Ben-Gurion University of the Negev

School of Electrical and Computer Engineering

Preliminary Design Review (PDR)

Implementing RISC-V Soft-core on an FPGA using Open Source Tools Only

Project Number:	**p-2025-033**		
Student:	**Guy Cohen**	**207881004**	**guyap@post.bgu.ac.il**
Supervisor:	**Dr. Guy Tel-zur**		
Submitting Date:	**17.11.2024**		

Table of Contents

1.Abstract

Implementing RISC-V Soft-core on an FPGA using Open Source Tools Only

1.I English version

This project aims to implement a RISC-V soft-core processor on the Ice Breaker FPGA (Lattice FPGA) using only open-source tools. A soft-core processor is a processor implemented using the FPGA fabric, in contrast to a hard-core processor, which is physically implemented as a structure within the silicon. The primary goal is to demonstrate that a fully functional processor can be developed without the need for expensive tools and commercial licenses. Open-source software, such as Yosys for synthesis and Gtkwave for simulation, will be utilized throughout the design and implementation process. After the processor is implemented, it will be tested by running encryption code to ensure its functionality. This project will showcase the feasibility of cost-effective FPGA development by utilizing only free, open-source tools for every stage, from simulation to debugging. The use of FPGAs for processor implementation offers a highly customizable and efficient platform, providing both educational and practical advantages in hardware design. Ultimately, the project will prove that open-source tools and FPGA technology can be effectively combined to develop a fully operational processor, offering a low-cost alternative to traditional development methods.

Keywords: RISC-V, FPGA, open-source tools, GTKWAVE, Yosys, Nextpnr, openocd, IceStorm, Icestudio ,soft-core processor, encryption code, Amaranth HDL.

1.II Hebrew version

פרויקט זה בא לממש מעבד RISC V בעל ליבה רכה על גבי רכיב FPGA של חברת LATTICE (דגם icebreaker) באמצעות שימוש בכלים חינמיים בלבד. מעבד בעל ליבה רכה הוא מעבד הממומש באמצעות שימוש בשערים הלוגיים הנמצאים בתוך שבב ה FPGA, בניגוד למעבד בעל ליבה קשה, המיושם פיזית כמבנה בתוך הסיליקון. מטרת הפרויקט היא להדגים כי ניתן לפתח מעבד אשר עובד במתכונת מלאה מבלי להשתמש בכלים יקרי ערך. כלים חינמיים כגון Yosys המשמש לסינתזה וGtkwave המשמש לסימולציה של המערכת, יבואו לידי ביטוי במהלך בניית הפרויקט. לאחר סיום פיתוח המעבד, הוא ייבחן על ידי הרצת קוד הצפנה אשר יבטיח את פעולתו התקינה. פרויקט זה יראה את היתכנות הפיתוח המוזל של FPGA תוך שימוש אך ורק בכלי קוד פתוח בחלקים השונים של הפרויקט, משלב הסימולציה ועד שלב הצריבה. השימוש ב FPGA למימוש מעבדים מציע פלטפורמה גמישה ויעילה מאוד, ומספק יתרונות לימודיים ופרקטיים בעיצוב חומרה. בסופו של דבר, הפרויקט יוכיח שניתן לשלב בצורה אפקטיבית בין כלים פתוחים

וטכנולוגיית FPGA כדי לפתח מעבד אשר פועל בצורה מלאה, מה שמציע אלטרנטיבה זולה לשיטות פיתוח מסורתיות.

<u>מילות מפתח:</u> RISC-V, FPGA , כלים חינמיים בלבד , GTKWAVE, Yosys , IceStorm, Nextpnr , openocd, מעבד בעל ליבה רכה, קוד הצפנה, Icestudio, Amaranth HDL.

2.Research Proposal

Problem:

The primary problem addressed in this project is implementing a processor on an FPGA using only free, open-source tools for all stages: simulation, programming, and debugging. implementing processors on FPGA using commercial tools remains expensive. These commercial tools provide high performance and extensive support, but they often require costly licenses, which increase the overall development cost. For instance, the cost of the cheapest license for synthesis software such as Xilinx Vivado is $2,995 [1].
The huge demand for IoT equipment makes low cost an important indicator for the sustainable operation of the entire IoT system [2].

Need:

Demonstrate the feasibility of implementing a processor on FPGA without requiring expensive software licenses.

Goal Definition:

The goal of this project is to successfully implement a RISC-V soft-core on an FPGA using only open-source tools, demonstrating the feasibility of cost-effective FPGA development without relying on expensive software licenses. Additionally, the choice of RISC-V as the processor architecture is revolutionary due to its free and open ISA (Instruction Set Architecture). Unlike traditional, proprietary ISAs, RISC-V allows developers not only to use open-source development environments but also to access complete processor implementations that can be downloaded and utilized at no cost. This approach provides a fully open-source pathway from the development environment to the system core.

Innovation:

Achieve a fully functional FPGA-based processor using only free, open-source software tools, showing that cost-effective FPGA development is possible.

Proposed Solution:

Use exclusively open-source software tools, such as GTKWAVE for simulation and Yosys for

programming and configuring the FPGA. GTKWAVE is basically a toolkit used for creating graphical user interfaces. It helps in viewing the VCD files [3].
Yosys, an open framework for RTL synthesis, provides extensive support for HDLs [4].
I will likely also use the following open-source tools: [5]

- IceStorm- tool for analyzing and creating bitstreams for iCE40 FPGAs.
- Nextpnr- vendor-neutral, timing-driven FPGA place and route tool.
- Icestudio- IceStorm-based visual editor for FPGA boards.
- Amaranth- provides an open-source toolchain for developing hardware.

Expected Results:
A functional RISC-V soft-core processor implemented on an FPGA, capable of executing light weight codes, such as basic encryption algorithms.

Chip and the development board [5]**:**

- **Lattice iCE40UP5k FPGA,** see Figure 1, Figure 2.
  - 5280 logic cells (4-LUT + Carry + FF)
  - 120 Kbit dual-port block RAM
  - 1 Mbit (128 KByte) single-port RAM
  - PLL, 2 x SPI, 2 x I2C hard IPs
  - Two internal oscillators (10 kHz and 48 MHz) for simple designs
  - Eight DSP multiplier blocks for signal processing such as audio synthesis and even software defined radio
  - Low power consumption ideal for battery-powered applications
  - Three 24 mA drive and 3 x hard IP PWM (can drive RGB LEDs and small motors)
- **Plentiful, fast storage**
  - 128 Mbit (16 MB) quad SPI double data rate (QSPI-DDR) flash
  - Example: can stream video to LED matrix
- **Lots of I/O**
  - Three pins (header) for RGB LED
  - Two on-board LEDs
  - One UART, RX pin and TX pin accessible via virtual (USB) serial port
  - One pushbutton
  - Two available Pmod connectors (16 x pins total)

- One breakaway Pmod (8 x pins)

- **Pre-wired, breakaway Pmod module**
  - Input and output user accessible and usable in your own hardware design
  - Five LEDs in a star pattern
  - Three push buttons
- **Capable enough to host RISC-V CPU softcore**
- **On-board FPGA programmer and USB-to-serial adapter**
  - Compatible with IceStorm iceprog tool
  - Easy, driverless connection as a serial device to host computer
- **USB high speed**
  - On-board FT2232 USB chip
  - Up to 480 Mbit/s interface to host computer

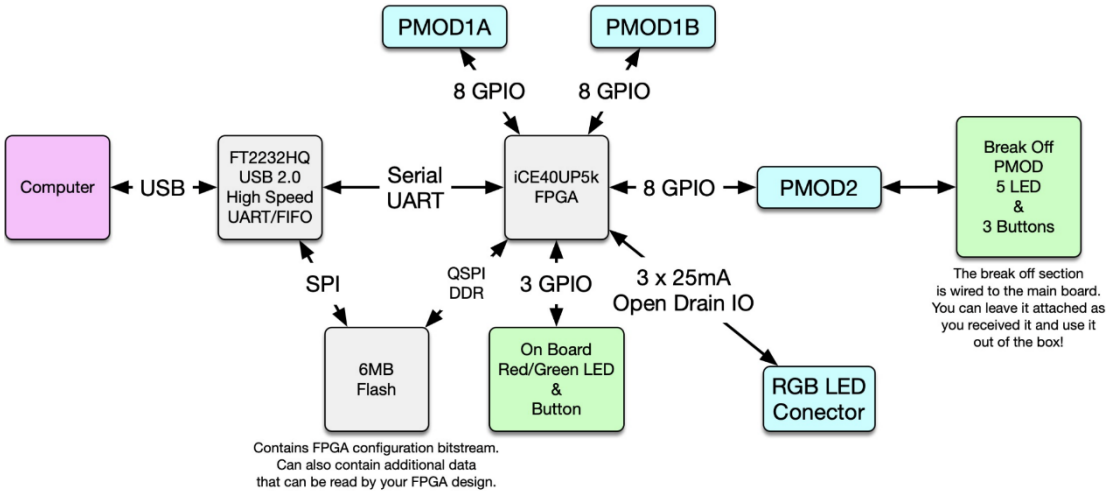

Figure 1: iCEBreaker block diagram.

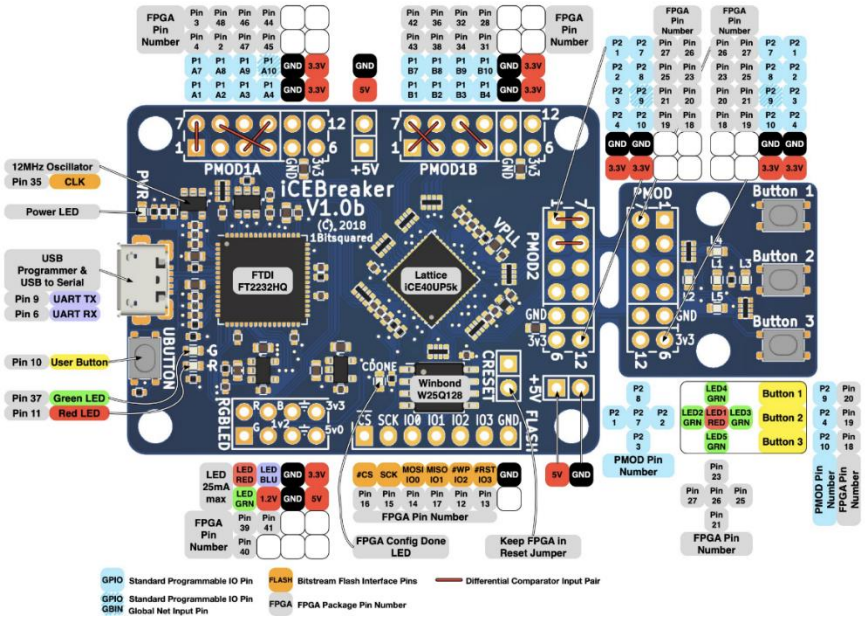

Figure 2: iCEBreaker V1.0b pinout legend.

3. Literature Review

In recent years, the RISC-V architecture has gained significant traction in both industry and academia. As an open-source instruction set architecture (ISA) introduced in 2010, RISC-V has become increasingly popular in research and commercial applications due to its flexibility and cost-effectiveness [6]. an implementation is considered open-source when it is published under a license that allows commercial use without imposed fees, enabling wider adoption in various projects [7]. Moreover, the use of FPGAs for processor implementation has been highlighted in multiple sources, including "JustAnotherElectronicsBlog.com" which defines a "Softcore CPU" as a CPU implemented within the FPGA's logic [8]. This approach provides a highly customizable and efficient platform for developing processors like RISC-V, offering both educational and practical benefits for hardware design.

4. Design Proposal

The RISC-V processor offers various implementation options, one possible design is a 5 stage pipeline processor. Below is a brief explanation of the processor's components:

ALU (Arithmetic Logic Unit)

- Responsible for performing core arithmetic and logical operations, such as addition, subtraction, AND, OR, and XOR.
- In a 5-stage pipeline, the ALU typically operates in the Execute stage, processing instructions and setting conditions for branch instructions.

RF (Register File)

The register file is a set of registers that temporarily holds data during instruction execution, accessible in the Decode and Write Back stages.

PC (Program Counter)

Holds the address of the next instruction to be fetched from memory, operates at the Fetch stage.

Control Unit

- Manages the overall flow of instructions and coordinates the activity of different components, enabling the processor to carry out complex tasks.

- Decodes each instruction in the Decode stage, determining the required operation and signals needed to drive the other modules.

IMM (Immediate Value Generator)

- Generates immediate values required by specific instructions, such as load/store or arithmetic with immediate operands.
- This component is used during the *Decode* stage to prepare values for the ALU when immediate operations are required.

Instruction Decoder

Decodes the binary representation of each instruction into a form that the control unit and other modules can understand.

Branch and Jump Unit

- Handles control flow instructions, including conditional branches and jumps, affecting the PC's behavior.
- Activated during the *Execute* stage if branch conditions are met, it updates the PC to redirect the instruction flow.

5. Bibliography

[1] AMD official website, https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado/vivado-buy.html

[2] Yiyang Chang, Yiming Liu, Chong Peng, Jiarui Guo and Yi Zhao (2024) Design of a Configurable Five-Stage Pipeline Processor Core Based on RV32IM, College of Electronic Science and Engineering, Jilin University, Changchun 130012, China.

[3] Geeta Yadav, Neeraj Kr. Shukla (2013) Pre-Eminance of Open Source Eda Tools and Its Types in The Arena of Commercial Electronics, Department of EECE, ITM University Gurgaon (Haryana), India.

[4] Kenneth B. Kent, Seyed Alireza Damghani (2022) Yosys+Odin-II: The Odin-II Partial Mapper with Yosys Coarse-grained Netlists in VTR, https://dl.acm.org/doi/abs/10.1145/3490422.3502344.

[5] CROWSSUPPLY website, https://www.crowdsupply.com/1bitsquared/icebreaker-fpga.

[6] Sarah L. Harris, Daniel Chaver , Luis Piñuel, J.I. Gomez-Perez , M. Hamza Liaqat , Zubair L. Kakakhel, Olof Kindgren, Robert Owen () RVfpga: Using a RISC-V Core Targeted to an FPGA in Computer Architecture Education, *Paper Title*.

[7] Mark Albers, Alexander Dörflinger, Benedikt Kleinbeck, Yejun Guan, Harald Michalik, Raphael Klink, Christopher Blochwitz, Anouar Nechi, Mladen Berekovic (2021) A Comparative Survey of Open-Source Application-Class RISC-V Processor Implementations, CF '21, May 11–13, 2021, Virtual Conference, Italy

[8] JustAnotherElectronicsBlog.com, *Just another electronics blog.pdf* (2020)

הערכה לשיחת סקר תכנון ראשוני (PDR)

מספר הפרויקט: P-20____-_____

שם הפרויקט:

שם המנחה החיצוני:

שם המנחה מהמחלקה:

שם הסטודנט/ית: ת.ז.:

%	מהות	ציון	הערות
100	שיחה + דו״ח - הבנת הנושא ומהות העשייה, הבנת הצורך, סביבת היישום, הגדרת מדדים, מקורות ועבודות דומות. הצגת התקצירים, מפרט טכני/הצעת מחקר והצעת תכנון מפורטים.		
	ציון סופי		

הערות: