IMPLEMENTATION OF PICORV-32 USING ASIC DESIGN

RITHESHWAR M
Department of Electronics and Communication Engineering
Vidyavardhaka College of Engineering
Mysuru, India

Dr. GEETHASHREE A
Associate Professor
Department of Electronics and Communication Engineering
Vidyavardhaka College of Engineering
Mysuru, India

R CHIRAG
Department of Electronics and Communication Engineering
Vidyavardhaka College of Engineering
Mysuru, India

Dr. C M PATIL
Professor and Head Department of Electronics and Communication Engineering
Vidyavardhaka College of Engineering
Mysuru, India

PRITHVI MK
Department of Electronics and Communication Engineering
Vidyavardhaka College of Engineering
Mysuru, India

Abstract— The VHDL hardware description language was used to create the core, which allows for the addition of customised, optimised hardware units for particular operation. The consequences on power consumption and area, and processing power are all investigated. RISC-V was created with the purpose of creating a practical ISA that was open-sourced, academically accessible, and royalty-free to use in any hardware or software design. A RISC-V processor core implementation for SoC design is presented.
In 0.6m and 0.13m technology, the distinctions between the hypothetical FPGA implementation and the standard cell realisations. The VHDL hardware description language was used to create the core, which allows for the addition of customised, optimised hardware blocks for particular operations. The consequences in terms of area and power are analysed.

Keywords: *RISC-V, VHDL, FPGA, ASIC*

Introduction

Deep sub-micron fabrication processes enables the design engineers to pack a lot of stuff into a single microchip, including microprocessors, memory, and interfaces. Hundreds of millions of transistors can now be integrated on a single chip, forming a parallel system/unit with thousands of particulars or components, thanks to the introduction of 100nm technologies.
Parallel processing is a promising solution for many applications, as new SoC devices must make more and more computing capability available while keeping power consumption and heat production to a low. Instead of a single high-clocked, high-power CPU, number of lower-clocked, the more energy-efficient processing blocks are constructed. This research focuses on development of a RISC-V processor core that may be used as a single or uni processor or as template for multi-processor SoC architectures. As previously stated, most SoC architectures have severe criteria for module size and power consumption. As a result, we've picked an

architecture that's well-suited to each of these requirements. The VLSI design flow is used as the guide for this project. The VLSI design flow is utilised as a project's guidance. Layout in physical design is the process of converting the optimised gate-level netlist into a geometrical representation of the design. A GSDII file is a binary file that contains cell references as well as cell geometry information. All tagged words, geometric forms of the cell, and other layout information are stored in the Graphical Database System (GDSII) file.
Low-cost hardware devices must have a range of components, including sensors, radio frequency (RF) communication modules, and security modules for safe data storage and transmission, in order to execute the function of reading, processing, and transmitting real-time data.

I. LITERATURE REVIEW

Dennis Agyemanh **[1]** Examining the hardware resources utilisation to ensure that it meets the device's specifications is the first step in selecting a synthesizable processor core for reduced-cost devices. There are multiple entities that are considered while choosing a processor.
1. Maximum Frequency- In a digital design, the maximum frequency is a key parameter that influences the circuit's performance. A design with a high maximum frequency is advantageous because it may be used in both low- and high-frequency applications without causing timing difficulties. Clocking challenges arise when a low-frequency design is used in a high-frequency application, and the design typically produces unwanted effects. Hummingbird, SCR1, Rocket, BOOM, and also Shakti-E all recorded frequencies below 0.1 GHz, with Shakti-E down to 0.13 GHz. All of the devices tested, including the SiFive E31,PICORV32 MRISCV, ORCA, Roa Logic RV12, recorded frequencies above 0.1 gHz, with the PICORV32 recording the highest at 0.2 GHz. PICORV32 has a greater maximum frequency than the other CPU cores, allowing it to perform at a higher level ideal for both low and high-speed applications.
2. LUT Utilization - The fabric of a configurable logic block is made up of LUTs, flip-flops, and adders (CLB). BOOM, Rocket, BOOM, and Shakti-E, all used over 5000 LUTs, with BOOM having the most at 50000. This accounts for 94.1 percent of the device's available LUTs. Developing BOOM would allow less LUTs for implementing other functions in IoT applications that require distinct functionality. SCR1, Hummingbird, ORCA, Roa Logic RV12, SiFive E31,PICORV32 and MRISCV all required fewer than 5000 LUTs, making them ideal for low-cost IoT devices. Only 904 LUTs were used by PICORV32, accounting for only 1.70 percent that is the total of LUTs in XC7Z020.
3. Use of Flip-Flops - Flipflops are the basic building units of synchronous circuit. The output of a flip-flop changes only at the rising or falling edge of a clock signal. The data flip-flop is one of the most commonly and widely used flip-flops (D). BOOM used 25000 flip-flops, accounting for 27 percent of all flip-flops accessible. few other cores used fewer than 3000 flip-flops, but PICORV32 utilised just 566, making it perfect for low-cost systems.
There are still many variables to consider, but we chose this one for our project because it allows us to complete it at a minimal cost.

The paper by Claire Xenia **[2]** It is critical to use optimised memory blocks for large memories when synthesising an ASIC design that comprises memories. Writing wrappers around customised memory blocks is a standard approach to handle customised memory blocks so that the design can be ported to a new technology by modifying the wrappers rather than having to rewrite the HDL code of the design itself. Due to their high area cost, large multiport memories are unlikely to be utilised in many FPGA systems. While such a memory should be more space efficient in an ASIC than in an FPGA, the cost of space will be prohibitive.
A read-only memory is another sort of memory that hasn't been considered yet. In FPGAs, whether a memory is read-only or not is rarely a problem (although some Altera FPGAs will implement small ROMs that are more efficient than small RAMs), but in an ASIC, the impact of employing a ROM rather than RAM will be significant. A 1024 16 bit RAM, for example, is nearly 7 times larger than a ROM of the same capacity in the ATC35 process. The disadvantage of employing read-only memory is that flexibility is lost.

G.Rajesh Babu **[3]** An application-specific integrated circuit (ASIC) is an integrated circuit (IC) that has been tailored to a specific application rather than being designed for general usage. ASICs are difficult to understand. Can have millions of transistors, making it impossible to complete the design only by schematic entry (and later custom layout).
The Full-custom flow and the Semi-custom flow are the two types of flow used in ASIC design. All custom logic cells and mask layers are included in the full custom flow. The manufacturing lead time is approximately 8 weeks, excluding design time. When there is no acceptable existing library, current libraries are not quick enough, or there are concerns with power or size, full-custom makes sense. Performance and affordability are two advantages of full bespoke design. The benefits come with certain drawbacks, such as design time, high risk, and complexity.

Standard cells, full-custom blocks, functional standard blocks cores, and other elements are used in a semi-custom flow. The "Cell-based design flow" methodology is used to design today's ASICs. Standard and large cells can be reused to save time and money. The only additional time required is for the standard library or cells to be designed, as well as time to produce all of the ASIC layers for the new design.

The paper by Clifford Wolf **[4]** Static random-access memory (SRAMs) are an essential component of any processor or integrated circuit (IC). However, hand-designing SRAMs takes a long time, but because they have a consistent structure, automation can quickly manufacture size and configuration variants. It is, however, difficult to apply these variants to a variety of technologies and instruments. Because memory design is so important to system performance and cost, optimization is equally crucial. These are ideal conditions for a memory compiler like OpenRAM to succeed.
OpenRAM is a memory compiler development framework that includes physical and reference circuit implementations. OpenRAM is written in Python and presently supports two processes: NCSU's Free PDK 45nm and MOSIS' fabricable SC1 MOS 0.35m (SCN4M SUBM).

Priya Singh**[5]** This study provided a thorough understanding of the PicoRV's architecture and functional pins. The RV32IMC ISA is implemented by the PicoRV32, a 32-bit RISC-V processor. PicoRV32, PicoRV32 axi, and PicoRV32 wb are the three variants of the core. PicoRV32 axi is a version of the PicoRV32 CPU with an AXI4-Lite interface, whereas PicoRV32 is the conventional PicoRV32 CPU. Finally, PicoRV32 wb is a Wishbone Master-compatible version of the CPU. Microprocessor Architecture.
The Arithmetic Unit (ALU), Booth's Multiplier, Control Unit, Register Bank, Memory, and Data Path make up the 32 Bit Asynchronous RISC-V CPU. There are five stages to the architecture. Instruction fetch (IF), Instruction decode (ID), Instruction Execution (IE), Memory Access (MA), and Write back are the steps of pipeline (WB).
There are three sorts of memories in the design: data memory, instruction memory, and register memory.
The 32 bit instruction is uploaded to Instruction register using the programme counter's location in the Instruction fetch stage (PC). Because each register only stores 8 bits and it takes four registers to hold a 32 bit instruction, the PC will increment by four addresses after the fetch cycle. The 32 bit instruction is separated in the Instruction Decode stage. Depends It decodes instructions based on the 6 bit opcode. The Opcode is delivered to the decoder, which subsequently generates the proper control signals. The ALU unit executes operations based on the decoded data at the execution step. The load and store instructions in Memory Access (MA) access memory, i.e. read and write operations on memory. The outcome of the ALU/Memory system is written back to the used or initially called register file in the Write Back step.

Instruction format
The instructions are divided into three categories. They really are.
1. Register (R) Format: The opcode is MSB5 bits long, Rs is 6 bits long, and Rt and Rd are each 5 bits long. Arithmetic and logical (ALU) operations are the most common uses for these Instructions.
2. Instantaneous(I) type: The opcode has six bits of MSB, five bits of Rs and Rt, and a 16-bit immediate address value. These are data transfer instructions, as well as immediate and conditional branch instructions.
3. Jump(J) type: This opcode has an MSB 6 bit word address and a 26 bit word address. For unconditional branch instructions, these instructions are utilised.
OP-The instruction's basic operation.
rs denotes the initial source operand register, rt denotes the second or the next source operand register, and rd denotes the destination register.
Shamt - the quantity of a shift.
Funct - Selects the opcode's specified variations.
The paper by C. Phipps**[6]** The table of contents provides details on the various factors that influence performance, unit costs, and communication methods.
The step-by-step procedures for fabricating at 45um are provided in this document.
The key idea is to look at the differences between 0.6um and 0.13um FPGA prototypes and ASIC implementations.
The parallel processing technique is adopted in this study, which means that instead of using a single high-clocked, high-power CPU, a group of lower-clocked, more power-efficient processing units is used. This study focuses on the development of a RISC processor that can be utilised as a single processor or as a template for multi-processor SoC architectures.
They created an S-core, which is essentially a RISC processor which is compatible with M-core architecture.
Motorola's M-core CPU is a low-power RISC processor designed for embedded system applications. M-core is based on the von Neumann architecture, which includes a common programme and data bus. FPGA prototypes are implemented utilising the RAPTOR 200 Rapid Prototyping System, which integrates all of the components

required for the realisation of circuits with system designs of complexity up to 100 million transistors using FPGA application-specific modules. They were able to attain a clock frequency of about 160Mhz with die size of 0.25mm by purchasing RAPTOR 200. In conclusion, we have an advantage over design verification and optimization by employing the FPGA prototype. We can obtain higher clock frequency performance, lower power consumption, and area optimizations by employing ASIC.

Lee Yong Key**[7]** Lower power consumption is necessary for longer battery life in mobile and handheld applications, as well as for low carbon footprint requirements in networking and storage systems. Clock power in future 45nm devices is expected to increase exponentially and make up for 65-75 percent of chip power. Because power is proportional to both voltage and clock frequency, this is the case.

Clock gating is one of the most prevalent techniques for reducing dynamic power. The clock signal is provided to the underlying sequential elements of a logic unit when it is clocked, regardless of whether they will toggle in the next cycle.

Designers often incorporate clock enabling signals during the system and clock design phases, when the interdependencies of the various functions are completely known. The interdependencies between the states of individual flip-flops are dependent on autonomously synthesised logic, making specifying such signals at the gate level, particularly in control logic, extremely difficult.

The clock gating approach was created to avoid inefficient power consumptions, such as the power wasted by timing components when the system is inactive. In the case of flip-flops, clock gating is the process of turning off the clock signal when the incoming data does not modify the stored data. It can be used to put the entire functional unit into sleep mode at the system level, or to put chosen parts of the circuit into sleep mode while the remainder of the block continues to work at the sequential/combinational circuit level.

Intelligent clock gating is a collection of methods that can detect and eliminate superfluous switching in a design. This totally automated solution adds a little bit of logic to the design to suppress and minimise non-essential activities, lowering the device's power consumption. A lot of industry sign-off tools, such as Cadence SOC Encounter, Altera, Xilinx, and others, have recently added an intelligent clock gating option to optimise the design's power usage. It is crucial to note that in such instances, the designer may not always be able to achieve the necessary amount of power reduction. As a result, to further reduce the dynamic, the designer might have to add possible clock gating approaches outlined previously at the RTL level.

II CONCLUSION

ASICs are difficult to understand. Can have millions of transistors, making it hard to complete the design at a single level of abstraction, such as by schematic entry or custom layout!

The contrasts between a normal cell realisation and a prototypical FPGA implementation in 45m and 90m technology, respectively. The VHDL hardware description language was used to create the core, which allows for the addition of customised, optimised hardware units for particular function. The consequences on dimensions, area, power/energy, consumption, and processing power are all investigated.

When it comes to the S-application Core, it's evident that the given processing block is best suited for SoC designs and, also a good fit for a lot of multiprocessor implementations. With 11 percent open opcode space designer will have ample architectural headroom for a few hardware-accelerated operations. As a consequence, the core processor is appropriate for applications that requires a lot of processing. We intend to create a SoC device with many S-Core units which would eventually become a multiprocessor.

The PicoRV32 processor is used to place and route OpenRAM SRAMs. It uses OpenRAM's.lib and.lef files, adds write masking, and proves that OpenRAM SRAMs can be placed and routed with a tiny processor. We achieve a design flow for the synthesis, placement, and routing of OpenRAM memories in a Verilog design using these adjustments. Additionally, write masking allows OpenRAM to support RISC-V processor memories. The PicoSOC with SRAM memories final configuration in Innovus takes up less space than the flip-flop equivalent. Furthermore, both architectures can meet setup and hold timing, though at different clock periods.

One of the most effective power optimization approaches for VLSI circuits is clock gating. Traditionally restricted to the synthesis, placement, and routing stages, power optimization has made its way up to the System level and RTL levels. Clock gating can thus be used by hardware designers to turn off inactive areas of the systemhence minimize the overall usage of dynamic power.

III REFERENCES

1. Dennis Agyemanh Nana Gookyi, Kwangki Ryoo Selecting a Synthesizable RISC-V Processor Core for Low-cost Hardware Devices Dept. of Information and Communication Engineering, Hanbat National University, Daejeon, Korea,

2. Claire Xenia, PicoRV32 - A Size-Optimized RISC-V CPU yosys -- Yosys Open Synthesis Suit 2012 – 2020.

3. DESIGN OF 32 BIT ASYNCHRONOUS RISC-V PROCESSOR USING VERILOG
G.Rajesh Babu, Asst.Proff. ,ECE Department, Usha Rama College Of Engineering and Technology , Telaprolu, JETIR March 2020, Volume 7, Issue 3

4. PicoChip: A full-chip ASIC implementation of the PicoRV32 "PicoSoC"
Tim Edwards, Mohamed Kassem,Clifford Wolf - Efabless Corporation San Jose, CA.

5. A comprehensive Power Optimization Technique for sequential Circuits, ISSN:2347-9817, Vol.2,Issue 2, Ver.2 June 2014 (IJARCST-2014)
Priya Singh, Ravi Goel Clock Gating.

6. C. Phipps, "The Early History of ICs at Texas Instruments: A Personal View," In IEEE Annals of the History of Computing, vol. 34, no. 1, pp. 37-47, Jan. 2012, doi: 10.1109/MAHC.2011.84.

7. Lee, Yong Ki, Herwin Chan, and Ingrid Verbauwhede. "Throughput optimized SHA-1 architecture using unfolding transformation." In IEEE 17th international conference on application-specific systems, architectures and processors (ASAP'06), pp. 354-359. IEEE, 2006.