CHAPTER 14
APIC-TIMER VIRTUALIZATION

This chapter describes a VMX extension called **APIC-timer virtualization**.

The new feature virtualizes the TSC-deadline mode of the APIC timer. When this mode is active, software can program the APIC timer with a deadline written to the IA32_TSC_DEADLINE MSR. A timer interrupt becomes pending when the logical processor's timestamp counter (TSC) is greater or equal to the deadline.

APIC-timer virtualization operates in conjunction with the existing virtual-interrupt delivery feature. With that feature, a virtual-machine monitor (VMM) establishes a virtual-APIC page in memory for each virtual logical processor (vCPU). A logical processor uses this page to virtualize certain aspects of APIC operation for the vCPU.

The feature is based on new guest-timer hardware that introduces two new architectural features: **guest-timer events** and a **guest deadline**. With APIC-timer virtualization, guest writes to the IA32_TSC_DEADLINE MSR do not interact with the APIC (or its timer) but instead establish a guest deadline to arm the guest-timer hardware. When a logical processor's TSC is greater than or equal to the guest deadline, a guest-timer event becomes pending. Processing of a guest-timer event updates the virtual-APIC page to record the fact that a new virtual interrupt is pending.

Section 14.1 presents the new guest-timer hardware, focusing on the guest deadline and guest-timer events. Section 14.2 identifies new VMCS support (a new control and new fields). Section 14.3, Section 14.4, and Section 14.5 detail the changes to VM entries, VMX non-root operation, and VM exits, respectively.

14.1 GUEST-TIMER HARDWARE

A logical processor supports APIC-timer virtualization using new guest-timer hardware. Software controls this hardware using an unsigned 64-bit value called the **guest deadline**. (There is a separate guest deadline for each logical processor.) If the guest deadline is non-zero, a guest-timer event will be pending when the timestamp counter (TSC) reaches or exceeds the guest deadline.

Section 14.1.1 describes how the guest-timer hardware responds to updates to the guest deadline. Section 14.1.2 presents details of the new guest-timer events.

14.1.1 Responding to Guest-Deadline Updates

Subsequent sections specify the operations that modify the guest deadline. The processor enforces the following:

- Modifying the guest deadline to have value zero disables guest-timer events. After this, no guest-timer event will be pending before the next modification of the guest deadline.
- Modifying the guest deadline to have a non-zero value less than or equal to the TSC causes a guest-timer event to be pending at the next instruction boundary.
- Modifying the guest deadline to have a non-zero value greater than the TSC arms the guest timer. After this, no guest-timer event will be pending before the TSC reaches the guest deadline (unless the guest deadline is modified again). A guest-timer event will become pending when the TSC reaches the guest deadline.

Races may occur if the guest deadline is modified when the value of the TSC is close to that of the guest deadline. In such a case, either of the following may occur:

- The TSC may reach the original guest deadline before the guest deadline is modified, causing a guest-timer event to be pended. Either of the following may occur:
  - If the guest-timer event is processed before the guest deadline is modified, the logical processor will clear the deadline (as part of event processing) before the deadline is modified. The new deadline may cause a second guest-timer event to occur later.
  - If the guest deadline is modified before a guest-timer event can be processed, no guest-timer event based on the original deadline will occur, and any subsequent guest-timer event will be based on the new guest deadline.

- The guest deadline may be modified before the TSC reaches the original guest deadline. In this case, no guest-timer event will occur based on the original guest deadline, and any subsequent guest-timer event will be based on the new guest deadline.

14.1.2 Guest-Timer Events

A guest-timer event becomes pending when the guest deadline is non-zero and is less than or equal to the TSC.

A logical processor in the wait-for-SIPI state or the shutdown state inhibits guest-timer events.

Guest-timer events have priority just below that of external interrupts (and above that of virtual interrupts or interrupt-window exiting).

A pending guest-timer event that is not inhibited or preempted by higher-priority events is processed by the logical processor as described in Section 14.4.2.

The remainder of this chapter should make clear that the guest deadline is always zero outside VMX non-root operation and thus a guest-timer event can become pending only if in VMX non-root operation.

14.2 VMCS SUPPORT

Section 14.2.1 identifies a new VM-execution control to enable the APIC-timer virtualization feature. Section 14.2.2 enumerates new fields added to the VMCS to support the feature.

14.2.1 New VMX Control

This feature introduces a new VM-execution control called "APIC-timer virtualization." It is tertiary processor-based VM-execution control 8.

Setting this control enables guest-timer events based on the guest deadline. See Section 14.3.2 and Section 14.4.1.

14.2.2 New VMCS Fields

This feature introduces three new VMCS fields:

- **Guest deadline** is a new 64-bit guest-state field. Software can access this field with VMREAD or VMWRITE using the encoding pair 2830H/2831H.
- **Guest deadline shadow** is a new 64-bit VM-execution control field. This is the guest deadline relative to the guest's virtualized view of the TSC. See Section 14.4.1 for details. Software can access this field with VMREAD or VMWRITE using the encoding pair 204EH/204FH.
- **Virtual timer vector** is a new 16-bit VM-execution control field. The low 8 bits of this field contain the vector used for virtual timer interrupts. Software can access this field with VMREAD or VMWRITE using the encoding 000AH.

14.3 CHANGES TO VM ENTRIES

This section describes changes to the operation of VM entries related to this new feature. Changes include new checking of certain VMX controls (Section 14.3.1) and possible loading of the guest deadline (Section 14.3.2).

14.3.1 Checking VMX Controls

If the "APIC-timer virtualization" VM-execution control is 1, VM entry ensures that the following all hold:

- The "virtual-interrupt delivery" VM-execution control is 1.
- The "RDTSC exiting" VM-execution control is 0.

- The value of the virtual timer vector is at most 255.

If any of those is not the case, VM entry fails. Control is passed to the next instruction, RFLAGS.ZF is set to 1 to indicate the failure, and the VM-instruction error field is loaded with value 7, indicating "VM entry with invalid control field(s)."

(This check may be performed in any order with respect to other checks on VMX controls and the host-state area. Different processors may thus give different error numbers for the same VMCS.)

14.3.2 Loading the Guest Deadline

If the "APIC-timer virtualization" VM-execution control is 1, VM entry loads the guest deadline from the corresponding field in the guest-state area of the VMCS. If the value loaded is non-zero, a guest-timer event may become pending, as described in Section 14.1.1.

If the "APIC-timer virtualization" VM-execution control is 0, the guest deadline is not loaded and its value remains zero. As a result, no guest-timer event will be pending after the VM entry.

14.4 CHANGES TO VMX NON-ROOT OPERATION

The 1-setting of the "APIC-timer virtualization" VM-execution control changes how a logical processor responds to accesses to the IA32_TSC_DEADLINE MSR. These changes are described in Section 14.4.1. In addition, the 1-setting of that control may result in the processing of guest-timer events, as is detailed in Section 14.4.2.

14.4.1 Accesses to the IA32_TSC_DEADLINE MSR

If the "APIC-timer virtualization" VM-execution control is 1, the operation of reads and writes to the IA32_TSC_DEADLINE MSR (MSR 6E0H) is modified:

- Any read from the IA32_TSC_DEADLINE MSR (e.g., by RDMSR) that does not cause a fault or a VM exit returns the value of the guest deadline shadow (from the VMCS).
- Any write to the IA32_TSC_DEADLINE MSR (e.g., by WRMSR) that does not cause a fault or a VM exit is treated as follows:
  - The source operand is written to the guest deadline shadow (updating the VMCS).
  - If the source operand is zero, the guest deadline (the value that controls when hardware generates a guest time event) is cleared to 0.
  - If the source operand is not zero, the guest deadline is computed as follows. The source operand is interpreted as a virtual deadline. The processor converts that value to the actual guest deadline based on the current configuration of TSC offsetting and TSC scaling.

  (See Section 14.1.1 for how a logical processor responds to such updates to the guest deadline.)

  Note that when the "APIC-timer virtualization" VM-execution control is 1, such writes do not change the value of the IA32_TSC_DEADLINE MSR nor do they interact with the APIC timer in any way.

When the "APIC-timer virtualization" VM-execution control is 0, reads and writes of the IA32_TSC_DEADLINE MSR operate as they would on processors that do not support the new feature. In this case, there is no way to read or write the guest deadline, and it is always zero.

14.4.2 Processing of Guest-Timer Events

As explained in Section 14.1.2, a pending guest-timer event that is not inhibited or preempted by higher-priority events is processed by the logical processor. This section provides details of that processing.

Processing of a guest-timer event updates the virtual-APIC page to cause a virtual timer interrupt to become pending. Specifically, the logical processor performs the following steps:

```
V := virtual timer vector;
VIRR[V] := 1;// update virtual IRR field on virtual-APIC page
RVI := max{RVI, V};// update guest interrupt status field in VMCS
evaluate pending virtual interrupts;// a virtual interrupt may be delivered immediately after this processing
Guest deadline := 0;
Guest deadline shadow := 0;
```

The following items consider certain special cases:

- If a guest-timer event is processed between iterations of a REP-prefixed instruction (after at least one iteration has completed but before all iterations have completed), the following items characterize processor state after the steps indicated above and before guest execution resumes:
  - RIP references the REP-prefixed instruction;
  - RCX, RSI, and RDI are updated to reflect the iterations completed; and
  - RFLAGS.RF = 1.
- If a guest-timer event is processed after partial execution of a gather instruction or a scatter instruction, the destination register and the mask operand are partially updated and RFLAGS.RF = 1.
- If a guest-timer event is processed while the logical processor is in the state entered by HLT, the processor returns to the HLT state after the steps indicated above (if a pending virtual interrupt was recognized, the logical processor may immediately wake from the HLT state).
- If a guest-timer event is processed while the logical processor is in the state entered by MWAIT, TPAUSE, or UMWAIT, the processor will be in the active state after the steps indicated above.
- A guest-timer event that becomes pending during transactional execution may abort the transaction and result in a transition to a non-transactional execution. If it does, the transactional abort loads EAX as it would had it been due to an interrupt.
- A guest-timer event that occurs while the logical processor is in enclave mode causes an asynchronous enclave exit (AEX) to occur before the steps indicated above.

14.5 CHANGES TO VM EXITS

This section describes changes to the operation of VM exits related to this new feature.

On a processor that supports the 1-setting of "APIC-timer virtualization" VM-execution control, every VM exit saves the value of the guest deadline into the corresponding field in the guest-state area of the VMCS and then clears the guest deadline to zero. This implies that, if "APIC-timer virtualization" is 0, a VM exit will overwrite the guest-deadline field in the VMCS with zero, and the previous value of that field will be lost.

Since VM exits always result in the guest deadline being zero and the guest deadline must remain zero until the next VM entry, guest-timer events are pending only VMX non-root operation (and only if the "APIC-timer virtualization" VM-execution control is 1).