Hello, and welcome to this presentation of the STM32 Global interrupt controller (GIC).
The Global Interrupt Controller embedded inside the STM32MP1 microprocessor provides up to 288 interrupt channels with support of TrustZone and Virtualization. Application can benefit from dynamic prioritization of the interrupt levels, ARM TrustZone security extension and, for multi-core product, the interrupt distribution to each core.
The Global Interrupt Controller is tightly coupled with the ARM Cortex-A7 processor and provides a dynamic reprioritization of interrupt requests, allowing an application to better serve the incoming events. Most of the peripherals have a unique interrupt line, making the development of the application easier (no need to determine the source of an interrupt during the interrupt handling). In case of multicore Cortex-A7 processor, each interrupt can be distributed to a single or both cores. GIC also fully supports ARM TrustZone security and virtualization extensions.
The GIC receives up to 256 interrupts from STM32MP15 peripherals shared between the two Cortex-A7 cores. In addition, up to 16 internal private peripheral interrupts (PPI) and up to 16 software generated interrupts (SGI) are managed by each Cortex-A7 core.

The GIC signals interrupt to each core with FIQ and IRQ. In a usual software implementation, FIQ is dedicated to secure interrupts.

The GIC also provides virtual control and interface for each Cortex-A7 core, helping hypervisor and virtual machine implementation.
Here is the list of the different type of interrupt sources managed by the Global Interrupt Controller.

- **Software Generated Interrupts (SGI)**
  - Generated by writing to the Software Generated Interrupt Register
  - A maximum of 16 SGIs, ID0-ID15, can be generated for each core interface
  - ARM recommend to use SGI ID0-ID7 for non-secure interrupts and SGI ID8-ID15 for secure interrupts

- **Private Peripheral Interrupts (PPI)**
  - A PPI is an interrupt generated by a peripheral that is specific to a single core
    - Secure Physical Timer event (PPI1)
    - Non-secure Physical Timer event (PPI2)
    - Virtual Timer event (PPI4)
    - Hypervisor Timer event (PPI5)
    - Virtual Maintenance Interrupt (PPI6)
The shared peripheral interrupts are the usual STM32MP15 interrupts coming from peripherals (e.g. UART). These interrupts are shared between the multiple Cortex-A7 cores and can generate interrupt to one or both cores. Note that an interrupt does not have programmable polarity and can be either rising-edge or high-level sensitive. Within FIQ or IRQ interrupt routine, interrupts pending is identified inside the GIC by an ID number from 0 to 287.
There is up to 32 interrupt priority levels. GIC includes a mechanism to avoid non-secure interrupt to get higher priority than secure interrupt. Using half of the priority range for non-secure interrupts allows the secure interrupts to use the full range of priority and choosing which secure interrupt could be protected from non-secure interruption. In that case, non-secure software can only see the lower 16 possible interrupt priority levels. Finally, the GIC distributor selects which priority is presented to each core.

- **Interrupt priority**
  - The GIC in STM32MP15x microprocessor implements a 5-bit version of the interrupt priority field for 32 interrupt priority levels
    - ARM strongly recommends that Group1 interrupts (non-secure) have assigned priority values in the upper half priority value range (highest number is lower priority)

- **GIC Distributor**
  - Centralizes all interrupt sources,
  - For each core, forwards the interrupt with the highest priority to the CPU interface for priority masking and preemption handling
There is a separated CPU interface for each Cortex-A7 core which presents either shared or banked register and either secure or non-secure register view.
The Global Interrupt Controller fully supports TrustZone security and allows a flexible implementation of secure and non-secure interrupts.
Virtualization is a system supporting more than one operating system at a time (or multiple independent instances of the same operating system). The GIC helps support of such virtualization, by example, by providing virtual interrupts to a virtual machine.
For detailed information, please refer mainly to the ARM Generic Interrupt Controller Architecture Specification (ARM IHI 0048).
Please visit also the MP1 Wiki pages for details.