MIPS-VZ Security Features as Compared to ARMv8-M CMSE **October 31, 2016**

Overview

ARMv8-M Security Extensions (CMSE) is based on Trust-zone (TZ). Similar to other TZ-based systems, there is a single Secure World and a single Non-Secure World. That means all code that is meant to be protected must mutually trust each other, as they all execute from the same shared single Secure World address space.

MIPS-VZ is based on hardware based CPU virtualization, which allows for multiple domains, each separate and protected from each other. This has the benefit that each piece of software that is meant to be protected need not trust other pieces of software, as each can have a private domain to itself. The MIPS-VZ architecture allows up to 255 separate domains, while particular CPU implementation might implement a lower number of domains.

Execution Modes

Both ARMv8-M CMSE and MIPS-VZ use additional execution modes to give additional privilege to the most trusted code and reduce privilege to less trusted code.

ARMv8-M adds the Secure-Handler and Secure-Thread (most trusted) execution modes to the NonSecure-Handler and NonSecure-Thread (least trusted) states.

Similarly, MIPS-VZ adds the Root-Kernel (most trusted) and Root-User execution modes to the Guest-Kernel and Guest-User (least trusted) modes.

To draw an equivalence between ARMv8-M CMSE and the MIPS-VZ model:
The MIPS model likewise provides a Secure World, and in addition, provides multiple "Normal" worlds, each isolated from the other.

Memory Management

ARMv8-M CMSE uses two blocks to resolve access to memory, the SAU (Security Attribute Unit), and an MPU (Memory Protection Unit) banked as secure and non-secure. The SAU resolves access by secure/non-secure mode. After this check is done, the MPU resolves access by privilege level within the mode.

MIPS-VZ uses an MMU that has two configurations:

1. A two-level TLB for larger systems.
    a. The Guest TLB is used by the Guest OSes.
    b. The Root TBL is used by the Root-kernel software (hypervisor).

This is the typical virtualization operating model, where Guests may be OSes and thus OS level isolation is provided.

2. A two-level MMU for smaller systems.

    a. The Guest portion can be a full TLB or a Fixed-Mapping-Translation (FMT) Unit.

    b. The Root-level portion is a stripped down TLB without the physical address, called the Root Protection Unit (RPU). Software management of the RPU is similar to a TLB. The model is for secure software to program the RPU before initializing any Guests (untrusted). There is no expectation that misses be resolved through the refill handler i.e., by demand-paging, though a handler may opt to do the same.

For the case where a Guest MMU is an FMT, and Root MMU is an RPU, application level isolation is provided. That is, the Guests are really partitioned applications/code all operating in the common privilege level of guest mode. This is a more static/constrained environment. An example of this is when an OEM manufactures and provisions a chip with its own proprietary software, which can be isolated either in root context or as a Guest. A VAR (value added reseller) will add its own software and perhaps even 3rd-party software, all of which is kept isolated from each other within the Guest context.

The MIPS-VZ MMU is CAM based, while the ARMv8-M SAU/MPU is base-bound based (meaning address comparators). The CAM-based MMU entry will be smaller in die-area than the comparator-based MMU entry. The MIPS Virtualization solution thus offers a fully scalable solution from application to OS isolation.

Routine Calls between Different Security Regions

ARMv8-M CMSE adds the ability of Non-Secure Code to call specific routines within the Secure World without the overhead of a Secure Monitor Call, which involves the high overhead of saving/restoring the architecture state.

This is done by:

a) Using the SG (Secure Gateway) instruction to denote a legal entry point in the Secure World address space.

b) The usage of the SAU to denote that the accessed routine resides in a region where execution in Non-Secure-World mode is allowed. Such regions are called Non-Secure-Callable (NSC) regions.

c) The SG instruction followed by a branch to a secure function in Secure memory, if allowed.

In MIPS-VZ, the equivalent ability is to share routines between multiple Guests/ Domains. This would be done by programming the Root portion of the MMU to create pages which are shared among the desired Guests. The code could be protected by making the shared page to be execute-only (neither readable nor writable). The Root MMU imposes permission checks in the form of Write-Inhibit, Execute-Only, and Read-Inhibit.

For execution speed and latency, the MIPS-VZ method is faster as there is no need for the SG instruction and the branch instruction that follows the SG instruction upon entry of the called routine.

Further, the protection provided by the MIPS Virtualization model is imposed by Root context software or secure boot code, and is thus for the most part transparent to the application. That is, the programming model is similar to that of a non-virtualized system, but with security controls. Root software completely monitors Guest activity without giving up control of the operating environment.

The use of the SG entry point instruction disallows jumping into the middle of routines. While this is a good feature, its impact on the security of the entire system is minimal. The reason for this is that this entry-point identification feature only exists for these small NSC regions. The vast majority of the system memory will be "regular" Non-Secure and Secure regions where such entry-point identification is not done.

The use of the SG entry point instruction allows for more granular use of the address space—some code being available to the Non-Secure world through the usage of the SG instruction and other code/data not accessible to the Non-Secure world, due to the lack of the SG instruction. In MIPS VZ, some of this granularity can be implemented by using the different page sizes that are available in the MIPS MMU.

Interrupt Handling

ARMv8-M CMSE supports the faster ARM interrupt schemes to be used when running in Secure execution mode.

Similarly, MIPS-VZ supports all of the faster MIPS interrupt schemes as well. Interrupts can be assigned to each Guest directly, where the interrupt can be handled without any intervention from Root-Kernel software/Hypervisor or additional security, interrupts can be directly assigned to Root.

SOC Security

ARMv8-M CMSE supports the traditional TZ HNONSEC fabric signal and HPROT memory attribute signal to tell IO devices which entity (secure or non-secure) is making the memory request.

MIPS-VZ supports the multi-bit GuestID signals for the same purpose. Since this is a multi-bit field, the individual Guest can be identified and the IO device can have specific behaviors for that particular Guest i.e., support for multiple protection domains can be extended throughout the SOC.

Context Switching

The TrustZone Secure World↔Non Secure World context switching requires saving/restoring the GPRs and FPRs.

The closest functional equivalent to this type of context switching in MIPS-VZ is Root mode↔Guest Mode context switching. Here there is no need for COP0 saving/restoring as both the Root and Guest have their own copies of COP0 (COP0 is the MIPS architecture structure holding the CPU privileged control/status registers, exception logic registers, MMU). For this case, the GPRs and FPRs are required to be saved and restored. As mentioned above, for the M5150, the Shadow Register Set feature can be used in combination with the CPU virtualization. A GPR Shadow set can be assigned to each Guest,

removing the need to save/restore the GPRs. Alternatively, the MIPS Architecture also defines virtualization of multi-threading capability in a core i.e., multiple independent contexts can be time-multiplexed in execution in a core while providing complete isolation without the need for saving/restoring any state.

Since MIPS-VZ allows multiple domains/guests, there is also the case of the Guest1↔Guest2 context switching. Each of these Guests can be an operating system—with each OS potentially managing COP0. For this case, the GPRs, FPRs and COP0 would have to be save/restored. Again, the Shadow Register Set feature can be used to mitigate the impact of GPR save and restore.

In contrast, TrustZone does not have the capability of running multiple Guest Operating systems. Instead, the TZ system is limited to running multiple Secure World applications under a shared TZ RTOS or TZ executive.

Summary

The MIPS Virtualization Architecture provides a simple but complete and robust security solution that scales from application to OS level isolation to flexibly meet the needs of customers in a world with ever-changing security threats.