Real-Time Extension to the RISC-V Advanced Interrupt Architecture

Alexey Khomich[1], Eugeniy Paltsev[1] and Paul Stravers[1]*

[1]Synopsys Inc.

Abstract

The Real-Time Interrupt Architecture (RTIA) is the evolution of RISC-V Advanced Interrupt Architecture (AIA) targeted to extend use cases to real-time latency critical scenarios and/or to resource constrained designs. The RTIA defines necessary mechanisms for interrupt nesting and low latency fixed overhead interrupts handling. Keeping compatibility with RISC-V Privileged ISA and RISC-V AIA, the RTIA provides the same programming model for light weight embedded applications and heavy feature rich systems allowing mixing it in virtual environments.

Introduction

RISC-V provides a wide range of interrupt architecture targeted to various application scenarios. This range includes Core Local Interrupter (CLINT) defined in RISC-V Privilege ISA [2], RISC-V Advanced Interrupt Architecture (AIA) [3], and number of application specific interrupt controllers for small real-time systems such as Core Local Interrupt Controller (CLIC) [4]. All interrupt architectures provide different capabilities and have architecture specific programming models, which makes it difficult to develop universal software and provide proper virtual environment.

Most feature rich interrupt architecture available as the RISC-V ratified standard is the RISC-V AIA [3]. It well supports single/multi-core designs including virtualization extension. Major disadvantages of the AIA are:

- Lack of real-time/safety features such as interrupt preemption, timing guaranties, high software overhead at interrupt enter/leave.
- Significant silicon area of external interrupt controller for single core designs.

This paper describes Synopsys proposed extensions to the RISC-V AIA tailored for real-time/safety applications and resource constrained systems, which enables high-end systems with real-time/safety requirements and as well as low-end real-time embedded systems in virtualized environment.

RISC-V Real-Time Interrupt Architecture

RISC-V Stack Pointer Management Extension

RISC-V ISA places full responsibility on the stack pointer management to the software. This cost significant software overhead at trap entry, when stack pointer may refer to some location that may not even be accessible from the trap handling context. The software trap handler is responsible for switching to an appropriate software stack. This stack pointer management can be moved to the hardware as an optional configurable feature for each privilege level capable of handling the traps. Hardware support for trap stack pointer can be used as the standalone feature and reduces the software overhead in all interrupt handling modes including standard direct and vectored mode.

RISC-V AIA Evolution

Bringing real-time capabilities and keeping full backward compatibility with RISC-V ISA specifications requires multiple updates to the original RISC-V Advanced Interrupt Architecture. The interrupt preemption had been introduced as an extended nested vectored interrupt handling mode in addition to the direct and vectored modes defined by RISC-V ISA. All real-time interrupt signals including major interrupts are routed via the IMSIC controller according to its priorities defined locally (for major interrupts) or externally (for external interrupts). Each external interrupt has an individual entry in the interrupt vector table, that hardware uses when jumps to the software interrupt handler associated with interrupt identity number. The IMSIC registers interface is the only one that is required in nested vectored interrupt handling mode. The IMSIC interrupt identity is treated as priority and preemption level which allows the higher priority interrupts to preempt lower priority handlers. Hardware assists the software by managing the interrupt priority threshold level, preventing lower priority interrupts to preempt higher priorities. Given modifications allow individual interrupt handler for each interrupt source with ability to minimize the interrupt prologue and epilogue depending on registers usage in the handler's body.

In addition to the nested vectored interrupt handling mode, the RTIA specifies the advanced MSI delivery mechanism with delivery timing guaranties. Instead of signaling the interrupt by simple memory operation, the Direct MSI (DMSI) delivery utilizes dedicated high-bandwidth low latency bus to deliver the interrupt signals from APLIC to the multiple IMSIC targets. For the non real-time signals the traditional MSI delivery is supported in parallel with DMSI.

RISC-V AIA Adoption for Resource Constrained Systems

Traditional AIA assumes three levels of controllers:

- External interrupt controller – APLIC.
- Incoming message signaling interrupt controller – IMSIC.
- Core local interrupt controller.

Having such complex interrupt architecture is reasonable for high-end systems but is not applicable for small resource constrained designs. However, it is possible to scale the AIA down to significantly reduce the cost, while maintaining the feature set and programming model. As the first step, the external interrupt gateway (APLIC) functionality is moved to the core. The core major interrupts (except major external interrupt) are replaced with the external interrupts driven by IMSIC-like controller. The local interrupt gateway distributes the external interrupt signals to an IMSIC-file according to the priorities placed into the iCSR-based priorities array. The standard ISA and AIA CSR/iCSR-based interface is used to handle the interrupts. All three RTIA interrupt handling modes are supported, including the nested vectored mode. Depending on target application case the number of external interrupt lines and required interrupt handling modes can be configured to balance the functionality and the design cost.

Table 1: Reduced RTIA Features.

	Direct	Vectored	Nested Vectored
Privileges	M/S-mode	M/S-mode	M/S-mode
Major IRQs	1	1	1
External IRQs	1..1023	1..1023	1..1023
IRQ Priorities	Fixed	Configurable	Configurable
IRQ Levels	1	1	1..255

Conclusions

Presented Real-Time Interrupt Architecture had been implemented in simulation environment, intensively tested and utilized by experimental software, including the real-time operating system and virtualization prototype. Developed software prototypes demonstrate high efficiency of the programming model, significant reduction of the interrupt latency, ability to support various use cases and software architectures. The same interrupt architecture and programming model is used across full range of RISC-V compliant processor designs from Synopsys and demonstrates real-life advantages over ratified RISC-V AIA version keeping full backward compatibility with it.

References

[1] The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA.

[2] The RISC-V Instruction Set Manual, Volume II: Privileged Architecture.

[3] The RISC-V Advanced Interrupt Architecture. Version 1.0

[4] RISC-V Core-Local Interrupt Controller (CLIC).