Serializing the Data Bus of the Sun OpenSPARC T1 Microprocessor Datapath for Reduced Power Consumption

R K Megalingam
Senior Lecturer, Department of ECE
+91-4762801280
rajeshm@am.amrita.edu

V Krishnan, M Nair
Student, Department of ECE
+91-9496341079
venkatkrishnan.b@gmail.com

V Sarma, R Srikumar
Student, Department of ECE
+91-9746658370
vineethisalways4u@gmail.com

ABSTRACT
Power consumption in processors have become a major issue in these days. This paper considers an efficient technique of serializing the datapath to reduce the power consumption as in [1]. We use the 64 bit datapath of Sun OpensSPARC T1 processor. This processor has four basic blocks for data manipulations which along with the register file and the bypass logic forms the datapath. Serialization is applied to all the four blocks which are the shift, multiply, divide, and Arithmetic and Logic (ALU) blocks. Power estimation and analysis are done for ALU and multiply blocks. We have introduced a module called serializer which is included as part of the bypass logic, to serialize the data path. Serializing can be brought about without much compromise in the speed but this paper emphasizes on the reduction in power consumption. The modified, bit serialized datapath of OpenSPARC T1 is implemented in Verilog HDL. Power analysis of original, parallel datapath and the modified, bit serialized datapath designs of OpenSPARC is done using Xilinx ISE 10.1 Power Analyzer. The results are discussed at the end of this paper.

Categories and Subject Descriptors
B.5.1 [**Register Transfer Level Implementation]**: Datapath design.

General Terms
Design

Keywords
Datapath, power consumption, serializing, bus lines, OpenSPARC T1.

1. INTRODUCTION
Datapath of a processor is one of the major power consuming blocks. When the speed of operation is being increased in various series of processors, power consumption is also a critical issue which needs a proper solution. Existing approaches to reduce power often achieve power reduction at the cost of increased design complexity, thus resulting in delay and area overheads. But any method that conserves power without compromising much in delay and area would be of great use [1]. Serializing the datapath discussed in this paper not only reduces the power consumption but also conserves the efficiency and effectiveness.

ICWET'10, February 26–27, 2010, Mumbai, Maharashtra, India.

The paper is organized in such a way that the first part gives a general introduction to the OpenSPARC T1 architecture and its datapath. The second part gives the logic behind the reduction in power consumption by reducing the number of data bus lines. The final part of the paper discusses the comparison of power analysis results for the original, parallel and modified, bit serialized data paths, which consist of ALU, multiply, divide and shift blocks of OpenSPARC T1 processor.

2. BACKGROUND OF THE PROPOSED WORK

2.1 OpenSPARC T1 Datapath
OpenSPARC T1 microprocessor implements eight cores with each core having four virtual processor or strands. The processor implements core multi threading or CMT. But what we concentrate is on the datapath of the OpenSPARC T1 architecture. This microprocessor implements a six stage pipeline and the six stages in the pipeline are Fetch (F), Thread selection (T), Decode (D), Execute (E), Memory access (M) and Write back (W) . Figure1 shows the six stage pipeline and the supporting units of the architecture [2]. The bit serialization is implemented in the execution unit which is explained in the later sections of this paper.

2.2 Effect in reducing the number of data bus lines
There are some factors that determine the power dissipation caused by parallel databus lines. In fact when the number of parallel databus lines is reduced, there is a reduction in the power consumption. It has been found and verified that, the main factors that contribute to power dissipation in bus lines are [1]:

1. Resistivity of the nanowires used for the architecture.
2. Energy dissipation in buses is proportional to the total coupling capacitance. The total coupling capacitance comprises of:

[1]Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Clappana P.O, Kollam, Kerala, India, PIN:690525.

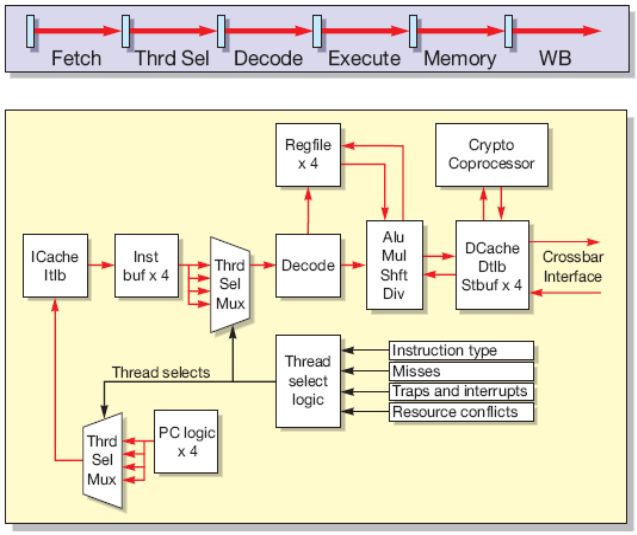

Figure 1. The various blocks involved in the six stage pipeline of the OpenSPARC T1 processor core [2].

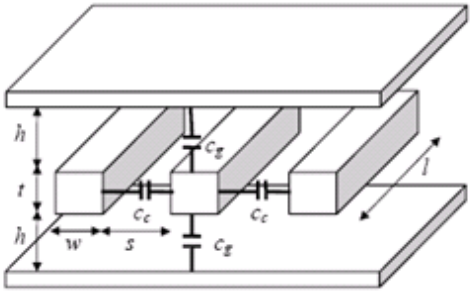

Figure 2. Parameters in a parallel line bus.

1. The line-to-ground capacitance per unit length to one of the adjacent metal layers (C_g)
2. The coupling capacitance per unit length between two neighboring lines(C_c)

MCF is the Miller coupling factor between the line of interest and its neighboring lines. Hence the total capacitance, C_t is given by

$$C_t = (2C_g + \Sigma (MCF.C_c)) \quad (1)$$

The multiplicative factor of two in equation (1) is due to the presence of two metal layers, above and below the line of interest as shown in Figure2 [1]. The *MCF* between any two neighboring lines depends on their relative switching activity: *MCF* = 2 for two oppositely switching neighboring lines; *MCF*=1 when only one line is switching while the other is quiet; and *MCF* = 0 for two similarly switching neighboring lines. Hence, assuming each interconnect is only coupled to its nearest neighbors, the average energy dissipation per unit length of an interconnect per transition is

$$E_{av} = 0.5C_{t,av}V_{DD}^2 = 0.5(2C_g + 2MCF_{av}.C_c)V_{DD}^2 \quad (2)$$

Where c_g and c_c are represented by

$$C_g = \varepsilon[(2w/h) + 4.08(t/(t - 4.531h))^{0.071}(s/(s - .5355h))^{1.773}] \quad (3)$$

$$C_e = \varepsilon[1.4116(t/s)e^{(-4s/s+8.014h)} + 2.3704(w/(w - 0.3078s))^{0.25724}(h/(h - 8.961s))^{0.7571}e^{(-2s/(s+6h))}] \quad (4)$$

Where t is the interconnect thickness and h is the inter -layer dielectric thickness (ILD) [2].

2.3 Power Dissipation

In CMOS technology there are two types of power dissipation namely,

i) Static Power Dissipation
ii) Dynamic power dissipation

Static power dissipation occurs mainly due to leakage currents or other currents drawn continuously from the power supply. Static power dissipation in pseudo NMOS transistors is due to direct path between VDD and GND. Total static power Ps is given by

$$P_s = \Sigma (I*V) \quad (5)$$

where I is the leakage current, V is the supply voltage.
Dynamic power dissipation occurs due to the fact that both nMOS and pMOS transistors are on, during the input transition from low to high or from high to low for a very short time. This results in short circuit from power supply to ground. The Dynamic power dissipation is given by the expression

$$P_d = (C*V_{dd}^2*f) \quad (6)$$

Where P_d is the dynamic power dissipation, f is the operating frequency and V_{dd} is the Voltage supply. The total power dissipation as per Xilinx ISE 10.1 is the sum total of all the below entity powers

1. Clock
2. Logic
3. Signals
4. Input-output
5. Static power
6. Dynamic power

It is the sum of all these entity power that we have furnished in the Tables as the total power. The power consumption comparison of the original OpenSPARC T1 datapath and the bit serialized datapath is done by considering total power of the designs.

2.4 Coupling Capacitances

In the proposed design of the data path, the power consumption is reduced due to the fact that the serialization reduces the coupling capacitances. The data bus lines in hardware are implemented using Carbon Nanotube (CNT) interconnects. The 64-bit ALU of OpenSPARC T1 processor has two 64 bit inputs and one 64 bit output. The number of output bus lines varies for multiplier. By the concept of serialization we are reducing the number of input data bus lines from 64 to 1 so that, the modified T1 ALU has two single bit input bus lines and a 64-bit parallel output bus. Between two parallel CNT interconnects there exist a coupling capacitance and the estimated value of coupling capacitance is as given in the Table 1. When the number of input bus lines is reduced to a single bit, we can say that the coupling capacitance is approximately zero[1].

The power dissipation for various bus line configurations is given in Table I. This shows that there is a significant decrease in the power consumption. The maximum power dissipation is calculated by the equation

$$P_D = (N-1)/2\ CV_{DD}^2F \quad (7)$$

Where N is the number of bus lines, C is the coupling capacitance between two adjacent wires, V_{DD} is the supply voltage and F is the frequency [1].

Table 1. Coupling capacitance values and the respective power dissipation for various input data bus line configurations

No. of bit lines	Clock frequency (GHz)	Coupling Capacitance, CC (fF)	Power due to CC(mW)
64	1	15100	320
32	1	12400	155
16	1	6000	75
8	1	2800	35
4	1	1200	15
2	1	400	5
1	1	0	0

Figure5 depicts the portion of the HDL code where the input data bus lines to the ALU block are serialized. A new block which is a serializer is added with the original HDL code. The serializer gives the serialized input to the ALU block. The similar process of serialization is applied for divider, multiplier, shift blocks.

2.4 Implementation

Figure3 shows the block diagram of a single core of an OpenSPARC T1 processor. The blocks within the grey area are the ones which we have modified for our purpose. The path between the register file and the ALU is serialized in the proposed implementation.

The modified design is done using Verilog HDL which includes the register file, the bypass logic, the divider, the multiplier, ALU and shift blocks. Serialization of the data bus is done in the path between the bypass logic and the four blocks (MUL, DIV, ALU and SHFT) as shown in the Figure4 OpenSPARC T1 processor has a 64 bit ALU. The execution control unit (ECL) co-ordinates the four blocks. The bypass logic implements operand forwarding, when one operation depends on the result of the previous operation, the result is directly given into the input bypassing the registers. The operand forwarding is done in D, E, W, M stages of the pipeline.

The shift block takes in 64 bit input and can produce a right/ left shift result. The divider and multiply unit is not included in the ALU. The results of the operations from all the blocks are given to the bypass logic.

The 64-bit multiplier is an integer multiplier, which has five clocks latency to compute the result. The 64-bit divider supports one divide operation per core. The multiplier is shared by the crypto co-processor which is also a part of the processor.

It may be slightly misleading that as we serialize the data bus there may be reduction in the speed. In parallel data bus the input to the ALU can be obtained in a single clock cycle. Where as in a bit serialized data bus it takes 64 clock cycles to obtain the 64-bit input. It takes one clock cycle to compute the result. The output is available in the 65th clock cycle.

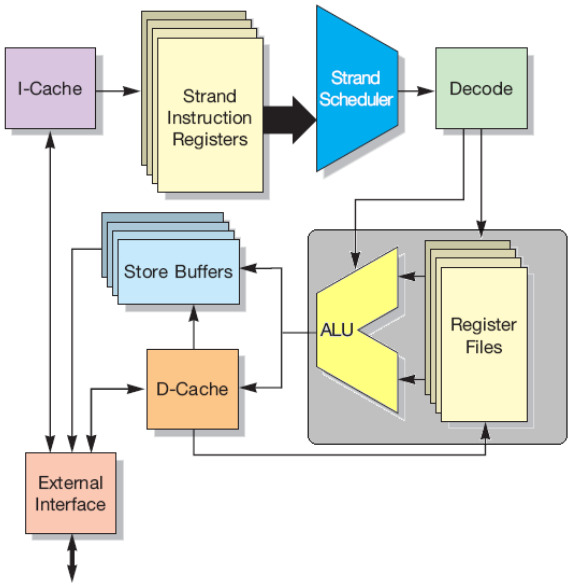

Figure 3. Single core of OpenSPARC T1 processor [2]

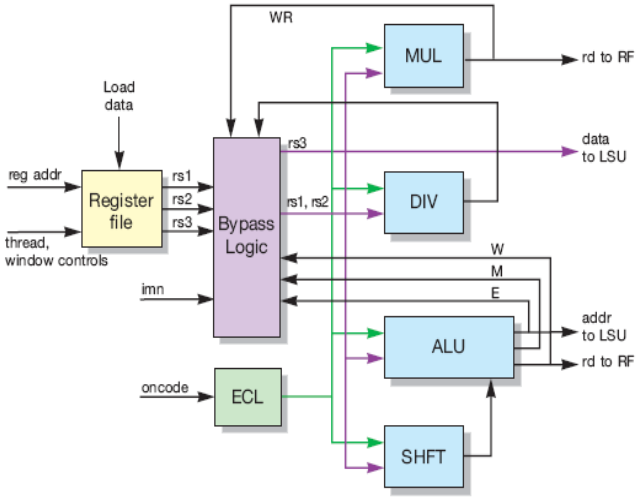

Figure 4. the various blocks in a core that make up the datapath of OpenSPARC T1[2].

But as we serialize the data bus, the coupling capacitances between the data bus lines decreases. Consequently, the operating frequency of the design increases [1]. This allows the increase of the input clock frequency and hence, the speed can be maintained to be a constant to a certain extend.

3. RESULTS & DISCUSSION

3.1 Power Analysis

The power analysis is done for the designs of the original ALU as well as for the bit serialized ALU. A similar power analysis is carried out for the MUL block. The HDL code is synthesized and implemented in Xilinx ISE 10.1. The power consumption of the device is found out using Xilinx Xpower analyzer. The power consumption of both serialized and non-serialized designs are discussed in the next section.

Serialization of data bus is done for all the four blocks, but we discuss the power consumption of only two blocks, the ALU and the MUL.

```
assign memrclk=rclk;
assign memrst=rst;
assign memse=se;
assign memsi=si;
assign membyp_alu_rs1_data_e=byp_alu_rs1_data_e[i];
assign membyp_alu_rs2_data_e_l=byp_alu_rs2_data_e_l[i];
assiign membyp_alu_rs3_data_e=byp_alu_rs3_data_e[i];
assign membyp_alu_rcc_data_e=byp_alu_rcc_data_e[i];
assign memecl_alu_cin_e=ecl_alu_cin_e;
assign memifu_exu_invert_d=ifu_exu_invert_d;
assign memecl_alu_log_sel_and_e=ecl_alu_log_sel_and_e;
assign memecl_alu_log_sel_or_e=ecl_alu_log_sel_or_e;
assign memecl_alu_log_sel_xor_e=ecl_alu_log_sel_xor_e;
assign memecl_alu_log_sel_move_e=ecl_alu_log_sel_move_e;
assign memecl_alu_out_sel_sum_e_l=ecl_alu_out_sel_sum_e_l;
assign memecl_alu_out_sel_rs3_e_l=ecl_alu_out_sel_rs3_e_l;
assign memecl_alu_out_sel_shift_e_l=ecl_alu_out_sel_shift_e_l;
assign memecl_alu_out_sel_logic_e_l=ecl_alu_out_sel_logic_e_l;
assign memshft_alu_shift_out_e=shft_alu_shift_out_e[i];
assign memecl_alu_sethi_inst_e=ecl_alu_sethi_inst_e;
assign memifu_lsu_casa_e=ifu_lsu_casa_e;

always @(posedge rclk)
 begin
   if(rst==1'b1)
       begin
           i<=64'd0;
       end
    else
        begin
            i<=i+1;
        end
end
```

→ Counter

Figure 5. The portion of HDL code for modified serialized ALU. The Counter used for serialization is marked in the figure.

We have assumed the clock frequency to be double of signal frequency [1] and it can be seen that the power dissipated increases with increase in frequency as explained in section IV.

From Table 2 and Table 3 we can compare the power consumption of a original OpenSPARC T1 ALU block and the bit serialized ALU block that we have designed. Let us take the case when the clock frequency is 60MHz. It can be inferred from the Table II and Table III that the bit serialized ALU block consumes a power of 1.7285 mW but the normal parallel data bus line ALU block consumes a much higher power which is equal to 4.411 mW. Let us consider the multiply block of the OpenSPARC T1 processor.

Figure6 shows the comparison of the power consumption of the original and the modified, bit serialized design of ALU. The increase in the power consumption in both the designs is evident from the graph. At lower frequencies the power saved is around 30%. As the frequency increases we see that the power saving increases upt0 70%.

Figure7 is the graph plotted to compare the power consumption of the original and the modified, bit serialized MUL block. Similar to the ALU block the power consumption tends to increase with

Table 2. Power Consumption for a normal OpenSPARC T1 ALU Block

Entity power (mW)	Clock frequency(C) & Signal frequency(S) (MHz)				
	C-20 S-10	C-40 S-20	C-60 S-30	C-80 S-40	C-100 S-50
Clock power	0.008	0.008	0.008	0.009	0.009
Sig. power	0.066	0.099	0.165	0.198	0.264
I/O power	0.858	1.749	2.64	3.498	4.389
Quiescent power	1.592	1.595	1.598	1.601	1.603
Dynamic power	0.932	1.856	2.813	3.705	4.662
Total power	2.524	3.451	4.411	5.306	6.265

Table 3. Power Consumption for a bit serialized OpenSPARC T1 ALU Block

Entity power (mW)	Clock frequency(C) & Signal frequency(S) (MHz)				
	C-20 S-10	C-40 S-20	C-60 S-30	C-80 S-40	C-100 S-50
Clock power	0.008	0.008	0.008	0.009	0.009
Sig. power	0.003	0.0045	0.0075	0.009	0.012
I/O power	0.039	0.0795	0.12	0.159	0.1995
Quiescent power	1.593	1.593	1.593	1.594	1.594
Dynamic power	0.05	0.092	0.1355	0.177	0.2205
Total power	1.643	1.685	1.7285	1.771	1.8145

increase in signal frequency, but the percentage of power saved is almost similar to the ALU block.

Similar to the results of the ALU block it can be seen that in MUL block also there is very high decrease in the power consumption. Here again, at the clock frequency 60MHz, it can be seen that the bit serialized implementation consumes a very less power of 1.8017 mW compared to 4.578 mW in the case of non-serialized

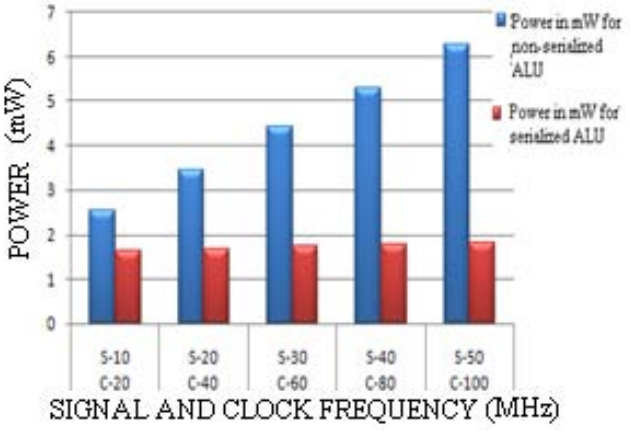

Figure 6. The graph showing the difference in the power consumption for a normal parallel bit input data bus line and a bit serialized input data bus line for the ALU block

Table 4. Power Consumption for a normal OpenSPARC T1 MUL Block

Entity power (mW)	Clock frequency(C) & Signal frequency(S) (MHz)				
	C-20 S-10	C-40 S-20	C-60 S-30	C-80 S-40	C-100 S-50
Clock power	0.072	0.076	0.080	0.084	0.088
Signal power	0.0015	0.0027	0.0037	0.0045	0.0053
I/O power	0.045	0.064	0.115	0.163	0.186
Quiescent power	1.601	1.601	1.603	1.603	1.604
Dynamic power	0.1185	0.1427	0.1987	0.2515	0.2793
Total power	1.7195	1.7437	1.8017	1.8545	1.8833

MUL block. These results given in Table 4 and 5 are without taking the coupling capacitances into consideration.

When the coupling capacitance values in Table I is also taken into consideration, the decrease in power consumption that can be achieved by serializing the datapath of the processor would be very high.

From the comparison shown in Table VI and Table VII, it can be inferred that power is significantly saved at least by 35%, (clock frequency is 20MHz and signal frequency is 10MHz) and at most by 70% (the clock frequency is 100MHz and signal frequency is 50MHz).

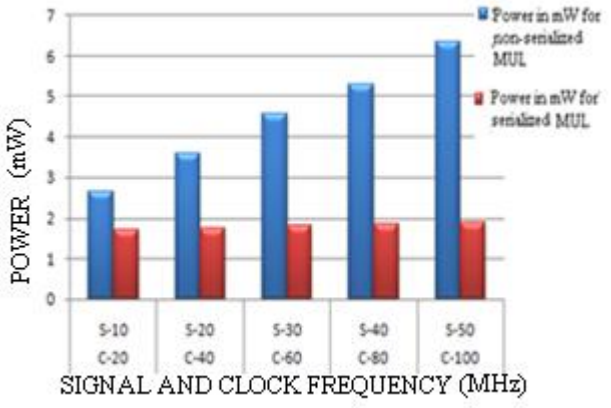

Figure.7 The graph showing the difference in the power consumption for a normal parallel bit input data bus line and a bit serialized input data bus line for the MUL block

Table 5. Power Consumption for a bit serialized OpenSPARC T1 MUL Block

Entity power (mW)	Clock frequency(C) & Signal frequency(S) (MHz)				
	C-20 S-10	C-40 S-20	C-60 S-30	C-80 S-40	C-100 S-50
Clock power	0.072	0.076	0.080	0.084	0.088
Signal power	0.033	0.060	0.081	0.098	0.116
I/O power	0.951	1.841	2.82	3.526	4.522
Quiescent power	1.596	1.596	1.597	1.597	1.600
Dynamic power	1.056	1.977	2.981	3.708	4.726
Total power	2.652	3.573	4.578	5.305	6.326

Table 6. Percentage Reduction in Power Consumption for ALU Block

Frequency (MHz)		Total power (non serialized)	Total powe (serialized)	%Saving
Clock	Signal			
20	10	2.524	1.643	34.90%
40	20	3.451	1.685	51.17%
60	30	4.411	1.7285	60.81%
80	40	5.306	1.771	66.22%
100	50	6.265	1.8145	71.03%

Table 7. Percentage Reduction in Power Consumption for MUL Block

Frequency (MHz)		Total power (non serialized)	Total powe (serialized)	%Saving
Clock	Signal			
20	10	2.652	1.7195	35.16%
40	20	3.573	1.7437	51.19%
60	30	4.578	1.8017	60.64%
80	40	5.305	1.8545	65.04%
100	50	6.326	1.8835	70.22%

Figure8 shows the power reduction percentage when serialization is applied in the ALU block and the multiplier block.

4. CONCLUSION

By estimating the power consumed by the original and the modified, bit serialized ALU and MUL blocks, we have arrived at a conclusion that the modified bit serialized ALU and MUL blocks shows a significant reduction in power. Serialization of bus lines brings about sufficient power conservation in CPU datapath. Our proposals have been implemented in Verilog and verified using Xilinx ISE 10.1 X-power analyzer.

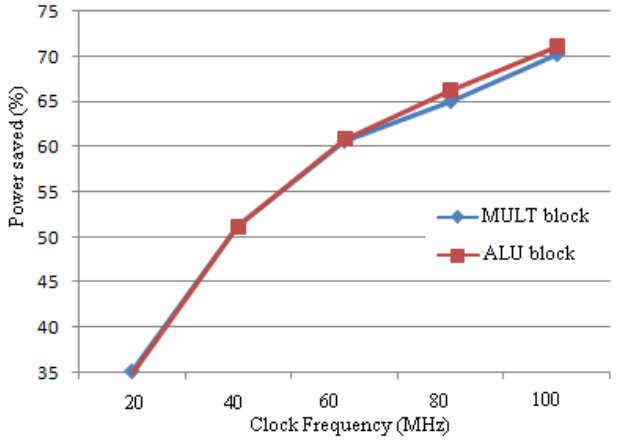

Figure 8. Percentage saving in power consumption brought about by serialization

5. ACKNOWLEDGMENTS

We gratefully acknowledge the Almighty GOD who gave us strength and health to successfully complete this venture. The authors wish to thank Amrita Vishwa Vidyapeetham, in particular the Digital library, for access to their research facilities.

6. REFERENCES

[1] Rajesh Kannan Megalingam, Venkat Krishnan. B, Mithun.M, Rahul Srikumar, Vineeth Sarma.V,"G ating and Serializing the Data Path of CPU for Low Power Consumption ", Proceeding of 2009 International Conference on Parallel Processing (ICPP '09)

[2] OpenSPARC T1 Microarchitecture Specification, Sun Microsystems , 2006.

[3] N. Banerjee, A. Raychowdhury, S. Bhunia, H.Mahmoodi, and K. Roy "Novel Low-Overhead Operand Isolation Techniques for Low-Power Datapath Synthesis", Proceedings of the 2005 International Conference on Computer Design (ICCD'05)

[4] Maged Ghoneima, Yehea Ismail, Muhammad Khellah, James Tschanz, and Vivek De "Serial-Link Bus: A Low-Power On-Chip Bus Architecture, 0-7803-9254-X/05

[5] Beom Seon Ryu, Jung Sok Yi,Kie Young Lee And Tae Won Cho,"A Design Of Low Power 16-B ALU "1999 IEEE TENCON, 0-7803-5739- 6/9

[6] Yu Zhou and Hui Guo "Application Specific Low Power ALU Design", 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 978-0-7695-3492-3/08

[7] C. Thimmannagari. *CPU Design: Answers to Frequently Asked Questions*. Springer, 2005.

[8] V. Tiwari, S. Malik, and P. Ashar "Guarded evaluation: Pushing power management to logic synthesis/design". In *Proceedings of the 9th International Symposium on Low Power Design*, pages 221–226, 1995.

[9] Computer system architecture, Third edition , M.Moris Mano.

[10] John.L.Hennessy, David.A.Patterson, "Computer Architecture – A quantitative approach" 4th edition.

[11] Dmitry Ponomarev, Gurhan Kucuk and Kanad Ghose "AccuPower: An Accurate Power Estimation Tool for Superscalar Microprocessors", Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition

[12] Enoch Hwang Frank Vahid Yu-Chin Hsu " FSMD Functional Partitioning for Low Power"

[13] Reducing Power Requirements of Instruction Scheduling Through Dynamic Allocation of Multiple Datapath Resources. Dmitry Ponomarev, Gurhan Kucuk, Kanad Ghose, 1072-4451/01

[14] Dmitry Ponomarev,Gurhan Kucuk,Kanad Ghose "Dynamic Resizing of Superscalar Datapath Components for Energy Efficiency", IEEE TRANSACTIONS ON COMPUTERS, VOL. 55, NO. 2, FEBUARY 2006

[15] Rajesh Kannan Megalingam, Shekhil Hassan, Vivek P, Ashwin Mohan, Tanmay Rao M, " Power Consumption Reduction in CPU Datapath using Novel Clocking Scheme".In proceedings of ICCSIT '2009

[16] Norman Yeung, Barbara Zivkov, Gulbin Ezer Unified Datapath: An Innovative Approach to the Design of a Low-Cost,Low-Power, High Performance Microprocessor , 1063-6390/94