Original Article

Microarchitecture based RISC-V Instruction Set Architecture for Low Power Application

Deepika R[1], Gopika Priyadharsini S M[2], Muthu Malar M[3], Vivek Anand I[4]

[1,2,3]Student, Department of Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu, India

[4]Assistant Professor, Department of Electronics and Communication Engineering, National Engineering College, Kovilpatti, Tamil Nadu, India.

Email: deepikarajaram128@gmail.com

DOI: 10.47750/pnr.2022.13.S06.051

Abstract

The goal of many contemporary and speculative applications is to create highly efficient CPUs and one among the architecture that is meeting out the above said condition is RISC V processor microarchitecture. The RISC-V Instruction Set Architecture [ISA] supports the microarchitecture. Microarchitecture and instruction set architecture are the two key elements of processor design. In comparison with other instruction execution stages, the high hardware complexity of multiplier and divider circuits must be included in any core microarchitecture. As a result, the construction of an appropriate hardware circuit capable of multiplication and division determines the total size, power, and performance of a core. With the exception of load and store, this core has four stages in which all instructions are executed. One clock cycle is used to complete the arithmetic operations. However, in order to reduce the critical path latency, the division and multiplication operations are repeatedly performed. In this project, the Baugh Wooley multiplier will be used because it can run in two clock cycles. Instructions for multiplication and division are included in RISC-V. The fundamental microarchitecture is designed to enable the execution of instructions with the minimal amount of structural risk, control and data. The ultimate objective is to create a low-power application, with only the most important functions. It was primarily concerned with optimizing the size, power, and efficiency of the system.

Keywords: Micro Architecture, Pipelining, RISC V.

1. INTRODUCTION

The demand for appliances such as wireless devices, medical gadgets, cellular phones and IOT devices is rising in the contemporary technological period. Each of these systems requires a strong CPU to carry out its duties. Furthermore, the bulk of these apps are portable and run-on batteries. These gadgets therefore need a low power and high-performance processor that can carry out the instruction while consuming the least amount of battery life. Over the past four decades, the CMOS scaling technology has enhanced processor performance. However, CMOS technology is nearing the end of its useful life as it approaches physical restrictions. As a result, in order to increase power and performance within the stated bounds, designers are constrained to think about other aspects of CPU design. instruction set architecture (ISA) and microarchitecture are the two key elements of processor design. The ISA is a load-store architecture in terms of RISC V architecture. There are numerous uses for the instruction set. The base instruction set consists of 32-bit naturally aligned instructions with a fixed length, however the ISA supports variable length extensions that let each instruction to be any number of 16-bits long.

2. Proposed Processor's Architecture

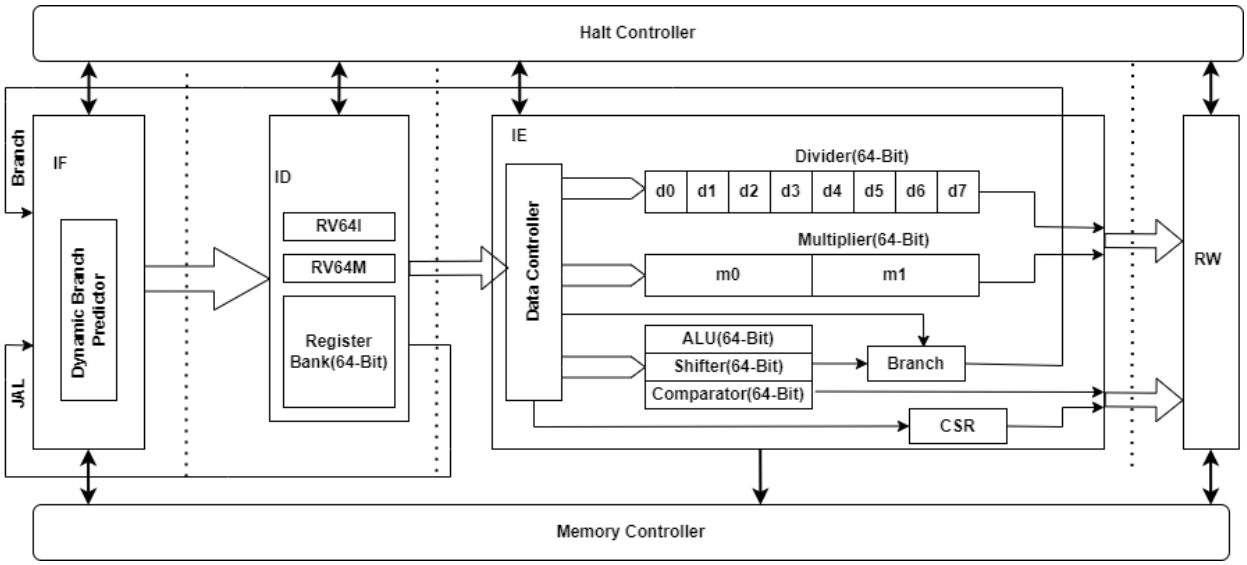

Fig 2 Architecture of the Proposed Device's Processor

This is the RISC processor design that we have suggested for the microarchitecture. We then constructed the ensuing building blocks and simulated the results.

2.1 Logical and Arithmetic Unit:

It is a combinational circuit that operates arithmetically and bitwise on integer binary integers. It is a crucial part of many different computing circuits. The arithmetic operations are addition and subtraction. Relational operations, which test conditions like less than, not equal to, greater than equal, greater than, less than and equal to, are used to compare data. Rotate right, rotate left, right shift and left shift are all examples of shifting procedures. The logical operations XOR, AND & OR are a few examples.

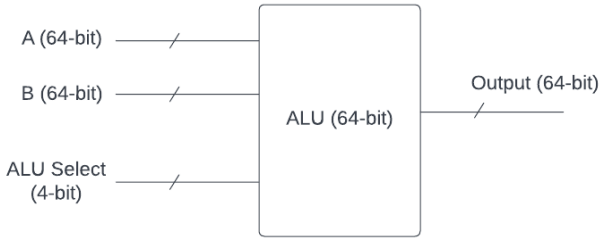

Fig 2.1 Arithmetic and Logic Unit

2.2 Memory device:

A storage space known as a memory unit houses the necessary memory for a system with 26 memory spaces.. The inputs are data, address, clk, and we, as illustrated in figure.2.2 (write- enable). The data comes from other blocks like the ALU, divider, multiplier, and so on. The output from such blocks was fed as input to the memory block after the relevant procedure was completed. The data to be saved in the desired location is denoted by the address pin. Then we use the write enable pin (1-bit) to help write the data to the correct spot.

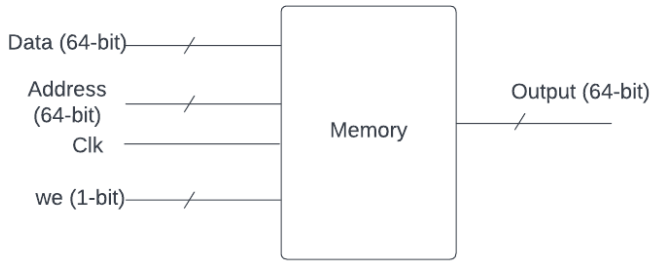

Fig 2.2 Memory Unit

2.3 Instruction Fetch :

The register values are originally assessed at the IE stage. Execution continues if they are prepared for reading; otherwise, values are obtained from the RW/IE stage based on preceding instructions. Then, these register values are passed to the various Functional Units (FU) of the IE stage for execution. There are five FUs in the IE stage. The following is a list of these FUs' operations and clock cycles:

- Arithmetic and Logical Unit - (1 clock), xor, sub, or, add, and
- Comparator block – Comparison of signed-to-unsigned conversion (1 clock)
- Logical Shifter – arithmetic and logical arithmetic shift (1 clock)
- Multiplier block – Multiplication carried out for signed or unsigned number (2 clocks)
- Divider block – Division carried out for signed or unsigned number (8 clocks).

The write/read and branch signals for memory are likewise produced by the IE stage. The second source register, rs2, supplies the memory read/write data while the Arithmetic and Logic unit gives the memory write/read address. The instruction ID stage generates the memory write/read enable signal. As a result, these three signals are combined to control memory write/read operations. Branch enable and branch address are the two signals that make up a branch. Depending on the kind of instruction, the comparator output turns on the branch enable signal and the ALU output assigns the branch address.

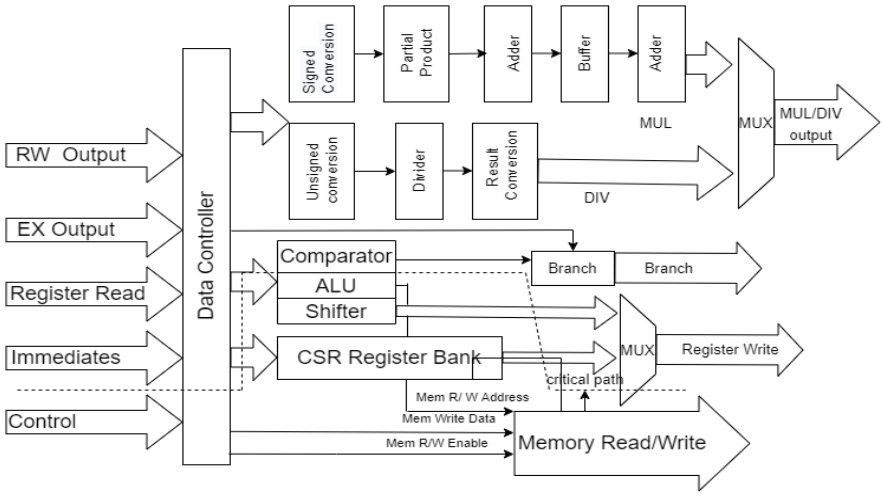

Fig 2.3 Instruction fetch Unit

2.4 Multiplier:

In digital electronics, such as a computer, a binary multiplier is an electrical circuit that multiplies two binary integers. Due to its low power consumption, the Baugh Wooley signed multiplier was found to be the ideal option for our micro-architecture. Baugh-Wooley multiplication is one of the most cost-effective methods for dealing with sign bits. This method was created to create regular multipliers that are suitable for 2's complement numbers. The Baugh Wooley signed multiplier circuit has a delay of 5.97 ns. If this circuit is used directly at the IE stage, the critical path for the entire system lengthens. As a result, the multiplier's hardware circuit is split into two stages. Partially computed products are calculated in the first stage. They are then separated into six parts and added together. The adder's output is then buffered for the subsequent step. The outcome is obtained in the second stage by again summing the six buffered results. There are 29 FAs and 1 HA (Half Adder) on the first stage's important route. There are 1HA (half adder) and 29 FAs on the first stage's important route (Full Adders). The first stage's delay is kept lower than the second stage's during the entirety of the partial product segmentation. This is because the multiplier's input data could originate from the register bank or from different pipeline stages. There will therefore be a multiplexer delay at the multiplier's inputs.

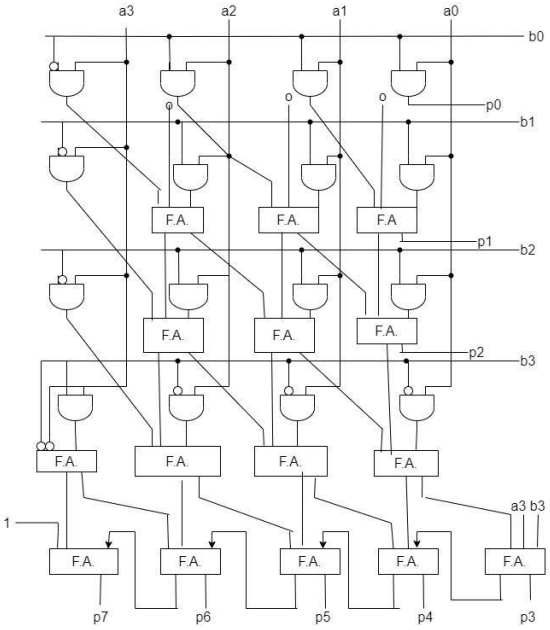

Fig 2.4 64 bit Baugh Wooley Multiplier

2.5 Divider :

The divider unit for the suggested micro-architecture is chosen to be Radix-16. The results of the initial constant multiplication of the divisor are saved in registers. It takes a lot of multipliers, each of which is expensive in terms of size, time, and power, to calculate these constant multiplications. As a result, shifters and adders are used to accomplish these multiplications in hardware circuits. For multiples of 2, 4, and 8, the divisor is shifted to the left, and the other multiples are found by averaging the results. The updating of the quotient and remainder is the next step in the division process. At each clock cycle, the 36-bits of the dividend that are the most important are compared to its multiples that have been previously calculated.

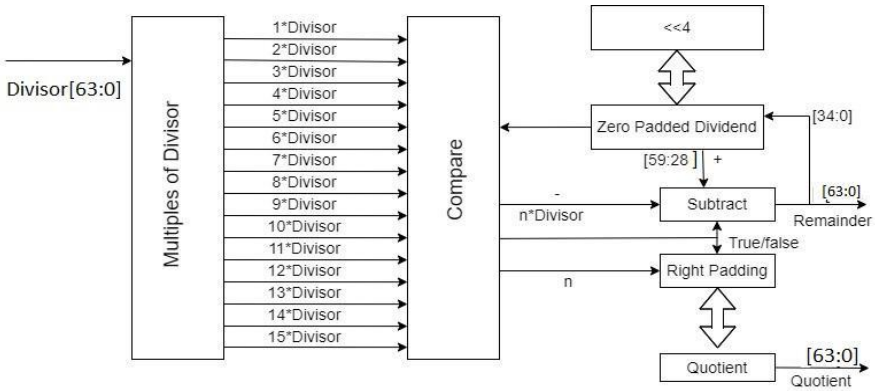

Fig 2.5 64-bit Radix-16 Divider

2.6 Shifter :

A shifter is a digital circuit that can change the order of bits in a data word. It has the ability to shift N-bit data in a single cycle. Data can be shifted left or right using a Shifter. A Shifter can perform arithmetic shifting, logical shifting, and rotation functions in general. Bits are multiplied or divided by powers of two by shifters. A shifter, as the name implies, shifts a binary integer a predetermined number of places to the right or left. An N-bit shifter can be made using N N:1 multiplexers. Depending on the value of the log2N-bit select lines, the input is shifted by 0 to N 1 bits. A shifter is used to move the bits in the partial product of the multiplication and division result.

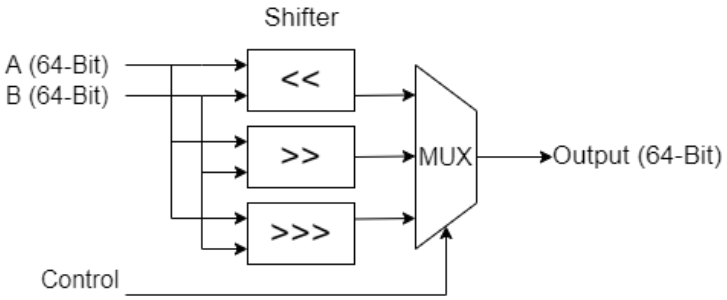

Fig 2.6 Shifter

2.7 Comparator :

A comparator is a circuit that evaluates the equality of two binary words. Some comparators also read the input words as an arithmetic relationship and unsigned or signed integers (greater or less than). Magnitude comparators are the common name for these devices. Based on the output of the dynamic branch predictor, comparators are used to forecast the branch and the next instruction to be performed.

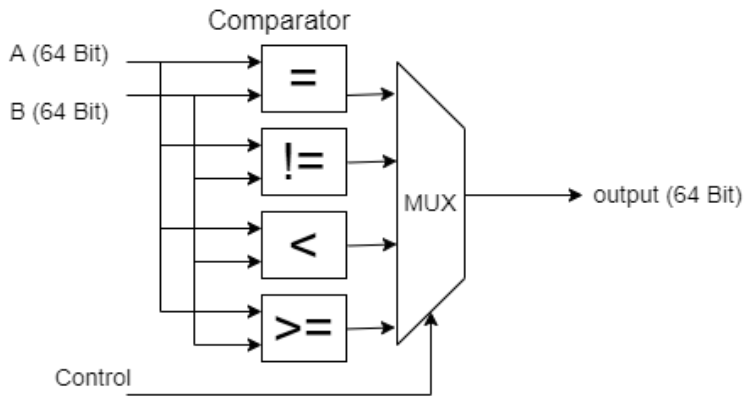

Fig 2.7 Comparator

2.8 Register Bank:

A 64-bit register bank in Figure 2.6 represents the programmable registers that assembly language programmers employ. It is equipped with two ports:

- Read port
- Write port

Caches and main memory both provide comparable interfaces for accessing and writing data. A register number acts as an index in a register bank. The index into main memory is a memory address.

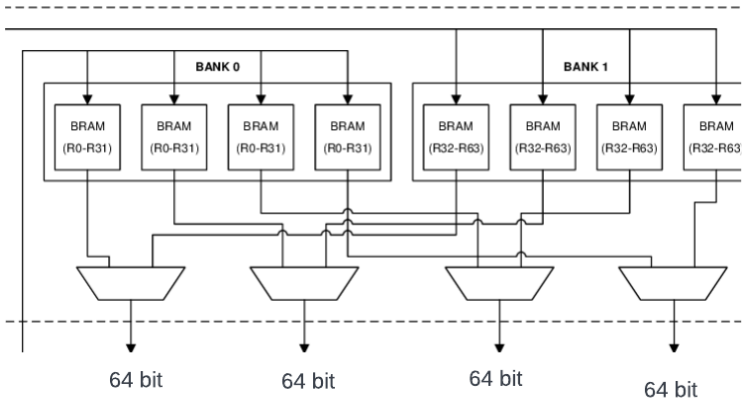

Fig 2.8 Register Bank

3. Result and Discussion:

In this paper after choosing the suitable multiplier and divider for the microarchitecture core design we have simulated the results for the internal blocks of the processor and the obtained results are listed below.

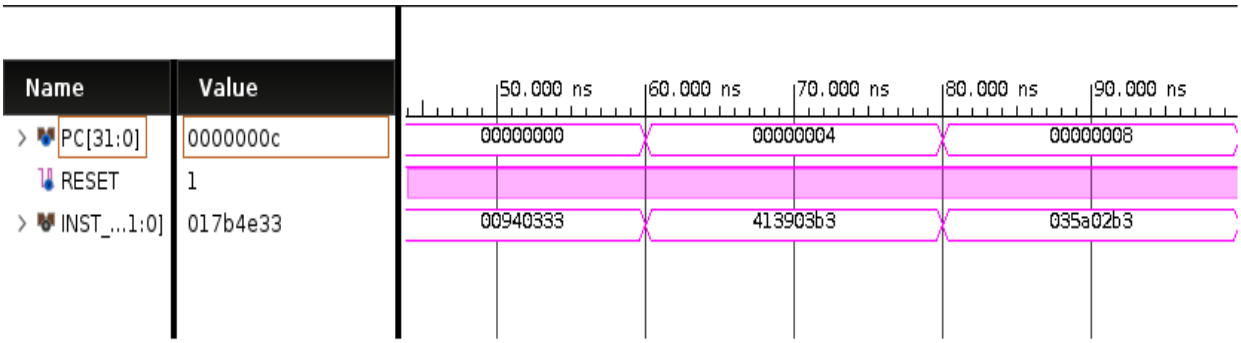

Fig 3.1 Simulation result of Memory Unit

Figure3.1 shows that data is fetched from the memory by using the program counter address.

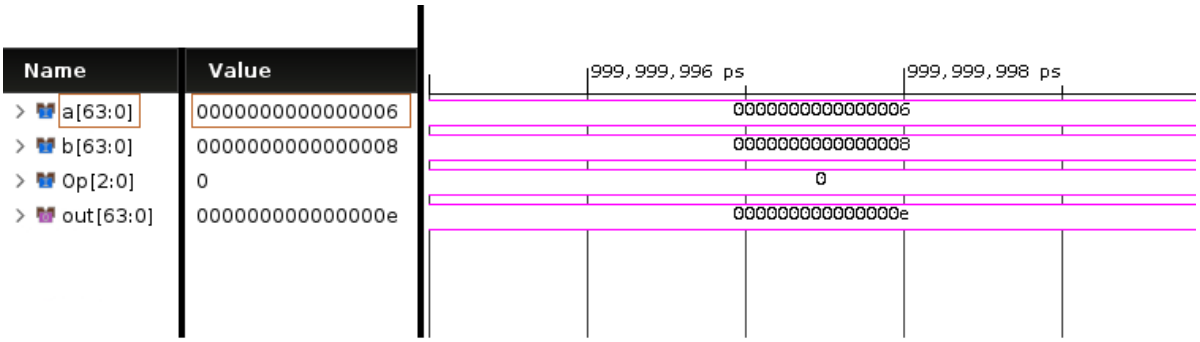

Fig 3.2 Simulation result of ALU for Addition Operation

Figure 3.2 shows that addition operation is performed for a ,b and the output is stored in out.

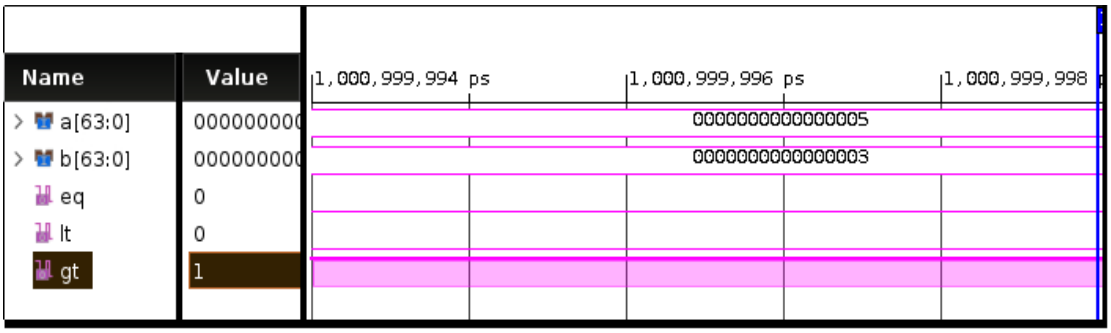

Fig 3.3 Comparator simulation outcome

Figure 3.3 shows the Simulation Result for Comparison between two inputs a and b.These are branch instructions which are sent into the dynamic branch predictor.

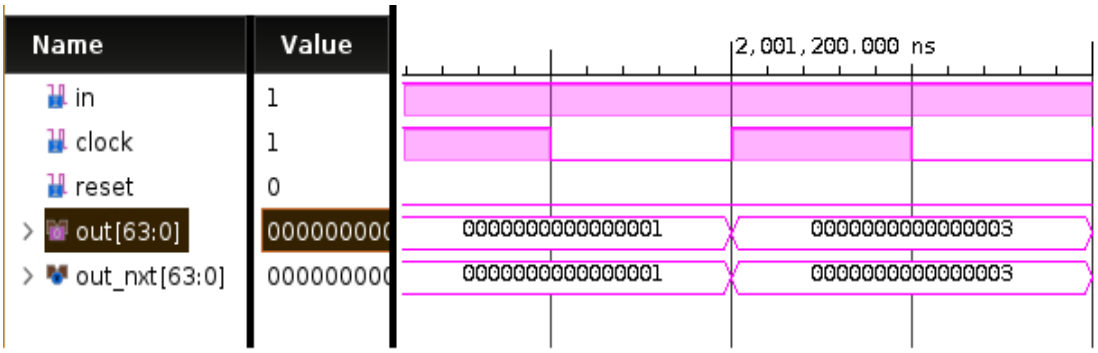

Fig 3.4 Shifter simulation results

The simulation result for shifting operations on a 64-bit input with clock and reset as additional inputs is shown in Figure 3.4. Shifted outputs and control and status register data are written in the register bank.

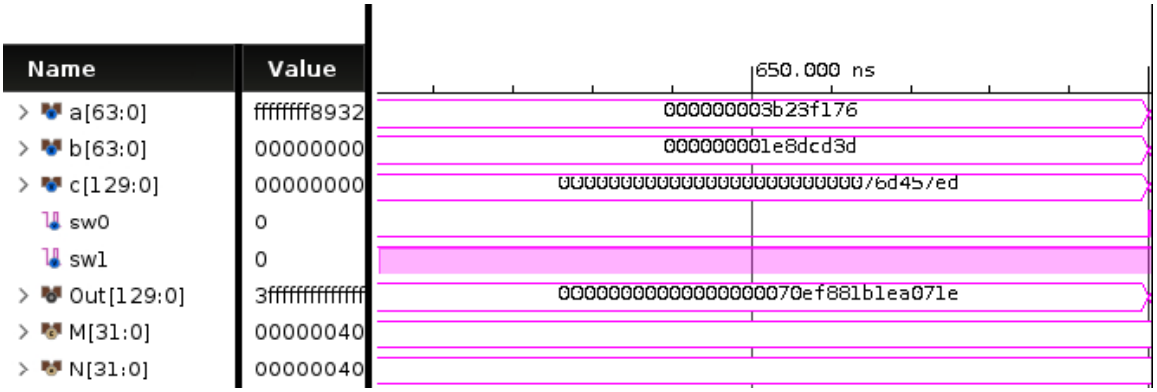

Fig 3.5 Multiplier simulation outcome

Figure 3.5 displays the simulation results for our suggested processor's Baugh and Wooley multiplier. It can perform multiplication of two signed or unsigned binary numbers.

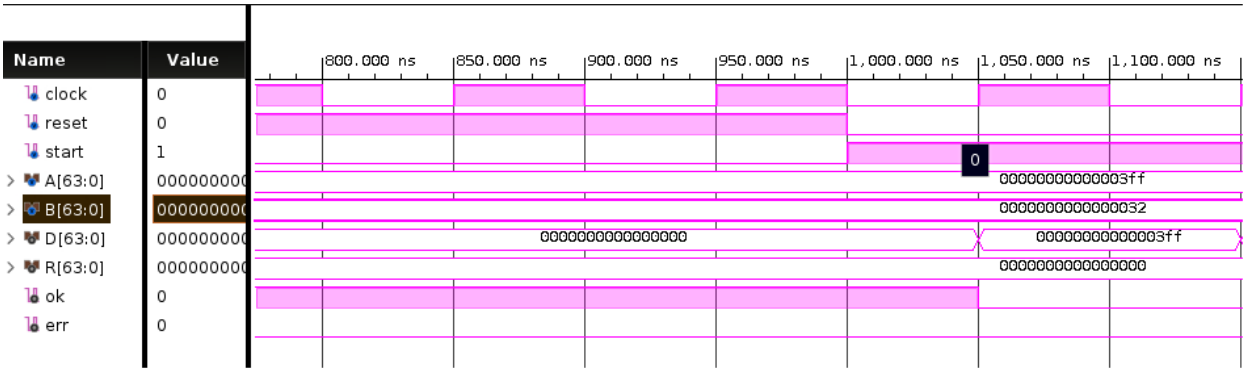

Fig 3.6 Divider simulation outcome

The division operation for two inputs, A and B, is performed by the proposed processor's divider, as shown in Figure 3.6, and the results are placed in the D and R registers. The divider implemented in the proposed processor is designed using the Radix-16 divider algorithm.

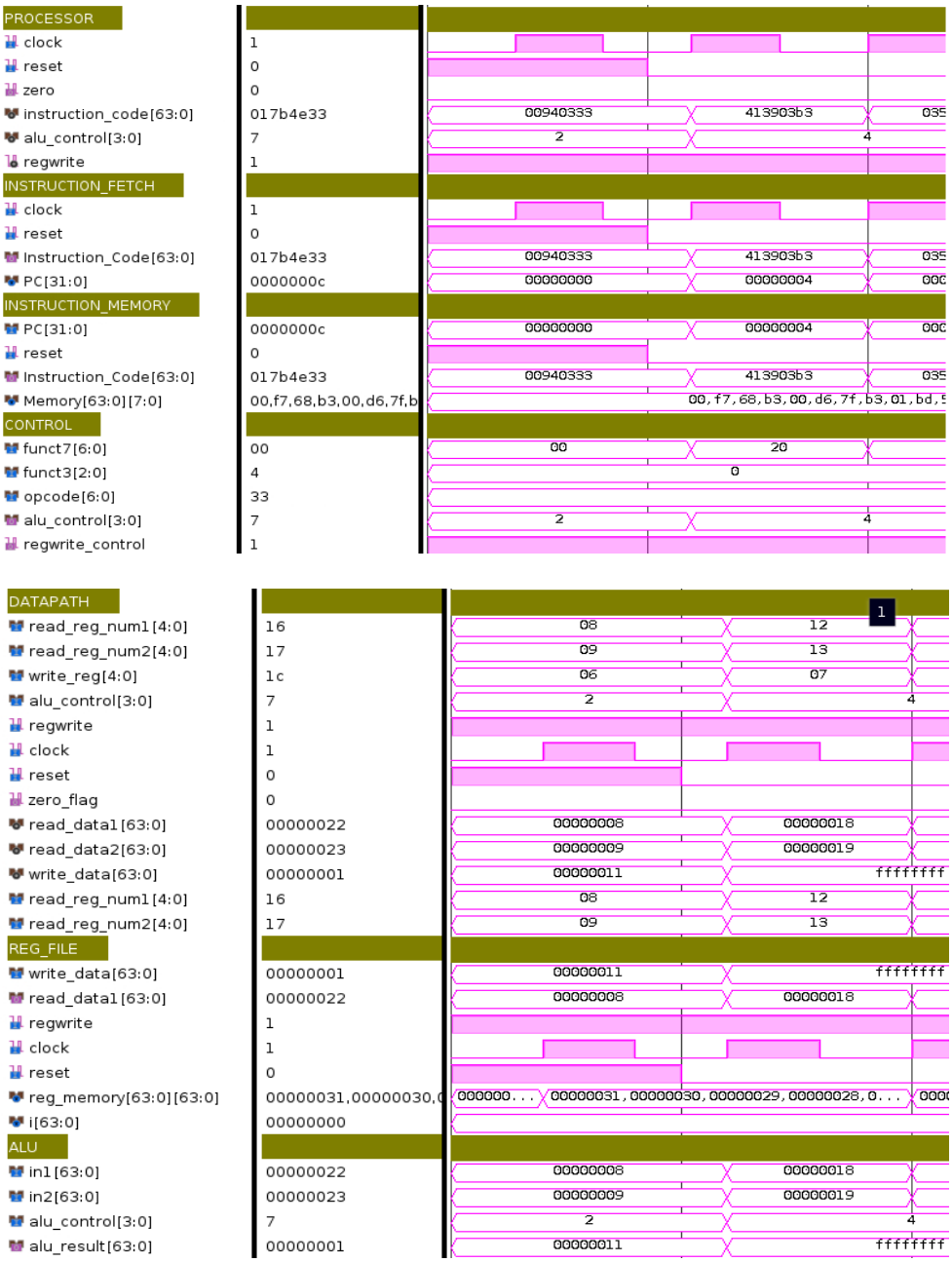

Fig 3.7 Simulation result of Proposed Processor

Figure 3.7 displays the simulation results for the 64-bit microarchitecture-based RISC processor we propose. The instructions are fetched from the instruction memory and data are processed in the ALU block by using a control signal and the outputs are written in the register.

3.8 Comparison of Power Report of Existing And Proposed Processor

The power consumption of the proposed CPU and the existing processor are contrasted in Table 3.8.

Power of Existing Processor(32 bit)	Power of Proposed Processor(64 bit)
0.702 mW	0.687 mW

4. Conclusion:

An energy-efficient 64-bit RISC processor is constructed in this study utilizing Verilog HDL. With this proposed CPU, we have achieved low power consumption thanks to the proposed architecture's careful design and consideration of power consumption. The CPU under consideration uses 0.687 mW in total. A microarchitecture was built using multipliers and divisors. In order to develop a RISC processor, one of the modules is the proposed microarchitecture. After the core's microarchitecture was implemented, the processor's sub-blocks were built to achieve low power consumption.

REFERENCES

[1] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., vol. C-22, no. 12, pp. 1045–1047, Dec. 1973.

[2] A. Waterman, Y. Lee, David A. Patterson, KrsteAsanovi, The RISC-V Instruction Set Manual,Vol. I: Base User-Level ISA. Tech. Rep. UCB/EECS-2011-62, EECS Department, UCB, May 2011.

[3] OpenRISC 1000 Arch. Manual, 1st Ed., OpenCores, 2012.

[4] R. Muralidharan and C. Chang, "Radix-4 and radix-8 Booth encoded multi-modulus multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 60, no. 11, pp. 2940–2952, Nov. 2013.

[5] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovic, "The RISC V instruction set manual, volume I: User-level ISA, version 2.0," EECS Dept., Univ. California at Berkeley, Berkeley, CA, USA, Rep. UCB/EECS-2014-54, May 2014.

[6] AndreyMokhov, Iliasov, Sokolov, Rykunov, Yakovlev, Romanovsky"Synthesis of Processor Instruction Sets from High-Level ISA Specifications", IEEE trans on computers, Vol:63, pp. 1551-1565, June 2014

[7] Liney Group article "The case of open instruction sets, open ISA would enable free competition in processor design". August 2014

[8] Y. Lee et al., "A 45 nm 1.3 GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators," in Proc. 40th Eur. SolidState Circuits Conf. (ESSCIRC), Sep. 2014.

[9] Lopez-Parrado, A.Valderrama-Cuervo,"OpenRISC-based System-onChip for Digital Signal Processing" Proc. of IEEE symposium on Image, signal processing and Artificial Vision, pp. 1-5, Sept 2014.

[10] S. Gupta, N. Gala, G. S. Madhusudan, and V. Kamakoti, "SHAKTIF: A fault tolerant microprocessor architecture," in Proc. IEEE 24th Asian Test Symp. (ATS), Nov. 2015.