DESIGNING OF LOW POWER GDI BASED 8-BIT BARREL SHIFTER

[1]Sejal Shah, [2]Latif Khan, [3]Vijendra Kumar Maurya
[1]Student, [2]Assistant Professor, [3]Assistant Professor
[1]VLSI, Electronics and Communication Department,
Geetanjali Institute of Technical studies, Udaipur, India

Abstract : A barrel shifter is a circuit that can shift a data word by a specified number of bits without the use of any sequential logic, only pure combinatorial logic. The main concern of this paper is to design and study about barrel shifter of 8-bit that is used in RISC processor using GDI (Gate-Diffusion-Input) technique. The main objective of this paper is to design a fully custom 8 bit barrel shifter using 8x1 multiplexer with the help of GDI technique and analyze the performance on basis of power consumption and number of transistors.
The tool used to fulfill the purpose is Tanner EDA15.1 using 180nm technology.

Keywords
Barrel Shifter,RISC,ARM,Pipeline,GDI,Multiplexer, 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.
Barrel shifter is only a bit rotating shift register. The bits shifted out of LSB end of the register are shifted back into the MSB side of the register. The number of multiplexers are required is decided n*log2(n), for an n bit word. Four common word length and the number of multiplexers essential 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
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.

1.1 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 8-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.

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

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

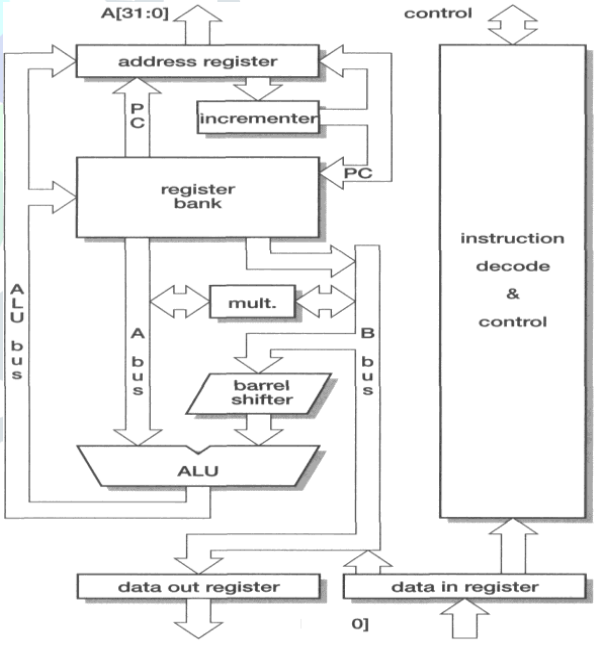

Figure1: Block diagram of RISC processor (ARM7)

II. BARREL SHIFTER OVERVIEW

2.1 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, where a 4 x 4 matrix is shown. (The ARM processors use a 8 x 8 matrix.)

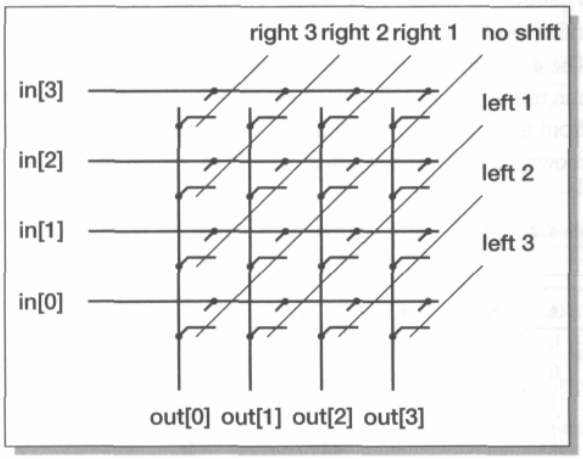

Figure 2 : The cross-bar switch barrel shifter principle (Steve Ferber, "ARM SoC Architecture", 2nd Edition)

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:

- 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.
- 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.
- 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.

2.2 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 8-bit barrel shifter requires 8 registers and thirty-two, 8-to-1 multiplexers, and so on. A schematic representation of an 8-bit barrel shifter is shown in Figure 3.

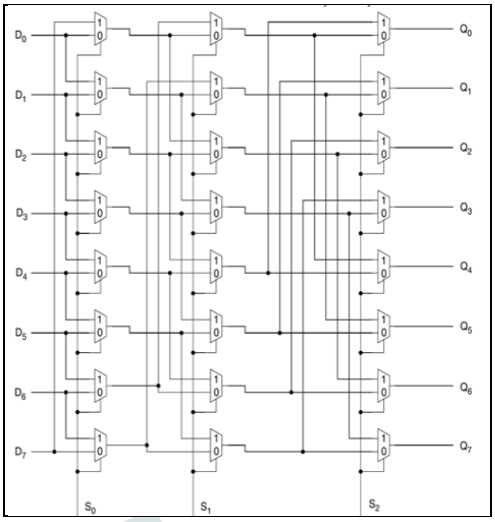

Figure 3: 8- bit barrel shifter

Table[1]: Truth Table for 8-Bit Barrel Shifter

VS2	VS1	VS0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	D0	D7	D6	D5	D4	D3	D2	D1
0	1	0	D1	D0	D7	D6	D5	D4	D3	D2
0	1	1	D2	D1	D0	D7	D6	D5	D4	D3
1	0	0	D3	D2	D1	D0	D7	D6	D5	D4
1	0	1	D4	D3	D2	D1	D0	D7	D6	D5
1	1	0	D5	D4	D3	D2	D1	D0	D7	D6
1	1	1	D6	D5	D4	D3	D2	D1	D0	D7

III. GDI METHOD AND COMPONENTS DESIGN

The Gate-Diffusion-Input (GDI) method is based on the use of a simple cell as shown in Figure 5. One may be think of the CMOS inverter in the first look of this circuit, but there are some major differences in the two:

(1) The GDI cell contains three inputs—G (common gate input of NMOS and PMOS), P (input to the source/drain of PMOS), and N (input to the source/drain of NMOS).
(2) Bulks of both NMOS and PMOS are connected to N or P (respectively), so it can be arbitrarily biased in contrast to CMOS inverter.

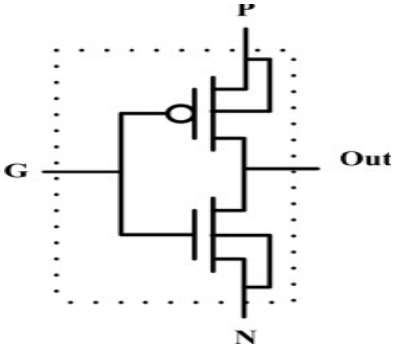

Figure 4: Basic Gate-Diffusion-Input cell [4]

The GDI cell with four ports can be recognized as a new Multi-functional device, which can achieve six functions with just different combinations of inputs G, P and N. Table 1 shows that simple configuration changes in the inputs G, P, and N of the basic GDI cell can lead to very different Boolean functions at the output Out. Most of these functions are complex (usually consume 6-12 transistors) in CMOS, while very simple (only 2 transistors per function) in the GDI design methodology. Meanwhile, multiple-input gates can be implemented by combining several GDI cells.

Table 2: GDI cell functions

Sr.no	Input			Output	Function
	G	P	N		
1	A	B	0	$\overline{A}.B$	F1
2	A	1	B	$\overline{A}+B$	F2
3	A	B	1	A+B	OR
4	A	0	B	A.B	AND
5	A	B	C	$\overline{A}.B+AC$	MUX
6	A	1	0	$\overline{A}$	NOT

The XOR gate made with the help of GDI cell is a application of the GDI technique. As it can be seen in Fig., the XOR made using GDI technique requires only four transistors. Obviously, the proposed GDI XOR gate use less transistors compared with the conventional CMOS XOR.

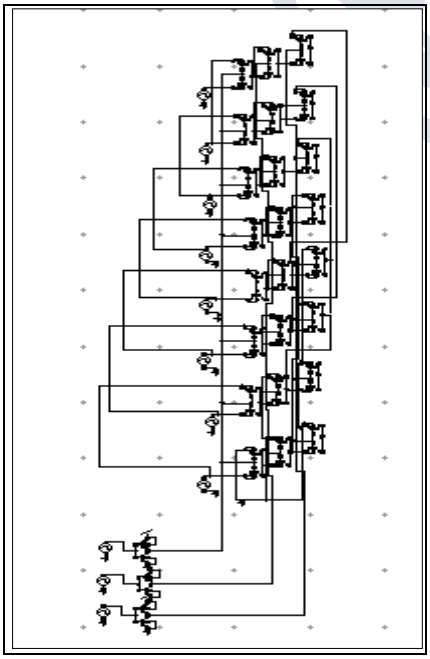

Figure 5: Design of 8-Bit Barrel Shifter using GDI

IV. SIMULATION RESULT-

The inputs to the shifter are the 8-bit input and a 3-bit shift count. The simulation waveforms are shown below:

To make a 1-bit right rotation when the control signals are configured from VS0 VS1 VS2 = 0 0 0 to VS0 VS1 VS2 = 1 1 1 and the simulation result for the 8-bit barrel shifter has shown in figure 6 at VS0 = 0, VS1 = 1 and VS2 = 0 with the input 10011110.

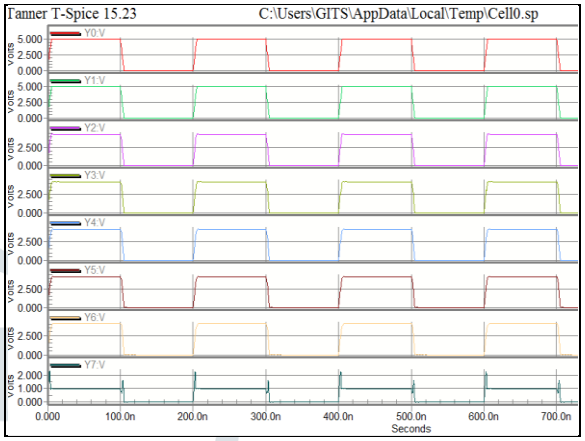

Figure 6: Output Waveform with count 1

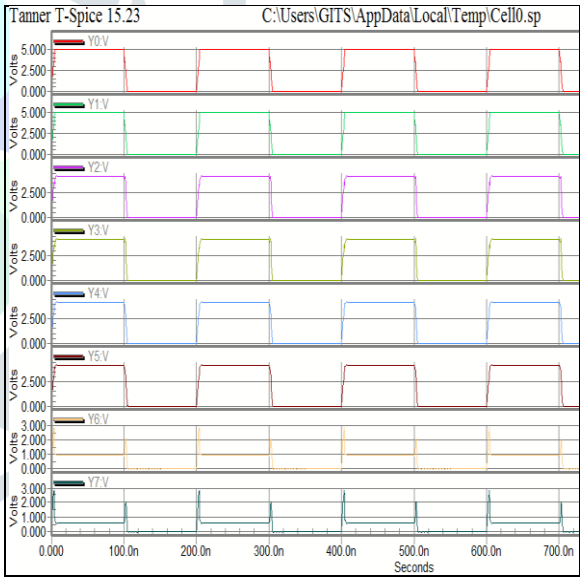

Figure 7: Output Waveform with count 2

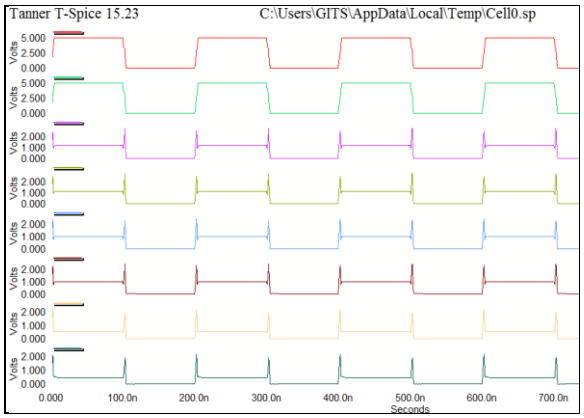

Figure 8:Output Waveform with count 3

V. CONCLUSION

We have proposed and designed a GDI implementation of based 8 Bit barrel shifter. Which produce appreciable results.It demonstrated that our approach yields & high speed barrel shifter.. The result is shown below in table

Table 3: Difference Table

8 bit Shifter Designs	Average Power (nW)	Delay (nS)
CMOS Logic	1926	89.87
Pass Transistor Logic	781.2	40.63
Proposed Technique	484.26	28.96

REFERENCES

[1] ARM v7 Instruction manual.

[2] RenukaVerma, Rajesh Mehra, "Area Efficient Layout Design AnalysisOf CMOS Barrel Shifter," International Journal of Scientific Research Engineering & Technology (IJSRET), pp. 84-89, March 2015.

[3] A Sharma, Rajesh Mehra, "Area and Power Efficient CMOS Adder Design By Hybridizing PTL and GDI Technique," International Journal of Computer Applications, Vol.66, No.4, pp.15-22, March 2013.

[4] Shilpa Thakur, Rajesh Mehra, "CMOS Design and Single Supply Level Shifter Using 90nm Technology," Conference on Advances in Communication and Control Systems, pp.150-153, 2013.

[5] R Singh, Rajesh Mehra, "Power Efficient design of Multiplexer using Adiabatic logic," International Journal Of Advances Engineering and Technology, Vol.6, Issue.1, pp.246-254, March 2013

[6] Steve Ferber, "ARM SoC Architecture", Addison Wesley Longman Publications, 2000.

[7] DA Patterson & J L Hennessy, "Computer Organization and Design", Morgan Kaufman Publications, 2005.

[8] A.Chandrakasan, W J Bowhill, "Design of High Performance microprocessor circuits", IEEE press.

[9] Bhaskar, "VDHL Primer", Prentice Hall inc.

[10] RMv7-M Architecture Application Level Reference Manual, ARM Limited.

[11] Paul Gigliotti, "Implementing Barrel Shifters Using Multipliers", XAPP195 (v1.1), August 17, 2004

[12] T. Thomson and H. Tam, "Barrel Shifter," U.S. Patent 5,652,718, July 1997.

[13] M.R. Pillmeier, "Barrel Shifter Design,Optimization, and Analysis", Lehigh University, January 2002.

[14] M. Diamondstein and H. Srinivas, "Fast Conversion Two's Complement Encoded Shift Value for a Barrel Shifter," U.S. Patent 5,948,050,