Implementation of Resource Sharing Strategy for Power Optimization in Embedded Processors

Kiritkumar Bhatt [1*] Prof. A I Trivedi[2]
1. Electronics & Communication Engineering Department, Sardar Vallabhbhai Patel Institute of Technology – Vasad -388306, Gujarat – India.
2. Electrical Engineering Department, Faculty of Technology & Engineering, M S University of Baroda – 390001, Gujarat, India.
*Email of corresponding author: Krbhattec@gmail.com

Abstract
The processors are better suited to the diverse applications in daily life ranging from small toys to complex automated systems. For better mobility and reliability; the longer battery life is an essential need. Improvement in power efficiency of the processor is achieved by implementing resource sharing logic at the hardware level. This paper discusses the modified architecture of a 32 - bit RISC processor having four - pipeline stages. Here power improvement is achieved by implementing the proposed technique called resource sharing at the hardware level and results are verified satisfactorily. The proposed work is simulated, synthesized, tested and verified by using tools such as VHDL simulator, Xilinx Sparten – 3E FPGA, ModelSim SE-6.5 and Xilinx ISE – 13.1 tool and XPower Analyzer for power estimation and analysis purpose.

Keywords: 32 – bit Low Power Processor , Power Efficient Embedded Processor, Resource Sharing, Low-Power Architecture, low – power design, low – power system, power minimization, power optimization, system design.

1. Introduction

The continuous increase in power consumption with each successive generation of processors has started to affect the system size and cost so badly that this power/performance trade-off has become increasingly difficult to justify in a competitive market. The processor complexity leads to demand more computation which in turn further increases total power consumption in the system and the power consumption impacts the system design. The trend in the desktop world of continuous growth in complexity and size of the underlying CPU in terms of instruction issue strategies and the supporting micro-architecture needs to be re-examined for these devices, as the tradeoffs in energy consumption versus the improved performance obtained may achieved by a different set of design choices. Power consumption arises as a third axis in the optimization space in addition to the traditional speed (performance) and area (cost) Dimensions.

Improvements in circuit density and the corresponding increase in heat generation must be addressed even for high-end desktop systems. Current trends in technology scaling of CMOS circuits cannot be reliably sustained without addressing power consumption issues. Environmental concerns relating to energy consumption by computers and other electrical equipment are another reason for interest in low-power designs and design techniques. Hence, designing a low-power processor is very important in today’s mobile devices. The greatest advantage of it is the extension of device battery life and another one is the reduction in switching current which leads to increase the reliability of the device.

This processor design is based on 32-bit RISC principle and it emphasis on load/store mechanism, which is used to eliminate the latency of memory operations. The processor has many registers, and operations are performed on data present in the registers. This paper focus on design and implementation of the processor based on pipelining and the modifications has been introduced in the data path at the hardware level to reduce the dynamic power dissipation which can be expressed by (K R Bhatt et al 2011,,Jevtic 2008).

$$P_{switching} = \frac{1}{2} C\, V_{DD}{}^{2}\, E_{(SW)} f_{clk} \qquad (1)$$

Where C is output node capacitance, VDD is supply voltage and E(sw) is the average number of transistors in the circuit per cycle and fclk is the clock frequency. In this design power reduction is achieved by sharing the common available resources frequently instead of deploying a dedicated resource in the hardware, this strategy reduces the unnecessary switching activity (Esw)(Collange 2009, Steve Furber 2007). Further sections discuss the implementation of this technique for which one has to redefine certain instructions and the relevant logic is to be implemented at the hardware level.

2. Low – Power CPU Design

The architecture of processor is given in Fig.1, which consists of four stage pipeline. These stages are instruction fetch, decode, and execute and write back. It also incorporates major units such as Data path, Control logic, Memory and register blocks, the Data Forward Unit and the Hazard Detection Unit to maintain the proper flow of data through the pipeline stages *(Kiritkumar Bhatt et al 2012, D Patterson 2005)*. The different stages of pipeline architecture are separated by the pipeline registers.

2.1 Data path:

The pipeline stages for different type of instructions mentioned in Table 1 are shown in the Fig. 2. In the fetch stage, instructions are fetched at every cycle from the instruction memory whose address is pointed by the program counter (PC). During the decode stage, the registers are read from the register file and the op-code is passed to the control unit which asserts the required control signals. Sign extension is taken care for the calculation of effective address. In the execute stage, for Register type instructions, the ALU operations are performed according to the ALU operation control signals and for load and store instructions, effective address calculation is done. The load and store instructions write to and read from the data memory using the data forwarding logic while the ALU results and the data read from the data memory are written in to the register file by the register type and load instructions respectively in the write-back stage (Pande et.al.2008,Gautam P 2009, L. Benini 2000).

The data-path contains pipeline registers in-between each of the stages. These registers are used to carry-over results from the previous to the following stages.

2.2 Control Logic:

The pipeline is controlled by setting control values during each pipeline stage. Each control signal is active only during a single pipeline stage and hence the control lines can be divided according to the four pipelined stages. These signals will be forwarded to the adjacent stage through the pipeline registers (D. Folegnani 2001).

2.3 Data Forward Unit:

It maintains correct data flow to the ALU also compares the destination register address of the data waiting in buffer and writes back pipeline registers to be written back to the register file with the current data required by ALU and forward the latest data to it(B.Mayer 2001,M.Krste et.al. 2002). By forwarding the data at the appropriate time, this unit assures pipeline does not stall as a result of the data dependencies, which arise when an instruction needs to use the result of one of the processes before that result has returned to the register file. The forwarding paths allow results to be passed between stages as soon as they are available. And the 4 – stage pipeline requires each source operand to be forwarded from any of intermediate result registers.

2.3 Memory and Register Blocks:

The data and code memories are 32 X 256 bytes and 32X1024 bytes respectively while the register block is 32X32 bytes. The code memory is implemented as on-chip distributed ROM while the data memory as block RAM inside the FPGA.

2.4 Hazard Detection Unit:

This module detects the conditions under which data forwarding is not possible and the pipeline stalls for one or two clock cycles in order to make sure that instructions are executed with the correct data set. When it detect stall, it disables any write operation in the instruction decode pipeline, stops PC from incrementing, and clears all the control signals generated by the control unit. By taking these steps it can delay the execution of any instruction by one clock cycle. It can do this as many times as necessary to ensure proper execution of instruction.

2.5 Instructions Supported by Architecture:

This processor architecture supports mainly three types of instructions: Register Type (R-Type), Immediate Type (I-Type) and Jump Type (J- Type). Table1 lists the total instructions implemented in the processor[7,12,17].

3. Proposed Resource Sharing Strategy for Power Optimization

This proposed technique is implemented during the designing of the decode stage of the processor. All most all the processors' design supports two types of the instructions' addressing modes out of many others which are immediate addressing and direct register addressing types. For example the addition operation can be performed by both the types as mentioned in Table 2.

Here, both the instructions performs almost the same operation i.e. addition and the operand 1 is common i.e. register r1. The operand 2 can be either immediate data value or a register depending on the type of instruction. To introduce resource sharing technique both the opcode is required to decode to same operation that is for addition Addi the opcode is 001110 and for Add the opcode is 000001 is decoded to perform the addition operation and assigned opecode is 000101*(Z.Zhu 2005,I.Ahmad 2011)*.

Instead of using the separate resources for both the instructions which perform the same operation, this technique channelize the instruction into one and hence the saving of half the resources is achieved from the decode stage which would have been required if both the instructions were executed differently using their separate resources. Thus in execute stage only one adder does the work for both types of instructions. Thus much amount of power was saved by resource sharing just by adding some control signals in the decode stage and eliminating many flip – flops, comparators and muxes which would have been required in the later stage. Through Resource sharing following instructions are channelized to one instruction and the considerable power saving was achieved. This technique can be applied to all the stages with different logic as applied to decode stage here. Thus from fetching, executing and write back stages considerable amount of resources are saved *(J Ayala2002, 2003, X Guan 2008, D. Brooks200)*. During the instruction fetch operation the Program Counter (PC) is either incremented by 1 or incremented/decremented by the content of branch address from the current position of PC. To perform this operation two adders and one subtractor is required. During the design of processor twos' complement logic was used for instruction fetching during the branching operation, which removes one subtractor.

Further, the resource sharing is implemented by using 2:1 MUX with inputs of value 1 and 2's complement branch address and a branch signal as select line. This MUX output is then added to current PC to generate the next PC value. This reduces further one adder as shown in following figure 3.

Where X1 is X "00000001", X2 is PC (31 downto 0) and Y1 is PC_incr signals. The Program Counter operation can be expressed as follows:

PC = PC + PC_incr;

In Execute Stage the ALU required 2 Adders for the operations such as addition and increment i.e. one for addition (Add, Addi) and one for Increment (Inc) and 2 subtractors for subtraction and decrement i.e. one for subtraction(Sub, Subi) and one for Decrement (Dec).

In this proposed work during execute stage instead of using 4 adders/subtractors dedicated to each operation one adder and a multiplexer have been implemented during the design of the CPU and the resource utilization have been optimized by which the power reduction has been achieved. The design and the functionality of the newer logic which is implemented in the CPU can be explained through the following example and implemented as shown in figure 4. Let us consider four operations are to be taken care*(D.Brookes 2000, A.Merkel 2006,K Scoones 2007)*:

(a) Add: a + b; (b) Inc: a + 1;

(c) Sub: a – b; (d) Dec: a – 1;

To achieve the resource sharing b register is given the input value through the 4:1 MUX with the opcode as select line, as shown in following figure 4. And after the implementation of this logic the single adder is used to generate the results and the logic for all the cases to perform C = a + b is as shown below.

4. Power Analysis of The Proposed Strategy

This section describes the power summary generated using the Xpower Analyzer. Two times the power analysis has been carried upon the system under consideration, once before the implementation of the resource sharing logic at the design level and the next time after its implementation. The Table 4 describes the power requirement for different clock frequencies upon which the power requirement is analysed *(K R Bhatt 2011, R Jevtic 2008, J H Anderson 2004)*.

It is noted from the table that the power consumption is increased with the increase in the clock frequency for both the architectures but the modified CPU always consumes less power compared to the conventional CPU structure for all the clock frequencies. Graphical view of this comparison can be represented as shown in figure 5.

First time the power analysis was carried on the conventional CPU and the power summary sheet is as shown in the figure 6 and the second time on the CPU with the modified logic structure in order to achieve the power improvement.

The power summary sheet generated by the system after the implementation of resource sharing strategy is as shown in figure 7.

The figure 8 shows the graphical comparison of power requirements of the various parts of the systems for both the standard CPU and for the CPU with modified logic structure.

It is noted that the power improvement is achieved by 3mW (2.65%) in the system after the implementation of the Resource Sharing logic in the conventional CPU at the hardware design level.

Thus the CPU with modified architecture performs the functionality with lesser power. The power improvement is achieved here is only because of the newly designed strategy, but further improvement can be better achieved by implementing many other strategies at the various abstraction levels of the system design.

The system under experiment has been implemented on Xilinx – SPARTAN – 3E FPGA and synthesized for the analysis purpose. The table 5 describes the summary of synthesis report.

5. Conclusion

A newly designed processor with 32 – bit 4 – stage pipeline based on RISC principal has been successfully implemented. This hardware system is simulated, synthesized and verified by using VHDL coding, Xilinx ISE 13.1, Modelsim 6.5and Xilinx SPARTAN – 3E FPGA as a target device. This paper suggested an innovative strategy for resource sharing for power optimization; by which 2.65% improvement in power consumption is achieved.

The power analysis of the design is done before and after the implementation of resource sharing strategy by using Xpower Analyzer and 3 mW corrections is offered by the design.

The resource sharing logic is also tried for different clock frequencies and concluded that the power consumption is increased with the increase in the operating frequencies but the modified design consumes less power for all the frequencies.

This design has been further optimized up to 14% of power saving by implementing various other techniques such as clock gating *(Kiritkumar Bhatt et al2012)*, coding techniques means one-hot coding, Gray coding, bus inversion, 2's complement versus sign magnitude which are not within the scope of this paper. Also at the memory level the memory sub-banking as the storage element and partitioning between sequential and combinational circuits can be taken care *(D. Brookes 2000,kScoones 2007,Gautam P 2009)*. The arithmetic instructions have an NOP stage in memory stage while there is no NOP during write back stage of the arithmetic instructions, so write back stage of the arithmetic instructions can be moved to a memory stage without causing any resource conflicts to reduce the transitions.

Here, only hardware and the logic level techniques have been considered and the techniques related to manufacturing process have not opted.

Thus, power consumption is a major factor that limits the performance of computers. Many techniques are available to reduce the total power consumption by a microprocessor system; these techniques are applied at various abstraction levels. Authors believe that the power management needs multidimensional inputs which are continually expanding with new techniques being developed at every abstraction levels.

Yet, it is too early to say that which technique will solve the problem of power consumption. But it is for sure that this work can be taken as base and one can keep developing and adding more and more techniques to the processor system under experiment at different levels and can achieve a power optimized processor.

References

K R Bhatt, A I Trivedi (2008), " Power Computation Model of CMOS based FPGA used for Power Optimization", Int. Journal of PDCS, Vol.3, No. 13, pp. 759 – 762.

R. Jevtic, C. Carreras, and G. Caffarena(2008), "Fast and accurate power estimation of FPGA DSP components based on high-level switching activity models," Int. J. Electron., vol. 95, no. 7, pp. 653–668.

J. H. Anderson and F. N. Najm(2004), "Power estimation techniques for FPGAs," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 12, pp. 1015–1027.

Intel(2011), "High-Performance Energy-Efficient Processors for Embedded Market Segments," http://www.intel.com/design/ embedded/downloads/ 315336.pdf.

S. Collange, D. Defour, and A. Tisserand(2009), "Power Consumption of GPUs from a Software Perspective," Proc. ACM Ninth Int'l Conf. Computational Science (ICCS).

Steve Furber(2007), " ARM – System on – Chip Architecture", IInd Edition, Pearson Education.

Kiritkumar Bhatt, Saurabh Patel, A I Trivedi(2012), " Power Optimized Embedded Processor Design with Parallel Pipelining", Int. Journal of PDCS, Vol. 4, No.1, pp 54 – 60..

D A Patterson, J L Hennessy(2005)," Computer Organization and Design, the Hardware/Software Interface", Morgan Kaufmann.

P. Pande, A. Ganguly, B. Belzer, A. Nojeh, and A. Ivanov(2008), "Novel Interconnect Infrastructures for Massive Multicore Chips—An Overview," Proc. IEEE Int'l Symp. Circuits and Systems (ISCAS).

D. Folegnani and A. Gonza´lez(2001), "Energy-Effective Issue Logic," Proc. 28th Int'l Symp. Computer Architecture, pp. 230-239.

Kiritkumar Bhatt, A I Trivedi(2011), " Power Estimation of Switching Activity for Low – Power Implementation on FPGA", Int. Journal of Programmable Devices Circuits and Systems, Vol. 3, No. 14, pp. 803 – 807.

Gautam P, Parthsarathy R, Karthi Balasubramanian(2009), " Low – Power Pipeliens MIPS Processor Design",

Proc. Of Int. Conf. On ISIC pp.462 – 465.
L. Benini, A. Bogliolo and G. Micheli(2000),"A Survey of Design Techniques for System – Level Dynamic Power Management", IEEE Trans. On VLSI Systems, pp. 299 – 316.
B. Mayer(2011), "Low – Power Design for Embedded Processors", Proc. Of IEEE Vol. 98, No.11.
F.J. Cazorla, A. Ramý´rez, M. Valero, and E. Ferna´ndez(2004), "Dynamically Controlled Resource Allocation in SMT Processors," Proc. 37th Int'l Symp. Microarchitecture, pp. 171-182.
Krste Asonavoic, Mark Hampton, Ronny Krashinsky and Emmett Witchel (2002), " Power Aware Computing", Kluwer Academic Publishers.
M. Pedram(2001), " Power Optimization and Management in Embedded Systems" , Proc. Of Int. Conf. On Asia South Pacific Design Automation Conference, pp. 239 – 244.
S. Lee and J. Gaudiot(2006), "Throttling-Based Resource Management in High Performance Multithreaded Architectures," IEEE Trans. Computers, vol. 55, no. 9, pp. 1142-1152.
Z. Zhu and X. Zhang(2005), "Look-Ahead Architecture Adaptation to Reduce Processor Power Consumption," IEEE Micro, vol. 25, no. 4, pp. 10-19.
I. Ahmad and S. Ranka(2011), Handbook of Energy-Aware And Green Computing. Taylor and Francis Group, CRC Press.
J L Ayala, A. Veidenbaum and M L Valejo(2003), " Power _ Aware Compilation for Register File Energy Reduction", Int. Journal of Parallel Program, Vol.31, No. 6, pp. 451 – 467.
J L Ayala, A. Veidenbaum(2002), " Reducing Register File Energy Consumption using Compiler Support", presented at workshop on Application – Specific Processors, Istabbul, Turkey.
X.Guan, Y.Fei(2008), " Reducing Power Consumption through Register File Partitioning and Compiler Support", Proc. Application Specific Systems, Architectures, Processors,pp.269 – 274.
D. Brooks, et al(2000)., "Power-Aware Micro-architecture: Design and Modeling Challenges for Next-Generation Microprocessors", IEEE Micro, vol. 20, No. 6, pp. 26– 44.
D.Brookes,V.Tiwari, Matronosi (2000), " Wattch: A framework for Architecture – Level Power Analysis and Optimization", proc. 27th Ann. Int. Symp. Computer Architecture, pp. 83 – 94.
A. Merkel, F.Bellosa(2006), "Balancing Power Consumption in Multiprocessor Systems", proc. ACM Eurosys Conf.
K.Scoones(2007), "Power Management Technology for Portable Devices", Austin Conf. Energy – Efficient Design.

Table 1 Instructions Supported by the CPU

Instruction Types	Instructions Implemented
Register	Add, Sub, Mul, Or, And, Xor, Mov, Inc, Dec, Cmp
Immediate	Addi, Subi, Muli, Ori, Andi, Xori, Movi
Branch	Bz, Bnz, Br,

Table 2 Addition Operation

Operation	Instruction format
Immediate Addressing Type Operation:	Addi r1 , 30 bit (Immediate Data)
Direct Addressing Type Operation:	Add r1 , r2

Table 3 List of Instruction on which Resource Sharing Applied

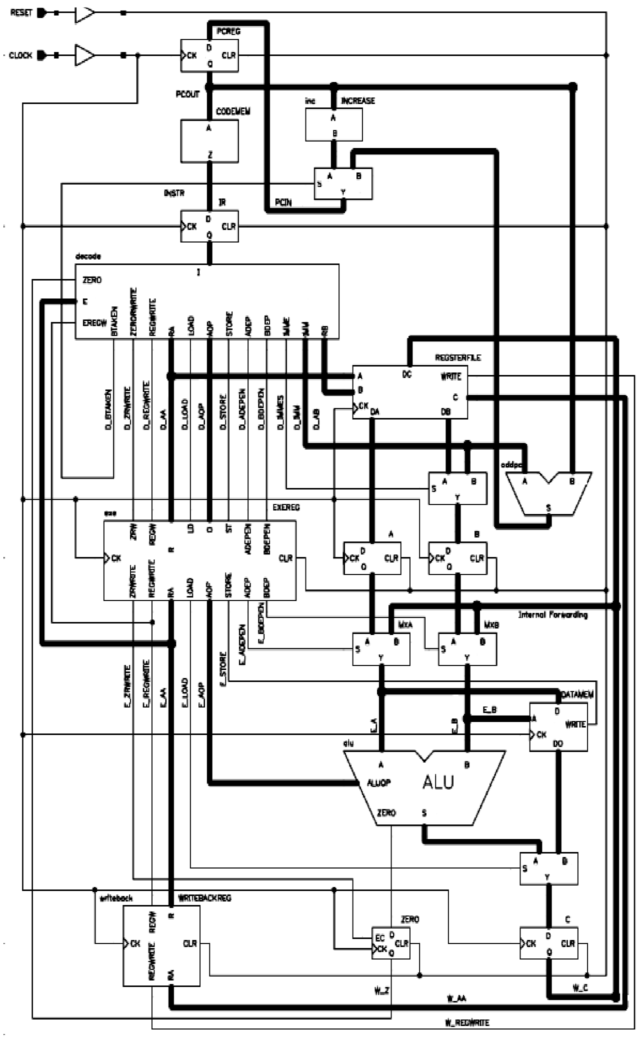
Instruction 1	Instruction 2	Operation	Decoded to Opcode
Add (000001)	Addi (001110)	Addition	Addition (000101)
Sub(000010)	Subi(001111)	Subtraction	Subtraction (000100)
Mul(000011)	Muli(010000	Multiplication	Multiplication (000110)
Or(000100)	Ori(010001)	OR	Orring (000001)
And(000101)	Andi(010010)	AND	Anding (000011)
Xor(000110)	Xori(010011)	XOR	Xorring (000010)
Mov(000111)	Movi(010100)	Move	Move(000000)
Ror(001000)	Rori(010101)	Rotate Right	Rotate Right(001000)
Rol(001001)	Roli(010110)	Rotate Left	Rotate Left(001100)
Srl(001010)	Srli(010111)	Shift Right	Shift Right(001101)
Sll(001011)	Slli(011000)	Shift Left	Shift Left(001110)
Load (001100)	Loadi(011001)	Load	Same Load Signal for Both
Store (001101)	Stori(011010)	Store	Same Store signal for Both

Table 4 Power Requirement at Different Clock Frequencies

Clock (MHz)	Power Consumption in Conventional CPU design (mW)	Power Consumption in Modified CPU design (4 – stage) (mW)
10	97	95
20	105	102
30	111	109
40	113	110
50	124	121
60	126	124
70	130	127
80	134	130
90	138	133
100	144	137

Device Utilization Summary			
Selected Device : XC3s500efg320-4(SPARTAN – 3E FPGA)			
Devices	Utilization		
	Devices Used	Out of	% Utilization
Number of Slices	1703	4656	36%
Number of Slice Flip Flops	656	9312	7%
Number of 4 input LUTs	3220	9312	34%
Number of bonded IOBs	2	232	0%
Number of BRAMs	9	20	45%
Number of MULT18X18SIOs	3	20	15%
Number of GCLKs	1	24	4%
Number of IOs	2		
Timing Summary			
Speed Grade	4		
Minimum period	18.768ns		
Maximum Frequency	53.283MHz		

Table 5 Summary of Synthesis Report



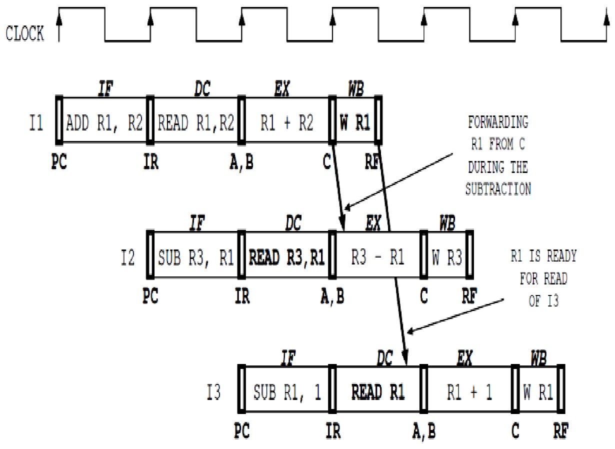


Fig. 2 Pipeline Stages for Various Instructions

Fig.1 Low - Power Architecture

of CPU

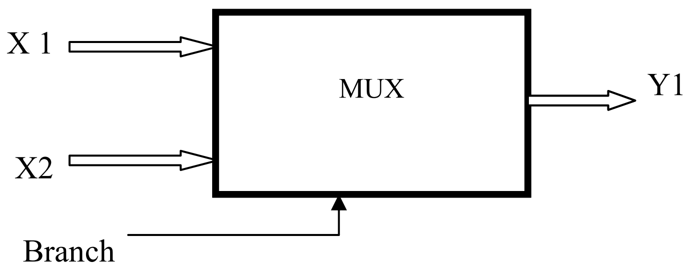


Fig.3 MUX Usage for Reducing Resources

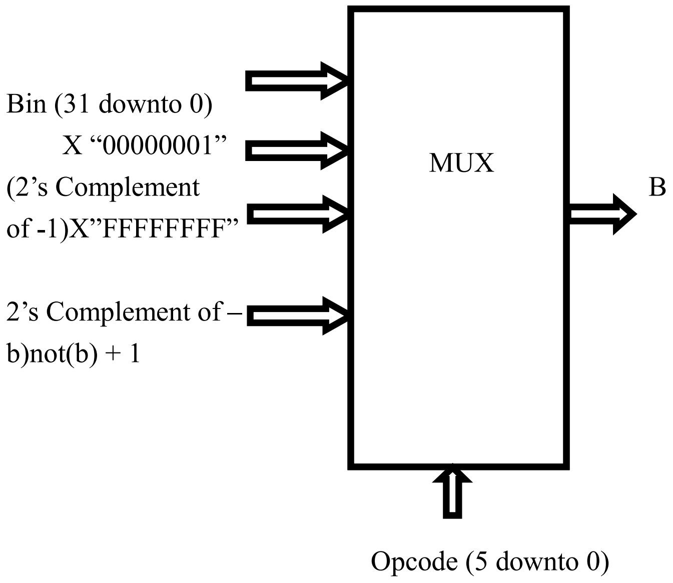

Bin (31 downto 0)
X "00000001"
(2's Complement of -1)X"FFFFFFFF"
2's Complement of – b)not(b) + 1
MUX
B
Opcode (5 downto 0)

Fig. 4 Resource Sharing Logic for above examples

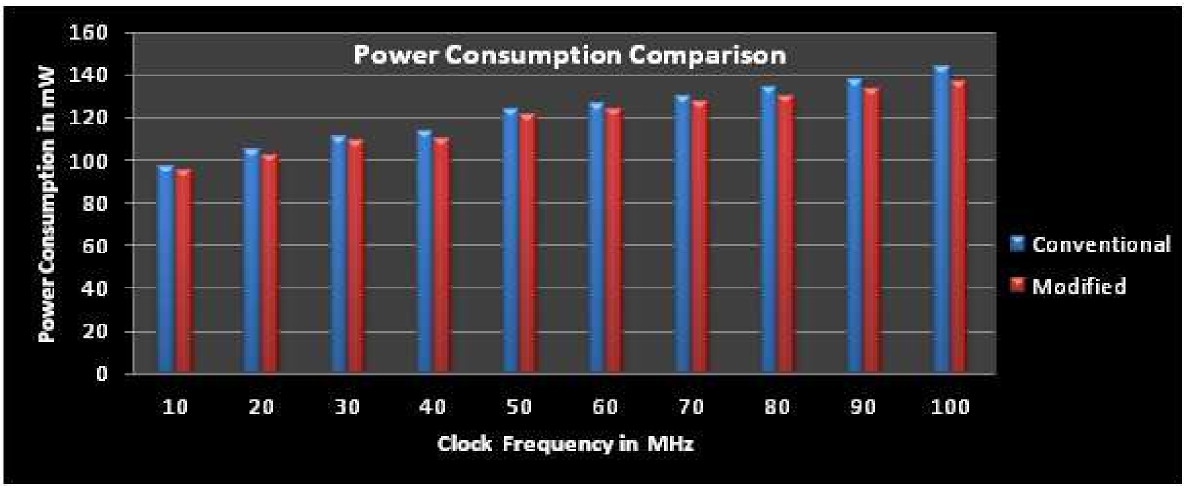
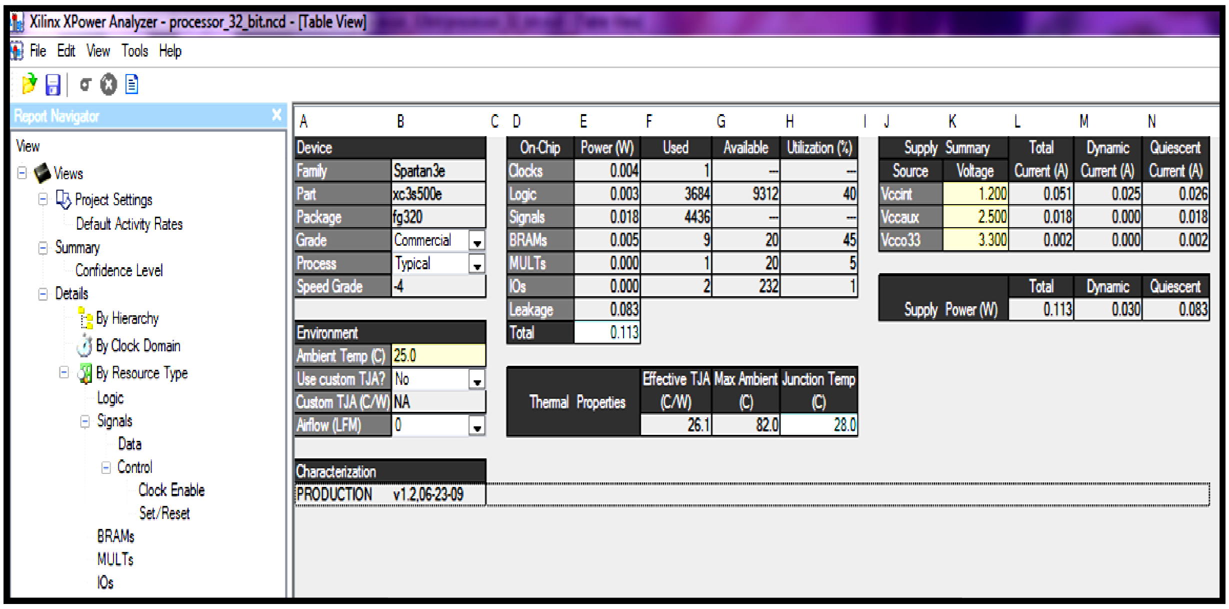


Fig. 5 Comparison of Power Consumption



Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

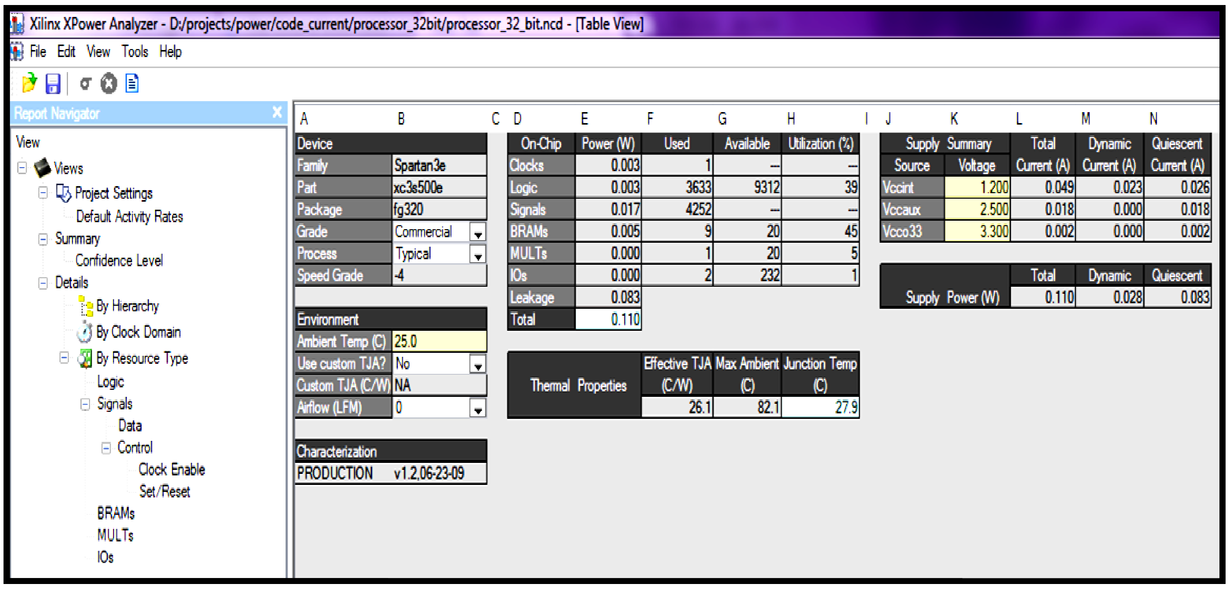
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.004	1	--	--
Logic	0.003	3684	9312	40
Signals	0.018	4436	--	--
BRAMs	0.005	9	20	45
MULTs	0.000	1	20	5
IOs	0.000	2	232	1
Leakage	0.083			
Total	0.113			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.0	28.0

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.051	0.025	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco33	3.300	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.113	0.030	0.083

Fig. 6 Power Summary sheet before Resource Sharing



Device	
Family	Spartan3e
Part	xc3s500e
Package	fg320
Grade	Commercial
Process	Typical
Speed Grade	-4

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Characterization	
PRODUCTION	v1.2,06-23-09

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.003	1	--	--
Logic	0.003	3633	9312	39
Signals	0.017	4252	--	--
BRAMs	0.005	9	20	45
MULTs	0.000	1	20	5
IOs	0.000	2	232	1
Leakage	0.083			
Total	0.110			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	26.1	82.1	27.9

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.049	0.023	0.026
Vccaux	2.500	0.018	0.000	0.018
Vcco33	3.300	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.110	0.028	0.083

Fig. 7 Power Summary sheet after Resource Sharing

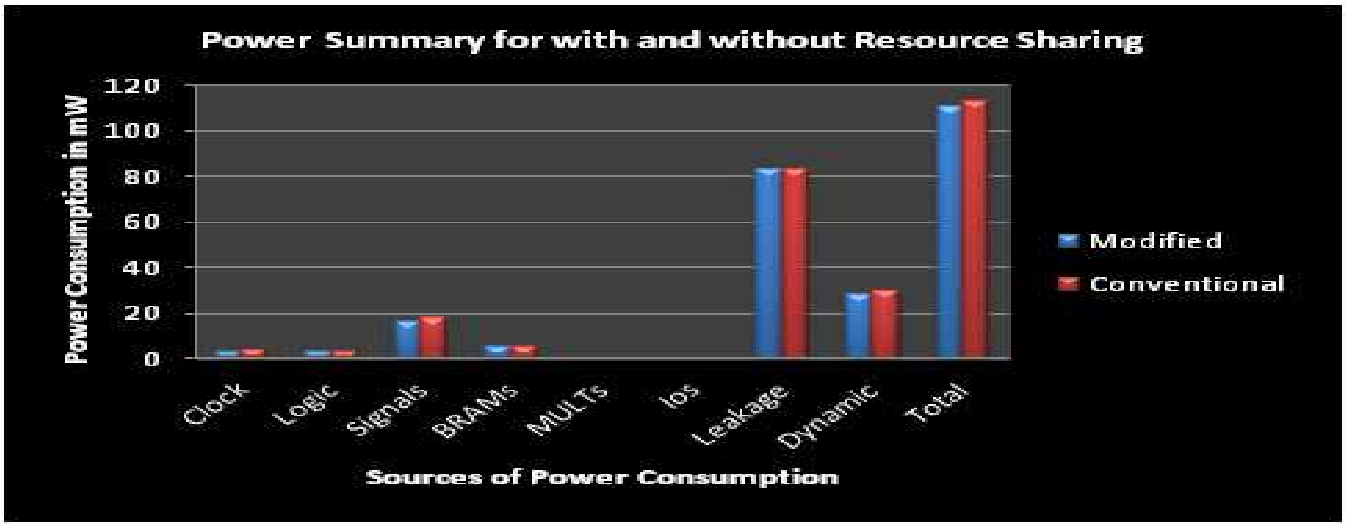


Fig. 8 Comparison of Power Requirement for both the structures

Kiritkumar Bhatt (Vadodara – Oct'1972) presently pursuing his research and has received Bachelor's (1997) and Master's (2002) Degree with specialization in Microprocessor Systems and Applications from the Elect. Engg. Dept., Faculty of Tech. & Engg, M S University of Baroda.

At present he is working as an Associate Professor in the E & C Engg. Dept.,at Sardar Vallabhbhai Patel Institute of Technology – Vasad, Gujarat – India. He has 14 years of experience as an academician. He authored number of papers in national /international journals/conferences of repute, also worked as designated reviewer for many IEEE conferences. He has guided and worked on many UG and PG projects. He is member of various professional bodies such as IETE, SPE, ISTE, IE(I), – India.

His research interest includes power estimation, optimization and system-level dynamic power management, low – power processor and embedded system design and implementation.

Prof. A.I. Trivedi is M.Phil, DIC, B.Tech (Hons) . He is professor in Electrical Engineering department in Factulty of Technology, M.S.University, Baroda. His research areas are Communication Systems, Image processing, DSP etc. He has guided four Ph.D. students and six are doing Ph.D. under his guidance. He has published many research papers and articles in referred journals and international conference proceedings.