Technical Committee Update

Kevin Chen
Co-Chair, Fast Interrupt Task Group
Senior Architect
Andes Technology

2019 RISC-V Workshop Taiwan

Introduction

- RISC-V is an open instruction set architecture
- Its instructions and features are openly discussed and defined by members
- If you have great ideas, you are welcome to propose them to the Technical Committee
- Once approved, a Task Group is formed to discuss and come up with a detailed proposal
- The final proposal will be ratified by all member companies to become the official specification

Task Group

- **Each Task Group (TG) is run by a chair and a co-chair**
  - It is responsible to collect ideas from task group members and form a consensus
  - There will be regular meetings or discussion threads through workspace/e-mail
- **This open discussion is the best way to combine the talents and expertise from different companies**
- **You are all encouraged to join task groups of interest to contribute your valuable ideas**
- **Currently, there are 17 active task groups and more are coming**

List of Task Groups

- **Base ISA**
  - Going through the ratification process and will soon be voted by member companies
- **Fast Interrupts**
  - Focuses on developing a low-latency, vectored, priority-based, preemptive interrupt scheme
  - Ideal for MCU and real-time systems
- **P Extension**
  - Defines Packed-SIMD DSP extension instructions operating on integer registers
  - Also defines compiler intrinsic functions for high-level programming languages

List of Task Groups (cont.)

- **Privileged Spec**
  - Just ended the public review period with no major issues
- **SW Tool Chain**
  - There is a dedicated session tomorrow
- **Vector Extensions**
  - This is very popular due to recent AI applications
  - Recently released a draft spec. 0.7
- **Processor Trace**
  - Standardize a hardware interface and a packet/data format for trace encoders

List of Task Groups (cont.)

- **Debug**
  - Ratified spec is now available on riscv.org
  - Errata rolled into ratified spec.
- **Trusted Execution Environment**
- **Cryptographic Extensions**
- **Bit Manipulation**
- **Compliance**
- **Formal Spec**

List of Task Groups (cont.)

- **J Extension**
  - aims to make RISC-V an attractive target for languages that are interpreted or JIT compiled
- **Memory Model**
- **Opcode Space Management**
- **Sv128**
  - Defines a 128-bit virtual address space

Update on Fast Interrupt TG

- **As the co-chair, I like to give abrief update on the Fast Interrupt TG**
  - The spec. is most finished with some details to be finalized
- **A new scheme Core-Local Interrupt Controller (CLIC) is proposed**
  - It has very low latency and is ideal for MCU or real-time systems
- **Main features**
  - Supports up to 4096 interrupt sources
  - Supports up to 256 priority levels
  - Automatic HW preemption
  - Selective Hardware Vectoring

Selective Hardware Vectoring

- **Andes has proposed this feature to let users select the behavior of each interrupt: either hardware vectored or non-vectored.**
  - It allows optimization for each interrupt to gain benefits from both behaviors
- **Hardware vectored**
  - Pro: faster interrupt response
  - Con: bigger code size (to save/restore contexts)
- **Non-vectored**
  - Pro: smaller code size (sharing same common code to save/restore)
  - Con: slower interrupt response
- **Typical usage: critical interrupts are vectored while others are non-vectored**

Additional Optimization

- **To avoid creating new instructions and hence extra work for toolchain, all new features only use special control and status registers (CSRs)**
- **A special CSR to find the pending interrupt with the highest priority (and disable/enable interrupts)**
  - This optimizes the performance of non-vectored interrupts
- **A special CSR to conditionally swap the stack pointer based on the privilege mode**
  - To speed up OS switching mode
- **A special CSR to conditionally swap the stack pointer based on the interrupt level**
  - For RTOS to switch between interrupt and non-interrupt code

Thank you