International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Design and Implementation of a softcore Risc-V RV32-IM, Dual Core Processor on a FPGA

Nididana Praveen Kumar
GITAM School of Technology Hyderabad Campus, GITAM University.
E-mail: pnididan@gitam.in

Abstract

For applications involving systems-on-chips and the Internet of Things, the RISC-V ISA is rising to the top of the list of instruction sets. RISC-V offers a scalable solution suitable for a wide range of devices, from simple embedded systems to high-performance systems. Its design centers around simplicity, enabling easy implementation, verification, and customization, thus in this paper, the Softcore processor RV32-IM, Dual Core with in-order superscalar is designed and implemented onto FPGA, As Dual-core processing which has a significant pivotal advancement in computer architecture, introducing heightened performance and independent multitasking prowess compared to their single-core predecessors. By integrating two autonomous processing units within a solitary chip, dual-core processors empower simultaneous execution of multiple tasks, thereby enhancing system responsiveness and operational efficiency. The softcore design of an open-source RISC-V processor utilizing contemporary hardware design methodologies is covered in this paper, along with details on how the softcore design was implemented on an FPGA using Xilinx Vivado software tool which later with Vitis IDE can be used to implement applications, as soft-core processors on FPGAs are often preferred as an alternative to ASICs due to their flexibility, shorter development cycles.

Keywords—RISC-V, open-source, soft-core, dual core, superscalar, FPGA

1. Introduction

The Instruction Set Architecture (ISA), which defines the set of instructions and associated encoding schemes, is the fundamental framework in computer architecture that software uses to communicate with hardware. The RV 32I of RISC V ISA which is used in this research as a fundamental component of computational arithmetic, representing the fundamentals of numerical representation and manipulation. The way 32-bit integers are treated within the RISC-V ISA strikes a careful balance between performance and simplicity, giving programmers the accuracy and productivity needed for a wide range of applications [1]. This is a new era for invention, cooperation, and accessibility which has been brought about by the introduction of open-source technologies. The integration of numerous processing cores on a single chip sends a signal as the paradigm shifts in the constantly changing field of computer architecture and the dawn of a new era of computational power and efficiency. Leading this change is the RISC-V Instruction Set Architecture (ISA), which is well-known for its flexibility, modularity, and openness. RISC-V, which makes use of the RISC design principles, opens the door for creative multi-core solutions, such as the use of dual-core processors in the proposed design. The capacity of dual-core processors with RISC-V ISA to greatly improve system performance and independent multitasking skills is one of its main benefits. These processors can run and processes concurrently because they have multiple processing cores on a single chip, which increases system responsiveness and efficiency. Additionally, dual-core processors provide improved flexibility and scalability. This project includes 5 stage pipelined architecture with a modified execution stage which is capable of superscalar execution of up to 3 integer operation in parallel, the quest for efficiency and performance has resulted in the creation of complex execution units that can carry out parallel execution at once. The most advanced of them are superscalar execution units, which allow several instructions to be executed simultaneously in a single clock cycle. Superscalar execution units, when combined with pipelined RISC-V processors, provide improved performance increases by utilizing instruction-level parallelism and optimizing processor throughput.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Then finally the Risc v processor is implemented onto FPGA board which is Driven by Xilinx's ZedBoard, ZedBoard is a flexible platform for embedded system prototyping that combines programmable logic fabric and the power of a dual-core ARM Cortex-A9 processor. For the purpose of creating applications for Xilinx FPGAs, SoCs, and ACAPs (Adaptable Compute Acceleration Platforms), lastly Xilinx developed the Vitis IDE which developers may use to design, create, and implement applications that take advantage of the programmable logic fabric on the Zynq device when utilizing the ZedBoard, a well-liked development board with the Xilinx Zynq-7000 SoC. With Vitis IDE, application functionality may be described in C, C++, or OpenCL by developers and automatically translated into resource- and performance-efficient hardware implementations. This implementation of soft-core FPGA makes it possible to quickly prototype and investigate hardware concepts without needing to manufacture HDL design like in the case of ASIC.

2. Architectural Design

The processor in this paper, is a 32-bit, in-order superscalar processor with two 5-stage pipelined processor. Thedesign uses the RV32IM extension of RISC-V ISA. This architecture is focused on improving the throughput by incorporating additional functionalities to the processor by implementing superscalar ALU unit. The design is implemented and the complexities that come with this is managed such that utilization of resource in FPGA is minimized. The processor in the design is dual core and has a separate instruction memory, which helps the processors work independent to each other.

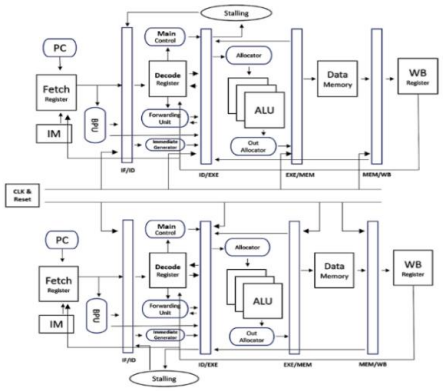


Fig 1. Proposed Processor Architecture

Proposed Processor Architecture is pipelined for 5 stages
The pipeline stages are:
Fetch Stage (IF Stage).
Decode Stage (ID Stage).
Execute Stage (EXE Stage)
Memory Stage (MEM Stage).
Write-back Stage (WB Stage)

3. METHODOLOGY

3.1 Instruction Set Architecture

This project's ISA is a reduced version of the RV32IM extension. The instructions available in RV32IM are based on the basic instruction set from "Computer Organization and Design RISC-V Edition [2].

R-type	ADD, SUB, MUL, AND, OR, XOR,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

	SLL, SRL
I-type	LW, ADDI
S-type	SW
B-type	BEQ, BGE, BLT

Table 1. The supported instructions

R-type: For arithmetic and logic operations involving two source registers and one destination register, RISC-V R-type instructions are typically utilized. Bitwise AND/OR/XOR, comparison, subtraction, and addition are common operations included in these instructions.

I-type: I-type instructions have one source register, one immediate value, and one destination register for use in immediate arithmetic and data transfer operations. These instructions are frequently used for tasks like immediate operand arithmetic, loading constants into registers, and employing immediate offsets to access memory addresses.

S-type: Data can be stored from a register into memory by using S-type instructions, which make memory store operations easier. One source register, one base register that indicates the memory address, and an instantaneous offset that indicates the memory offset are all involved in these instructions. Data is frequently stored into memory regions with a fixed offset in relation to a base address using S-type instructions.

B-type: Branch instructions, also known as B-type instructions, enable the processor to branch to a target address in response to the outcome of a comparison operation. They are utilized for conditional control transfer activities.

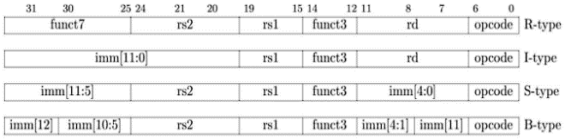
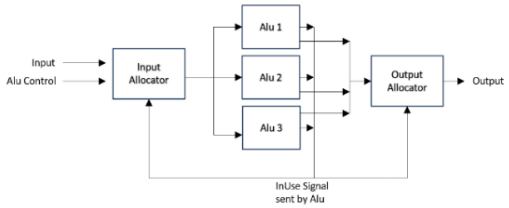


Fig 2. Risc V instruction format

3.2 Superscalar Execution Unit

The Simplicity and parallelism come together in an In-Order Superscalar Execution Unit designed for RISC-V processors. This execution unit adheres to the in-order execution model characteristic of RISC architectures, but it exploits the power of instruction-level parallelism (ILP) in a way that traditional scalar processors do not: it executes instructions sequentially. RISC-V In-Order Superscalar Execution Units provide significant performance increases without compromising the RISC design's signature architectural simplicity and elegance by carrying out several instructions concurrently in a disciplined, in-order manner

Up to three integer instructions can be issued simultaneously with this design. In order to set up the multiplexers for data forwarding in the EXE stage, the decode stage decodes the instruction and generates determine signals. Arithmetic procedures (addition, subtraction, multiplication, division), logical procedures (AND, OR, NOT), data movement (load, store), and control flow (branching, jumping) are a few examples of these instructions. resulting in higher throughput compared to a scalar exe unit



International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Fig 3. Superscalar Exe unit Architecture

3.3 Implementation onto FPGA

Software tool Vivado is used for implementation, this software Vivado is produced by Xilinx, a major supplier of Field-programmable gate arrays (FPGAs) and programmable logic devices (PLDs).

System on Chips (SoCs) and Xilinx FPGAs are the main platforms for which Vivado is used for designing, developing, and debugging digital systems. Specifying any other limitations that are necessary for the design, including pin assignments, clock limitations, and I/O standards. Throughout implementation, these limitations ensure proper timing and functioning. Generating the bitstream, which is the binary file that holds the FPGA's setup information. The FPGA is programmed and configured for using design utilizing this file. FPGAs and SoCs are programmed and debugged using the Hardware Manager. It enables users to utilize on-chip debug features to do real-time debugging and download the resulting bitstream onto the Zed Board the board upon which the Risc V processor is implemented onto.

Fig 4. Floorplanning Layout of FPGA after implementation.

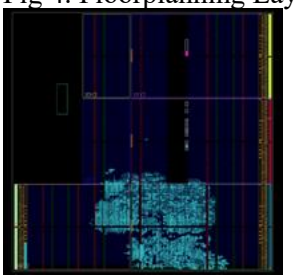


Fig 5. Blue-led light on the FPGA signifies the completion of Implementation.

Implementation of Application

Vitis™ Unified Software Platform, the software used after implementation which is designed to speed up the production of accelerated applications and embedded software for Xilinx platform, in the context of software programs: Vitis's integrated developmentenvironment (IDE) can be used to write C/C++ application. Example of a simple Traffic light application project being used after programing the ZedBoard after programing it using the Bitstream file of RV32IM Dual Core processor.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

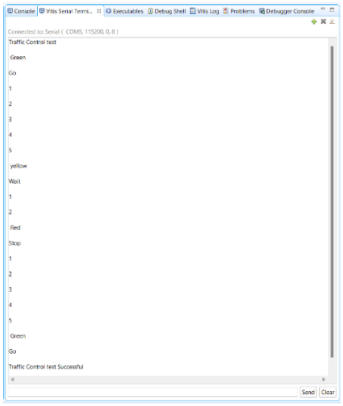


Fig 6. Simple Traffic Light Application on Vitis IDE

4. Results

The processor operates on a clock frequency of 300Mhz with all timing constraints is implemented on an AMD Xilinx Zynq®-7000 All Programmable SoC FPGA based board, and the proposed architecture was written with Verilog and synthesis, implementation was done using Vivado software tool.

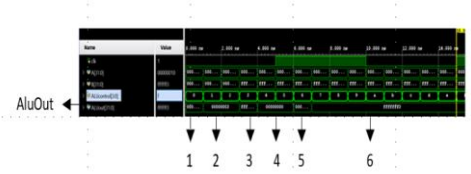


Fig 7. Behavioral Stimulation of Scalar EXE unit Which uses 1 ALU unit.

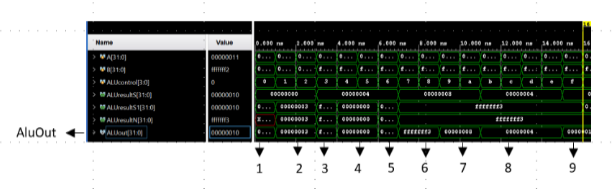


Fig 8. Behavioral Stimulation of Superscalar EXE unit. Which is capable of using 3 ALUs parallelly.

From the fig 5 it can be observed that a Scalar Exe Unit capable of executing 6 Alu outputs from 0.00ns to 16.00nsand from fig 6 it can be observed that a Superscalar Exe unit is capable of executing 9 Alu outputs from 0.00ns to 16.00ns superscalar EXE unit increases throughput as a high throughput suggests that the processor can effectively handle operations, which improves performance all around.

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	26.312 ns	Worst Hold Slack (WHS):	0.104 ns	Worst Pulse Width Slack (WPWS):	15.250 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	1041	Total Number of Endpoints:	1033	Total Number of Endpoints:	487

All user specified timing constraints are met.

Fig 8. Summary of Timing analysis.

Resource	Utilization	Available	Utilization %
LUT	5214	53200	9.80
LUTRAM	178	17400	1.02
FF	3466	106400	3.26
BRAM	2	140	1.43
DSP	6	220	2.73
IO	3	200	1.50
BUFG	12	32	37.50

Fig 9. Netlist propertyresource used inFPGA.

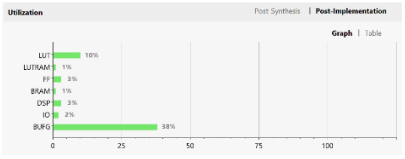


Fig 10. Resource utilization Report of Zed Board FPGA afterimplementation.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

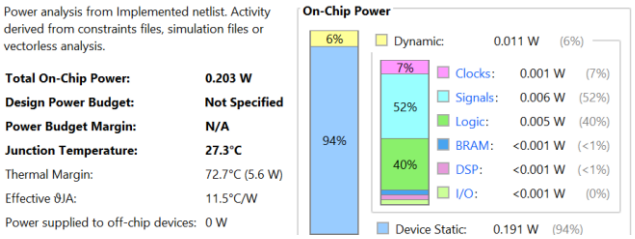


Fig 11. Power Analysis using Vivado

From fig 9 and 10 the proposed processor has a lower utilization, which conserves FPGA resources, allowing for the integration of additional functionality or future design improvements without demanding a larger FPGA. Which can show that the soft-core design can also be used in enabling the selection of smaller, less costly FPGA versions while preserving power efficiency as observed in fig 10, it also helps reduce costs.

5. CONCLUSIONS AND FUTURE SCOPE

In conclusion, the implementation of design onto FPGA process includes critical components such as timing, power analysis, and resource usage which makes it a preferable alternative to ASIC manufacturing,specifically for prototype or low-to-medium operation production. A 32-bit, dual-core, five-stage pipelined in-order superscalar architecture has been developed on Verilog using Xilinx Vivado 2022.2 and is implemented on AMD Xilinx Zynq®-7000 All Programmable SoC FPGA based board.The CPU has a low power consumption of about 0.203 W on-chip and is implemented on ZedBoard FPGA. Superscalar Execution Unit has improved throughput compared to scalar architecture. Integration of small-time application such as the traffic light system serves as a tangible demonstration of the potential applications achievable with the Vitis IDE and Zed Board or any other FPGA. Furthermore, the possible integration of renewable solar technology with soft-core implemented FPGA-based system presents several opportunities, such as the implementation of small-scale applications like a traffic signal system. The design can run independently on solar power, even in isolated or off-grid areas, with careful power management and optimization. Soft-core implemented FPGA technology's scalability and flexibility make it possible to seamlessly integrate new features or improvements in order to meet changing needs with respect to manufacturing ASIC Chips. For the future of this design, it can be configured to support Out of Order and Dual Issue Instructions, which can be incorporated into this core to further enhance performance.

6. REFERENCE

1. David A. Patterson and John L. Hennessy “Computer Organization and Design RISC-V Edition: The Hardware Software Interface.” Page 22 Morgan Kaufmann Publishers Inc., 2017, San Francisco, CA, USA.
2. Andrew Waterman, Krste Asanovi, and Five Inc. “The RISC-V Instruction Set Manual Volume I: User-Level ISA Document Version 2.” CS Division, EECS Department, University of California, Berkeley 2017.
3. L. Poli, S. Saha, X. Zhai and K. D. Mcdonald-Maier, "Design and Implementation of a RISC V Processor on FPGA," 2021 17th International Conference on Mobility, Sensing and Networking (MSN), Exeter, United Kingdom, 2021, pp. 161-166
4. S. Hiremath, S. Chickerur, J. Dandin, M. Patil, B. Muddinkoppa and S. Adakoli, "Open-source Hardware: Different Approaches to Softcore implementation," *2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Shivamogga, India, 2022, pp. 76-83
5. Goossens, B. (2023). Setting up and Using the Vitis_HLS, Vivado, and Vitis IDE Tools. In: Guide to Computer Processor Architecture. Undergraduate Topics in Computer Science. Springer