2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)
Galgotias University, Greater Noida, UP, India. Oct 30-31, 2020

Design and Implementation of a 32-bit ISA RISC-V Processor Core using Virtex-7 and Virtex-UltraScale

Aslesa Singh
Amity University,
Uttar Pradeesh
aslesasingh@gmail.com

Neil Franklin*
Amity University,
Uttar Pradeesh
neilofranklin64@gmail.com*

Nidhi Gaur
Amity University,
Uttar Pradeesh
ngaur@amity.edu,

Paursuh Bhulania
Amity University,
Uttar Pradeesh
pbhulania@amity.edu

***Abstract*—In this paper, a 32-bit Datapath with RISC-V instruction set architecture based on RV32I CPU instruction set has been designed. Furthermore, through analysis of function and theory of RISC-V CPU instruction set, the processor has been optimized by designing a six staged folded pipeline core. Finally, the design has been examined on the current industrial standards FPGA boards. Synthesis and implementation on two Boards of Virtex family has been compared on the basis of Utilization reports and Power Reports. The comparison shows the more heavily configurable board i.e. the Virtex Ultrascale delivers better power – performance tradeoff at higher costs.**

Keywords— RISC, RISC-V, RTL, Datapath, Pipeline

I. INTRODUCTION

RISC-V is an open source ISA enabling a novel world of processor design through open collaborative domain, creating a pool of technology, with 32- or 64-bit address space, consisting of a small core Instruction set and a compressed ISA designed for special and standard purpose extensions. It is effectively a base ISA, which is present in any implementation, and optional extensions can be added for more functionality. The base integer is the same as the previous generation RISC ISAs but there is an absence of branch delay slots and support of variable length encoded instructions. Base is a reasonable target to the compilers, linkers, assemblers etc. or operating systems with supervisor level operations, because of a restricted minimal instruction set. The processor design is based on industry standard instruction set by RISC-V organization. It can be parameterized as 32- or 64-bit data. The processor designed in this project is based on the open source RV12 processor by RoaLogic. The processor is single core and based on RISC-V ISA version 2.2. [1] Here the design is focused on the core of the processor which is heavily configurable, has precise interrupts, single cycle execution, parameterized cache memory, size, architecture. The most important feature of the processor is:

- Optimizing folded 6 staged pipeline
- Optional Branch Prediction Unit

The Implementation of this processor is supposed to have a small silicon footprint. Gated clock design reduces power. Since the design is extensively configurable, it gives power, utilization and performance trade-offs making the core highly optimizable. This is the focus of this project where power and utilization of the design is being compared.

II. DESIGN OF THE CORE

A. Execution Pipelining

The processor comprises a six staged folded pipeline which is different from the basic pipelining of RISC. A classic RISC pipeline has five stages – Fetch, Decode, Execute, Memory Access and Write back. [2] This core design has six stages with an additional "Pre-Decode" stage (Fig.1) and is conveniently modified where the instruction fetch takes two cycles to encode 16-bit compressed instructions again and jumps or branches are predicted.

Fig. 1. Core Execution Pipelining

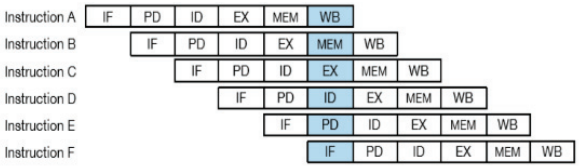

Fig. 2. Execution stages overlapping

The major modification in the execution pipelining is that the Execute and Write–Back stages are folded into the Memory Access Stage. The instruction bits are optimized by the Decode Stage to allow stalls and CPI of the processor. [3]

This modified pipeline overlaps the execution stages and executes one instruction per clock cycle, i.e. at the same time five operations can be performed. (Fig. 2)

The "classic RISC processor" consist of Instruction fetch, Instruction decode, Execute, Memory access and Writeback.

The target of 5 stages was to perform one instruction per cycle. Since increasing the pipeline increases the instruction throughput of the Central processing unit, work has been done on the six stages pipelined processor.

During decoding, the instruction determines whether the register from which the instruction needs to be decoded is being written to by a different operation in the execution stage, if this happens, the instruction is stalled by the Control

978-1-7281-6324-6/20/$31.00 ©2020 IEEE

unit by 1 clock cycle as well as in the FETCH stage so that no data is overwritten This is known as CPU stall.

Here, decode helps to hide the CPU stalls and increase the CPU cycle instruction by letting CPU stalls, mem access and execution to overlap.

The instruction stage also is succeeded by the pre-decode stage which is heavily based on the compression component of the RISC-V RV32I ISA. [4]

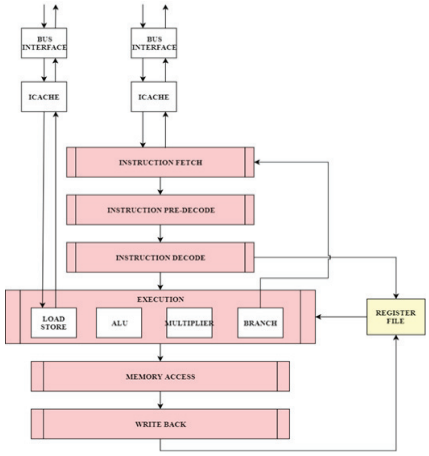

Fig. 3. Execution Pipeline

The pipelining (Fig. 3) in this core is as follows:

- Instruction fetch

The Fetch instruction piles up a package from the program memory. This package is a field that involves more than one instruction. The location of these (addresses) are known to the program counter. The Program counter can get refreshed at any point unless there is a CPU stall. It is either a 32-bit or 64-bit. After flushing the pipeline, the Program counter starts again fresh with the given default location

- Pre-Decode

The given instructions need to be first decoded into the RISC V instruction set from its 16-bit compressed state. After that, the program counter is processed, changing different instructions such as link, branch and jump hence the dependence on execution stage for waiting for the change is avoided and also reduces the need for flushing the pipeline. The address of the destination is predicted by either using an extension of branch predictor or on the basis of the offset value. [5]

- Instruction Decode

The instruction decode accesses registers, checks values in it, checks whether opcodes are correct, checks the immediate values and confirms that the operands are available for the executions.

- Execute

The Execution stage carries out necessary operations for the information given by former Decode unit stages and here is where the actual computation occurs. This stage has the Arithmetic and Logical unit and a bit shifter. This has different execution units and each has a novel capacity. The ALU is responsible for Boolean operations and also for integer arithmetic operations bit Shifter is for shifts and rotations. The Multiplier unit computes marked/unsigned increase. The Divider unit manages marked or unmarked division. The Load / Store Unit enables the data memory to load or store data. The Branch Unit figures hop and branch addresses and approves the anticipated branches. [6]

- Memory-Access

At this unit the data memory is accessed. The memory stage provided a gap for fetching and load/store to finish. Data, and control signals Memory accessing and address, being working during the execution pipeline stage are simultaneous. The memory stores signals and so forth completes the true access after memory read access is complete. The read data will not be up until a sole clock cycle later. This would happen after the Write-Back stage, and henceforth late for the results to be written. Therefore, the MEM stage is essential. [7]

B. External Bus Interfaces

1) AHB – Lite Memory using AMBA 3

The process core design uses an AHB Lite Memory. It is a memory access bus interface designed which has a very wide range of parameter support. All ports on the chip are supported by AMBA 3 AHB - Lite. The generic use of this allows physical implementation on a board with a controlled synthesis for hardware and simulation support according to the behavioural HDL.

The AHB-Lite Memory is a parameterized and configurable IP that allows a designer to connect internal device memory to AHB-Lite based hosts. The dimensions of the memory with a registered output stage, are set via parameters.

2) Interrupts

The processor here supports multiple various external interrupts and operates in concomitance with an external PLIC (platform Level Interrupt Controller).

Specific pins on the processor core provide the interrupt to the processor which then makes the identifier source interrupt by the PLIC at the efficient interrupt vector upon a call.

III. IMPLEMENTATION

The design has been examined on two different FPGA boards,

Virtex-7 (Product Part: xq7vx690trf1930-2I) and Virtex Ultrascale. (Product Part: xcvu440-flga2892-2-e)

The schematic diagram generated of the design on Virtex-7 (Fig. 5) and Ultrascale (Fig. 6) along with the RTL design of the core (Fig. 4) has been displayed.

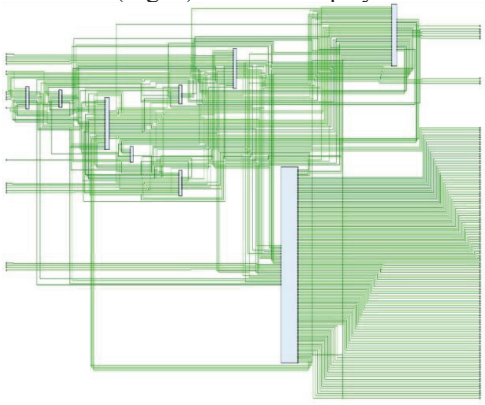

Fig. 4. RTL design of the core

Fig. 5. Implemented design of the core on Virtex-7

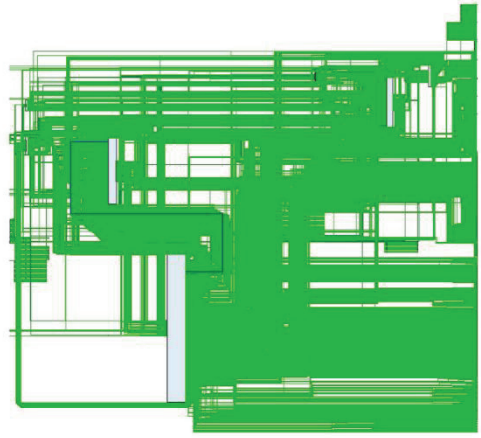

Fig. 6. Implemented design of the core on Virtex-Ultrascale

IV. RESULTS

The comparison of power reports for both the FPGA boards used in this Project – Virtex-7 and Virtex Ultrascale are with the same environment setting.

- Temp Grade: Industrial
- Ambient temperature: 25*c
- Board Selection: Large
- Number of Board Layers: 12 to 15 Layers

After a bit of Switching adjustments and toggle rate alterations, the power reports were generated in all the cases.

Power reports can be broadly classified into Physical Domain and Functional Domain. Physical Domain of power deals with the board shape, size, power deliverance, Thermal power dissipation system. Functional Domain is more based on the design i.e. the utilization of area, I/O signal interference etc. This bit theory is highly essential in understanding the core difference between the two boards used in this project.

The utilization reports have a great impact on the power usage or on-chip power of both the processors. The Power report taken out during synthesis explains the Power estimated to be used or dissipated whilst using a specific device or vendor.

Power reports have been monitored and refined in Synthesis of the project without a placement procedure. This is also known as Post Synthesis in Vivado.

The power report received after Implementation is called 'Post Placement' and is done when netlist components are actually placed into the FPGA board resources. This is more precise due to a detailed utilization report and configuration. This stage also verifies the routing and best- and worst-case gate and delays. Basically, power analysis is most accurate at this stage before testing on an actual board. [8]

The utilization reports have a great impact on the power usage or on-chip power of both the processors. The Virtex-Ultrascale processor is a huge board with a very powerful set of specifications. This highlights the immense I/O resource provided on the board, with 1456 I/O pins available but the design of the core takes up 941 pins only, giving a 64% utilization. Compared with this Virtex-7, the 7 has 1000 available I/O pins which gives an efficient 94% utilization. [9].

TABLE I. POWER OF THE CORE

Power Report Details	**Virtex 7**	**Virtex UltraScale**
Total on Chip Power	0.491 W	3.028 W
Junction Temperature	25.4 C	26.2 C
Thermal Margin	83.7 W	178.5 W
Dynamic Power	0.166 W	0.389 W
Signals, Logic, BRAM	0.018 W	0.014 W
I/O	0.148 W	0.376 W
Device Static Power	0.326 W	2.639 W

TABLE II. UTILIZATION OF THE CORE

	Virtex – 7			**Virtex - UltraScale**		
Resource	Utilization	Available	Utilization %	Utilization	Available	Utilization %
LUT	3309	433200	0.76	3284	2532960	0.13
LUTRAM	96	174200	0.06	96	459360	0.02
FF	2230	866400	0.26	2852	5065920	0.06
BRAM	0.50	1470	0.03	0.50	2520	0.02
I/O	942	1000	94.20	942	1456	64.70
BUFG				1	1440	0.07

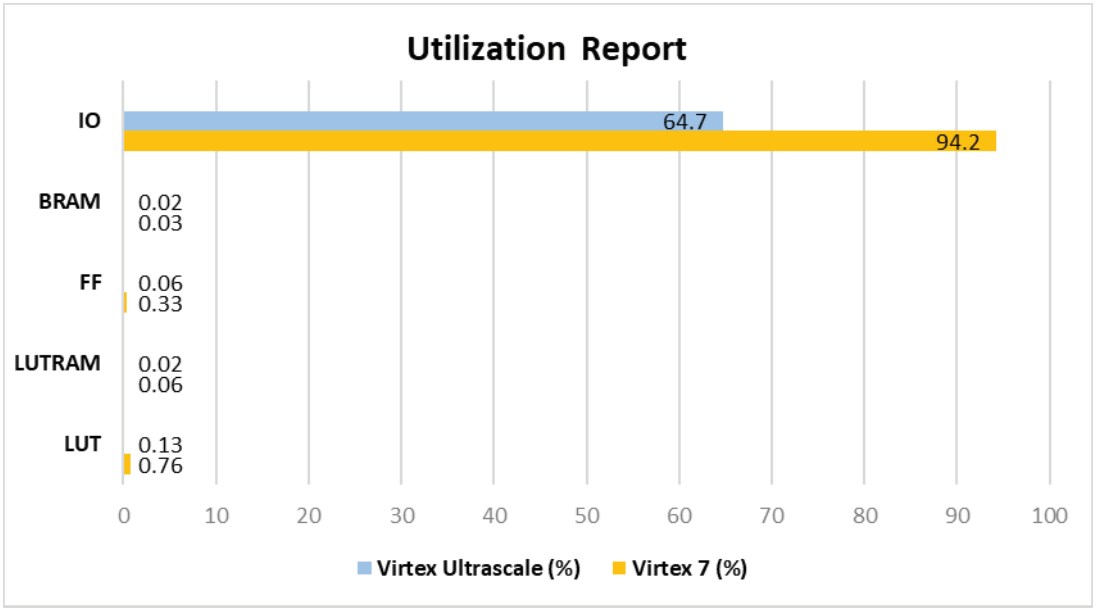

Fig. 7. Utilization report

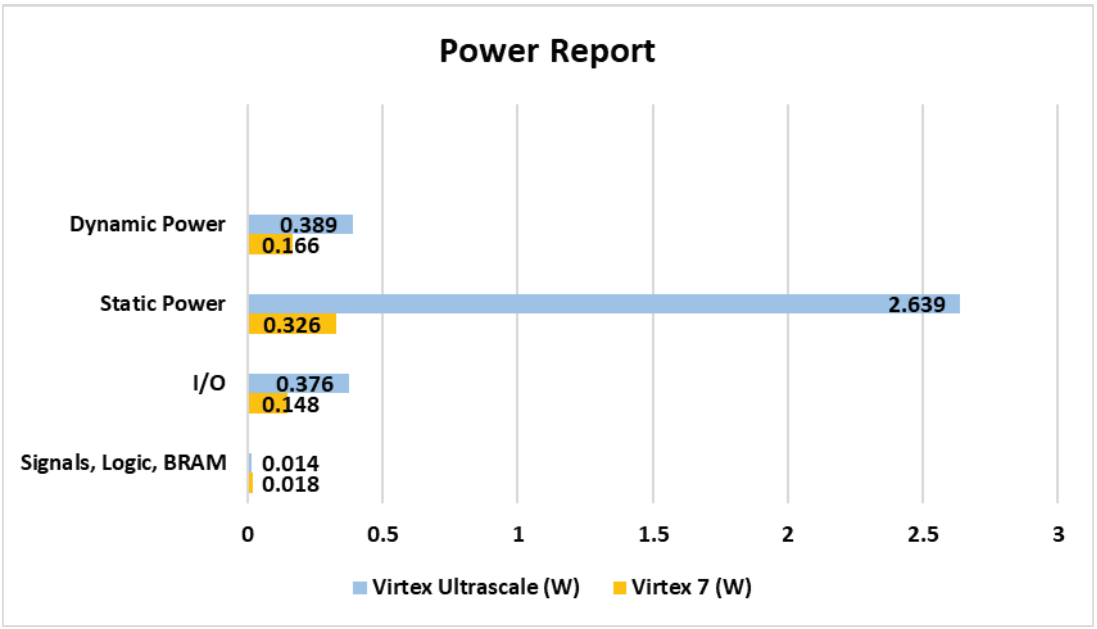

Fig. 8. Power report

V. CONCLUSION AND FUTURE SCOPE

The two FPGA Boards used here are of Virtex family provided by Xilinx product Vivado. Implementation on these boards were possible due to their size.

Even though the total chip power is greater in the case of Ultrascale, the dynamic power is significantly lower as compared to Virtex-7 which means that the Ultrascale board consumes low power when it is in dynamic mode. The reason can be summarized to the evident range of the design to be parametrized and hence consuming more power on a bigger board but essentially takes performance into consideration while functioning. Virtex-7 kit is comparatively less accommodating to the vast configurations and hence uses less static power.

The utilization reports give a clear picture regarding the future usage of this processor design as it is highly effective in terms of Power / Performance tradeoff as the design of the core is very much parameterized and user defined. The core has about more than a dozen options, those which are user defined and can exceedingly control to give more power efficiency or higher performance and usability. The core design has a dense Branch Prediction unit, Cache on both data and instruction side working seamlessly with an AHB lite3 bus, all optional and in the pure control of the user.

The RISC-V gives an open source environment to allow us to tweak to a great extent, giving a plausible error free placement on both the devices.

The design in the project can be also extensively used in the upcoming processor designs with a RISC-V ISA due to its unique design pipelining component and extremely parametrized core.

REFERENCES

[1] A. Waterman, K. Asanovic, "The RISC-V Instruction Set Manual" in User-Level ISA, Berkeley:SiFive Inc. and CS Division, EECS Department, University of California, vol. I, 2017.

[2] P. Maillard, J. Arver, C. Smith, O. Ballan, M. J. Hart and Y. P. Chen, "Test Methodology & Neutron Characterization of Xilinx 16nm Zynq® UltraScale+™ Multi-Processor System-on-Chip (MPSoC)," 2018 IEEE Radiation Effects Data Workshop (REDW), Waikoloa Village, HI, 2018, pp. 1-4.

[3] I. Kuroda, E. Murata, K. Nadehara, K. Suzuki, T. Arai and A. Okamura, "A 16-bit parallel MAC architecture for a multimedia RISC processor," 1998 IEEE Workshop on Signal Processing Systems. SIPS 98. Design and Implementation (Cat. No.98TH8374), Cambridge, MA, USA, 1998, pp. 103-112.

[4] K. Patsidis, D. Konstantinou, C. Nicopoulos, G. Dimitrakopoulos, "A low-cost synthesizable RISC-V dual-issue processor core leveraging the compressed Instruction Set Extension", Microprocessors and Microsystems, vol. 61, pp. 1-10, Sept. 2018.

[5] Mukherjee et al., A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor Proceedings of the 36th Annual International Symposium on Microarchitecture (MICRO), pp. 1-5, 2015S. Islam, D. Chattopadhyay, M. K. Das, V. Neelima and R. Sarkar, "Design of High-Speed-Pipelined Execution Unit of 32-bit RISC Processor," 2006 Annual IEEE India Conference, New Delhi, 2006, pp. 1-5.

[6] S. Islam, D. Chattopadhyay, M. K. Das, V. Neelima and R. Sarkar, "Design of High-Speed-Pipelined Execution Unit of 32-bit RISC Processor," 2006 Annual IEEE India Conference, New Delhi, 2006, pp. 1-5.

[7] A. Oleksiak, S. Cieślak, K. Marcinek and W. A. Pleskacz, "Design and Verification Environment for RISC-V Processor Cores," 2019 MIXDES - 26th International Conference "Mixed Design of Integrated Circuits and Systems", Rzeszów, Poland, 2019, pp. 206-209.

[8] D. K. Dennis et al., "Single cycle RISC-V micro architecture processor and its FPGA prototype," 2017 7th International Symposium on Embedded Computing and System Design (ISED), Durgapur, 2017, pp. 1-5.

[9] Y. Jing, C. Y. Meng, M. T. Reyes, J. O. Yang, P. F. Salinas and G. Tan, "Electrical diagnosis of temperature-dependent global clock failures using probeless isolation and pattern commonality analysis," 2012 19th IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits, Singapore, 2012, pp. 1-6.