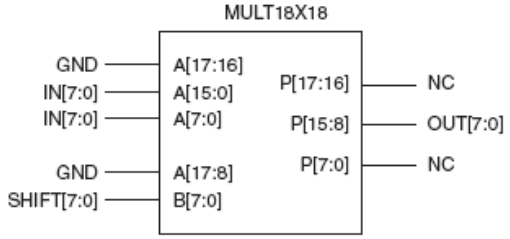

Figure 4: MULT18X18 (Paul Gigliotti," Implementing Barrel Shifters Using Multipliers")

C. Single clock 32-bit Barrel shifter-:

As above mentioned, a 32-bit barrel shifter requires thirty-two, 32-to-1 multiplexers. A 32-to-1 multiplexer can be implemented in a Vertex -II device using two CLBs. Only sixty-four CLBs are required to accomplish all the required multiplexing. By using a Vertex-II multiplier based barrel shifter, a 32-bit barrel shifter is built using four 8-bit barrel shifters and thirty-two 4-to-1 multiplexers. The diagram on the left side of Figure 3 is a single-cycle, 32-bit barrel shifter. The input bus is broken down into four 8-bit words. The data is "processed" in two stages. The first stage is built out of the 8-bit barrel shifters. This stage provides the "fine" shifting, moving the bits from adjoining bytes. After the first stage the appropriate bits are stored in a byte, but the bytes need to be reordered. The reordering of the bytes, or "bulk" shifting, is provided in the second stage, shown on the right in Figure 3. As previously mentioned, the 8-bit barrel shifter requires the shift amount to be one-hot encoded. Also, the three LSBs are used to control the fine shifting, and the two MSBs are used to control the bulk shifting.

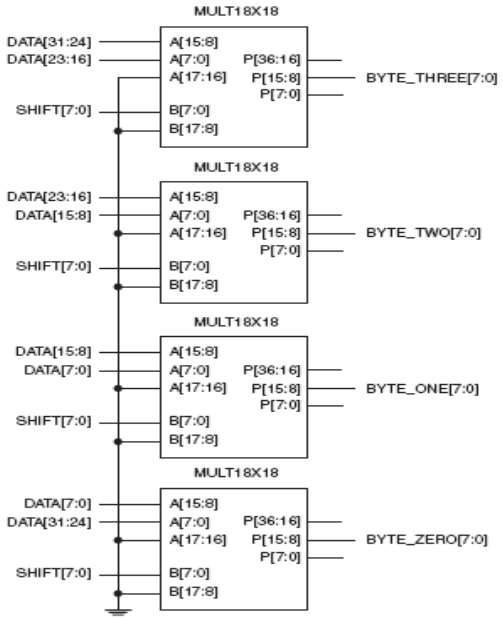

Figure 5: Single-Cycle, 32-bit Barrel Shifter (Paul Gigliotti," Implementing Barrel Shifters Using Multipliers")

III. DESIGN METHODOLOGY

The design flow shown below in figure 2.1 is typically used by the designer who use HDL's. In any design, specifications are written first. Specification describes abstractly the functionality, interface and overall architecture of the digital design circuit to be designed. Behavioral Description is then created to analyze the design in term of functionality, performance, compliance to standards and other high level issue.

The behavioral description is manually converted into RTL description in an HDL. The designer has to describe the data flow that will implement the desired digital circuit .Logic synthesis tools convert the RTL description to a gate level net list. The gate level net list is input to the automatic placement and route tool, which

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

create the layout. The layout is finally verified and the fabricated or implemented on the Chip.

Hardware design is done with the related CAD tools. The first step in the hardware design is to prepare the specification of the design (microcontroller) the architecture and the instruction set must be understood thoroughly. The design ideas are the describe with VHDL in text editor complied and simulated. Then, the VHDL code the synthesized with any Synthesis' Tool. If synthesized successfully, it will generate a net list files (EDF files). Then file is then send to active HDL for compilation and simulated. Results are verified by simulation. The Hardware design is repeated until the microcontroller is complete without any errors.

Hardware implementation is performed by downloading the design into the any FPGA device. The hardware implement ion tests the design in real physical environment by some control application. For every application, different programs must be written and store into the program ROM of the microcontroller before it can do the job. So, before the microcontroller is downloaded into FPGA device, the specific program for the application must be written. The Program is put into the ROM of controller and FPGA is programmed for application must be written.

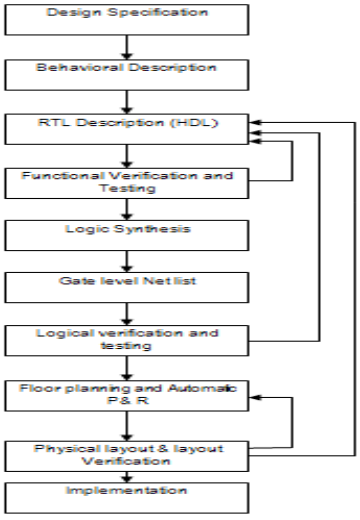

Figure 6: Typical Design flow

IV. SIMULATION RESULT

The inputs to the shifter are the 32-bit input and a 5-bit shift count. The simulation waveforms are shown below

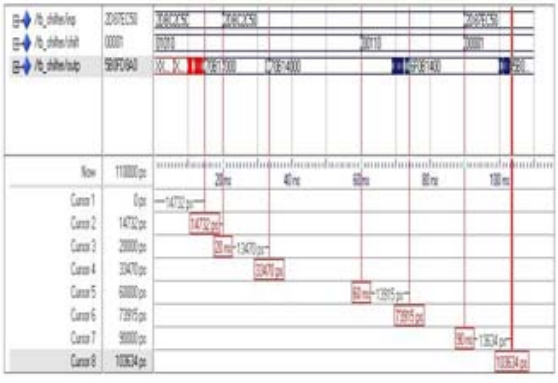

Figure 7:Simulated waveform of barrel shifter

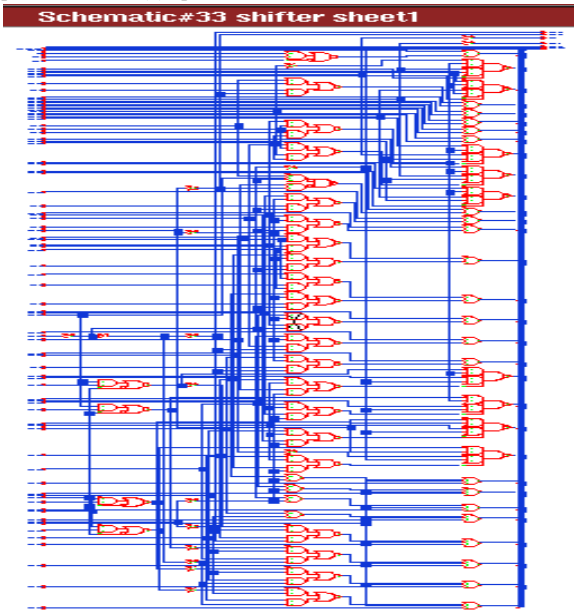

Figure 8: Schematic for Barrel Shifter

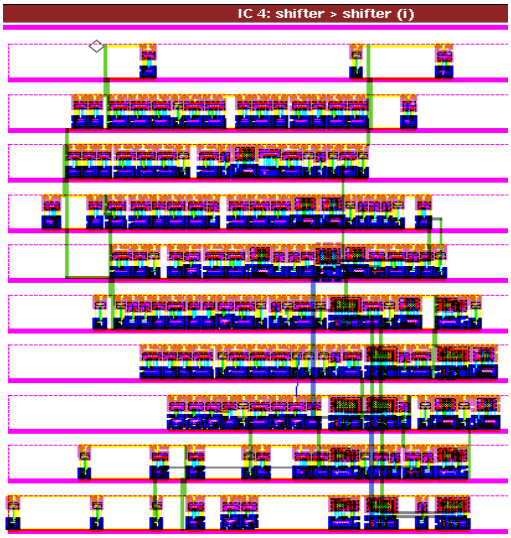

Figure 9: The layout for Barrel Shifter

V. CONCLUSION

So finally we find that our all module is designed successfully and working properly after running a program. Barrel shifter is working properly but the parameters calculated by me and values given by RISC processor (ARM7) datasheet is different, the reason behind it that I am using map file of 350nm and in ARM7, 180nm is used. The result is shown below in table

Table 1: Difference Table

Parameter	Risc processor (arm7)	Proposed
Process	.18u	.35u
Voltage	1.8v	3.5v
Power	<0.25 mW	1.79mW
Frequency	>40Mhz	20Mhz
Critical Path	-	19.47 s

VI. FUTURE WORK

Constrain synthesis of RISC Processor for Area, Power and Timing Analysis.

Test Vector generation for circuit using test vector that consists of fault Modeling.

Future work for RISC Processor

a. Implementation of delay testable enhanced scan flip-flop in the benchmark circuits.
b. Area overhead due to Delay Testable Enhanced Scan Flip-flop for benchmark circuits.
c. Power efficiency for benchmark circuits using Xilinx Power compiler.

VII. REFERENCES

[1]. ARM v7 Instruction manual.
[2]. Steve Ferber, "ARM SoC Architecture", Addison Wesley Longman Publications, 2000.
[3]. DA Patterson & J L Hennessy, "Computer Organization and Design", Morgan Kaufman Publications, 2005.
[4]. A.Chandrakasan, W J Bowhil "Design of High Performance microprocessor circuits", IEEE press.
[5]. Bhaskar, "VDHL Primer", Prentice Hall inc.
[6]. ARMv7-M Architecture Application Level Reference Manual, ARM Limited.
[7]. Zainalabedin Navabi, "VHDL Modular Design and Synthesis of Cores and Systems", TMH.
[8]. Paul Gigliotti, "Implementing Barrel Shifters Using Multipliers", XAPP195 (v1.1), August 17, 2004
[9]. Xilinx ISE Quick Start Tutorial
[10]. Daniel Tabak, RISC Systems, Research Studies Press Ltd.: Taunton, Somerset, England TA1 1HD, 1990
[11]. T. Thomson and H. Tam, "Barrel Shifter," U.S. Patent 5,652,718, July 1997.
[12]. M.R. Pillmeier, "Barrel Shifter Design, Optimization, and Analysis", Lehigh University, January 2002.
[13]. M. Diamondstein and H. Srinivas, "Fast Conversion Two's Complement Encoded Shift Value for a Barrel Shifter," U.S. Patent 5,948,050, September 1999.
[14]. Abhijit Asati1 and Chandrasekhar "VLSI Implementation of a High Performance Barrel Shifter Architecture using Three Different Logic Design Styles" International Journal of Recent Trends in Engineering, Vol 2, No. 7, 22 November 2009.
[15]. G.M. Tharakan and S. M. kang, "Anew design of a fast barrel network,"IEEEJ. Solid–State Circuits, Vol. 27, pp. 217- 221, February 1992.
[16]. R.Pereira, J. A. Michell and J. M. Solana, "Fully Pipelined TSPC Barrel Shifter for Circuits, Vol. 30, pp.686- 690, June 1995. High-speed Applications," IEEE J. Solid –State