International Journal of Gender, Science and Technology
ISSN: 2040-0748

UGC Care Group I Journal.
Vol-13 Issue-02 July 2024

https://ijgst.com.2024.v13.i2.pp1061-1066

DESIGN & ANALYSIS OF 32 B IT RISC PROCESSOR USING LOW POWER PIPELINING

DR T.KRISHNA MOORTHY[1] MACHERLA YAMINI SARASWATHI[2] PRADEEP PANUGANTI [3] GUNTURU KRANTHI KUMAR[4]

[1]ASSOC.PROFESSOR, DEPARTMENT F ELECTRONICS & COMMUNICATION ENGINEERING,
[2] ASSST. PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[3]ASST.PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[4]ASST.PROFESSOR, DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING,
[1,2,3,4] SRI MITTAPALLI COLLEGE OF ENGINEERING

Abstract - In the architectural point of view, the processor has 3-stage pipeline, 6 register banks, 32-bit ALU, and 4-cycle MAC. The core described here was designed by latch base for low power and low complexity. Its functional operation was verified by comparison the results of logic simulation with those of the commercial simulator. These RISC or Reduced Instruction Set Computer is a design philosophy that has become a mainstream in Scientific and engineering applications. The main objective of this paper is to design and implement of 32 – bit RISC (Reduced Instruction Set Computer) processor using XILINX VIRTEX4 Tool for embedded and portable applications. The design will help to improve the speed of processor, and to give the higher performance of the processor. The most important feature of the RISC processor is that this processor is very simple and support load/store architecture.

Index: Processor, Reduced Instruction Set Computer (RISC), VHDL, Xilinx 12.1, FPGA.

I. INTRODUCTION

Now a day RISC Processor is increasing widely used in every field. Most microprocessor in today's market is based on either RISC or CISC architecture technologies. The RISC architecture boost the computer speed also used in control pipelined outline. The pipelined architecture is used which minimizes the latency and increases the speed and reduce the stalling in instruction. The whole architecture of RISC processor work on the 2 cycle .the fixed size of instruction allows the given instruction to be easily piped. RISC processor has a flexible architecture. With the rapid development of silicon technology RISC includes extensions to RISC concepts that help achieve given levels of performance at significantly lower cost than other systems. The main features of RISC processor are the instruction set can be hardwired to speed instruction execution.

In the present work, the design of a 32-bit data width Reduced Instruction Set Computer (RISC) processor is presented. It has a complete instruction set, program and data memories, general purpose registers and a simple Arithmetical Logical Unit (ALU) for basic operations. In this design, most instructions are of uniform length and similar structure, arithmetic operations are restricted to CPU registers and only separate **load** and ***store*** instructions access memory. The architecture supports 16 instructions to support Arithmetic, Logical, Shifting and Rotational operations. The remainder of this paper is organized as follows. Section II explains System architecture of a 32 – Section III presents RISC Instruction set format. Section IV results with the implementation of the RISC design topology. The final section presents the Conclusion and References.

II. ARCHITECTURE OF 32 BIT RISC PROCESSOR

https://ijgst.com.2024.v13.i2.pp1061-1066

The system architecture of a 32 - bit RISC processor is shown in Fig. 1. The RISC processor architecture consists of Arithmetic Logic Unit (ALU), Control Unit (CU), Barrel Shifter, Booth's Multiplier, Register File and Accumulator. RISC processor is designed with load/store (Von Neumann) architecture, meaning that all operations are performed on operands held in the processor registers and the main memory can only be accessed through the load and store instructions. One shared memory for instructions (program) and data with one data bus and one address bus between processor and memory . Instruction and data are fetched in sequential order so that the latency incurred between the machine cycles can be reduced. For increasing the speed of operation RISC processor is designed with five stage pipelining. The pipelining stages are Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Data Memory (MEM), and Write Back (WB).

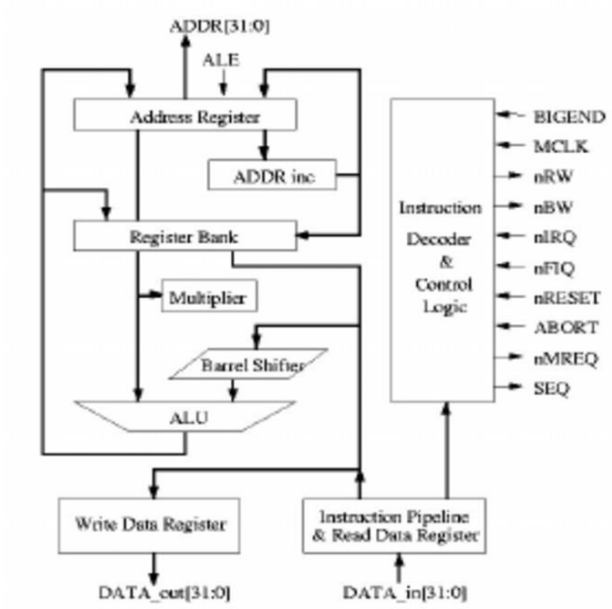


Fig 1 Architecture of 32 bit RISC processor

- Fetch- Instruction memory- in the fetch33 bit of instruction are drawn from instruction memory, in the instruction memory it contains the instruction that are executed by the processor.
- Program counter- The program counter contains the address of the instruction that will be fetched from the instruction memory during the next clock cycle. The pc is incremented by one during each given clock cycle. The fetch unit is work in the positive edge of the cycle.
- Decode- In the decoder consist of 2 addressing mode as input immediate addressing mode or a register addressing mode the three operation ALU ,universal shifter ,shifter rotator are performed. The two decoder are exist in the control unit the first decoder perform the arithmetic and logical operation And second decoder performs rotating and shifting operations. The decoder operation is performed in negative cycle.
- Execute Unit – The execution stage is where actual computation take place. All the operations that are perform by the given control signals are executed. The ALU operation , shifter operation and shifter rotator operation are performed exactly given control signal. This operation is performed in the positive clock edge of the cycle.
- Write back- In this stage, their result is written into the register file by both single and two cycle instructions.

The instructions opcode field bits [31-26] are sent to a control unit to determine the type of instruction to execute.

International Journal of Gender, Science and Technology
ISSN: 2040-0748

UGC Care Group I Journal.
Vol-13 Issue-02 July 2024

https://ijgst.com.2024.v13.i2.pp1061-1066

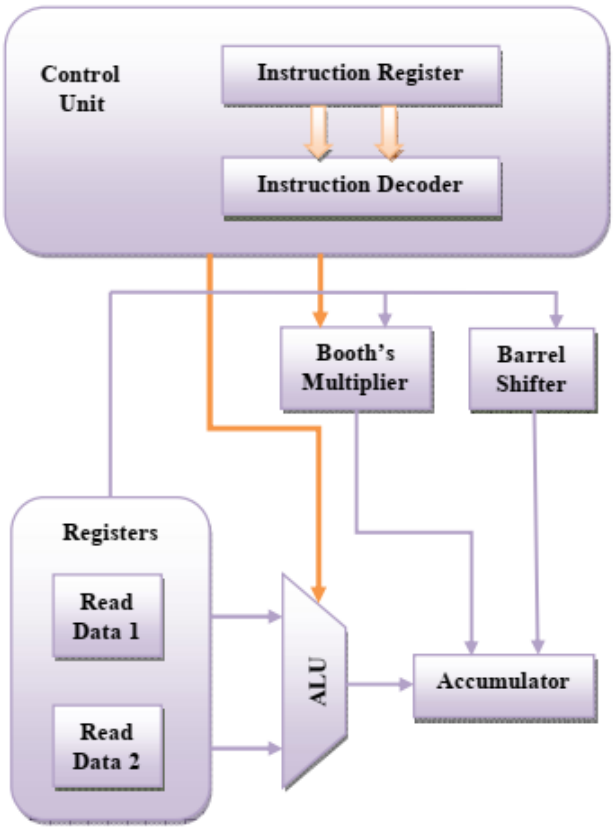


Fig 2 System architecture of 32 bit RISC processor

The control unit has two instruction decoders that decode the instruction bits and the decoded output of the control unit is fed as control signal either into Arithmetic logic unit (ALU) or Barrel shifter or Booth's Multiplier. If the instruction decoded is arithmetic, the ALU result must be written to a register. If the instruction decoded is a load or a store, the ALU result is then used to address the data memory. The final step writes the ALU result or memory value back to the register file.

Operations

1) ALU Operation- The ALU design has 16 operations are using ,comprises of 2 units one is for logical operation such as AND, NAND, OR, NOR, XOR, XNOR, NOT, BUFFER, Parity Generator are used and another one is arithmetic operation is used CSA Adder , subtract ,efficient multiplier and divider, reminder, square functions are operated. In the arithmetical logic unit the operation of Arithmetic operation such as ADD and SUBTRACT Based on the control unit operation like when

Cin=0 (Add Operation)
Cin=1 (Subtract Operation)

Opcode 33 bit	Source1 9 bit	Source2 9 bit	Destination 9 bit

Table I ALU operations

SELECT LINES					DECODER															OPERATION PERFORMED
S4	S3	S2	S1	S0	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	ADD
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	SUB
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	MULTI
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	%REM
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	AND
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	NAND
0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	OR
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	NOR
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	XOR
0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	XNOR
0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	NOT
0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	BUFFER
0 0	1 1	1 1	0 0	0 1	0 0	0 0	0 1	1 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	PARITY GENERATOR DIVIDE
0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SQURE
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NO OPERATION

2) Universal Shift Register- The 9 operations are performed in the Universal Shift Register .The right shift operation, left shift operation and arithmetic left shift operation is performed. Now according to the 9 select line operation performed and it is worked as follows by the table
3) Barrel shift Rotator- The input of the next multiplexer is connected to the output of first multiplexer in this way it can perform the rotation operation as the input get shifted in each multiplexer. In the barrel shifter 6 operations are performed. The left rotation and the right rotation is performed in this rotator.

Table II Universal shifter operations

SELECT LINES									OPERATIONS PERFORMED OUTPUT
8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	1	Right Shift 1 Bit
0	0	0	0	0	0	0	1	0	Right Shift 2 Bit
0	0	0	0	0	0	1	0	0	Right Shift 3 Bit
0	0	0	0	0	1	0	0	0	Left Shift 1 Bit
0	0	0	0	1	0	0	0	0	Left Shift 2 Bit
0	0	0	1	0	0	0	0	0	Left Shift 3 Bit
0	0	1	0	0	0	0	0	0	Arithmetic Left shift1 bit
0	1	0	0	0	0	0	0	0	Arithmetic Left shift2 bit
1	0	0	0	0	0	0	0	0	Arithmetic Left shift3 bit

Table III Shifter rotator operations

SELECT LINES						OPERATIONS PERFORMED OUTPUT
5	4	3	2	1	0	
0	0	0	0	0	1	Left Rotate 1 Bit
0	0	0	0	1	0	Left Rotate 2 Bit
0	0	0	1	0	0	Left Rotate 3 Bit
0	0	1	0	0	0	Right Rotate 1 Bit
0	1	0	0	0	0	Right Rotate 2 Bit
1	0	0	0	0	0	Right Rotate 3 Bit

III. RISC INSTRUCTION SET

RISC only has three instruction types: I-type is used for the Load and Stores instructions, R-type is used for Arithmetic instructions, and J-type is used for the Jump instructions as shown in Fig.3**.**

Arithmetic instructions or R-type include: ALU Immediate (e.g. addi), three-operand (e.g. add, and, slt), and shift instructions (e.g. sll, srl). The J-type instructions are used for jump instructions (e.g. j). Branch instructions (e.g. beq, bne) are I-type instructions which use the addition of an offset value from the current address in the address/immediate field along with the program counter (PC) to compute the branch target address; this is considered PC-relative addressing. The most significant feature of instruction set is that all of instruction may be executed conditionally according to the condition flags.

A condition field within all instructions is compared with the condition flag in program status register and the result of the comparison determines whether the instruction can be executed or not. This reduces the need for forward branches and allows very dense in-line code (without branches) to be written. In addition, shift operation is performed in serial with ALU in all data path cycles. This enables data processing instructions to execute various arithmetic operations using shift operation in one clock cycle .

https://ijgst.com.2024.v13.i2.pp1061-1066

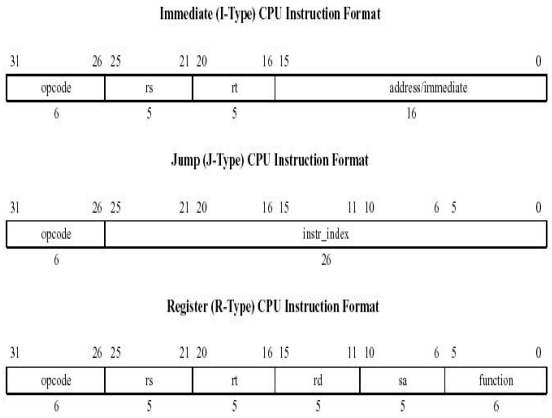
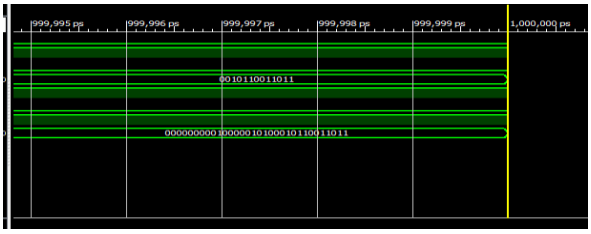


Fig 3 RISC instruction types

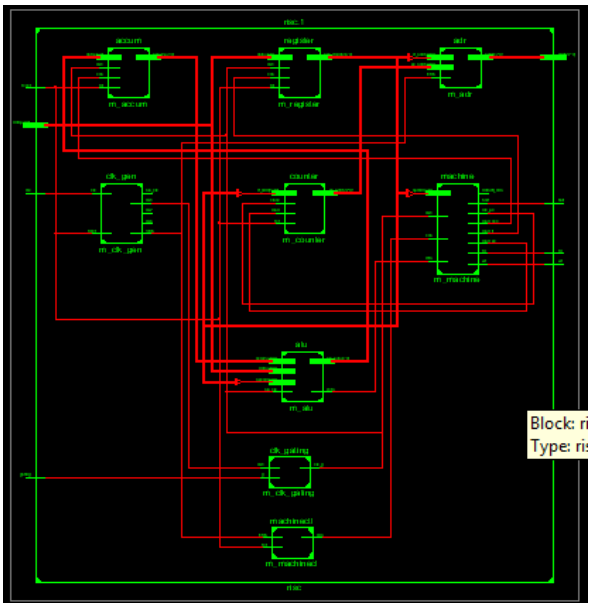
IV. RESULTS

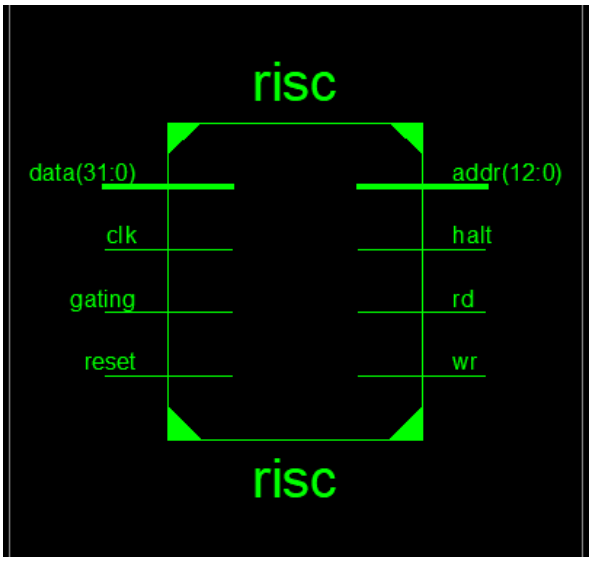
Simulation result :



Synthesis result:

RTL schematic:





Design summary:

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	233	4656	5%
Number of Slice Flip Flops	122	9312	1%
Number of 4 input LUTs	407	9312	4%
Number of bonded IOBs	51	232	21%
Number of MULT18X18SIOs	6	20	30%
Number of GCLKs	2	24	8%

V. SUMMARY

This section presents the performance of the processor in terms of its Total Power and Delay that are obtained using the Virtex4 XC4VLX15 XILINX

tool. Table IV presents the maximum power dissipation, area occupied and time taken by each module to operate the processor. Table IV Delay, Total Power and Area Calculation **.**

Table IV Summary

Topology	Delay (ns)	Slices Utilized (Area)	AT	Power Dissipation 10^{-9}W	Total Power (W)
Control Unit	0.905	15	13.575	0.159	0.165
ALU	4.495	60	269.7	0.159	0.168
Barrel Shifter	2.905	262	761.11	0.159	0.164
Booth's Multiplier	4.495	60	269.7	0.159	0.166
Register File	5.443	74	402.782	0.159	0.166
Total	18.243	471	8592.453		0.829

VI. CONCLUSION

A 32-bit RISC processor with 16 instruction set has been designed. Every instruction is executed in one clock cycles with 5-stage pipelining. The design is verified through exhaustive simulations. The processor achieves higher performance, lower area and lower power dissipation. This processor can be used as a systolic core to perform mathematical computations like solving polynomial and differential equations. Apart from this it can be used in portable gaming kits.

VII. FUTURE SCOPE

The floating-point multiplication is not a complete 32-bit operation as it incorporates only an 11-bit mantissa. This can be improved by multiplexing the result or, by using the concept of pipelining. A compiler to this processor can be realized in software to facilitate easy input and output to this processor. The 32-bit processor can be made to a 64-bit processor making minor modifications to the code. This increases the data handling capability of the ALU. It also increases the range of numbers that can be operated by the processor. The exception cases in the arithmetic operations can be flashed to the user using seven segment displays. Using the compiler can facilitate these operations.

BIBLIOGRAPHY

1) Digital System Design Using VHDL by Charles H Roth Jr.
2) A VHDL Primer 3rd edition by J. Bhasker.
3) VHDL by Douglas L. Perry.
4) A VHDL Synthesis Primer by J.Bhasker. Computer Architecture and Organization by John P. Hayes.
5) Computer Organization and Architecture by William Stallings.
6) www.aldec.com/products/tutorials.
7) www.toolbox.xilinx.com/docsan.
8) www.csold.cs.ucr.edu/content/esd/labs/tutorial/vhdl_page.html.
9) www.psc.edu/general/software/packages/ieee/ieee.html..
10) www.leepoint.net/notescomp/data/numbers/ieee754.html.
11) www.cs.berkely.edu/
12) www.mecaulay.ac.uf/fearlus/floating-point
13) www.carlstedt.se/hyldo/datorarkitektur/tidigare