An SDPF RISC-V Processor with Two-stage Pseudo-pipelined Architecture for IoT Applications

Wenji Mo, Yuchen Wang, Haoning Sun, and Jingjing Liu*

School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China

* Email: mowj28@mail2.sysu.edu.cn, liujj77@mail.sysu.edu.cn

2024 IEEE 17th International Conference on Solid-State & Integrated Circuit Technology (ICSICT) | 979-8-3503-6183-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSICT62049.2024.10832035

Abstract—**Internet of Things (IoT) nodes are required to execute lightweight tasks for sensing information and simple signal processing. Thus low-power processors serve as crucial components of highly integrated IoT smart sensors. This paper proposes a low-power RISC-V instruction set architecture (ISA) based RV32I processor for embedded IoT nodes, which follows the serial data path. To enhance the performance of the serial data path followed (SDPF) processor, a pseudo-pipelined architecture is proposed. By partitioning and combining a portion of the instruction lifecycle tasks into two stages, a two-stage pseudo-pipelined structure is achieved, reducing the number of cycles per instruction (CPI) of the SDPF processor. The proposed processor is designed based on Verilog HDL, the implementation results on FPGA demonstrate that the performance is improved by 58% compared to the conventional SDPF RV32I processor. Besides, the synthesis is performed using a standard 0.18 μm CMOS process technology. Post-layout simulation results demonstrate that, the System on Chip (SoC), which consists of the proposed processor, a 2 kB IMEM and a 4 kB DMEM, achieves an average CPI of 36, an area of 0.988 mm^2, and an average power consumption of 182 μW/MHz.**

Keywords—RISC-V, serial data path, pseudo pipelined, low power, IoT

I. Introduction

Embedded IoT nodes employed for information sensing and signal processing are required to operate continuously for extended periods in energy-constrained environments [1]. These devices have relatively low performance requirements and are highly sensitive to power consumption. Consequently, a low-power processor is necessary to manage the operation of the entire system. The RISC-V, an emerging open-source ISA [2], offers flexibility that allows users to remove redundant hardware and implement the instruction set in a modular manner. This enables the design of low-power processors specifically tailored for IoT sensor nodes. In the low-power design of RISC-V processors, retaining only the RV32I integer instruction set is an important approach. Moreover, power consumption and area can be reduced through various architectural designs. The single cycle design can effectively decrease power consumption but may lead to timing criticality in critical paths [3]. An alternative strategy is to apply a multi-cycle non-pipelined design, which reduces dynamic power by decreasing the toggle rate [4]. However, this approach introduces more registers, thereby increasing the chip area.

Traditional processors follow parallel data transfer techniques, but they face limitations in reducing power consumption and area. On the other hand, the processors based on serial data paths offer an attractive alternative [5]. By processing instructions in a serial data path with reduced register utilization, this approach can achieve lower power consumption and smaller area compared to parallel path processors. However, converting 32-bit wide data into serial form and transmitting it to the serial path require at least 32 cycles, which makes the implementation of a conventional pipeline architecture challenging, leading to a further decrease in core performance. This paper proposes an RV32I processor that follows a serial data path and features a pseudo-two-stage pipelined structure. By employing additional shift registers (SRs), the processor divides and combines a portion of the tasks within the instruction cycle into two stages, simulating a two-stage pipeline structure and reducing the CPI of the proposed SDPF processors. While delivering superior performance, the proposed design maintains low power consumption and a small area footprint. This paper is organized as follows: Section II introduces the pipeline simulation scheme and the data path design. Section III discusses the simulation results and Section IV concludes the paper.

II. Design of The Proposed SDPF RISC-V Processor

A. *Two-Stage Pseudo-Pipelined Structure*

In previous work, a RISC-V processor adhering to a serial data path was first introduced in SERV, which is an open-source RV32I processor. In SERV, multiple SRs are employed to convert the data to be processed into a serial format and transmit it to other modules for serial execution. The core computes one bit of data per clock cycle, and the results of operations are written back to the register file in a serial manner. Consequently, in the RV32 baseline, the core requires at least 32 clock cycles to complete the computation of data or address. The architectural paradigm poses significant challenges in implementing a conventional pipelined processor structure within a serial data path. Furthermore, the overall architectural design of SERV resembles that of a traditional multi-cycle, non-pipelined processor. This implies that SERV requires the complete execution of the current instruction before fetching a new one, which leads to a significant degradation in performance. In order to solve this problem, this paper introduces a two-stage pseudo pipeline architecture aimed at optimizing the core implementation. The proposed architecture is illustrated in Fig. 1. The structures A and B represent two pseudo-pipeline architectures employed for processing different instructions in RV32I instructions set. The instruction processing flows of both the A and B structures commence with the *Instructions Fetch* (IF) stage. Each IF and *Instruction Decode* (ID) occupy one clock cycle respectively. Based on the decoding results, the core determines the subsequent execution flow. If the current instruction is a conditional-dependent instruction, it is processed according to the B structure. Otherwise, it follows the structure A for processing. In the structure A, the core determines the subsequent execution order based on the decoding results. If the current instruction is a non-source-

This work is supported by National Natural Science Foundation of China with project number 6217418.

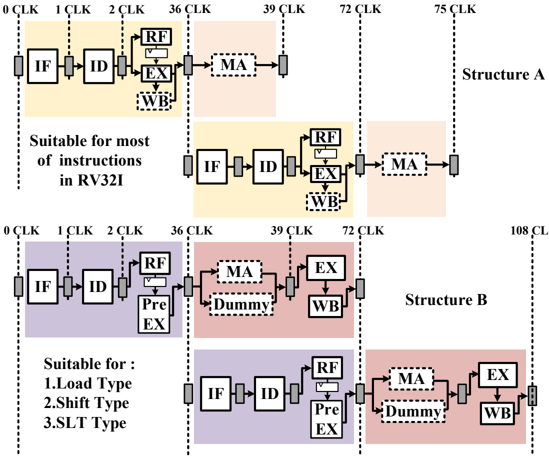

Fig. 1. The proposed two-stage pseudo-pipelined structure.

register-dependent instruction, it proceeds directly to the *Execution* (EX) stage; otherwise, *Register Files* (RF) reading need to be performed first. The RF, EX, and write-back (WB) are embedded within the same time frame. The register file transmits data bit-by-bit starting from the least significant bit (LSB), and it takes two cycles for the first bit of data to reach the EX module. During this phase, the core performs bit-wise computations on the serial input data in each clock cycle and continuously writes the new execution results back to the register file bit-by-bit. This entire process requires 32 clock cycles. After completing the WB, the core transitions into the second stage. The second stage of the structure A involves only *Memory Access* (MA), which is executed exclusively when processing S-type instructions. The core spend two cycles accessing the memory via the bus and writes data to the memory in the third cycle.

The structure B is a pseudo-pipelined architecture that is followed when processing load type (L-type), shift type and the *Set Less Than* (SLT) type instructions. Unlike the structure A, after reading the register files in the first stage, Pre-Execution (Pre–EX) is required, and the EX and WB is scheduled in the second stage. The Pre-EX is responsible for calculating the data that will be used in the EX. This stage requires 32 clock cycles to compute and determine the conditions necessary for the instruction. During this stage, if processing L-type instructions, the core calculates the address for MA. If processing SLT-type instructions, the core compares the two specified numbers. For shift-type instructions, although condition evaluation is not required, 32 clock cycles are needed to retrieve the data to be shifted and store it in the SR before performing the subsequent specified shift operations in the second stage. The second stage pseudo-pipeline encompasses three tasks: MA, EX, and WB. Among the three types of instructions involved in Structure B, only L-type instructions require the inclusion of the MA stage. Similar to S-type instructions, L-type instructions also necessitate three clock cycles to complete data retrieval. Furthermore, when processing the remaining two types of instructions, three dummy clock cycles are inserted during the original MA time slot to align with the timing of L/S-type instructions, thereby maintaining the consistency of the pipeline structure. The fundamental difference between processing instructions using Structure A or Structure B lies in handling condition-dependent instructions. Structure A is unable to simultaneously perform condition evaluation and the calculations of write-back data within the serial data path. To maintain the same pipeline structure, the two stages that consume the most clock cycles are placed separately in the two stage of the pseudo-pipeline, resulting in two distinct methods of instruction processing.

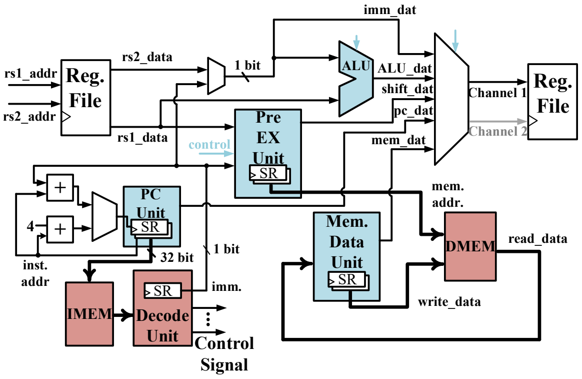

Fig. 2. The data path of the proposed SDPF RISC-V processor.

Since the SERV processor executes instructions in either one or two stage and its design resembles a multi-cycle non-pipelined architecture [5], the CPI of SERV can be represented as Equation (1).

$$CPI_{SERV} = \frac{37f_1+71f_2}{f_1+f_2} = 37 + \frac{34}{1+\frac{f_1}{f_2}} \tag{1}$$

where f_1 denotes the number of one stage instructions, while f_2 represents the number of two stage instructions. The proposed processor, on the other hand, has a CPI of 36. This shows that the proposed processor have higher performance by reducing the CPI.

B. Data Path of the Proposed Processor

The data path architecture based on the aforementioned pseudo-pipelined structure is illustrated in Fig. 2, modules with a blue background indicate that data will be processed and computed serially within these modules, while modules with a red background signify that data is stored or computed in a 32-bit format. The thick lines in the figure represent data transmission in a 32-bit parallel manner, whereas the thin lines denote data transmission in a 1-bit serial manner. The instructions are first fetched from the instruction memory (IMEM) and then sent to the Decode Unit for decoding, which subsequently generates a series of control signals to drive the operation of the remaining core modules. Besides, Decode Unit converts the immediate value contained in the instruction signal via the SR and serially transmits it to the data path for subsequent module usage. All data that needs to be written back to the RF is connected to a multiplexer, which is controlled by the control signals generated by the Decode Unit. Based on the instruction type, the multiplexer selects the required data and serially writes it back to the register file. The MA operation is mainly handled by the Pre-EX Unit and the MEM Data Unit. The address for MA is generated by the SR in the Pre-EX Unit, and the data required for S-type instructions is generated by the SR in the MEM Data Unit. The core sends address signals, data signals, and handshake signals to the data memory (DMEM) via the bus. Unlike cache, the DMEM consists of an SRAM directly connected to the bus. Ideally, it takes three clock cycles from the core initiating the handshake to the DMEM returning a response signal. Additionally, when processing L-type instructions, the data returned from the DMEM is loaded into the SR of the MEM data Unit and converted to serial output, waiting to be written back to the register file.

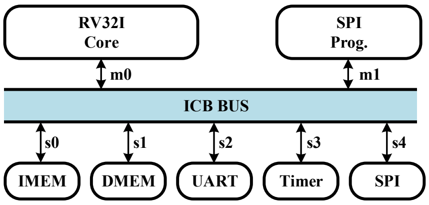

Fig. 3. Block diagram of SoC implementation for processor measurement.

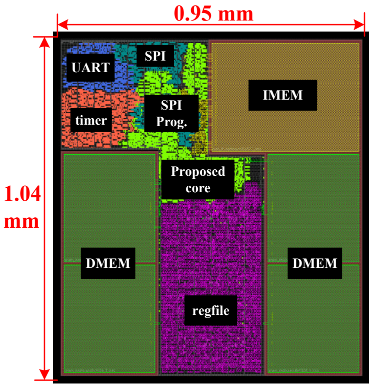

Fig. 4. The layout of the minimalist SoC including the proposed processor.

III. Implementation Results and Analysis

A. FPGA Implementation Result

To obtain meaningful experimental results, the proposed SDPF RISC-V processor has been tested within a minimalist SoC. As shown in Fig. 3, the SoC contains a 2 kB IMEM and a 4 kB DMEM, which are enough for the target application. Several communication interfaces are employed to transmit the results to the host computer. The internal interconnect of the SoC adopts the Internal Chip Bus (ICB) [6] from the Hummingbird E203 processor, as highlighted in blue. This is a custom bus suitable for low power processor cores with only two independent handshake channels, and users can customize priorities to implement multi master and multi slave peripheral mounting. SERV, an open-source RISC-V processor that also follows a serial data path, will be used as a suitable benchmark for comparison. The SERV core will be placed in an identical SoC and configured with its default settings.

Table I presents the test results obtained from a testing environment based on the Xilinx KC705 device. Dhrystone and CoreMark are two benchmark programs whose scores are positively correlated with performance. Compared to the SERV based SoC, the proposed SoC based on the SDPF RISC-V processor consumes more LUTs and FFs. This can be attributed to the additional SRs used in the pseudo-pipeline structure for jump address calculation and data computation in the Pre-EX module, which necessitate additional control logic. Consequently, this leads to higher static and dynamic power consumption compared to the SERV. However, compared to SERV based SoC, the proposed processor based SoC achieves an average performance improvement of 58%. Therefore, the additional power consumption and area overhead are acceptable in light of the performance improvements achieved.

B. Area and Power of the Proposed Processor

The proposed SDPF RISC-V processor were synthesized and placed-and-routed using standard 0.18 μm CMOS process with 1.8 V standard cell library. Fig. 4 shows the layouts of the minimalist SoC. The SoC incorporating the proposed processor occupies an area of 0.988 mm^2. The post-layout simulation results indicate that, under an operating condition of 1.8 V, its average dynamic power consumption is 182 μW/MHz. Compared to the parallel data path followed (PDPF) RV32I processors, such as [3], the proposed processor reduces area by 33% and dynamic power consumption by 37%, and in comparison to [7], the proposed processor reduces area by 25.7% and dynamic power consumption by 73%. In comparison to SERV, the proposed processor exhibits an advantage in terms of average CPI. Based on the Coremark and Dhrystone results, it can be estimated that the actual CPI of SERV is approximately 62. The proposed processor, on the other hand, has a CPI of 36, which indicates that the proposed processor has a higher instruction processing efficiency. These characteristics make the proposed processor well-suited for operation in low-power IoT nodes.

TABLE I. Experimental Results of RISC-V Minimalist SoC Implementation on FPGA.

SoC	LUTs	FFs	Static Power (mW)	Dynamic Power (mW)	Dhrystone	Coremark
SERV	493	533	157	114	0.034	0.023
This work	903	795	158	115	0.053	0.037

IV. Conclusion

This paper proposes a method to improve the performance of RISC-V processors that follow a serial data path. A two-stage pseudo-pipeline structure is proposed to reduce the CPI of the SDPF processors. The implementation result shows that the proposed processor improves computational performance by 58% compared to the SERV, while reducing dynamic power consumption by at least 33% compared to the PDPF processors. This validate the effectiveness of the pseudo-pipelined structure. The proposed SDPF RISC-V processor exhibits a high degree of suitability for IoT smart sensor applications.

References

[1] C. Seok, M. M. Mahmud, M. Kumar, O. J. Adelegan, F. Y. Yamaner, and Ö. Oralkan, "A Low-power Wireless Multichannel Gas Sensing System Based on a Capacitive Micromachined Ultrasonic Transducer (CMUT) Array," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 831–843, Feb. 2019.

[2] A. Waterman, Y. Lee, R. Avizienis, D.A. Patterson, and K. Asanovic, The RISC-V Instruction Set Manual Volume II: Privileged Architecture Version 1.9, Tech. Rep. UCB/EECS–2016–129, EECS Department, University of California, Berkeley, 2016.

[3] S. Shukla, P. K. Jha, and K. C. Ray, "An Energy-efficient Single-cycle RV32I Microprocessor for Edge Computing Applications," *Integration*, vol. 88, pp. 233–240, 2023.

[4] J. Saussereau, C. Leroux, J. B. Begueret, and C. Jego, "AsteRISC: A Size–Optimized RISC-V Core for Design Space Exploration". *2023 IEEE International Symposium on Circuits and Systems*, Monterey, CA, USA, 2023.

[5] M. Sarmiento, K. Nguyen, C. Duran, and R. Serrano, "Systems on a Chip With 8 and 32 Bits Processors in 0.18–μm Technology for IoT Applications," *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, vol. 69, no. 5, pp. 2438–2442, May 2022.

[6] "Open-source Hummingbirdv2 E203 RISC-V processor core and SoC," github.com. [Online]. Available: https://github.com/riscv-mcu/e203_hbirdv2.

[7] F. F. Nascimento, R. N. Wuerdig, A. F. Ponchet, and B. Sanches, "RISC-V SoC Physical Implementation in 180 nm CMOS with a Quark Core Based on FemtoRV32," *2023 IEEE Seventh Ecuador Technical Chapters Meeting*, Ambato, Ecuador, 2023, pp. 1–6.