Intel® Virtualization Technology FlexMigration Application Note

This document is intended only for VMM or hypervisor software developers and not for application developers or end-customers. Readers are expected to be knowledgeable about Intel® Architecture and Intel® Virtualization Technology.

323850-003a

July 2011

CHAPTER 1 INTRODUCTION

1.1 OVERVIEW

Intel® Virtualization Technology (**Intel® VT**) refers to a suite of hardware assists in Intel silicon (the processors, chipsets and networking devices) that make virtualization software simpler and robust and enable it to deliver better performance.

Intel® Virtualization Technology FlexMigration (Intel® VT FlexMigration) is a part of Intel VT that allows a **virtual-machine monitor** (**VMM**) to report a consistent set of available processor features to guest software running in a **virtual machine** (**VM**), thereby broadening the live-migration compatibility pool across generations of Intel® processors. The eventual live-migration solution built across servers using different generations of Intel processors is defined by the VMM vendor and validated by the VMM provider. Intel primarily provides the hardware hooks to enable the VMM vendors to implement such a solution. Hence Intel VT FlexMigration, combined with support from a virtualization-software provider, allows IT to maximize flexibility across a single pool of virtualization resources using multiple generations of Intel server products.

CHAPTER 2 INTEL® VIRTUALIZATION TECHNOLOGY FLEXMIGRATION AND CPUID VIRTUALIZATION

Software discovers the processor identification and feature information using the **CPUID** instruction. Intel VT FlexMigration is an umbrella term that refers to processor features that allow a VMM to **virtualize** the CPUID instruction. That is, these features give a VMM control over the values reported by CPUID to guest software running in a VM.

2.1 CPUID VIRTUALIZATION AND A LIVE-MIGRATION COMPATIBILITY POOL

Without CPUID virtualization, a VMM may be challenged in its ability to support a live-migration compatibility pool that includes processors that support different features. Suppose, for example, that such a pool contains processor P1, which supports feature X, and processor P2, which does not. If guest software begins operating in a VM running on processor P1, the CPUID instruction will report feature X (which is supported on processor P1). If the VM is later migrated to processor P2, guest software will not operate correctly if it attempts to use feature X (which is not supported on processor P2).

A VMM can avoid this problem with appropriate CPUID virtualization. Specifically, a VMM can virtualize the CPUID instruction so as not to report any feature that is not supported by every processor in the live-migration compatibility pool. In the example above, the VMM would virtualize the execution of CPUID on processor P1 so as **not** to report feature X. (It would do so, despite the fact that processor P1 does support feature X, because processor P2 does not.) As a result, guest software will not attempt to use feature X and will thus continue to operate correctly after being migrated to processor P2.

Support for a live-migration compatibility pool does not require arbitrary CPUID virtualization. As noted in the previous paragraph, it is sufficient to conceal features that are not supported by all processors in the live-migration compatibility pool. It is not necessary change the reporting of other processor capabilities (e.g., cache sizes).

Intel VT FlexMigration provides features that enable CPUID virtualization so as to support a live-migration compatibility pool that includes processors that support different features. It includes features both for VMs that operate in VMX non-root operation and those that do not.[1] Intel VT FlexMigration supports **CPUID-induced**

1. VMX non-root operation is a feature of Intel VT for Intel® 64 and IA-32 Architectures (Intel® VT-x).

VM exits for VMs that operate in VMX non-root operation. On certain processors, Intel VT FlexMigration may support either of two other features for VMs that do not operate in VMX non-root operation: **CPUID masking** and **CPUID faulting**. VMM providers can claim support of Intel VT FlexMigration if at least one of these features is enabled to implement a cross-generation live-migration capability.

2.2 Intel® Virtualization Technology FlexMigration for VMs That Operate in VMX Non-Root Operation

VMX non-root operation is a feature of Intel VT for Intel® 64 and IA-32 Architectures (Intel® VT-x). VMs that are supported by Intel VT-x operate in VMX non-root operation. Intel VT-x is defined so that any execution of the CPUID instruction in VMX non-root operation causes a transition to the VMM. These transitions are called **CPUID-induced VM exits**.

The CPUID instruction reports processor identification and feature information to software by returning values in the registers EAX, EBX, ECX, and EDX. A VMM handling CPUID-induced VM exits can emulate execution of the CPUID instruction by placing whatever values it chooses in those registers. When the VMM returns control to the VM, it will appear to guest software that the CPUID instruction returned the values that were chosen by the VMM.

Because every execution of the CPUID instruction in VMX non-root operation causes a VM exit, and because the VMM may put any values it wants in the registers EAX, EBX, ECX, and EDX, a VMM has complete control over execution of the CPUID instruction in any VM supported by Intel VT-x. If the VMM is supporting a live-migration compatibility pool, it can emulate the CPUID instruction in such VMs so as not to report any feature that is not supported by every processor in the live-migration compatibility pool. As explained in Section 2.1, this approach supports correct guest operation even after live migration.

CPUID-induced VM exits are an integral part of Intel VT-x and are an architecturally committed feature. They are and will be supported on all Intel processors that support Intel VT-x. CPUID-induced VM exits are documented in *Intel® 64 and IA-32 Architectures Software Developer's Manual* (SDM) in Volume 3B, Chapter "VMX Non-Root Operation", Section "Instructions That Cause VM Exits Unconditionally". This volume may be downloaded at `http://download.intel.com/design/processor/manuals/253669.pdf` .

2.3 Intel® Virtualization Technology FlexMigration for VMs That Do Not Operate in VMX Non-Root Operation

A VMM might choose not to use Intel VT-x for some or all of its VMs. It might instead support them using software techniques based on binary translation or paravirtualization. (Such techniques can also be used for VMs that are supported by Intel VT-x.) Because such VMs do not run in VMX non-root operation, the VMM cannot use CPUID-induced VM exits to emulate executions of the CPUID instruction in those VMs. To compensate for this, certain Intel processors support **CPUID masking**, described in Section 2.3.1. More recent processors may instead support **CPUID faulting**, described in Section 2.3.2.

2.3.1 CPUID Masking

CPUID masking is a hardware assist that enables such VMMs to implement a cross-generation live-migration capability on platforms based on supporting processors. Unlike CPUID-induced VM exits, CPUID masking does not give a VMM complete control over guest executions of the CPUID instruction. It does, however, allow partial virtualization of the CPUID instruction so as to prevent reporting any feature that is not supported by every processor in the live-migration compatibility pool. As noted in Section 2.1, this ability is required to implement a cross-generation live-migration capability.

The CPUID instruction reports information to software using a set of **CPUID functions**. Each function reports four (4) 32-bit values, and these are reported in the registers EAX, EBX, ECX, and EDX.

The value in the EAX register at the time the CPUID instruction is executed determines the function that the instruction reports.[1] The notation CPUID.*nn*.*reg* refers to the value returned in register *reg* by CPUID function *nn*. For example, CPUID.01H.EDX is the value that the CPUID instruction places in the register EDX when the register EAX contained the value 01H at the time the instruction was executed.

CPUID function 0DH reports information using multiple sub-functions that are selected by the value of ECX. The notation CPUID.(EAX=0DH,ECX=*nn*):*reg* refers to the value returned in register *reg* by sub-function *nn* of CPUID function 0DH.

The CPUID instruction returns information about ISA features in the following values: CPUID.01H.ECX, CPUID.01H.EDX, CPUID.(EAX=0DH,ECX=01H):EAX, CPUID.80000001H.ECX, and CPUID.80000001H.EDX. (Not all of the processors that support CPUID masking use all of these CPUID functions.) Further details about the CPUID instruction are given in *Intel® 64 and IA-32 Architectures Software Developer's Manual* (SDM) in Volume 2A, Chapter "Instruction Set Reference, A-M", Section "Instructions (A-M)". This volume may be downloaded at

`http://download.intel.com/design/processor/manuals/253666.pdf` .

As examples, Figure 2-1 identifies the features reported by CPUID.01H.ECX, and Figure 2-2 does the same for CPUID.01H.EDX.

1. For some CPUID functions, the ECX register at the time the CPUID instruction is executed selects a sub-function, determining with the EAX value the values returned in EAX, EBX, ECX, and EDX. There is no CPUID masking for such functions and, for that reason, the remainder of this document does not consider CPUID sub-functions.

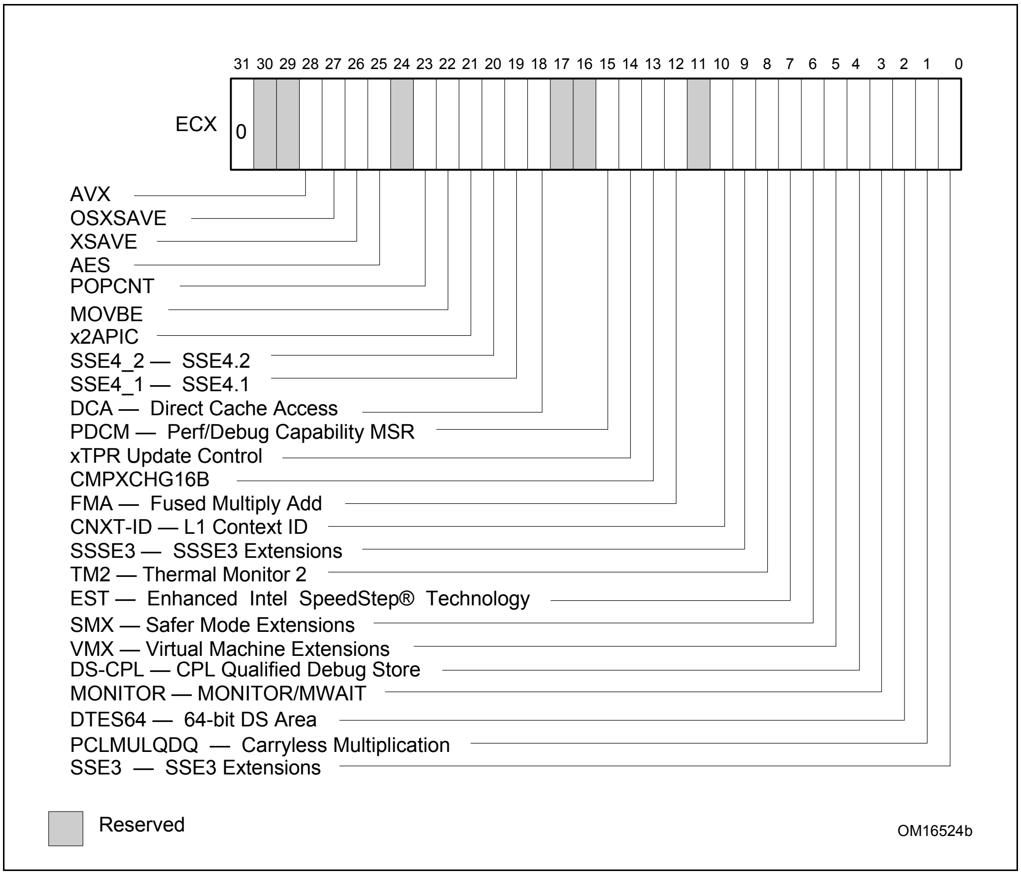

Figure 2-1. Feature Information Returned by CPUID.01H.ECX

CPUID masking allows VMM software to prevent the processor from reporting selected features in the values CPUID.01H.ECX, CPUID.01H.EDX, CPUID.(EAX=0DH,ECX=1H):EAX, CPUID.80000001H.ECX, and CPUID.80000001H.EDX. VMM software controls CPUID masking by programming the model-specific registers (MSRs). Specifically, there are three **CPUID-masking MSRs**:

- CPUID1_FEATURE_MASK MSR.
  This MSR is supported by all processors that support CPUID masking. Its contents affect the values reported by CPUID function 01H:
  - — The value loaded into ECX is the logical-AND of the value that would normally be reported for CPUID.01H.ECX and bits 31:0 of this MSR.
  - — The value loaded into EDX is the logical-AND of the value that would normally be reported for CPUID.01H.EDX and bits 63:32 of this MSR.

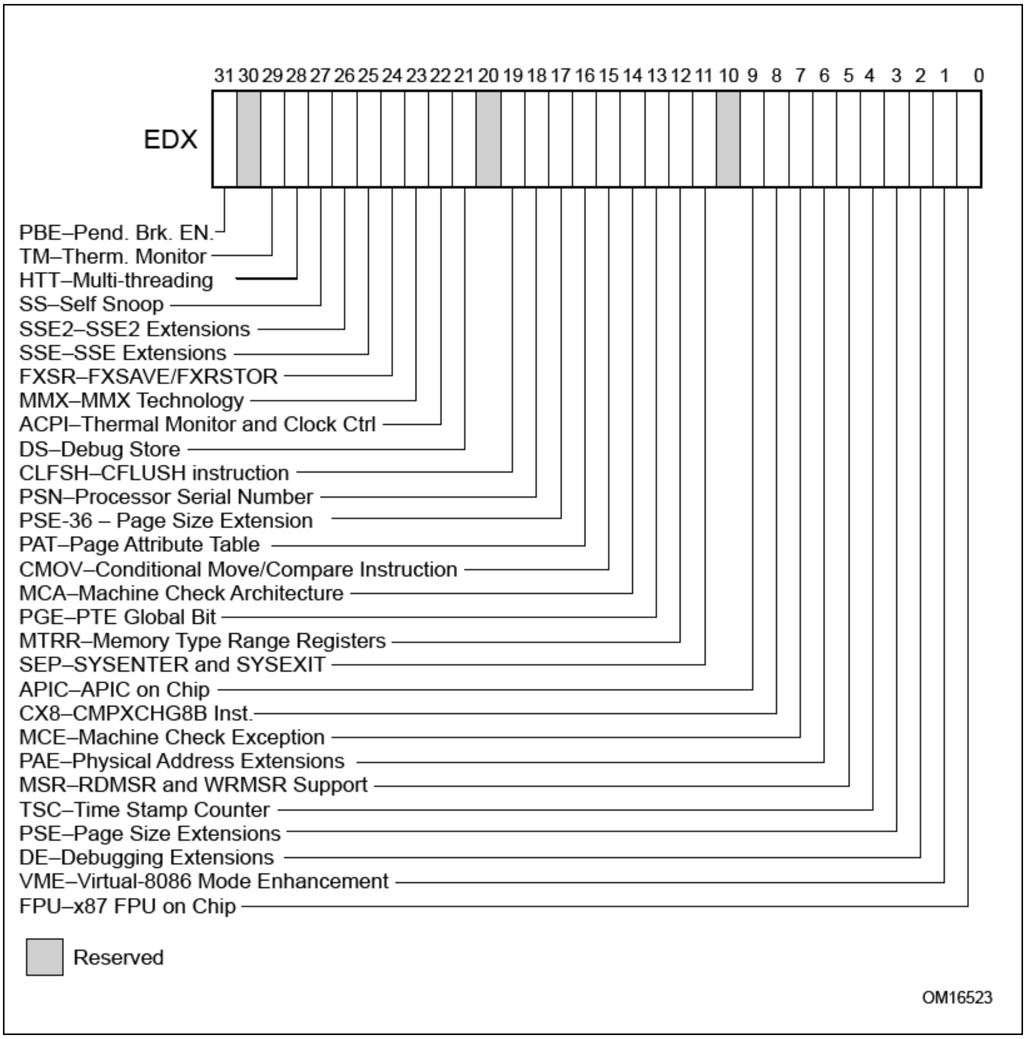

Figure 2-2. Feature Information Returned by CPUID.01H.EDX

All processors that support CPUID masking support CPUID1_FEATURE_MASK MSR.

- CPUIDD_01_FEATURE_MASK MSR.
  This MSR is supported by some but not all processors that support CPUID masking. Its contents affect the values reported by sub-function 01H:
  - The value loaded into EAX is the logical-AND of the value that would normally be reported for CPUID.(EAX=0DH,ECX=01H):EAX and bits 31:0 of this MSR.

    - Bits 63:32 of this MSR are reserved and software should not change its initial value.
- CPUID80000001_FEATURE_MASK MSR.
  This MSR is supported by some but not all processors that support CPUID masking. Its contents affect the values reported by CPUID function 80000001H:
    - The value loaded into ECX is the logical-AND of the value that would normally be reported for CPUID.80000001H.ECX and bits 31:0 of this MSR.
    - The value loaded into EDX is the logical-AND of the value that would normally be reported for CPUID.80000001H.EDX and bits 63:32 of this MSR.

(The locations of the CPUID-masking MSRs are model-specific; they are provided later for specific processors.)

CPUID masking does not give VMM software full control over CPUID virtualization. While it allows a VMM to prevent the processor from reporting a feature that is supported by the processor, it does not allow a VMM to cause the processor to report a feature that is not supported by the processor. In addition, a VMM cannot use CPUID masking to prevent the processor from reporting native hardware capabilities that are reported in other CPUID functions (e.g., cache sizes). Even with these limitations, CPUID masking provides the capability identified in Section 2.1 as required to implement a cross-generation live-migration capability.

Software can identify support for CPUID masking by consulting CPUID.01H.EAX, which reports the following version information: (1) extended family ID in bits 27:20; (2) family ID in bits 11:8; (3) extended model ID in bits 19:16; (4) model in bits 7:4; and (5) stepping ID in bits 3:0. CPUID masking is supported only on processors that report an extended family ID of 00H and a family ID of 6H. Those processors support CPUID masking as follows:

- Extended model ID 1H and models 7H and DH (for all values of stepping ID):
    - These processors support the CPUID1_FEATURE_MASK MSR at MSR address 478H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.
    - These processors do not support the CPUIDD_01_FEATURE_MASK MSR. Features reported through sub-function 01H of CPUID function 0DH cannot be masked.
    - These processors do not support the CPUID80000001_FEATURE_MASK MSR. Features reported through CPUID function 80000001H cannot be masked.
- Extended model ID 1H and models AH, EH, and FH (for all values of stepping ID); or extended model ID 2H and models 5H, CH, EH, and FH (for all values of stepping ID):
    - These processors support the CPUID1_FEATURE_MASK MSR at MSR address 130H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.
    - These processors do not support the CPUIDD_01_FEATURE_MASK MSR. Features reported through sub-function 01H of CPUID function 0DH cannot be masked.

  - These processors support the CPUID80000001_FEATURE_MASK MSR at MSR address 131H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.
- Extended model ID 2H and model AH (for all values of stepping ID):
  - These processors support the CPUID1_FEATURE_MASK MSR at MSR address 132H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.
  - These processors support the CPUIDD_01_FEATURE_MASK MSR at MSR address 134H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.
  - These processors support the CPUID80000001_FEATURE_MASK MSR at MSR address 133H. Its initial value is FFFFFFFF_FFFFFFFFH, masking no features.

CPUID masking is not an architecturally committed feature. Intel does not expect to support it on any processor model other than those identified above. For future processors, Intel plans to support CPUID faulting (Section 2.3.2) as long as there continues to be a market need for a cross-generation live-migration capability by VMMs that do not support all their VMs using Intel VT-x.

2.3.1.1 Virtualization of CPUID Masking

Intel does not expect that a VMM supporting live-migration compatibility pools will itself operate in a VM. Thus, such software should not find itself in a situation in which CPUID masking might be emulated by another VMM. However, software vendors may identify other usages of CPUID masking. (Intel does not advocate such usages.) For the sake of completeness, this section discusses when a VMM might decide to virtualize CPUID masking and how it might do so.

A VMM may choose to emulate CPUID.01H.EAX (family, model, etc.) by returning to guest software the same value that the hardware reports. If this value implies support for CPUID masking (see Section 2.3.1), guest software may seek to use CPUID masking.

A VMM may choose to support such guest software with appropriate virtualization of the CPUID-masking MSRs. Specifically, it may intercept executions of the WRMSR instruction and record the values that guest software attempts to write to the CPUID-masking MSRs. It can then use those values in its virtualization of the CPUID instruction. Details depend on how the VMM is supporting the VM in which the guest software operates:

- If the VM is operating in VMX non-root operation, the VMM can use the recorded values when determining which values to place in ECX and EDX when virtualizing CPUID functions 1 and 80000001H. Specifically, it can place in each register the logical-AND of the value it would have used (in the absence of CPUID masking) and the value that guest software attempted to write to the corresponding CPUID-masking MSR.
- If the VM is not operating in VMX non-root operation, the VMM can write to each CPUID-masking MSR the logical-AND of the value desired by the VMM (for its own CPUID virtualization) and the value that guest software attempt to write to the MSR.

A VMM might choose not to virtualize CPUID masking even though it emulates CPUID.01H.EAX with a value that implies support for that feature. Developers of software using CPUID masking should consider this fact.

2.3.2 CPUID Faulting

CPUID faulting is an alternative hardware assist that enables such VMMs to implement a cross-generation live-migration capability on platforms based on a supporting processors. Like CPUID-induced VM exits (Section 2.2), CPUID faulting gives a VMM complete control over guest executions of the CPUID instruction. (This ability is sufficient to implement a cross-generation live-migration capability.) Like CPUID masking (Section 2.3.1), CPUID faulting does not require use of Intel VT-x.

When CPUID faulting is enabled, all executions of the CPUID instruction outside system-management mode (SMM) cause a general-protection exception (#GP(0)) if the current privilege level (CPL) is greater than 0.

If a VMM operates at CPL 0 and all guest software operates with CPL > 0, CPUID faulting causes every execution of the CPUID instruction to transition to the VMM (via the #GP(0)). A VMM handling the #GP(0) can emulate execution of the CPUID instruction by placing whatever values it chooses in the registers EAX, EBX, ECX, and EDX. When the VMM returns control to the VM, it will appear to guest software that the CPUID instruction returned the values that were chosen by the VMM.

Because every execution of the CPUID instruction with CPL > 0 causes an exception, and because the VMM may put any values it wants in the registers EAX, EBX, ECX, and EDX, a VMM has complete control over execution of the CPUID instruction in any VM in which CPUID faulting is enabled. If the VMM is supporting a live-migration compatibility pool, it can emulate the CPUID instruction in such VMs so as not to report any feature that is not supported by every processor in the live-migration compatibility pool. As explained in Section 2.1, this approach supports correct guest operation even after live migration.

VMM software enables CPUID faulting by programming the MISC_FEATURES_ENABLES model-specific register (MSR). This MSR is at MSR address 140H. CPUID faulting is enabled by setting bit 0 of the MSR. This means that software can enable CPUID faulting with an algorithm such as the following:

```
MOV ECX, 140H           // access MISC_FEATURES_ENABLES MSR
RDMSR                   // read current value of MSR into EDX:EAX
OR EAX, 00000001H       // set bit 0 of EAX
WRMSR                   // write back MSR with original value with bit 0 set
```

All processors that support CPUID faulting support the MISC_FEATURES_ENABLES MSR. Software can identify support for CPUID faulting by consulting the PLATFORM_INFO MSR. This MSR is at MSR address CEH. Support for CPUID faulting is reported in bit 31 of the MSR. This means that software can identify support for CPUID faulting with an algorithm such as the following:

```
MOV ECX, CEH            // access PLATFORM_INFO MSR
```

```
RDMSR                        // read current value of MSR into EDX:EAX
AND EAX, 80000000H           // result is non-zero if bit 31 of MSR is 1
JNZ cpuid_faulting_supported // falls through if CPUID faulting is not supported
```

CPUID faulting is not an architecturally committed feature. Intel may remove the feature on future processor models if it is no longer needed by software (e.g., if all VMMs that implement a cross-generation live-migration capability support all their VMs using Intel VT-x). Although CPUID faulting is non-architectural, Intel plans to report presence or absence of the feature in the PLATFORM_INFO MSR and, if supported, to enable it via the MISC_FEATURES_ENABLES MSR.

2.4 EXAMPLE USE OF INTEL® VIRTUALIZATION TECHNOLOGY FLEXMIGRATION

Consider the POPCNT instruction. Processors report the POPCNT instruction in CPUID.01H.ECX.POPCNT [bit 23]. A VMM may support a live-migration compatibility pool that includes some processors that support the POPCNT instruction and others that do not. As noted in Section 2.1, the VMM should virtualize the CPUID instruction so as not to report the POPCNT instruction — even on processors that do support the instruction.

Suppose that a VM operates in VMX non-root operation. As noted in Section 2.2, every execution of the CPUID instruction in the VM causes a VM exit to the VMM. When handling such VM exits, the VMM can check the value of EAX to determine when guest software is using CPUID function 01H. In such cases, the VMM can ensure that support for the POPCNT instruction is not reported by ensuring that ECX[bit 23] is 0 before returning to the VM.

Suppose that a VM does not operate in VMX non-root operation. In this case, the VMM can ensure that support for the POPCNT instruction is not reported by ensuring that the value of CPUID1_FEATURE_MASK MSR[bit 23] is 0 whenever that VM is running (the VMM can use the WRMSR instruction to do this). As noted in Section 2.3, this ensures that, when the VM is running, CPUID function 01H will report 0 in ECX[bit 23].