Power and Delay Optimization of Customized 16-Bit Low Power RISC Processor Using VHDL

Surendra Bajia, Research Scholar at JNU Jaipur, Email- sbajia02@gmail.com

Abstract: This paper presents the behavioral design and the function characteristics of a general purpose RISC processor. The RISC processor design is based on the study and evaluation of a variety of assembly instruction sets. The designed RISC processor is a pipeline RISC processor with 5 stages of pipelining. A variety of instructions have been implemented for the proposed design and care has been taken to provide a control mechanism for structural, data and timing hazards. The processor's architecture features 8 internal general-purpose registers, 16 bit instruction words, each of which can hold a 16 external address lines and 16 bit data word, to external memory. The entire processor was modeled as a bottom up approach in the design methodology. The design has been done in VHDL and synthesized using hardware tool Xilinx 13.1. The features of this processor include 16-bit architecture and multi-cycle implementation of the data path. The design has been done at the behavioral levels of VHDL. A number of MIPS instruction set have been implemented. The individual components were analyzed, designed, synthesized and tested at each level of implementation. Finally the individual components integrated in a top-level simulation by appropriate port mapping.

Key Words: RISC, MIPS, VHDL, CISC.

I. Introduction: Processors are divided into 3 categories: - 8, 16 and 32-bit processor, depending upon the demand of cost, power, performance and programmability. 8-bit processors consume less power and have extreme low cost for simple control system. In contrast to 8-bit, 32-bit processors have high performance, high programmability, and are widely used in PDA and cellular phone that need high computation but it has high power consumption. On the other hand 16-bit processors have high performance and power than 8-bit processor and low power consumption than 32-bit processor. They are often used in 16-bit applications such as cellular communication, disk driver controller, and airbags. The 16-bit fully functional single cycle processor is applicable for real tasks and also used for assembly language programming. We need to participate in the process of processor design and to understand the inner structure of processor. Therefore its architecture is well structured and simple enough so that it can be built by first grade students, without any design experience. These all requirements can be obtained by the FPGA based processor with Very High Speed IC Hardware Description Language (VHDL) [1]. Figure 1 shows the design of basic steps to the processor. In the mid-1970's advances in semiconductor technology began to reduce the difference in speed between processor chips and main memory. As high-level languages displaced assembly language and memory speed increased the major reasons for CISC began to disappear and computer designers began to look at ways computer performance could be optimized beyond just making faster hardware. Realizations one of their key was that a sequence of simple instructions produces the same results as a sequence of complex instructions, but can be implemented with a simpler (and faster) hardware design.

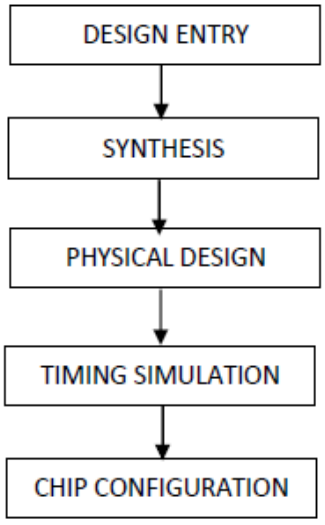

Fig.1. Proposed 16 bit processor Design Method

Memory design advances to achieve the desired speed up have been critical in the development of RISC architectures and basic characteristics of RISC processors are:

- **Simple instruction set:**

The instruction set contains simple, basic instructions, from which more complex instructions can be composed in a RISC machine.

- **Same length instructions:**

Each instruction fetched in a single operation due to same length.

- **One machine-cycle instructions:** Complete most instructions in one machine cycle, which allows the processor to handle several instructions at the same time.

This pipelining may be a key technique wont to speed up RISC machines. RISC designer's area unit involved primarily with making the quickest chip potential, then they use variety of recent directions till the slow instruction has captive on to subsequent stage. Since the processor is sitting idle once stalled, each the designers and programmers of RISC systems build an aware effort to avoid stalls. To do this, designers use many techniques that primarily enhance the performance of the processor.

II. General of RISC Processor

A processor incorporates most or all of the functions of a computer's Central process Unit (CPU) on one IC or semiconductor unit. So as to accomplish these many Innovative and unconventional style tradeoffs are created, while not compromising the goals. The final design of 16-bit teaching processor is shown in figure 2. It contains range of basic items. There's a register array of 8-bit and 16-bit, a 16-bit ALU, a 16-bit shifter, a program counter, associate degree instruction register, a 16-bit comparator, associate degree address register and management unit. All of those units communicate through a standard 16-bit tri-state knowledge bus.

The top level style consists of the processor block and a memory block human action through a bifacial knowledge bus, associate degree address bus, and few management lines. The processor fetches directions from external memory and executes these directions to run a program. These directions area unit hold on in instruction register and decoded by management unit. The management unit causes the suitable signal interaction for processor unit to execute the instruction.

If the instruction is associate degree add of 2 registers, the management unit would cause the primary register price to be written in operational register (OpReg) for temporary storage. The second register price would then be placed on knowledge bus. The ALU is currently set at add mode and result are hold on in output register (OutReg). Output register stores the ensuing price till it's traced to the ultimate destination. once death penalty associate degree instruction, range of steps takes place. Program counter holds the address in memory of the present instruction. once associate degree instruction has finished execution, the program counter is advanced to wherever subsequent instruction is found. If the processor is death penalty a linear stream of directions, this is often subsequent instruction. If a branch is taken, the program counter is loaded with next instruction location directly. The processor values the address register, which supplies output as new address on the address bus. At a similar time, management unit sets the R/W (read write signals) to '0' for scan operation and sets signal VMA (Valid Memory Address) to '1'; sign the memory that the address is currently valid. Memory decodes the address and places the memory knowledge on knowledge bus. Once knowledge has been placed on knowledge bus, memory set the prepared signal to '1' indicating that the memory knowledge is prepared for consumption.

Control unit causes the memory knowledge to be written into the instruction register. The management unit is currently access and decodes the instruction. The decoded instruction executes, and method starts all over again.

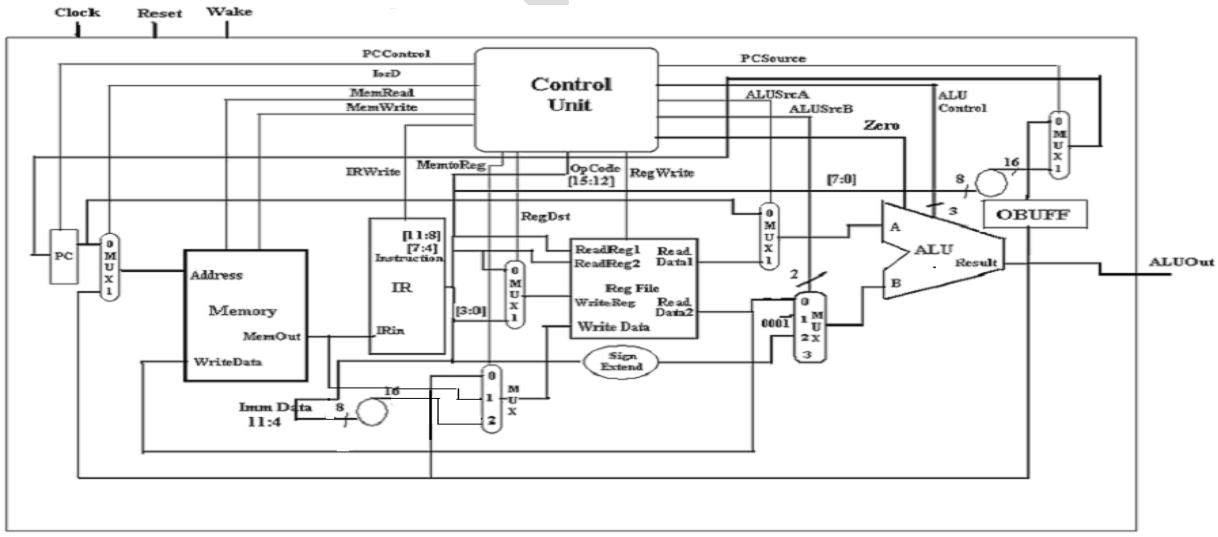

Figure 2:- Internal architecture of 16-bit processor

A. Data Path

There square measure sixteen range of sixteen bit registers in register file. So Register file information path consists of 4-bit address bus and 16-bit data bus. This can be a 2 port register file which might perform 2 concurrent browse and one write operation. It contains sixteen 16-bit general purpose registers. The registers square measure named R0 through R15. Once the Register Write signal is high, a write operation is performed to the register indicated by the write address, otherwise it outputs the worth contained within the register indicated by the browse address. The ALU is chargeable for all arithmetic and logic operations that happen among the processor. These operations will have one quantity or 2, with these worth's returning from either the register file or from the immediate value from the instruction directly. The output of the ALU goes either to the memory (in the case wherever the output is associate degree address) or through an electronic device back to the register file. The ALU has 2 16-bit knowledge inputs for 2 operands and 16-bit output for result. It will perform Booth multiplication, restoring division, addition and subtraction functions. The tiny electronic device of the info path has 3 16-bit inputs and one 16-bit output. It chooses one among the 16-bit inputs and sends to the output looking on the condition of two-bit electronic device select signal. The number unit has 2 inputs and one output. The interior structure consists of 4 units particularly Booth Encoder, Partial Product generator, Carry Save Adder, and Carry Look Ahead Adder [3, 4]. The highest style of Booth Encoder has one input and 3 outputs. The Booth Encoder generates the booth codes to cipher the number into the partial product. The outputs of the Booth Encoder square measure fed to the inputs of the partial product generator (PPG) module. The PPG module reads the booth code signals generated from the Booth Encoder to cipher the number into the partial product. The outputs of the partial product generators square measure taken because the inputs of the Wallace tree. The Wallace tree uses the three to a pair of carry save adder to implement the Wallace tree. The outputs of the Wallace tree square measure fed to the inputs of the Carry Look Ahead (CLA) adder. The CLA is employed to perform the addition of the ultimate total and carry vector. When implementing all the four parts of the quick number unit, the required multiplication results obtained. In restoring division, the divisor is positioned suitably with relevance the dividend. Then the divisor is deducted from the dividend. If the remainder is zero or positive, a quotient little bit of '1' is set. The rest is extended by another little bit of the dividend and therefore the divisor is repositioned and another subtraction is performed. If the rest is negative, a quotient little bit of '0' is set and therefore the dividend is restored by adding back the divisor. Then the divisor is repositioned for one more subtraction and output looking on the condition of two-bit electronic device choose signal. The number unit has 2 inputs and one output. the interior structure consists of 4 units particularly Booth Encoder, Partial Product generator, Carry Save Adder, and Carry Look Ahead Adder. The highest style of Booth Encoder has one input and 3 outputs. The Booth Encoder generates the booth codes to cipher the number into the partial product. The outputs of the Booth Encoder square measure fed to the inputs of the partial product generator (PPG) module. The PPG module reads the booth code signals generated from the Booth Encoder to cipher the number into the partial product. The outputs of the partial product generators square measure taken because the inputs of the Wallace tree.

The Wallace tree uses the three to a pair of carry save adder to implement the Wallace tree. The outputs of the Wallace tree square measure fed to the inputs of the Carry Look Ahead (CLA) adder. The CLA is employed to perform the addition of the ultimate total and carry vector. When implementing all the four parts of the quick number unit, the required multiplication result's obtained. In restoring division, the divisor is positioned suitably with relevance the dividend. Then the divisor is deducted from the dividend. If the rest is zero or positive, a quotient little bit of '1' is set. the rest is extended by another little bit of the dividend and therefore the divisor is repositioned and another subtraction is performed. If the rest is negative, a quotient little bit of '0' is set and therefore the dividend is restored by adding back the divisor. Then the divisor is repositioned for one more subtraction.

B. Control Unit

Control unit composed of controller, program counter, instruction register, and electronic device. The controller provides the required signal interaction to create the info flow through the processor and perform expected operate. The 16-bit program counter indicates the address of ensuing location wherever ensuing instruction is to be fetched. Its 16-bit program counter input, 16-bit program counter output, and a few management signals like read, write, and clear. The Program Counter (PC) contains the address of the instruction that may be fetched from the Instruction Memory throughout ensuing clock cycle. Commonly the computer is incremented by one throughout every clock cycle unless a branch instruction is dead. Once a branch instruction is encountered, the computer is incremented or decremented by the number indicated by the branch offset. The instruction register (IR) has one 16-bit input and 2 16-bit output [5]. The instruction set describes the

bit-configurations allowed within the IR .The instruction consists of operation codes and quantity. The processor support direct, register, registers indirect and immediate addressing modes. In immediate addressing; the quantity field contains the info itself. In registers addressing, the quantity field contains the address of a knowledge path register within which the info resides. In registers indirect addressing, the quantity field contains the address of a register that successively contains the address of a memory location within which the info resides. In direct addressing, the quantity field contains the address of a memory location within which the info resides. In indirect addressing, the quantity field contains the address of a memory location that successively contains the address of a memory location within which the info resides.

III. Reduced Instruction Set Computing Microinstruction Set:

The unit of measurement instruction set is easy like every alternative reduced instruction set computing styles. Unit of measurement square measure a load/store design, which implies that solely load and store directions access memory. Alternative directions will solely operate values within the registers [8]. Generally, the MIPS directions per second of measurement directions are broken into 3 classes: the memory-reference instructions, the arithmetic- logical directions, and therefore the branch directions. Also, there square measure 3 completely different directions format (as shown in Fig.2) in MIPS architecture: R-Type instructions, I-Type directions and J-Type directions. The sixteen bit reduced instruction set computing silicon chip that was coded in VHDL has the subsequent specifications: -

1. 4- bit op-code for a most of sixteen directions
2. 16 bit register bank of sixteen registers 256*16 bit memory house available victimization eight bit address.
3. 16 bit ALU to perform the logical operations

The reduced instruction set computing silicon chip was created with the subsequent instructions: -

(1) Arithmetic & amp; Logical – ADD, SUB, OR, AND, XOR
(2) Memory connected – Load from memory and Store to memory address
(3) Immediate – Move immediate knowledge
(4) Conditional – Jump to memory address if condition is true
(5) Unconditional– Jump to a memory address to loop a program
(6) Halt to prevent the program
(7) Sleep instruction

The reduced instruction set computing processor was tested employing a program to get the Fibonacci series. The sleep instruction is employed to place the processor in a very state wherever neither the inputs nor outputs of any of the registers or memory locations modification. The clock cycle is prevented from reaching the processor by employing a latch controlled by an indication that is generated once the instruction is dead. The processor comes out of the sleep state once associate degree external wake signal is declared.
The simulations were in deep trouble a complete run-time of concerning fifty microseconds out of that the processor was idle when the execution of the program (to generate ten Fibonacci series numbers) was completed at concerning twenty six microseconds.

Table 1. Shown Instruction Set of 16-bit RISC Processor.

S.No.	Instruction	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	ADD	0	0	0	0	Source Reg 1				Source Reg 2				Destination Reg			
2	SUB	0	0	0	1	Source Reg 1				Source Reg 2				Destination Reg			
3	AND	0	0	1	0	Source Reg 1				Source Reg 2				Destination Reg			
4	OR	0	0	1	1	Source Reg 1				Source Reg 2				Destination Reg			
5	XOR	0	1	0	0	Source Reg 1				Source Reg 2				Destination Reg			
6	Store Word	0	1	0	1	Reg with Mem Address				Reg With Data to store				4 bit Mem. Offset			
7	Load Word	0	1	1	0	Reg with Mem Address				Reg to load data to.				4 bit Mem. Offset			
8	Jump address	0	1	1	1	Jump Address								X	X	X	X
9	Move Imm.	1	0	0	0	8 bit data extended to 16 bit								Destination Reg			
10	Jump on Zero	1	0	0	1	Jump Address								Reg to check			
11	Sleep	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
12	Halt	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

A clock cycle with fundamental quantity of 24ns was used with a five hundredth duty cycle. the primary state is that the reset state once the external input reset goes high for the primary 24ns. Then ensuing clock cycle is employed to fetch the instruction from memory. The fetch is competed at the start of third clock cycle that is that the op-code deciphers cycle. On the idea of the opcode applicable management signals square measure issued within the next clock cycle to start the command execution. The Logical commands take 2 a lot of clock

cycles to browse the info, perform the calculation then to store it within the register file. However if the instruction is memory connected then the primary clock cycle when the opcode is decoded is employed to calculate the memory address, ensuing to browse the worth to be loaded or keep & the ultimate cycle to store in memory or Register file. The conditional directions check whether or not the contents of the required register square measure zero and consequently browse the memory address to leap to. The bit-wise Instruction Set design and therefore the processor design square measure shown on Figure a pair of and Figure three. The distinction between sleep associate degreed halt is that the processor will begin of the sleep state once an external wake signal is issued and continue its execution [7, 8].

IV. Power Estimation

Clock-gating is the most common register transfer level (RTL) optimization for reducing dynamic power. In clock gating method, clock is applied to only the modules that are working at that instant. Clock-gating support adds additional logic to the existing synchronous circuit to prune the clock tree, thus disabling the portions of the circuitry that are not in use.

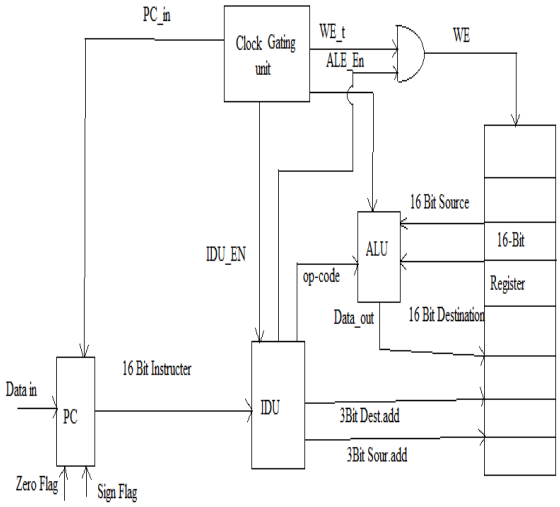

Fig.3. Shows clock gating circuit

Here, an additional circuit called clock gating circuit is introduced before the data path unit which provides clock inputs to only working modules based on the clock selection logic. Thus, the unnecessary dynamic power consumption is reduced using clock gating technique home. Depends on the SMS received the device control unit will control the corresponding device. This system provides ideal solution to the problems faced by home owners in daily life. The system is wireless therefore more acceptable and cost-effective.

The reduced instruction set interrupt controller (RISI Controller) architecture mainly consists of ALU (Arithmetic and logical Unit), Port Controller, Interrupt controller and Register Array and its block diagram is shown in the Fig. 1. It contains RISC CPU, Interrupt controller, Port controller and Program flow controller. These blocks are connected by internal buses.

V. Simulation and Result synthesis

I have synthesized and simulated the VHDL code of the proposed processor using Xilinx Integrated Software Environment tool (Version 13.1). The synthesis results and simulation results of processor are presented for justification.

The proposed 16 bit RISC processor is coded with VHDL (very high speed integrated circuit hardware description language). Using Xilinx ISE 13.1 software the code is tested and checked for error. When there is no error, the code is synthesized and simulated using Xilinx ISE 13.1 software. The synthesis and simulation results are compared with the theoretical results. Before the start of simulation, the memory is loaded by writing the instructions and data into the memory. The completed processor with memory is tested for addition, subtraction, multiplication, and division program The operating setting for the planning is:

1. Target Device: Spartan 3xc3s1000-4fg320.
2. Tool Version: Xilinx ISE 13.1
3. Optimization Goal: Speed and Low Power Consumption
4. Style Strategy: Balanced
5. Total Slices: 3,820
6. Total LUTs: 3,227

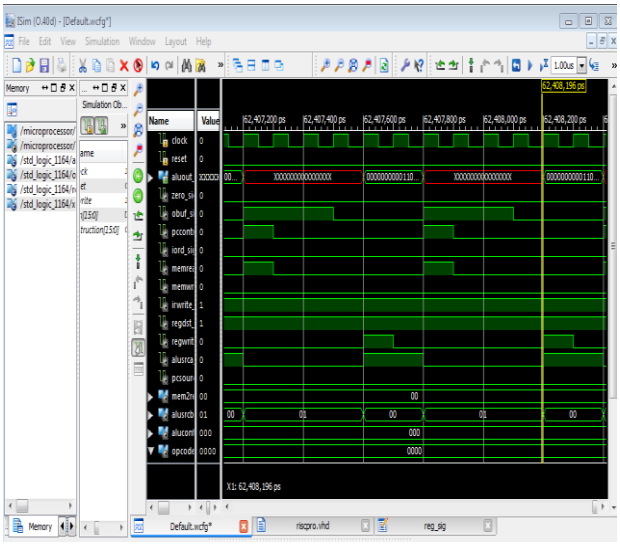

Fig.4. Shows Simulation Waveforms Add of Two 8-Bit Resisters Value

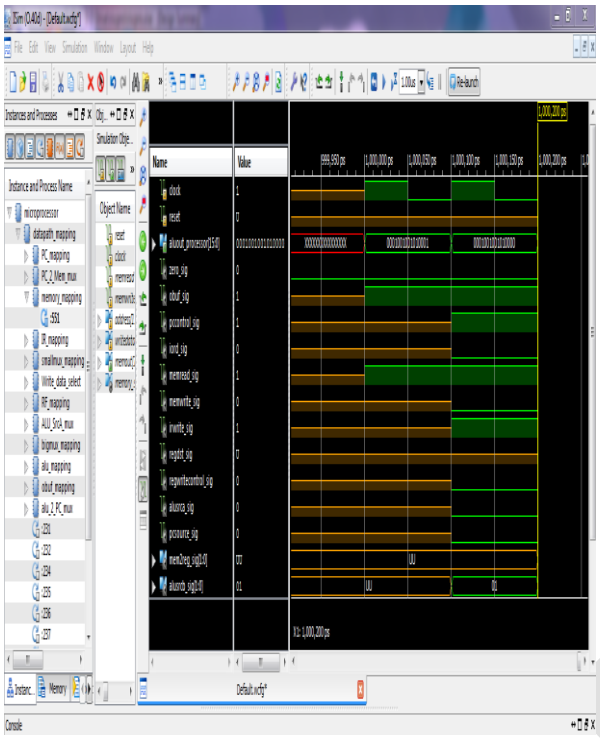

Fig.5. Shows Simulation Waveforms for SUB of Two 8-Bit Resisters Value

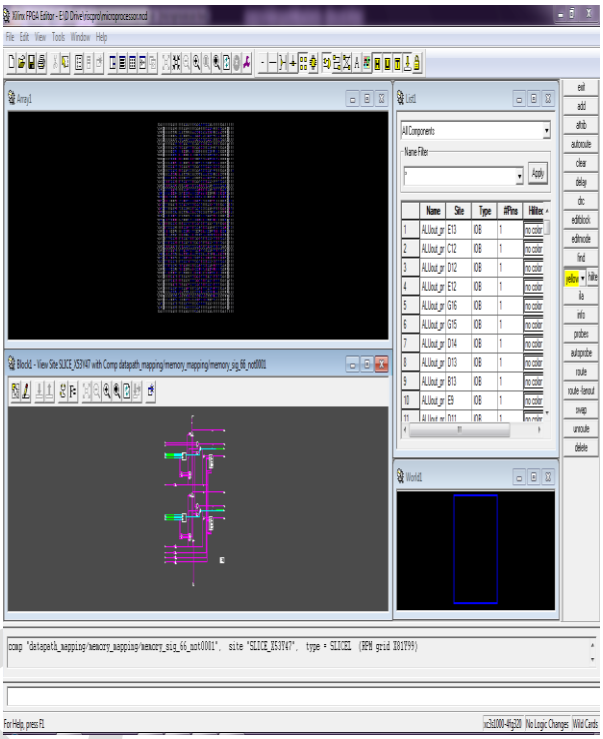

Fig.7. Shows Xilinx FPGA Editor of RISC Processor

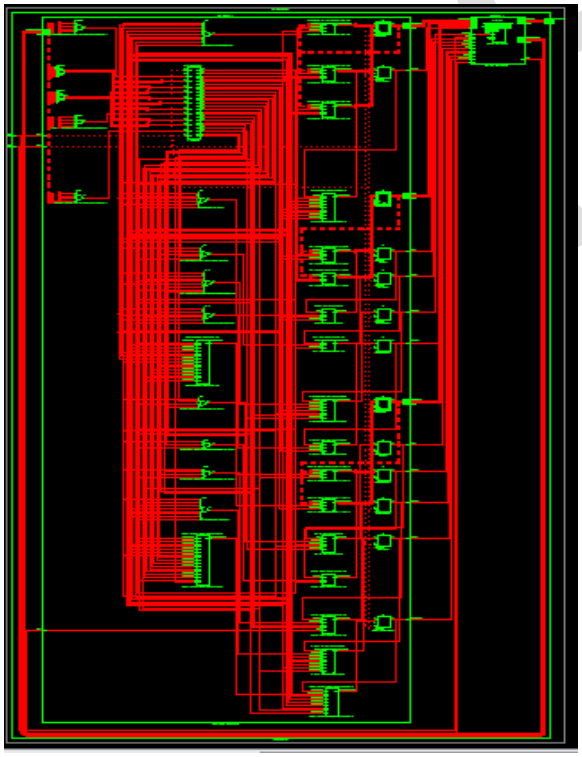

Fig.6. Shows RTL Schematic View

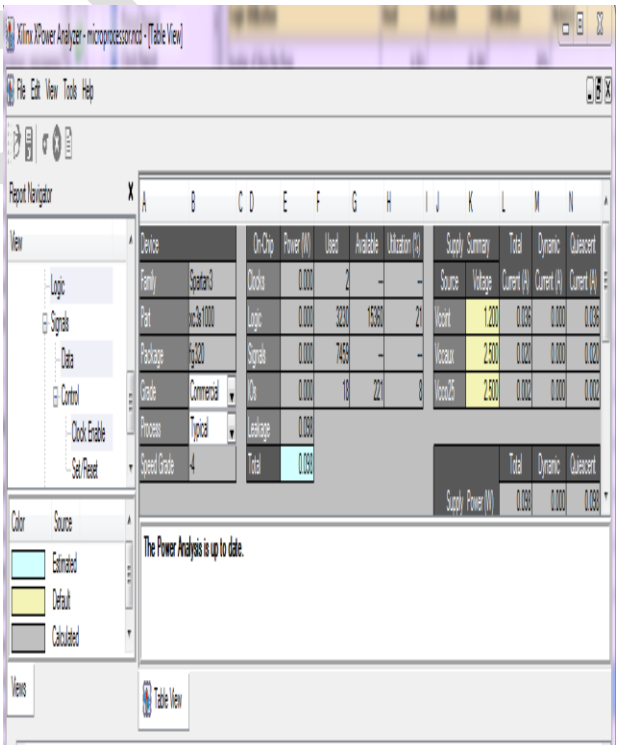

Fig.8. Shows Power Report

Table 2. Shows Comparison between different processors

	8-bit RISC Processor	16-bit RISC Processor	32-bit RISC Processor	Proposed 16-bit RISC Processor
Target Device	Xilinx SPARTAN 3E XCS500E	Xilinx spartan 3S500EFG320	Xilinx Virtex4 XC4VLX15	Spartan 3xc3s1000 -4fg320
Slice Utilized(Area)	2.19%	2.017%, 2..2%	6.77%	35.46%
Delay (ns)	77.4ns	22.323ns, 18.622ns	14.348ns	16.732ns
Maximum Frequency	300MHz	44.797MHz, 53.700MHz	179.092MHz	59.767MHz
Power Dissipation (W)	6.25	0.87	0.829	0.098

Timing Summary of the Proposed 16-Bit RISC Processor

1. Speed Grade: -4
2. Minimum period: 16.732ns (Maximum Frequency: 59.767MHz)
3. Minimum input arrival time before clock: 15.268ns
4. Maximum output required time after clock: 18.570ns
5. Maximum combinational path delay: 12.800ns
6. Total Number of used pin in RISC processor :320
7. Power desipation: 0.098W

VI. Conclusion and Future Work

The RISC processor was enforced and also the operating was verified at the highest level and at the part levels victimization Xilinx 13.1 tool. Separate modules are enforced for the 5 pipeline stages i.e. instruction-fetch, instruction decrypt, instruction execute, memory and input access, and memory write-back. Processors area unit divided into three categories: - 8-bit, 16-bit and 32-bit processor, relying upon the demand of performance, cost, power and programmability. 8-bit processors have extreme low price and consume less power for easy system. In distinction to 8-bit, 32-bit processors have high programmability, high performance and area unit wide employed in telephone and personal organizer that require high computation however it's high power consumption.

On the opposite hand 16-bit processors have high performance and power than 8-bit processor and low power consumption than 32-bit processor. They're typically employed in 16-bit applications like disk driver controller, cellular communication and airbags.

Future Work

A number of future proposals is recommended supported the Thesis. They include:

1. Development of a 32-bit RISC processor.
2. Development of separate hardware for memory and implementing memory management Sub-routines.
3. Hardware implementation and Development of a full cache memory.

Reference

[1] Galani Tina, R. D. Daruwala, "Performance Improvement of MIPS Architecture by Adding New Features", IJARCSSE, Volume 3, Issue 2, February 2013; ISSN: 2277 128X.

[2] R. Uma, "Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool", IJERA, ISSN: 2248-9622 Vol. 2, Issue 2, Mar-Apr 2012

[3] "Low Power Code Generation for a RISC Processor by Register Pipelining" Stefan Steinke, Ruediger Schwarz, Lars Wehmeyer, Peter Marwedel, University of Dortmund, Dept. of Computer Science, Otto-Hahn-Strasse 1644221 Dortmund, Germany.

[4] Madhavi Anupoju, M. Sunil Prakash, "Performance Evolution of 16 Bit Processor in FPGA using State Encoding Techniques". IJSR, India Online ISSN: 2319-7064.

[5] Viral Makwana ,Vandana Shah, Roma Patel " 32*32 Bit Matrix Coprocessor Design Using VHDL FPGA Device For Advanced RISC Machine Operations" International Journal of Advanced Engineering Technology E-ISSN 0976-3945.

[6] Sivrama P. Dandamudi "Guide to RISC Processor for programmers and engineers", ISBN 0-387-21017-2, Springer, 2005.

[7] Gaganpreet Kaur "VHDL Basic to Programming", ISBN 978-81-317-3211-3, Pearson Education, 2011

[8] J. Bhaskar "A VHDL Primer- Third Edition", ISBN-13 978-81-203-2366-7, PHI Learning, 2009

Author Profile

Surendra Bajia is pursued Diploma in (EF) Electronics Fiber optics Communication from BTTI, PILANI. B. Tech. in ECE from Govt. ECB Bikaner under Rajasthan Technical University and He is pursued his M. Tech. in Embedded System from JNU JAIPUR. His interested research area in Analog and Mixed signal circuit, Low power VLSI Design, Embedded System Design, VLSI circuit testing & Signal Processing, Robotics, Mathematical modeling and simulation.