

Power Optimized Data Path Unit for A RISC Processor

Dr. M. Kamaraju

Professor & Director-AS&A
Department of Electronics and
Communication Engineering
SeshadriRao Gudlavalleru Engineering
College Gudlavalleru, Krishna District,
521356 Andhra Pradesh, India

Shaik Haseena

Dept of Electronics &
Communication Engineering
SeshadriRao Gudlavalleru Engineering
College Gudlavalleru, Krishna District,
521356 Andhra Pradesh, India

S. Nandini

Dept of Electronics &
Communication Engineering
SeshadriRao Gudlavalleru Engineering
College Gudlavalleru, Krishna District,
521356 Andhra Pradesh, India

P. Lohith

Dept of Electronics & Communication Engineering
SeshadriRao Gudlavalleru Engineering College
Gudlavalleru, Krishna District, 521356
Andhra Pradesh, India

Shaik Ayesha Begum

Dept of Electronics & Communication Engineering
SeshadriRao Gudlavalleru Engineering College
Gudlavalleru, Krishna District, 521356
Andhra Pradesh, India

ABSTRACT - This paper is about a way to design the inside of a RISC-V processor. The goal is to use power. Most processors use a clock network that uses a lot of power because it is always switching on and off. The new design uses a method to control the clock. It uses flip-flops to make the clock for each part of the processor. This means that only the parts that need to be working can be turned on. The New Design is based on a five-stage pipelined RISC-V processor. It has five parts: Instruction Fetch, Instruction Decode, Execute, Memory Access and Write Back stages. Most designs use a clock that controls everything. This new design uses small clocks made from D flip-flops. These small clocks can turn on and, off parts of the processor like the Arithmetic Logic Unit, the register file and the memory units. This way parts of the processor that are not being used do not waste power. The RISC-V processor still works correctly. It uses less power when it is not doing anything. The RISC-V processor is more efficient because it only uses power when it needs to.

The design is made using Verilog. Implemented on a special kind of computer chip called an FPGA platform using the Xilinx Vivado design suite. We do a lot of tests and checks to make sure the system works correctly and efficiently. We look at things like how power it uses how much space it takes up and how long it takes to do things. We compare these things to ways of doing it like the usual RISC-V datapath implementations. When we try it out we see that our way uses a lot power and it does not take up much more space or time, than the other ways. The design and the FPGA platform and the Xilinx Vivado design suite all work together to make this happen. We use Verilog to make the design and the Xilinx Vivado design suite to implement it on the FPGA platform.

KEYWORDS - RISC-V Processor, Low Power Design, Clock Gating, Flip- Flop Based Clocking, Datapath Architecture, FPGA Implementation, Verilog HDL, Power Optimization, Pipelined Processor, Energy Efficient Computing.

I. INTRODUCTION

The need for energy- processors is really going up because of all the new embedded systems and Internet of Things devices and portable electronics coming out. When we talk about these Internet of Things devices and portable electronics the amount of power they use is a deal. This is because power consumption affects how long the battery lasts, how hot the system gets and how well the whole system works. Nowadays people who design processors are trying to make them use less power without making them slower which is a big challenge, for energy-efficient processors. RISC-V, an open source instruction set architecture is becoming an option for making processors that can be customized and scaled. It has an flexible structure that allows designers to create processor architectures that fit specific needs. This makes RISC-V very suitable for use in devices that have limited power and resources. The RISC-V architecture can be tailored to meet the requirements of applications. It is a platform for designing processors that are both customizable and scalable. RISC-V's modular structure and simplicity are key benefits for designers. This instruction set architecture is highly suitable, for power and resource-constrained environments because of its flexibility. Traditional datapath designs use a global clock distribution network. This network sends the clock signal to all components all the time. The clock signal keeps going to all components even when they are not doing anything. This means that components that are not being used still switch on and off which uses power. The clock network uses a lot of power too. This is because it has a capacitive load and the clock signal is always turning on and off. The datapath designs and the clock network are the reasons for high dynamic power consumption, in these systems. The clock network and datapath designs are important to consider when trying to reduce power consumption. This paper is trying to

solve a problem. It does this by using a kind of design for the datapath that uses flip-flops to control the clock. The datapath is the part of the system that does the work. The idea here is to make the clock signals only go to the parts of the datapath that need them. This means that the other parts of the datapath do not get the clock signal and so they do not use any power. By turning off the clock signal to the parts of the datapath that are not being used we can reduce the amount of power that is being used. This is because the datapath components are not switching on and, off all the time. The datapath design is power-optimized, which means it is designed to use power. The flip-flop-based clock gating technique is what makes this possible.

II. LITERATURE SURVEY

The design of RISC-V processors that use power has become really important to people who do research. This is because we need systems that use energy. People have done a lot of studies on how to make these processors better. They have looked at how to make them work faster how to make them smaller and how to make them use power. RISC-V processors are a part of this because we want to make sure they are good, at saving energy.

The authors of the paper[1] showed how to make a RISC-V processor using the RV32I instruction set and put it on an FPGA. They wanted to make it easy to understand and use for learning and for projects. The RISC-V processor is good for people who are just starting out because it is simple and easy to work with. However the people who made the RISC-V processor were mostly concerned with making sure it worked correctly and was easy to build. They did not think as much about how to make it use less power. The RISC-V processor is still a tool for learning and, for basic projects.

A RISC-V core called AsteRISC was suggested in [2] to use hardware. The main goal was to minimize the use of hardware resources. This design helps put AsteRISC on FPGA platforms and allows for exploring different design options. The approach reduces the area used significantly. However it does not solve the problem of power use caused by clock switching, in AsteRISC. The AsteRISC core still uses a lot of power when its clock switches. This is an issue that needs to be addressed in AsteRISC design.

The people who made this thing called a hazard- pipelined architecture for an RV32I RISC-V processor they told us about it in [3]. They did some things to make it work better like forwarding and hazard detection so that the instructions can go through faster and the pipeline does not stop. This makes the RV32I RISC-V processor work efficiently.. What they really care about is making sure the RV32I RISC-V processor does things correctly and quickly not so much, about using less power.

The people who made [4] showed us a RISC-V soft-core processor that has five stages. This processor is written in VHDL. It is good because it is modular and flexible which is great, for systems that use FPGA. The RISC-V soft-core processor has the pipeline stages and it can handle basic hazards but it does not have good ways to save power like stopping the clock when it is not needed or reducing the

amount of switching activity.

Research in [5] looked at how to make a high-speed UART module for embedded systems. This module is important for communication. It should use as few resources as possible. Even though this is not directly about the design of a processor datapath it shows that optimizing the parts, around the processor like the UART module is crucial for the whole system to work well and use fewer resources. The UART module is a part of this and it needs to be optimized for better performance and efficiency of the whole system, including the processor datapath and the UART module itself.

These studies also looked at ways to make things use power like stopping the clock when it is not needed keeping parts separate and using lower voltage. One of the ways to reduce power use is to stop the clock when it is not needed. This method really works because it stops switching in circuits that need a clock to work.. A lot of the time people use a simple way to stop the clock, which can cause problems, with how things work together and when things happen. From what I have read it is clear that many efforts have been made to improve RISC-V processors in terms of performance, area and functionality. However there seems to be focus on optimizing power consumption at a detailed level. This work aims to fill that gap by suggesting a method that uses flip-flops to control the clock and reduce power usage in a pipelined RISC-V processor without causing any glitches.The goal is to make RISC-V processors more efficient in terms of power consumption.RISC-V processors are being improved in areas but power optimization at the datapath level is an area that needs more attention.Therefore we propose a flip-flop-based clock gating technique, for RISC-V processors.This technique ensures that RISC-V processors operate efficiently and use power.

A. Overview

III. PROPOSED WORK

- Instruction Memory: This is where the program instructions are stored and it gives them out when it is time to fetch them.
- Register File: This has some purpose registers that we use

The proposed work is about creating a datapath unit for a RISC-V processor that uses power. It does this by using a clock control method that turns off parts of the processor when they are not, in use. This method is called clock gating. It helps to reduce power consumption.Normally all parts of the processor are always on even when they are not doing anything. This causes power consumption. The new design turns on the clock for the parts of the processor that are actually working. This helps to save power. The goal is to make the processor use power by turning off parts of it when they are not needed.

The RISC-V processor is going to have a datapath unit that can switch parts on and off. This is really useful because it helps the RISC-V processor to save power. We are looking at the datapath unit of the RISC-V processor. The RISC-V processor datapath unit is going to use something called

a flip-flop-based clock gating mechanism. This clock gating mechanism is important for the RISC-V processor because it helps the RISC-V processor use power when it needs to. The new datapath unit design will be good, for the RISC-V processor.

B. Datapath Architecture

The processor is made using a five stage system. This system is called RISC-V architecture. The datapath has parts like the Program Counter, Instruction Memory Register File, ALU, Data Memory and pipeline registers. The RISC-V architecture is what makes the processor work. It uses the Program Counter and other parts like the Instruction Memory and Register File to do its job. The ALU and Data Memory are also parts of the datapath, in the RISC-V architecture.

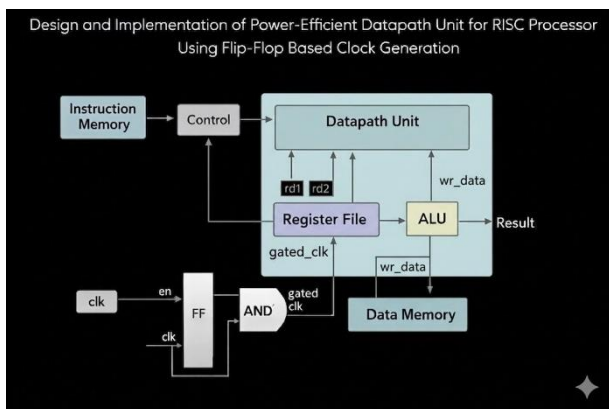


Fig. 1. Block Diagram of Proposed RISC-V Datapath with Flip-Flop Based Clock Gating

The processor is made using a five stage pipeline system that follows the RISC-V architecture. The datapath has some parts:

- Program Counter (PC): This keeps track of the address of the current instruction and updates it one by one or based on what the program says to do. The processor has an other key components.
- to store the things we need to work on and the results of what we do.
- Arithmetic Logic Unit (ALU): This does the math and logic work that the instructions need. * Data Memory: This takes care of reading and writing data when we need to load or store something.
- Pipeline Registers: These are, like temporary storage spaces that separate the stages of the pipeline and let us do many things at the same time.

The processor works in the following steps:

- Instruction Fetch: The processor gets an instruction from memory. It uses a counter to keep track of where the program's. This counter helps the processor know what to do.
- Instruction Decode: The processor figures out what the instruction means. It reads data from the registers. Then it creates signals to control what the processor does.

- Execute: The processor does the math or logic. It checks if it needs to branch to a part of the program. The processor is like a smart calculator.
- Memory Access: The processor goes to memory to get or store data. This is where the processor gets the information it needs.
- Write Back: The processor writes the result back into the register. The processor can work on one instruction in the Instruction Fetch step while it works on another instruction, in the Execute step and so on. This makes the processor work efficiently. The processor can do things at the same time because of this structure.

C. Flip-Flop Based Clock Gating

The main idea of this project is to use a technique to save power. This technique is called flip-flop based clock gating. It helps to reduce the power that is used when the system is working. Of sending the main clock signal to all parts of the system we use something called D flip-flops to create a special clock signal for each part.

- The control unit decides when to send a signal to turn on the clock for each part.
- Each part of the system gets its special clock signal that only turns on when it is actually needed.
- When a part is not being used the clock signal is turned off so it does not waste any power.

This technique is better than the way of doing things because it uses flip-flops to make sure the clock signal is stable and reliable. The flip-flop based clock gating technique also makes it easier to send the clock signal to all the parts of the system and reduces the load, on the clock network. This means that the system can work smoothly and use less power. The flip-flop based clock gating technique is a way to improve the system.

D. Instruction-Aware Clock Control

The new design has a control system that helps make it use less power. This control system looks at what the computer's being asked to do and only turns on the parts that are really needed. For example:

- The math part, which is called the ALU only turns on when the computer is doing math or comparing things.
- The data memory only turns on when the computer is getting or storing information.
- The register file, which is where the computer stores information it is using now only gets updated when it really needs to.
- The pipeline registers, which help the computer do things in the order turn off when the computer is not doing anything or is waiting.

This way of using what is needed helps reduce wasted energy and makes the computer use less power when it is running. The design of the computer is made to use power by only using the instruction-aware control logic when it is really needed which is a big part of making the computer

use less power.

E. Key Features

The new datapath design has a lot of things about it. The datapath design offers advantages.

- Reduced Switching Activity: the clock works when it is needed for the datapath design
- Glitch-Free Clock Operation: the datapath design uses flip flops to control the clock
- Improved Power Efficiency: the datapath design uses a lot less power
- Architecture: the datapath design is very flexible
- Efficient Hardware Utilization: the datapath design makes sure that the hardware is used in a smart way

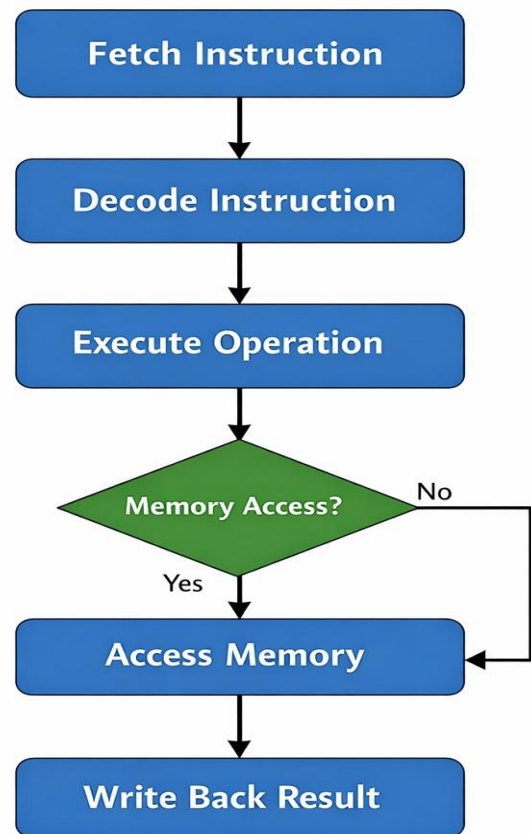
IV. IMPLEMENTATION

A. Flow Chart & Algorithm

The RISC-V processor works in a way. It follows a sequence based on a five-stage pipeline architecture. Each instruction goes through stages. This lets the processor do things in parallel and get work done.

Here is how the processor executes instructions:

- Fetch Instruction: The Program Counter (PC) figures out the address of the instruction. Then it gets that instruction from the instruction memory.
- Decode Instruction: The processor looks at the instruction it just fetched. It determines what kind of operation it is. It also figures out what operands it needs and where it will put the result. It decides what control signals are required too.
- Execute Operation: The Arithmetic Logic Unit (ALU) does the work. It does logical operations based on the instruction. If it is a branch instruction the ALU checks the condition at this stage.



RISC-V Processor Pipeline

■ Processing Stage ◆ Decision Point

Fig. 2. Flowchart of Proposed RISC-V Processor Operation

- Access Memory: If the instruction needs to get data from memory or store data in memory the processor does that now.
- Write Back Result: The final result is stored in the register file. This result comes from the ALU or, from memory.

This way of working makes the processor faster. The RISC-V processor also has a clock gating mechanism. This mechanism only turns on each stage when it is needed. That means it does not waste power when it is not doing anything. The RISC-V processor gets work done and uses less power.

B. Verilog Implementation

The whole datapath design is made using Verilog. We use it to describe the hardware. Each part of the processor is its module. This makes it easy to work with and find problems.

- The Program Counter module changes the

instruction addresses one, by one or when a branch happens.

- The Instruction Memory module gives us the instructions that go with the Program Counter value.
- The Register File lets us read and write the operands and results.
- The Arithmetic Logic Unit module does the math and logical work based on the control signals.
- The Control Unit makes the control signals that the datapath needs to work.

We added a clock gating logic to each module. This logic uses flip-flops and enable signals. These signals decide if a module gets the clock input. This means that modules that are not being used do not waste power.

We tested the design using Xilinx Vivado. We made test benches to check if everything works correctly. We looked at the simulation waveforms to make sure the instructions are executed properly and the clock gating works right. The datapath design and the clock gating logic are checked to ensure they work together correctly.

C. FPGA Implementation

The design we made is put on a kind of computer chip called a Field Programmable Gate Array. This chip is really good for trying out our design because it is easy to use and does not cost a lot of money. Target Platform:

- Field Programmable Gate Array: We did a things to get our design ready.

Design Flow:

- First we wrote the design in a computer language.
- Then we tested it to make sure it works right.
- After that we changed the design into something the computer chip can understand.
- Next we put the design on the chip. Connected all the parts together.
- Finally we made a file that tells the chip what to do and we put it on the chip.

Using the Field Programmable Gate Array is really helpful because we can see how the design works in time. We can also see how well it is doing by looking at things like how power it uses and how fast it is. The results we got were really good. The design we made uses a lot power than other designs because it can turn off parts of the chip when they are not being used. This is really good because it saves power but the design still works correctly and does what it is supposed to do. The Field Programmable Gate Array implementation is really useful, for testing the processor and seeing how well it works.

D. Testbench and Functional Verification

To check if the proposed datapath is correct a complete test system is made using Verilog HDL. This test system tries out different kinds of input to see how the datapath handles different types of instructions like R-type, I-type, load and store and branch instructions.

- Module-Level Testing: Each part of the datapath like the Arithmetic Logic Unit, the register file and the control unit is tested on its own to make sure it works right.
- System-Level Testing: The whole datapath is tested to make sure all the parts work together properly.
- Pipeline Verification: Many instructions are run at the time to check that the pipeline is working correctly.

The results of the simulation are looked at closely to confirm that the instructions are being executed correctly and that the control signals are being passed properly from one stage to the next, in the datapath. The datapath is checked to make sure it is working as it should and the test system helps to validate the datapath.

E. Clock Gating Verification

The flip-flop based clock gating mechanism works well. We looked at the waveform analysis to see how well it works. Clock signals are checked for each part of the datapath. We make sure the enable signals are working correctly so that the modules turn on and off when they should. We also looked at how the switches are turning on and off. This helps us see if the mechanism is reducing switching activity. What we found out is that the flip-flop based clock gating mechanism is very effective. The modules that are not being used do not get clock signals. This means that the dynamic power consumption is lower and the operation is more efficient. The flip-flop based clock gating mechanism helps with this.

F. Timing Analysis

We do the timing analysis after we have finished the synthesis and implementation. This is to make sure that the design we have made meets the required timing constraints. We check the Setup Time and the Hold Time for all the elements. We also look at the Critical Path Delay to find out the operating frequency of the design. The Pipeline stages are checked to make sure they are working together properly. The results show that using clock gating in the design does not cause any timing problems and it keeps the operation of the design stable. The integration of clock gating in the design is good because it does not introduce any timing violations and the design works as it should.

G. Resource Utilization

The FPGA resource usage is checked to see how well the hardware works.

- Look-Up Tables or LUTs are used for logic that happens in an order
- Flip-Flops or FFs are used for logic that happens one step, at a time
- Slices and routing resources are also used Adding clock gating does not use extra hardware resources.

This shows that the FPGA resources are used in a way. The. Clock gating work well together. The FPGA resource utilization is good because of clock gating. Clock gating helps to use FPGA resources efficiently.

H. Power Analysis Methodology

We use tools like Xilinx Vivado to do power analysis for our FPGA design. We look at a few types of power.

- Dynamic Power: this is based on how the signals are switching on and off
- Static Power: this is because of the tiny currents that are always leaking
- Clock Power: we look at this on its own to see how turning the clock on and off affects things Our new design is really good because it uses a lot less dynamic power.

This is because we made sure the clocks are not switching when the modules are not being used. We did this by minimizing the clock switching, in the modules that're not active. The FPGA design tools help us see where we can make these changes to reduce the power of the design.

I. Discussion on Implementation Results

The results of the implementation show that the datapath design we came up with works. It helps reduce the power that it uses. It still works well. We used a design so it can be changed and improved later. Here are some good things about this design:

- It reduces power without taking up much space
- It is just as fast as designs
- Using clock gating makes the whole system work better We used an FPGA to test our design and it works well for real world applications that need to use less power. The datapath design is a choice, for these kinds of applications.

V. RESULTS AND DISCUSSION

This section is about the RISC-V processor design with clock gating. How well it works. We look at the RISC-V processor design with clock gating in terms of how it's verified how much power it uses how much space it takes up and how it uses the resources of a field programmable gate array. We also compare the RISC-V processor design, with clock gating to a traditional design to see how it stacks up.

A. Simulation Results

We did a test to see if the design we came up with works like it is supposed to. The test results show that all the parts of the design work together properly. These parts are Instruction Fetch, Instruction Decode, Execute, Memory Access and Write Back. The processor does what it is supposed to do with logical and memory instructions. It also makes the right control signals. The test results also show that the clock signals only turn on when they are really needed. This means that the design does not waste energy when it is not doing anything and it still works the way it is supposed to.

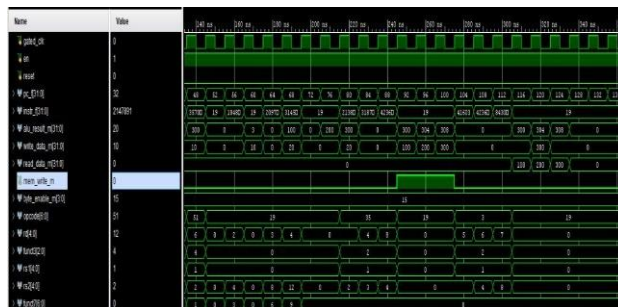


Fig. 3. Simulation Waveform

The RISC-V processor design that we came up with does a lot better when it comes to using power. It is more power efficient. The RISC-V processor design uses power and works just as well as other designs. We looked at how power the RISC-V processor design uses and it uses a lot less. The power used went from 145.131 mW to 0.661 mW. This is because we used clock gating. Clock gating helps by stopping parts of the RISC-V processor design from using power when they are not needed. The RISC-V processor design is a little bigger because we added some parts.. It is not much bigger. It went from 3042 units to 3043 units. This does not make the RISC-V processor design more expensive. We also looked at how

Name	Stack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	299	299	146	dp1d_w_reg_reg22C	dpbranch_e_regD	72.409	49.358	23.051
Path 2	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg109D	72.385	49.351	23.025
Path 3	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg159D	72.385	49.351	23.025
Path 4	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg169D	72.385	49.351	23.025
Path 5	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg179D	72.385	49.351	23.025
Path 6	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg189D	72.385	49.351	23.025
Path 7	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg199D	72.385	49.351	23.025
Path 8	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg1F0D	72.385	49.351	23.025
Path 9	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg229D	72.385	49.351	23.025
Path 10	∞	299	299	153	dp1d_w_reg_reg22C	dpimm_ext_e_reg319D	72.385	49.351	23.025

Fig. 4. Delay analysis results of the proposed RISC-V processor

fast the RISC-V processor design works. It works as fast as it used to. The time it takes for the RISC-V processor design to do things is still 72.409 ns. This means that our new RISC-V processor design works as well as the old one but uses less power. So the RISC-V processor design we came up with is good because it uses power and works well. It is good for things that need to use power and work fast. The RISC-V processor design is a choice, for these kinds of things.

B. RTL Schematic

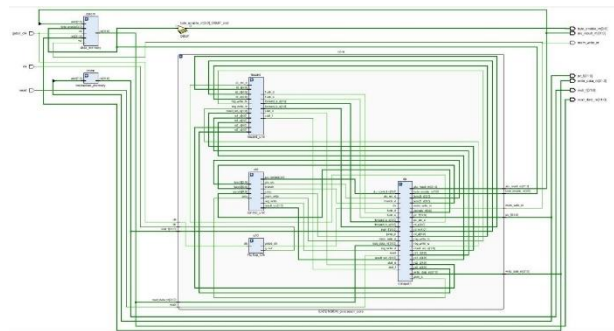


Fig. 5. RTL schematic of the proposed RISC-V processor

The processor design is made into a picture after all the parts

are put together. This picture is called the Register Transfer Level. It is used to check if the parts of the processor are connected correctly. The picture shows that the important parts like the Arithmetic Logic Unit, the register file, the control unit and the memory interfaces are all connected properly. The design also includes a kind of logic that controls the clocks. This means that the clocks only work when they are really needed. This helps to reduce the amount of activity in the processor. The Register Transfer Level schematic is very important for the processor design. It shows that the datapath and control units are working correctly. The Arithmetic Logic Unit and the other parts are all connected in the proper way.

C. Power Analysis

Power consumption is very important in designing processors especially for small devices and applications that use less power. Our design uses a technique called clock gating to reduce the power wasted when the processor is not doing much. The results from our tests and analysis show that our design uses less power than the usual RISC-V design. The power used by parts is less because we turn off the clock signals for parts that are not being used which prevents unnecessary changes.

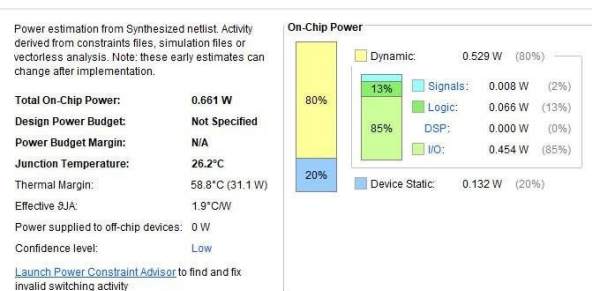


Fig. 6. Power analysis results of the proposed RISC-V processor

- As you can see from the comparison power consumption goes down a lot from 145.131 mW in the design to 0.661 mW in our design.

This shows that clock gating is good at reducing power used for switching without changing how the system works. The clock gating technique helps to save power. The proposed method is more effective, in reducing power consumption. The power consumption is decreased in the proposed design.

D. Area Analysis

The clock gating logic adds a bit more space to the chip. The synthesis report shows that a few extra flip-flops and gating elements are needed. These extra parts make the total area go from 3042 units to 3043 units. That is not much of an increase. The proposed design saves a lot of power without using extra space. It is very good, for hardware. The extra space is not a problem. Clock gating logic helps to save power. Furthermore the small area overhead makes sure the proposed design still works well for FPGA and ASIC implementations, where saving resources very important. The results show that

Name	Slice LUTs (134600)	Slice Registers (269200)
EXTENSION_processor_...	3043	395
dp (datapath)	3042	394
alu_unit (alu)	2560	0
rf (register_file)	49	0
s10 (Flipflop_Clk)	0	1
d0 (dff)	0	1

Fig. 7. Area utilization results of the proposed RISC-V processor

adding clock gating logic gives a balance between saving power and hardware cost. This makes it a practical and scalable solution for low-power processor designs that use clock gating logic and save power. The incorporation of clock gating logic provides a balance between power savings and hardware cost. It is a solution, for modern low-power processor designs.

E. FPGA Utilization

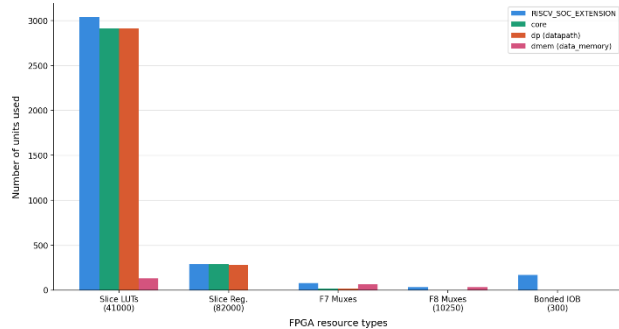


Fig. 8. FPGA resource utilization of RISC-V SOC EXTENSION across design modules

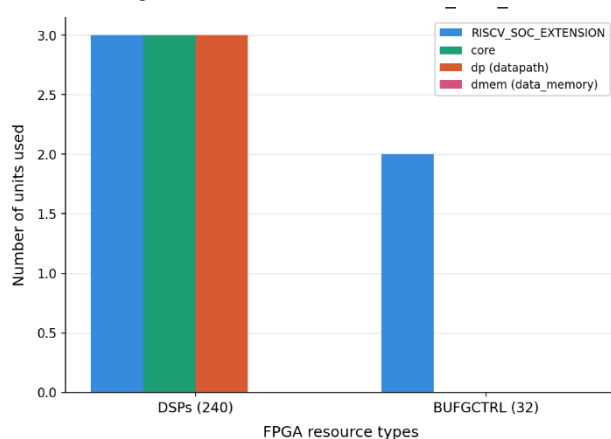


Fig. 9. DSP and BUFCTRL utilization of RISC-V SOC EXTENSION across design modules

The design is built on an FPGA platform to check if it can work in life. The FPGA results show that it uses resources like Look-Up Tables (LUTs) flip-flops and routing elements efficiently. The proposed architecture uses resources optimally.

Also includes techniques to save power. The small increase in elements shows that the design can be scaled up and is suitable for real-time applications, without overloading the

FPGA. In addition to using resources our design also shows stable timing performance. This means that all operations are completed within the required time limits. The use of flip-flop based clock gating greatly reduces activity. This directly leads to power consumption during operation. This improvement is especially important, for applications where power usage's a major concern and we focus on the clock gating and power consumption. The proposed design and clock gating help achieve this.

F. Comparative Analysis

We can see a comparison between the RISC-V processor and the new clock-gated design in Table 1. This comparison looks at some things like how much space the RISC-V processor takes up how long it takes to do things and how much power it uses. The comparison of the RISC-V processor and the clock-gated design is really, about these things. From

TABLE I
COMPARISON OF CONVENTIONAL AND PROPOSED RISC-V DESIGN

Parameter	Conventional RISC-V	Proposed Method
Area	3042	3043
2 Delay	72.409 ns	72.409 ns
Power	145.131 mW	0.661 mW

the results we can see that:

- Power consumption goes down a lot without affecting performance.
- Delay stays the same which means speed is not affected.
- Area overhead is very small so design efficiency is maintained.

Overall the proposed method gets a balance between using less power and hardware cost, which makes it very suitable, for applications that need low power and high performance.

VI. CONCLUSION

This paper talks, about designing a part of a RISC-V processor that uses power. The main goal was to reduce the power used by the clock system without slowing down the processor. We did this by turning off the clock when not needed and making the clock locally. The new design was written in Verilog. Tested on a special computer chip called FPGA. The tests showed that the processor works well in all its parts. These parts include getting instructions decoding them executing them accessing memory and writing back results. The design diagram also shows that all parts are connected correctly. The whole system works reliably.

The performance evaluation shows that the proposed method really reduces power consumption. It goes from 145.131 milliwatts to 0.661 milliwatts. The critical path delay stays the same at 72.409 nanoseconds. The area overhead is very small. It only goes from 3042 units to 3043 units. This means the design is very good at using area. These results show that using flip-flop based clock generation is a way to reduce power usage without making things slower or more expensive. Furthermore the results from implementing the design on an FPGA show that it uses resources like LUTs, flip-flops and routing elements efficiently. The design makes use of resources and has minimal

overhead, which means it can be scaled up and used in real-time applications. This architecture provides a solution for modern systems that need to use low power, where saving energy is a major constraint. In addition the proposed design keeps a balance between power, area and performance. This balance is really important for digital systems. The datapath has a structure. This makes it easy to add features and integrate them. As a result the design can be easily improved in the future. The reduction, in switching activity helps with heat performance. Makes the device last longer. Future work will probably look at adding ways to save power like changing the voltage and speed of the computer chips turning off parts of the chip when not in use and adjusting the clock to use less energy. This will make the computer chips use less power. Future work may also focus on making the design work with complicated instructions and multiple cores so it can be used in really fast computers. This will make the design more useful, for high-performance computing systems like the ones used for projects.

In summary the RISC-V processor design achieves a balance between power consumption, performance and size. Using clock gating along with flip-flop-based clock generation is a way to reduce power dissipation in digital systems. This makes the RISC-V processor design suitable for applications, like embedded systems, Internet of Things devices, portable electronics and other environments where energy's limited.

REFERENCES

- [1] W. Zhang, Y. Zhang, and K. Zhao, "Design and Verification of Three Stage Pipeline CPU Based on RISC-V Architecture," in *2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*, Haikou, China, 2021, pp. 697–703. [Online]. Available: <https://ieeexplore.ieee.org/document/9731161>
- [2] M. H. Banna Saif, N. U. Sadad, and M. N. Islam Mondal, "FPGA Implementation of Educational RISC-V Processor Suitable for Embedded Applications," in *2023 International Conference on Electrical, Computer and Communication (ECCE)*, Chittagong, Bangladesh, 2023, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/10101508>
- [3] J. Saussereau, C. Leroux, J.-B. Begueret, and C. Jégo, "AsteRISC: A Size-Optimized RISC-V Core for Design Space Exploration," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, Monterey, CA, USA, 2023, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/10181330>
- [4] I. Thanga Dharsni, K. S. Pande, and M. K. Panda, "Optimized Hazard Free Pipelined Architecture Block for RV32I RISC-V Processor," in *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2022, pp. 739–746. [Online]. Available: <https://ieeexplore.ieee.org/document/9952122>
- [5] A. E. Phangestu, I. T. Mujiono, M. I. Kom, and S. Ahmad Zaini, "Five Stage Pipelined 32-Bit RISC-V Base Integer Instruction Set Architecture Soft Microprocessor Core in VHDL," in *2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia, 2022, pp. 304–309. [Online]. Available: <https://ieeexplore.ieee.org/document/9855292>
- [6] S. Harris and D. Harris, *Digital Design and Computer Architecture: RISC-V Edition*, Morgan Kaufmann, 2021. [Online]. Available: <https://www.elsevier.com/books/digital-design-and-computer-architecture/harris/978-0-12-820064-3>