2024 Design, Automation & Test in Europe Conference (DATE 2024)

I²SR: Immediate Interrupt Service Routine on RISC-V MCU to Control mmWave RF Transceivers

Jimin Lee[a], Sangwoo Park[a], Junho Huh, Sanghyo Jeong, Inhwan Kim, and Jae Min Kim[b]
S.LSI, Samsung Electronics, Hwaseong, Republic of Korea
{jimin15.lee, sw0113.park, huhjunho, shyo.jeong, inhwan1.kim, jmy.kim}@samsung.com

***Abstract*— In 5G technology, millimeter Wave (mmWave), known as Frequency Range 2 (FR2), is one of the candidates for next-generation mobile communications. However, the adoption of mmWave technology in transceiver systems faces challenges in terms of strict latency requirements, due to the significantly increased number of control and compensation tasks. In this paper, we propose a novel interrupt architecture, Immediate Interrupt Service Routine (I²SR) on a RISC-V MCU, enabling zero-latency context switching. Applied to the mmWave transceiver, our I²SR MCU efficiently handles the mmWave tasks by removing context switching overhead. In our experiment, I²SR architecture reduces the overall MCU utilization by 30.6% compared to the baseline MCU, thereby demonstrating that I²SR architecture satisfies the strict latency requirements essential for mmWave applications.**

Keywords—RISC-V, MCU, ISR, Context switching, Interrupt latency, 5G mmWave, FR2

I. Introduction

As the requirements for mobile communication continues to grow both in terms of latency and bandwidth, millimeter Wave (mmWave) technology is being thought as the viable solution. However, the application of mmWave technology is still suffering various challenges, postponing its global adoption. One notable challenge involves the significant increase in the number of control and compensation tasks required for Radio Frequency (RF) transceivers. These tasks come with fine-grained timing requirements, demanding a minimum timing window as short as 4.17μs due to the Orthogonal Frequency-Division Multiplexing (OFDM) [1]. To mitigate the timing-critical control challenge, several studies suggest the use of embedded MCU which provides much faster and flexible controllability [2], [3].

With the adoption of MCU, numerous interrupt-based applications are being developed to enhance performance and they tend to be divided into smaller sub-tasks to fit into symbol-wise control windows. Accordingly, it necessitates the adoption of a nested-interrupt to prioritize the tasks. This requirement causes a severe performance challenge, having that each of the tasks is running at an extremely fine-grained timing granularity of a few microseconds. Note that nested interrupt architecture causes a non-negligible overhead and is still unable to fully resolve the execution time uncertainties, due to the critical sections in the context switch period.

There has been a fair amount of studies that challenge in reducing the interrupt overhead. The previous studies concentrate in Operating System (OS) level optimization [4], or focus on a compromised hardware optimization scheme, that provides a sub-optimal performance with limited addition of hardware [5]. Further, we aim directly at achieving the best interrupt performance for the mmWave applications, with the addition of hardware. In this paper, we propose a novel interrupt architecture, I²SR implemented on a Rocketchip [6] based MCU. We integrate the MCU on our mmWave platform. Then, we evaluate and analyze the performance using the state-of-the-art mmWave applications. Finally, we conclude our paper proposing unlimited opportunities for fine-tuning the interrupt architecture with RISC-V, to fulfill the diverging requirements of domain-specific applications.

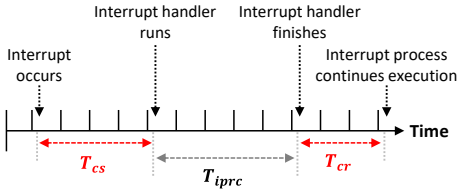

Fig. 1. Illustration of context switching overhead, on an execution of ISR.

II. I²SR Architecture

This paper proposes the I²SR architecture which delegates all control to the hardware to achieve zero-latency context switching in exchange for some additional hardware cost. Fig. 1 describes the conventional interrupt handling architecture, where each Interrupt Service Routine (ISR) execution has context storing time (T_{cs}) and context restoring time (T_{cr}). During context switching, MCU requires not only to preserve register values, but also to configure interrupt preemption based on interrupt levels. These are handled by the software in conventional architecture. In contrast, I²SR architecture efficiently reduces context switching time to zero by implementing additional hardware. Thus, I²SR enables MCU directly jumps to ISR, without any software control. In addition, I²SR allows the preemption control among timer, software, and external interrupts to be flexibly applicable to domain-specific environments. Note that the typical RISC-V architecture does not have the consideration between these types of interrupts.

I²SR architecture utilizes additional sets of General-Purpose Registers (GPRs) and Floating-Point Registers (FPRs). To achieve zero-latency transition when switching ISR for a specific interrupt level, the GPRs and FPRs should be dedicated to that level. To preserve conventional architecture pushes registers onto the stack memory. In our implementation, the use of dedicated register sets removes the requirements of storing the registers of the previous context, ensuring the original values of the registers are not contaminated during interrupt servicing. Additional Control Status Registers (CSRs) are also implemented, to preserve the status of each interrupt. I²SR architecture utilizes the CSRs to support the hardware-based preemption control.

Upon receiving an interrupt signal, the interrupt controller checks to see if there are any pending interrupts. When multiple interrupts are pending, the interrupt controller forwards information about the interrupt with the highest level to the core. In addition to this information, both the interrupt ID and level are also transmitted in the I²SR architecture. The

[a] Equal contribution
[b] Corresponding author

979-8-3503-4859-0/DATE24/© 2024 EDAA

core preempts the current task to handle the incoming interrupt if the incoming interrupt has a higher level than the current one. Based on the interrupt level, the core dynamically selects the appropriate register file that corresponds to each incoming interrupt level. When context switching occurs in the interrupt level utilizing the dedicated register set, the core jumps to the target ISR without the context save/restore.

III. Evaluation

Fig. 2 shows the overall control system of 5G mmWave implemented in 5G User Equipment (UE). It consists of a baseband modem, an Intermediate Frequency (IF) transceiver, and multiple Phased-Array (PA) transceivers. Our latest IF transceiver design supports the connection of up to three PA transceivers. Once data transfer is initiated, further transfer information is dispatched to the UE. This information is first received by PA, followed by IF and the modem in order. Control is executed in the opposite direction, where the modem sends the message to the IF transceiver in order to control the IF. During the control, it is required to execute the mmWave applications on IF and PA at the proper symbol or slot boundary, the timeframe of mmWave communication. Consequently, we design our IF transceiver to have HW-driven tick generators that trigger MCU to execute the application at the required timeframe.

Our I²SR MCU is integrated on the IF transceiver. The IF transceiver receives the control message from a baseband modem, and controls the multiple PA transceivers. The message not only configures the accurate timing of slot and symbol but also schedules the suitable task for each application at a specific timeframe. To transfer and receive the data with high accuracy in mmWave communication, it is essential to execute the assigned application at the symbol or slot boundary with precision. Our I²SR MCU supports message processing and execute applications without any context switching overhead; thus the overall operation is executed in time, satisfying the strict requirements of mmWave communication.

In our IF transceivers, total of 9 applications are embedded. There are 5 applications triggered by the modem. One is dedicated to receive the control messages (A_{Msg}). In turn, one of two scheduling applications is executed to configure the slot-wise (A_{Sch1}) or symbol-wise (A_{Sch2}) tasks by parsing the received message. Other two applications (A_{SW1}, A_{SW2}) are triggered by the modem to control the HW setting. Additionally, three timing-critical applications (A_{TRX}, A_{TX}, A_{RX}) are implemented to process the slot and symbol tasks. These applications are triggered by the tick generators and configured at relatively higher interrupt levels to support its timing-critical behavior, compared to the other applications. The last application (A_{Timer}) is configured at the lowest level to avoid blocking other applications, focusing on processing the trivial tasks.

Our evaluation conducted during the most demanding mmWave communication scenario where data is given every symbol. Compared to the conventional architecture, the utilization of the MCU is reduced by 30.6%, due to the I²SR architectural features and characteristics of mmWave applications. This is because of the current mmWave control architecture, triggering the slot-wise and symbol-wise applications at every symbol boundary. This causes the applications to be executed even without the scheduled tasks but with context switching overhead. Fig. 3 shows that I²SR reduces the average execution time of A_{RX} more than 30%, by removing the software context switching. Consequently, I²SR reduces the MCU utilization by removing this unavoidable context switching overhead.

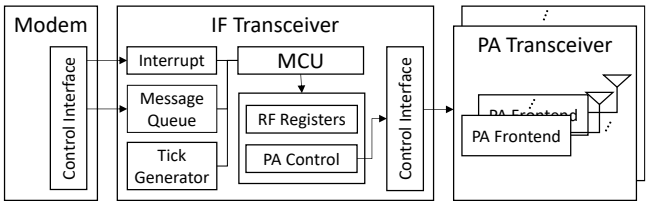

Fig. 2. Overall 5G system block diagram

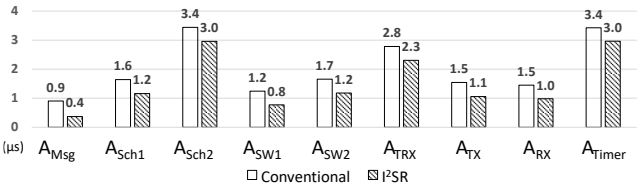

Fig. 3. Average execution time of mmWave applications. Execution time is measured during which the MCU is actively engaged the corresponding application, including context switching, but excluding preempted time.

Lower utilization of I²SR provides opportunities for not only HW, such as power or clock; but also performance by introducing more control on MCU. For instance, in mmWave, it is effective to control multiple PAs for higher performance. However, it makes MCU extremely challenging, considering that the execution count of symbol-wise applications is directly tied to the number of PA. Our mmWave platform supports up to three PA; while the evaluation is conducted only with single PA. Note that most frequent symbol-wise application takes 1.5μs with context switching and the minimum symbol length of mmWave is 4.17μs, I²SR is effective for meeting the latency requirements.

IV. Conclusion

In this paper, we propose a novel I²SR architecture that guarantees zero-latency context switching for the selected interrupt level in a flexible manner. We integrated the I²SR architecture into the transceiver system to satisfy the strict latency requirements of the mmWave application. Our experiment shows that the I²SR architecture reduces the MCU utilization by 30.6% compared to the MCU with conventional context switching. Based on the flexible characteristics of I²SR architecture, we expect I²SR as a solution for any domain suffering from strict timing requirements.

References

[1] NR - Physical channels and modulation – Rel. 15, 3GPP TS 38.211, 2019.

[2] J. Kim et al. "A Flexible Control and Calibration Architecture Using RISC-V MCU for 5G Millimeter-wave Mobile RF Transceivers," 2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), IEEE, 2020.

[3] C.C. Chan, et al. "Open-source and low-cost test bed for automated 5G channel measurement in mmWave band." Journal of Infrared, Millimeter, and Terahertz Waves 40 (2019): 535-556.

[4] R. Balas and L. Benini, "RISC-V for Real-time MCUs - Software Optimization and Microarchitectural Gap Analysis," 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2021.

[5] J. Yiu, The Definitive Guide to the ARMCortex-M3. Elsevier, 2015.

[6] K. Asanovic, et al. "The rocket chip generator." EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17 4 (2016): 6-2.