

Design of 32-bit Processor for Embedded Systems

Hyun Woo Oh, Kwon Neung Cho and Seung Eun Lee*

Dept. of Electronic and IT Media Engineering
Seoul National University of Science and Technology
Seoul, Korea

*seung.lee@seoultech.ac.kr

Abstract— In this paper, we propose a 32-bit processor for the embedded system. In order to provide less area and low power operation, we adopt MIPS instruction set architecture (ISA) to our processor. The processor consists of five pipeline stages to reduce the critical path. In order to solve the data hazard in pipeline stages, we design the data forwarding unit and stall unit with optimized bubble insertion. The processor is implemented on a field programmable gate array (FPGA), and we verify the functionality of the processor and measure the performance by using the Dhrystone benchmark. The Dhrystone MIPS (DMIPS) is measured at 27.71 at 50 MHz operation.

Keywords; embedded system; MIPS; pipelining; data forwarding; stall;

I. INTRODUCTION

Recently, the embedded systems are used for many applications with the increased complexity of the system [1-2]. Due to the limitations of area and power, the embedded system requires a small area and low power consumption [1]. For this reason, the processor that provides low power operating conditions with high performance is necessary for the embedded systems. In the point of the processor, there are two types of instruction set architecture (ISA) called RISC and CISC. RISC has a fixed length of instruction and requires fewer number of instructions compared to CISC [3]. These properties enable the processors to be area-efficient because of the simplified hardware architecture. By adopting the load-store architecture, all the arithmetic operations are executed by only using general-purpose registers (GPRs). The load-store architecture leads the processor to operate in high-speed through RISC [4]. RISC with load-store architecture such as MIPS, ARM, and SPARC is widely applied to various embedded processors [3-4]. In order to improve the performance of the processor by optimizing throughput, reducing the critical path by pipelining is one of the methods [5]. However, in the pipelining process, the data hazard occurs when the operating instructions are dependent on the previous one. Although inserting the bubble instruction solves the data hazard, the bubble insertion causes an increase in program size and throughput degradation. In order to reduce the bubble insertion, the processor needs additional pipeline control units.

In this paper, we propose a 32-bit processor that is compatible with MIPS I ISA. A 5-stage pipelining applied to the proposed processor. Each stage is called instruction fetch (IF), decode (DEC), execute (EX), memory (MEM), and write back (WB). In addition, we design the data forwarding unit and

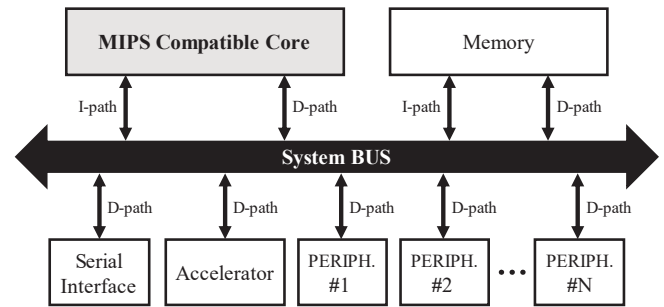


Figure 1. Entire architecture of the proposed processor

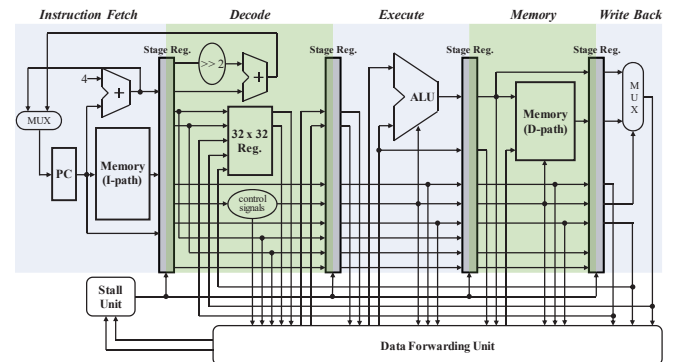


Figure 2. Block diagram of the MIPS core

the stall unit to deal with the data hazard with optimized bubble insertion. The processor is described in Verilog HDL, and we implemented the processor on a field programmable gate array (FPGA). We verify the functionality and measured the performance of the processor by the Dhrystone benchmark.

II. PROPOSED SYSTEM

Our proposed system consists of MIPS I ISA compatible core, system bus, memory, and peripherals such as serial interface and accelerator as shown in Fig. 1. All the components of the processor are connected through the system bus, which contains the instruction path (I-path) and data path (D-path) to avoid the structural hazards by reading the instruction codes and data separately. The serial interface is included to provide communication with external devices.

Fig. 2 presents the MIPS compatible core architecture. The core includes the 32 of 32-bit GPRs, five pipeline stages, data forwarding unit, and stall unit. Each pipeline stage has a

register that stores operands, addresses of operands, and control signals to execute. In the IF stage, the program counter (PC) points the address of the instruction to read, and the instruction is stored in the DEC stage register. The PC is changed to a certain value by the condition of the DEC stage. In the DEC stage, the core performs decoding instruction, reading operand data from GPR, calculating branch condition for PC, and generating control signals for the other stages. In the EX stage, the arithmetic logic unit (ALU) calculation is executed. A 32-bit multiplier and divider, which calculates in one clock cycle and four clock cycles respectively, are also included in the core. In the MEM stage, the core performs memory-related tasks such as load and store. In the WB stage, the core writes the data acquired from previous stages to the GPR.

In the pipelining process, the data hazard occurs when the data to be read from the GPR in the current stage are dependent on the data to be written to the GPR in the next stage. The data forwarding unit detects the data hazard by comparing the addresses for reading the GPRs in each current stage and the addresses for storing the data to the GPR in each next stage. In addition, the data forwarding unit checks the write enable signal for the GPR in each stage to prevent forwarding the data that not to be written to the GPR. The data hazards are detected only when the write enable signal of each stage is set.

The data forwarding unit forwards the data from the closest hazard-detected stages to the current stage to deal with the data hazards when the data are able to be forwarded. On condition that the data hazard occurs but the data is unable to be forwarded, the data forwarding unit generates the stall unit control signal. There are two types of stall that the core is unable to forward the data of the next stage. Each condition is called stall type 0 and stall type 1. The stall type 0 signal is generated when the operand in the EX stage is dependent on the data read from memory in the MEM stage. The stall type 1 signal is generated when the instruction of the DEC stage is a kind of branch with comparing GPR values and that GPR values are dependent on data to write at the EX stage or MEM stage. There are two conditions that the stall type 1 signal is generated. The first condition is when the instruction of the EX stage is one of the ALU or the memory load operations.

TABLE I. DHRYSTONE BENCHMARK RESULT

Entity	Value
Run Count	131,072
Total Execution Time (in seconds)	2.69
Dhrystones per Second	48,685.48
Dhrystone VAX MIPS (DMIPS)	27.71
DMIPS / MHz	0.554

Compiled with mips-elf-gcc-5.3.0 and binutils-2.34

TABLE II. DEVICE UTILIZATION SUMMARY

Entity	Resource Usage
Total Logic Elements	6,032 / 24,624 (24 %)
4-input LUTs	3,439
3-input LUTs	1,406
<=2-input LUTs	860
Register only	327

Implemented on Altera Cyclone III EP3C25Q240 FPGA

The second condition is when the instruction of the MEM stage is one of the memory load instructions. The stall unit is operated by the control signals generated by the data forwarding unit. In case of both stall type 0 and stall type 1 signal are generated, the stall type 0 signal has a higher priority than the stall type 1.

III. IMPLEMENTATION

We verify the functionality of our processor through the FPGA implementation and measure the performance of the processor by running the Dhrystone 2.1 benchmark. In order to check the execution time of the Dhrystone benchmark, we add the clock counter to the system bus. Table I presents the result of the Dhrystone benchmark. The benchmark result shows that our processor provides 27.71 Dhrystone VAX million instructions per second (DMIPS) at 50 MHz operation. Table II presents the resource utilization of the processor including MIPS I ISA compatible core, system bus, serial interface, and clock counter, implemented on Altera Cyclone III FPGA. The result shows that the performance and area usage of the processor is suitable for the embedded system.

IV. CONCLUSION

In this paper, we proposed the 32-bit processor for the embedded system compatible with MIPS ISA. The processor is pipelined with five stages. In addition, the processor has the data forwarding unit and stall unit that provides reduced bubble-inserting conditions to improve the throughput of running instruction codes and decrease the program size. We implemented the processor on FPGA and the processor is verified and measured by using the Dhrystone benchmark. Our processor provides 27.71 DMIPS at 50 MHz operation. In the future work, we are planning to attach various accelerators to the processor. We expect that our processor is suitable for a variety of embedded systems.

ACKNOWLEDGMENT

This work is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). No. 2019R1F1A1060044, 'Multi-core Hardware Accelerator for High-Performance Computing (HPC)'.

REFERENCES

- [1] Y. H. Yoon, S. Y. Jang, D. Y. Choi and S. E. Lee, "Flexible Embedded AI System with High-speed Neuromorphic Controller," International SoC Design Conference (ISOC), 2019, pp. 265-266.
- [2] Z. Zhong and M. Edahiro, "Model-Based Parallelizer for Embedded Control Systems on Single-ISA Heterogeneous Multicore Processors," International SoC Design Conference (ISOC), 2018, pp. 117-118.
- [3] C. Venkatesan, M. T. Sulthana, M. G. Sumithra and M. Suriya, "Design of a 16-Bit Harvard Structure RISC Processor in Cadence 45nm Technology," International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 173-178.
- [4] Prasanth, V., Sailaja, V., Sunitha, P., & Vasantha, B., "Design and implementation of low power 5 stage pipelined 32 bits MIPS processor using 28nm technology". International Journal of Innovative Technology and Exploring Engineering, Vol. 8, pp. 503-507, March 2019.
- [5] Indira, P., Kamaraju, M. & Vyas, V., "Design and Analysis of A 32-bit Pipelined MIPS RISC Processor," International Journal of VLSI Design & Communication Systems, Vol. 10, pp. 1-18, October 2019.