A REVIEW ON ANALYSIS OF 32-BIT AND 64-BIT RISC PROCESSORS

Saroj Ashishkumar Ramprakash[1], Professor Dharmesh Patel[2]

[1]*PG Scholar, Department of Electronics & Communication Engineering, GTU, Dr. S. & S.S. Ghandhy Government Engineering College, Surat, Gujarat, India*
[2]*Professor, Department of Electronics & Communication Engineering, GTU, Dr. S. & S.S. Ghandhy Government Engineering College, Surat, Gujarat, India*

Abstract– *Reduced Instruction Set Computer or RISC is a microprocessor that runs small, similar messages running over the same amount of time. It's not yet a business right now and has raised some notable technology barriers. This is the main reason for the development of 32-bit and 64-bit RISC processes and concept tubes. In this article, our goal is to analyze the 32-bit and 64-bit RISC processor model and the independent sets of its use. RISC is a 32-bit processor based on non-Harvard architecture pipelines that contains separate CPU memory and instruction memory. RISC is a 64-bit processor pipeline that borrows from the architecture of MIPS implementation plans. GPR (general purpose register) processors and common frames (gateway, zero, etc.). The optimization algorithm will simulate the optimized multiplier model discussed and will attempt to optimize the data in such a way that arithmetic and logic operations consume more energy with a large amount of execution time. The document aims at a comparative study between the implementation models present in their installation and the operational elements, such as acceleration, power dissipation, etc. Each sample was designed and simulated and finally integrated into a high-level CADENCE TOOL module along with power analysis.*

Key Words: *RISC (Reduced Instruction Set Computer),32-bit, 64-bit, pipelining, Verilog, Cadence, delay, power analysis, operating frequency, system architecture*

1.INTRODUCTION

As the name suggests, the RISC microprocessor has a limited number of instructions that can be executed quickly because it is smaller and simpler. They require fewer transistors, making them cost-effective in terms of design and production. The idea behind the implementation is that RISC implements simple instructions rather than complex instructions that require complex design. The RISC instruction set consists of simple and basic instructions, while complex instructions can be realized by this combination of basic instructions. Because instructions are complete in a single cycle, this allows the processor to handle multiple instructions at the same time. Instructions are based on registers, so data transfer is done from one register to another. The instruction register is programmed to accept a combination of binary numbers that can be called operands or Opcode bits. There is an Opcode for each type of instruction that can be executed on the CPU in a particular classification. This makes the RISC processor instruction set easier to understand. [1].

The development of advanced digital systems around the world has created a huge demand for fast, comprehensive processors. Processor performance has greatly improved since its inception in 1970. As the name suggests, RISC microprocessors have a limited number of commands that can be executed quickly because they are small and simple. They require a small number of transistors which makes them inexpensive in terms of design and production. Effective processors now have a major impact on the commercial market.

The high growth rate of the processor is possible due to impressive technical advancements in computer architecture, circuit design and manufacturing processes. various processor structures have been upgraded and achieve to optimum performance. The RISC philosophy appeals to a large extent to microprocessor designers. Most of the computer engineers employed today in various fields such as servers, networking, and signal processing are based on the RISC philosophy. Multi-functional requirements, Multiple user resources in high-performance systems, 32-bit generic processor architecture is based on RISC philosophy.

1.1 The System Architecture of A 32-bit RISC Processor

The Harvard architecture-based design of 32-bit RISC the processor shown here incorporates 8 general purpose registers, a basic arithmetic logic unit (ALU) for operations such as addition and change operations, data memories and an instruction set of 14 instructions. This supports the load storage architecture where's all operations are done in the records. Since it's not a pipeline processor, it is much easier to understand and implement.

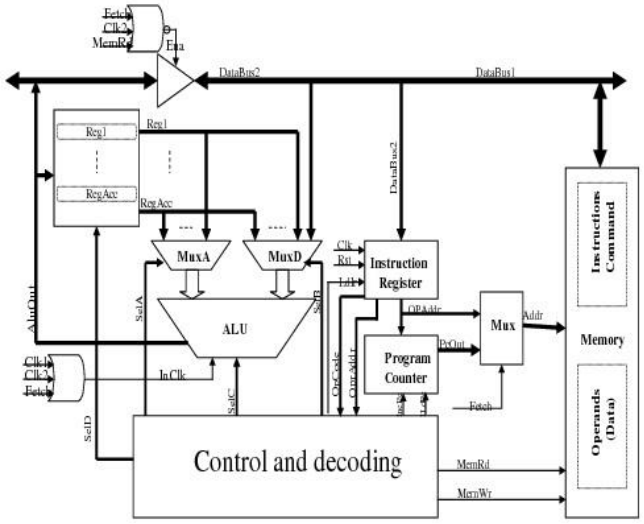

Fig-1: Architecture of 32-bit Pipelined RISC Processor[2]

Fig.1 offers a complete view of the 32-bit RISC processor. The architecture mainly consists of 5 logical blocks [3]. These are the Control Unit, Arithmetic Logic Unit (ALU), Execution Unit, Program Counter (PC), and Memory Unit. The register file contains all the operands that are retrieved from the memory unit and also with the appropriate signal from the control unit, the operations are executed in the ALU and each read/write operation is executed and the program counter goes to the next instruction.

A. Program counter:

The purpose of the 32-bit button is to take the address Subsequent instructions are present in memory. Every time an executed instruction, executed program counter and more instructions are loaded for execution. But until then the first second executed instruction could not be loaded. Is connected to the control unit as most of the instructions to operate the PC is present in the instruction set. But in at the same time, it also complicates the hardware, creating more power loss in the circuit then it is suitable for load/save architecture[4].

B. Arithmetic and Logic Unit(ALU):

The ALU executes most of the instructions in the instruction set. The ALU is controlled by the ALU Output signal from the ALU control unit. It implements arithmetic operations such as addition and subtraction, logic operations such as AND, NOT (inversion) of binary numbers, and rotation operations which are useful during signal processing operations such as correlation [5]. Depending on the type of operation, the respective number of registers/data elements is used. While the basic addition operation will use 3 registers (2 source and 1 destination), the immediate addition uses only 2 registers (1 source and 1 destination) along with a data element in binary available for calculation directly in the Opcode [6]. The impact of the operations is applied to the output of the ALU and the flags present indicating any by-product of the operation performed.

C. Control Unit:

The control unit acts as the processor brain and receives stored in memory. Based on that, it toggles on/off the signal that determines which operation to perform.

The instruction type format belongs below;

1. Memory Access
2. Data Processing
3. Branch
4. Jump

Memory Access : Load

Opcode	Source Register Rs	Destination Register Rd	Offset
4	3	3	6

Memory Access : store

Opcode	Source Register Rs1	Source Register Rs2	Offset
4	3	3	6

Data Processing

Opcode	Source Register Rs1	Source Register Rs2	Destination Register Rd	Offset
4	3	3	3	3

Branch

Opcode	Source Register Rs1	Source Register Rs2	Offset
4	3	3	6

Jump

Opcode	Offset
4	12

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycles}} \times \frac{\text{cycles}}{\text{instructions}} \times \frac{\text{instructions}}{\text{program}}$$

Fig-2: Performance Equation of Processor

Though the 32-bit RISC processor is pipelined yet it is able to execute simple instructions effectively and overcomes the shortcomings of the traditional CISC processor where the number of instructions are minimized at the cost of CPI (cycles per instruction). RISC follows a different strategy by reducing the CPI at the cost of the number of instructions per program. This is depicted in Fig. 2. The next section discusses what the

62-Bit RISC processor brings in with increasing technological requirements.

1.2 The System Architecture of a 64-bit RISC Processor

The 64-bit RISC processor is a 5-stage pipelined Harvard architecture-based implementation MIPS processor that has been a commercial success. A 64-bit machine outperforms a 32-bit machine because it can not only address twice as many unique memory addresses, but can also access data in memory in larger "chunks". The upside is that, all other things being equal, a 64-bit computer will be functionally a little less than twice as fast as its 32-bit counterpart.

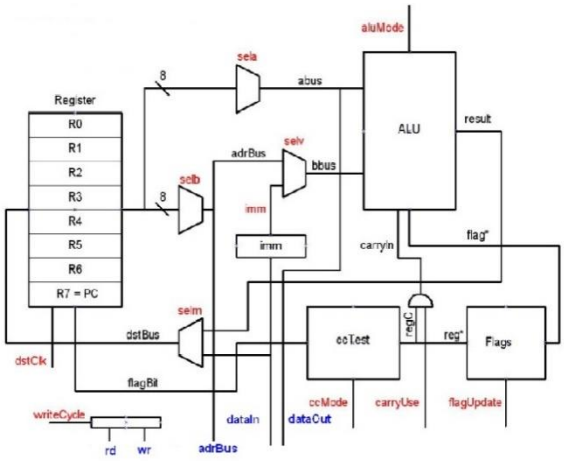

Fig-1: Architecture of 64-Bit RISC Processor

A. Pipelined

As the name suggests, it allows to store and execute instructions in a sequential manner. The traditional CPU solved the problem of excessive lag because no other task was performed when one of them was being processed. This is a scenario similar to that of an operating system in an idle state when an I/O device is functional. This causes excessive delays. This problem is solved by the principle of pipelining that allows parallel execution, i.e. while Opcode 1 is fetched and executed, Opcode 2 can be fetched and decoded, leading to parallel execution of multiple instructions simultaneously.

The processor receives the following advantages due to pipelining:

- Processor cycles per instructions (CPI) are reduced while speed (in theory) is increased by a factor of the number of pipeline stages.
- Pipelining also reduces the delay between completed instructions, which is called throughput. This is because multiple instructions are processed at the same time, which reduces the average time it takes to complete an instruction.
- Pipelining allows to include more complex instructions in the ALU, since the design is much faster than the previous one. This also increases the scope of applications in which such a processor can be used. For example, DSP (digital signal processing) applications such as convolution.
- Pipelining allows CPUs to run at a higher frequency than RAM, greatly increasing the overall performance of the computer. This is because the net combinational circuit has been simplified, reducing the net time period (delay).

Due to the availability of more general-purpose registers and the increased length of the Opcode, 6 bits are used to define an instruction type, which means that up to 63 instructions can be executed. The proposed 64-bit MIPS architecture contains complex instructions, such as multiplication, comparison instructions, such as register content equal to zero.

There are majorly 3 types of instruction encoding present in this processor.

1. R-type instruction encoding where the ALU operates only on the source and destination register. The 5 bits from 10 to 6 contain the amount of change to shift and rotate, while the last 6 bits from 5 to 0 contain the Opcode extension (function) to perform additional functions.
2. I-type instruction encoding where the ALU operations are performed on one source register data and on direct data (specified in the Opcode) and the result is stored in the destination register. The last 15 bits are for the direct data on which the operation is to be performed, as specified by the user.
3. J-type command encryption where command encryption is used for flight commands that change the flow of command execution on the processor and move the program sequence to a specific memory address. Here the instructions are 32 bits. The first 6 bits from the left have an Opcode of the operation to be performed, while the next 26 bits contain the direct data address to which the software string is to be transferred.

C. Flag Register

Flag registers are available in the 32-Bit RISC processors as well but their reach and usage are limited due to simpler computations and a limited instruction set. When an instruction is executed, the

output of the ALU fetches a biproduct which can be evaluated by the changes in the flag register. 5-Bits are being used out of 64- Bits.

1. Parity Flag (PF)
   The 1st bit of the flag register is occupied by PF whose objective is to count the number of 1s in the result. If the number of 1s is even then PF is set (1) else it remains reset (0).

2. Zero Flag (ZF)
   The 2nd bit of the flag register is occupied by ZF whose objective is to determine whether the result of the ALU or the instruction (BEQZ, BNE etc.) is zero or not. If the result is 0; the ZF is set else it remains reset.

3. Sign Flag (SF)
   The 3rd bit of the flag register is occupied by SF whose objective is to determine whether the result (signed number) obtained is within the range of signed numbers. The MSB of the result is used for this purpose. If it is 1 then the number is negative while if it is 0 then the number is positive.

4. Auxiliary Carry Flag (ACF)
   The 4th bit of the flag register is occupied by ACF which denotes the carry generated every time an arithmetic operation (addition, subtraction) takes place (except for MSB). This stores the extra generated carry or borrow and is used for computation bit by bit and keeps updating itself once used.

5. Carry Flag (CF)
   The 5th bit of the flag register is occupied by CF which stores the carry generated by MSB or the borrow generated for MSB. If there is a carry/borrow, the CF flag is set else reset.

The previous studies related to 32-BIT RISC and 64-BIT RISC Processor is discussed as follows:

BHARAT BHUSHANET al. (2021) proposed to compare the parsing performance between 16-bit RISC processor and 32-bit RISC processor. They experimented on this article and proved the excellent performance of the XILINX ISE Design Suit model. By analyzing all the data, they can reduce cost, cycle time and operating power. The proposed new schematic architecture demonstrates better performance without introducing excessive advances in simulation models. Statistics show that the complexity of the instructions is lower. Through a detailed study of the processor makes a choice between the two processors based on the requirements of the application. By simulating similar programs on both processors and then testing them on different fronts. For this, a simple adder is implemented on a 16-Bit RISC processor and then on a 32-Bit RISC processor. While the power analysis highlights the power consumption of each processor, the analysis of the maximum operating time and frequency is available. Emphasize the extent to which the minimum time period and joint delay are affected.[1].

M. KAMARAJU et al. (2019) Improvements have been made to the 6-stage process pipeline of the RISC processor. There are the take Instruction step, the decoding Instruction step, the register memory read access step, the data memory step and the write back step. Instructions are carried out consciously and systematically at all stages. Each step performs predetermined tasks and contributes to all tasks. Registers between each step help buffer during the pipeline process. The time allocated for each step is indicated as a clock cycle. If something doesn't fit, danger can arise. The control unit generates a NOP (store) signal to prevent line flushing. Hazards are expected to be addressed using various hardware and software protection techniques. To minimize throughput, each element of the piping process is carefully selected and implemented. In addition, low power techniques, i.e. power gating, are used on all units in the pipeline to minimize electricity consumption. Likewise, balancing pipelining reduces latency and increases process speed.[2].

SHOFIQUL ISLAM et al. (2006) The entire processor is modeled in Verilog HDL. RTL layout syntax is controlled by LEDA tools. For the functional verification of the design, the processor is modeled in high-level languages - System Verilog [5]. The layout is verified both at the block level and at the top level. Block-level test cases are generated in System Verilog on a specific and random basis. For higher level verification, assemblers are written and the associated hexadecimal code is passed from the assembler to the RTL layout and model. The control module captures and compares the model signals and messages for signal values that do not match. For the completeness of the verification, several coverage matrices were reviewed. Formal design verification is also planned. The design is synthesized aiming at standard 0.13-micron cell technology using the Cadence PKS tool. The complete design along with all time constraints and optimization options are described using the TCL script. When the maximum time angle in the library is taken into account, the synthesizer reports the worst path time of 1.4 ns in the entire design. After synthesis, verilog netlist is functionally verified against the 'sdf' file generated by the synthesizer. [3].

PRIYANKA TRIVEDI et al. (2015) Proposed building design RISC, in which 2-stroke operation is realized by means of a piping system. Piped architecture is used which minimizes latency and increases speed and in new innovation 2 stage pipeline which acts on both positive and negative edge minimizes latency and increases

speed and also reduces instruction hang up. The entire architecture of the risk processor operates on cycle 2. The fixed size of the instruction makes it easy to pipeline the given instruction. The RISC processor has a flexible architecture. The clock gate technique is used to minimize power. This is the most popular method of reducing dynamic power consumption. The power is consumed by the combination rationale, whose properties change on each edge of the clock so that the portrait enters the Image and starts the clock. In the current research work, the scheme for the 16-bit RISC processor is presented, implemented for high efficiency and low consumption. It operates on max 100 MHz frequency range and 16-bit RISC processor which uses 2 stage pipeline which reduces latency and increases speed. An energy-efficient design technique called clock synchronization technique was used to reduce power consumption. This method drastically reduced the dynamic power to 0.071 W and the idle power was reduced to 0.149 W. The total power consumption to 0.220 W, which is much lower than the conventional one, and also the final latency was found to be 1.5 cycles.[4].

THABSERA SULTHANA et al. (2021) The 16-bit RISC processor with Harvard architecture and 4-step pipeline structure is implemented in one clock cycle. This design can work for portable devices such as laptops, mobile phones and tablets, etc. The RISC architecture was simulated and synthesized using a Cadence RTL compiler. The design was held at the Cadence SOC Encounter. In the future, this design bit could be increased to 64 bits. We can compare these RISC structures in different technology libraries, such as slow, fast, and usually with different technologies, so that the area, power, and delay of the designer can be improved. attenuating the background image effect on the detector The RISC processor was synthesized using Cadence RTL compilers with slow Cadence 45nm library technology. The results are visible in area, force and retardation. They perform pre-design steps that do the processor design for the netlist. Verilog synthesized netlist files and associated constraint layout files are imported into the Cadence SOC Encounter window to generate layouts based on automated standard cells, pre-CTS, post-CTS, placement, and routing. After writing, a Nano path sdf file is generated. Save web lists and Meeting designs. This is the last step in designing the layout.[5].

2. METHODOLOGY

In the final stage of a 3-stage pipeline architecture where not only arithmetic logic operations are performed, but also branch operations are performed by calculating branch addresses, load/store instructions are used to give and receive data from data memory. A shifter and multiplier are also available at this stage to perform shift /rotate and multiply/divide operations. The results stored in the bank register are hidden at this stage. Software tool for performing 3-stage pipelining through a suite of CADENCE TOOL architectures and instruction sets. For 32-bit RISC processors, reduce power and delay time by running algorithms and calculating power analysis. To make the processor core suitable for high-speed applications, in this article I propose a new implementation unit architecture with 5 pipelines. The processor core is modeled by Verilog HDL. In This processor there are two operating modes, user mode and supervisor mode (safe mode). Dependency-Resolver detects and resolves data threats in the pipeline. The execution unit is coupled to the instruction channel and the data channel. Both channels operate in parallel and communicate with external devices via a common Bus Interface Unit (BIU). The flow of external data through the instruction channel and the data channel is controlled by the respective state machine controller. The processor also has seven interrupt requests (IRQs) and one non-maskable interrupt input (NMI). The Exception Processing Unit handles interrupts and exceptions on the chip.

2.1 Datasheet:

The proposed model has incorporated a total of 20 instructions, which can be further expanded depending on the utility of the processor. It is obvious from Table 2 that there is scope for expansion in the proposed instruction set. In addition to the basics available in the instruction set, there are shift commands that are used to multiply/divide the contents of a register by 2. This is a better method compared to the traditional multiplication operation available in the instruction set to retrieve results when the data in a register must be multiplied by 2 powers.

Table. 2 Instruction Set

Instruction Set		
Opcode	ALU Operation	Instruction
000000	ADD	R Type: Add
000001	SUB	R Type: Subtract
000010	AND	R Type: AND
000011	OR	R Type: OR
000100	SLT	R Type: Set Less Than
000101	MUL	R Type: Multiply
000110	XOR	R Type: XOR
000111	INVERT	R Type: Invert

001000	LW	I Type: Load
001001	SW	I Type: Store
001010	ADI	I Type: Add Immediate
001011	SUBI	I Type: Subtract Immediate
001100	SLTI	I Type: SLT Immediate
001101	BNEQZ	I Type: Content not equal to 0
001110	BEQZ	I Type: Content equal to 0
001111	BNE	I Type: Contents not equal
010000	BEQ	I Type: Contents are equal
010001	J	J Type: Jump to Segment
011111	HLT	R Type: Halt
100000	SGT	R Type: Set Greater Than

Datasheets play a pivotal role as future technologies take shape. Learn how to start with purchasing data right, and the best way to master this area is to get your hands on the basic datasheets.

There

3. CONCLUSION

In general, a 32-bit processor costs less than a 64-bit processor, due to the fact that the internal data path is narrower, requiring fewer transistors to make it, so there is a fair amount of space available. This space can be used to house functions (chip memory, peripheral interfaces, etc.). This analysis describes the pipeline execution device architecture for the 32-bit RISC processor and the 64-bit RISC processor. The design runs at 700 MHz after synthesis with 0.15-micron technology. Currently, the preparation takes place through the back-end flow. A higher frequency leads to a greater number of operations during a cycle, further resulting in dynamic power consumption. In the case of arithmetic calculations, while the 64-bit processor can provide decent processing speed on a small system at minimal cost, they should be preferred in the case of applications requiring high efficiency, such as floating-point calculations. The 32-bit processor is now limited to a specific situational need, because working in 32-bit on the one hand increases code density compared to a fixed 64-bit format, instruction set performance and the ability to run 64-bit.bit processor to improvise a much better choice overall.

4. Future work

Our future work includes adapting the core infrastructure so that it can create multiple threads and effectively support network operations. It works well up to the entire organization plus the pipeline system. In the future of components, it could be a pipeline operation, but it could also be extended with a higher scalar design to make a much smoother command line. Although word length increased, pipeline effects still occurred, reducing connection delay.

Abbreviations –

RISC: Reduce Instruction Set Computer

Bit: Binary digit

CPU: Central Processing Unit

GPR: General Purpose Register

MIPS: Million Instructions Per Second

ALU: Arithmetic Logic Unit

HDL: Hardware Description Language

BIU: Bus Interface Unit

NMI: Non Maskable Interrupt

IRQs: Interrupt Requests

CF: Carry Flag

ACF: Auxiliary Carry Flag

SF: Sign Flag

ZF: Zero Flag

PF: Parity Flag

MSB: Most Significant Bit

ACF: Advance Communication Function

PC: Program Counter

CPI: Cycles Per Instructions

CISC: Complex Instruction Set Computer

REFERENCES

[1] Animesh Kulshreshtha, Anmol Moudgil, Abhishek Chaurasia, Bharat Bhushan; “Analysis of 16-Bit and 32-Bit RISC Processors” 2021 7th International Conference on Advanced Computing & Communication Systems (ICACCS).

[2] P. Indra, M. Kamaraju and Ved Vyas Diwivedi, “Design and Analysis of a 32-bit Pipelined MIPS RISC Processor” International Journal of VLSI design & Communication Systems (VLSICS) Vol 10, No 5, October 2019.

[3] Shofiqul Islam, Debanjan Chattopadhyay, Manoja Kumar Das, V Neelima, and Rahul Sarkar, "Design of High-Speed-Pipelined Execution Unit of 32-bit RISC Processor" September 2006 Annual IEEE India Conference

[4] Priyanka Trivedi and Rajan Prasad Tripathi, " Design & Analysis of 16 bit RISC Processor Using Low Power Pipelining" International Conference on Computing, Communication and Automation (ICCCA), 2015.

[5] Chandran Venkatesan, Thabsera Sulthana M, Sumithra M.G, Suriya M, "Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology" 2019 5 th International Conference on Advanced Computing & Communication Systems (ICACCS).

[6] G M am un B, Shabiul I. and Sulaim an S, "A Single Clock Cycle MIPS RISC Processor Design using VHDL".

[7] Sneha Mangalwedh, Roopa Kulkarni S, Kulkarni Y. Low Power Implementation of 32-Bit RISC Processor with Pipelining. Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems. 2017; 307-320.

[8] Iro Pantazi-Mytarelli; "The history and use of pipelining computer architecture: MIPS pipelining implementation" 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 3 May 2013.

[9] J.B. Dennis; G.R.Gao; "An efficient pipelined dataflow processor architecture" IEEE Supercomputing '88:Proceedings of the 1988 A CM /IEEE Conference on Supercomputing, Vol. I, 6 Aug. 2002.

[10] Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran. "16-Bit RISC processor design for convolution application", 2011 International Conference on Recent Trends in Information Technology (ICRTIT), 2011.

[11] Chandran Venkatesan, M. Thabsera Sulthana, M. G .Sumithra, M. Suriya. "Design o f a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology", 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019.

[12] Guang-Ming Tang; Pei-Yao Qu; Xiao-Chun Ye; DongRui FanR. Nicole, "Logic Design o f a 16-bit Bit-Slice Arithmetic Logic U nit for 32-/64-bit RSFQ microprocessors" IEEE Transactions on Applied Superconductivity, vol. 28, issue no. 4 - 31,Jan. 2018.

[13] S. P. Ritpurkar; M. N. Thakare; G. D. Korde; "Design and simulation of 32-Bit RISC architecture based on MIPS using VHDL" IEEE 2015 International Conference on Advanced Computing and Communication Systems, 12 Nov. 2015.

[14] Norman P. Jouppi, Improving Direct-Mapped cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers, Digital Equipment Corporation Western Research Lab, 1990 IEEE

[15] N. Sureka, R. Porselvi and K. Kumuthapriya, "An Efficient High Speed Wallace Tree Multiplier", 2013 International Conference on Information Communication and Embedded Systems (ICICES).

[16] Vivado Design Suite User Guide: Implementation (UG904).