

midel __X64

LoadGsBaseToRbp

%else

mov ebx, [base]

%endif

mov ax,

处理器 虚拟化技术

邓志 著

test DWORD

;; 检测是否支持 VMX

bt DWORD [ebp + PCB.FeatureExcl], 5

mov eax, STATUS_UNSUCCESS

jnc vmx_operation_enter_done

;; 开启 VMX operation 允许

REX.Wrxb

mov eax, cr4

REX.Wrxb

bit eax, 13

REX.Wrxb

and cr4, eax



处理器 虚拟化技术

邓志 著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书针对在Intel处理器端的虚拟化技术（Intel Virtualization Technology for x86，即Intel VT-x）进行全面讲解。在Intel VT-x技术下实现了VMX（Virtual-Machine Extensions，虚拟机扩展）架构平台来支持对处理器的虚拟化管理。因此，VMX架构是Intel VT-x技术的核心。本书内容围绕VMX架构实现细节展开全面讲解。但Intel VT-d（Virtualization Technology for Directed I/O）和Intel VT-c（Virtualization Technology for Connectivity）技术并不在本书的描述范围。同时，也不针对AMD-v技术进行讨论。

全书共分为7章，书的整体结构也较为规整，可读性比较强。本书共提供14个例子，对VMX架构的一些特色功能进行辅助讲解。

读者阅读本书，可以学习Intel VT-x技术的VMX架构知识，并且对整个x86/x64体系有更深入的了解！可以说，不了解VMX架构，根本算不上对x86/x64体系熟悉，因为，在处理器的虚拟化技术里需要使用全方位的体系知识，对处理器在非常细节的地方进行虚拟化处理。

因此，本书适合有一定x86/x64体系知识基础或者想更深入学习x86/x64体系知识的读者。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

处理器虚拟化技术 / 邓志著. —北京：电子工业出版社，2014.6

ISBN 978-7-121-23019-6

I. ①处… II. ①邓… III. ①微处理器—虚拟技术 IV. ①TP332

中国版本图书馆 CIP 数据核字(2014)第 080434 号



策划编辑：李 冰

责任编辑：李 冰 高洪霞

印 刷：北京京科印刷有限公司

装 订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：41.75 字数：1076 千字

印 次：2014 年 6 月第 1 次印刷

印 数：2000 册 定价：109.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

虚拟化技术大概可以分为软件虚拟化和硬件虚拟化两大类。软件上的虚拟化实际上很多地方都可以看到，例如常见的系统虚拟机软件，Java 虚拟机和 Android 上的虚拟机可以算是广义上的软件虚拟机了。

在硬件虚拟化技术出现之前，虚拟机只能靠软件来模拟实现。硬件上的虚拟化技术可以算是比较热的一门技术了，但它并不是什么新鲜的技术。据资料显示，实际上 Intel 第一代 VT-x 技术 Vanderpool 在 2005 年就已经推出了，但 Intel 发展得很快，从处理器端到芯片组端，再到网络端都已经可以布署硬件虚拟化技术了。

《处理器虚拟化技术》一书围绕 Intel VT-x 处理器端的虚拟化技术而写，可以算是前作《x86/x64 体系探索及编程》的下册。尽管此书只围绕一个主题，但笔者认为它的知识面并不比《x86/x64 体系探索及编程》一书窄，甚至还要超出。因为，完整地虚拟化一个“虚拟处理器”，必须结合原来的 x86/x64 体系知识并扩展开来。编写本书所花的时间比前作还要多许多，从开始动笔到完稿用了 11 个月的时间，期间不可谓不艰辛。

本书内容

全书共 7 章。

第 1 章 系统平台

本书的代码使用汇编语言编写，运行在裸机平台上。本章主要介绍如何实现一个简易平台来支撑书中所有例子的运行。这个平台可以被编译生成 32 位或者 64 位环境。也实现了一些特色机制，例如，开启了多处理器环境，并且为了更好地、直观地了解代码的执行流程而加入了调试信息机制。

第 2 章 VMX 架构基础

VMX 架构是 Intel 处理器端虚拟化技术 (Intel VT-x) 的实现框架。本章对虚拟化技术进行概括讲解，全面了解 VMX 架构的基础知识。首先介绍了 VMX 架构下引入 VMX root operation 与 VMX non-root operation 模式，以及这两个模式之间的切换。然后介绍了

VMX 架构下的各种处理器能力，以及这些能力的检测。最后全面讲解新引入的 VMX 指令集，以及 VMX 指令产生的各种失败的情形。

第 3 章 VMCS 结构

VMCS 结构是 VMX 架构下最基本、最重要的数据结构。VMCS 结构内分为 6 个区域，每个区域有若干个字段。整个 VMX operation 模式的运行环境由 VMCS 结构内的这些字段来配置与管理，例如配置 guest 与 host 的运行环境。全面地讲解了这 6 个区域内每个字段的设置与使用。

第 4 章 VM-entry 处理

在 VMX 架构下，guest 运行在 VMX non-root operation 模式下，而 host 运行在 VMX root operation 模式下。如果需要运行 guest 软件，则处理器需要从 VMX root operation 切换到 VMX non-root operation 模式运行，这个行为被称为“VM-entry”。

本章详细地讲解处理器在执行 VM-entry 操作时的每个流程。例如，在 VM-entry 时会检查 VMCS 结构内字段配置是否合理，并且加载 guest 的运行环境。处理器也会根据 VMCS 内字段来控制 guest 软件运行。

第 5 章 VM-exit 处理

在 guest 运行过程中，会因为某些事件而被迫返回 host 环境。处理器由 VMX non-root operation 模式切换回 VMX root operation 模式，这个行为被称为“VM-exit”。

本章详细地讲解处理器在执行 VM-exit 操作时的每个流程。例如，在 VM-exit 处理器的相应状态信息会得到更新，guest 的环境信息会保存在 VMCS 结构内的 guest-state 区域，并且从 host-state 区域加载 host 环境信息。最后转入 host 入口执行 VMM 的管理代码。

第 6 章 内存虚拟化

VMX 架构引进了 EPT 机制来实现对物理内存的虚拟化管理，使得每个 VM 在物理平台上拥有自己独立的物理内存区域，而不受 VMM 及其他 VM 的干扰。同样也避免了 VM 对 VMM 的干扰。

本章全面讲解了 EPT 机制的实现细节，也着重介绍了 VMX 架构下的 cache 管理。在随书例子中，我们实现了 EPT 机制来管理 VM 内存。

第 7 章 中断虚拟化

VMX 架构下也实现了对 local APIC 的虚拟化管理，引进了两个重要的页面：APIC-access page 页面与 virtual-APIC page 页面。因此而形成了 virtual-APIC 的概念。它是物理平台上的 local APIC 的 shadow。

在这一章里，我们将看到处理器如何处理 APIC-access page 与 virtual-APIC page 之间的关系，将对 APIC-access page 的访问转化为对 virtual-APIC page 的访问。从而实现访问和管理 virtual-APIC 组件。

本章也着重介绍了 VMM 应该如何处理 guest 产生的异常与任务切换。最后以实际例子来展现 VMM 如何监控 INT 指令及 NMI 与外部中断。

如何阅读本书

本书将理论结合实际地进行讲解，但每章内容还是会有侧重点。以章节的篇幅来说，第 2 章至第 5 章偏重于理论知识。而第 1 章、第 6 章和第 7 章偏于实际例子。

因此，如果读者并不想通篇阅读，可以选择偏重理论知识的章节来看。第 6 章与第 7 章介绍的重点知识主要是 EPT（扩展页表）与 local APIC 虚拟化。这两个内容也是必须掌握的。

在第 1 章中并不涉及 VMX 架构知识，而是通过代码的讲解来介绍基础平台。书中的全部例子将构建于这个基础平台之上。如果读者对此章不感兴趣，可以直接跳过。但是，如果读者对一些 OS 的基础元素感兴趣，大可详细加以阅读。

读者对象

读者最好有一些 x86/x64 体系知识基础，如果感到这方面的基础知识相对薄弱，可以阅读作者的另一本著作《x86/x64 体系探索及编程》或者 Intel 官方手册。

如果读者只是想了解 VMX 的理论知识，那么可以选择相应章节，对汇编知识的要求不高。如果读者想深入全面地了解整个 VMX 架构知识，希望能够仔细阅读所有章节（第 1 章可以略过），那么需要对汇编有一定了解。

随书例子

书中的每个章节有若干例子，全书共 14 个例子。读者请登录网址 <http://www.mouseos.com/books/vt/index.html>，选择相应的栏目下载源码包。

源码包解压后有 7 个 chapXX 目录，对应于每一章。还有 commom、inc 以及 lib 目录，存放所有例子共用的代码。每个 chapXX 目录下有多个 ex 实例目录，对应于每个例子。以第 1 章为例，chap01 目录下有 ex1-1 和 ex1-2 实例目录。

每个实例目录里有下面的文件。

- Bs: bochs 的配置文件。
- Build: 一个 DOS 批处理的编译工具，用来编译源码。
- c.img: 硬盘与 U 盘映像文件。
- demo.img: 软盘映像文件。
- ex.asm: 例子实体代码的汇编源文件。
- ex.inc: 例子实体代码头文件。

在时间宽裕的情况下，我会在 Windows 平台上用 C 语言来实现代码示例。目前已经写了大部分代码，但为了保证代码的正确性，力求在所有代码完成后再放出来。

勘误反馈

由于作者学识有限，尽管在书写时已经力求认真细致，但难免有错漏之处。读者如果发现有不妥之处，敬请不吝指出，邮件请发至：mik@mouseos.com，或登录 www.mouseos.com 网站留言。

参考资料

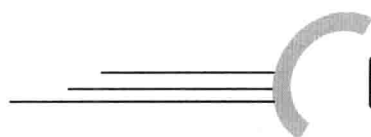
- 《Intel® 64 and IA-32 Architectures Software Developer's Manual》volumes 1, 2A, 2B, 2C, 3A, 3B and 3C
- 《Intel Virtualization Technology for Directed I/O Architecture Specification》

致谢

感谢电子工业出版社博文视点公司的全体工作人员，特别感谢本书的策划编辑李冰，正是有你们的辛勤劳动，才使得本书可以顺利出版。

邓志

2014年5月



目 录

第 1 章 系统平台	1
1.1 环境及工具	1
1.1.1 使用 VMware	2
1.1.2 使用 Bochs	4
1.1.3 在真实机器上运行	4
1.1.4 Build 工具	4
1.2 64 位与 32 位代码的混合编译	7
1.2.1 使用符号 __X64	7
1.2.2 指令操作数	8
1.2.3 64-bit 模式下其他指令处理	11
1.2.4 函数重定义表	15
1.3 地址空间	17
1.4 数据结构	23
1.4.1 PCB 结构	23
1.4.2 LSB 结构	37
1.4.3 初始化 PCB	38
1.4.4 SDA 结构	42
1.4.5 初始化 SDA	56
1.4.6 DRS 结构	57
1.5 系统启动	59
1.5.1 Boot 阶段	59
1.5.2 stage1 阶段	62

1.5.2.1	stage1 阶段的多处理器初始化	66
1.5.2.2	BSP 的收尾工作	68
1.5.2.3	APs 的 stage1 阶段工作	70
1.5.3	stage2 阶段	73
1.5.3.1	BSP 在 stage2 最后处理	80
1.5.3.2	APs 在 stage2 阶段收尾工作	81
1.5.4	stage3 阶段	83
1.5.4.1	BSP 在 stage3 阶段的最后工作	87
1.5.4.2	APs 在 stage3 阶段收尾工作	88
1.5.5	例子 1-1	90
1.6	系统机制	91
1.6.1	分页机制	91
1.6.1.1	PAE 分页模式实现	91
1.6.1.2	IA-32e 分页模式实现	98
1.6.2	多处理器机制	102
1.6.2.1	调度任务	102
1.6.2.2	处理器切换	109
1.6.3	调试记录机制	113
1.6.3.1	例子 1-2	120
1.6.3.2	运行结果	121
第 2 章	VMX 架构基础	122
2.1	虚拟化概述	123
2.1.1	虚拟设备	124
2.1.2	地址转换	125
2.1.3	设备的 I/O 访问	125
2.2	VMX 架构	126
2.2.1	VMM 与 VM	127
2.2.2	VMXON 与 VMCS 区域	127
2.2.3	检测 VMX 支持	128
2.2.4	开启 VMX 进入允许	128
2.3	VMX operation 模式	129

2.3.1	进入 VMX operation 模式	130
2.3.2	进入 VMX operation 的制约	131
2.3.2.1	IA32_FEATURE_CONTROL 寄存器	131
2.3.2.2	CR0 与 CR4 固定位	133
2.3.2.3	A20M 模式	135
2.3.3	设置 VMXON 区域	135
2.3.3.1	分配 VMXON 区域	135
2.3.3.2	VMXON 区域初始设置	135
2.3.4	退出 VMX operation 模式	136
2.4	VMX operation 模式切换	137
2.4.1	VM entry	138
2.4.2	VM exit	139
2.4.3	SMM 双重监控处理下	140
2.5	VMX 能力的检测	141
2.5.1	检测是否支持 VMX	141
2.5.2	通过 MSR 组检查 VMX 能力	141
2.5.3	例子 2-1	146
2.5.4	基本信息检测	147
2.5.5	允许为 0 以及允许为 1 位	149
2.5.5.1	决定 VMX 支持的功能	150
2.5.5.2	控制字段设置算法	150
2.5.6	VM-execution 控制字段	151
2.5.6.1	Pin-based VM-execution control 字段	151
2.5.6.2	primary processor-based VM-execution control 字段	152
2.5.6.3	secondary processor-based VM-execution control 字段	152
2.5.7	VM-exit control 字段	152
2.5.8	VM-entry control 字段	153
2.5.9	VM-function control 字段	153
2.5.10	CR0 与 CR4 的固定位	154
2.5.10.1	CR0 与 CR4 寄存器设置算法	155
2.5.11	VMX 杂项信息	156
2.5.12	VMCS 区域字段 index 值	157

2.5.13	VPID 与 EPT 能力	157
2.6	VMX 指令	158
2.6.1	VMX 指令执行环境	159
2.6.2	指令执行的状态	159
2.6.3	VMfailValid 事件原因	160
2.6.4	指令异常优先级	161
2.6.5	VMCS 管理指令	161
2.6.5.1	VMPTRLD 指令	162
2.6.5.2	VMPTRST 指令	162
2.6.5.3	VMCLEAR 指令	162
2.6.5.4	VMREAD 指令	163
2.6.5.5	VMWRITE 指令	165
2.6.6	VMX 模式管理指令	166
2.6.6.1	VMXON 指令	167
2.6.6.2	VMXOFF 指令	167
2.6.6.3	VMLAUNCH 指令	167
2.6.6.4	VMRESUME 指令	168
2.6.6.5	返回到 executive monitor	168
2.6.7	cache 刷新指令	169
2.6.7.1	INVEPT 指令	170
2.6.7.2	INVVPID 指令	170
2.6.8	调用服务例程指令	171
2.6.8.1	VMCALL 指令	171
2.6.8.2	VMFUNC 指令	172
第 3 章	VMCS 结构	173
3.1	VMCS 状态	173
3.1.1	activity 属性	174
3.1.2	current 属性	174
3.1.3	launch 属性	174
3.2	VMCS 区域	175
3.2.1	VMXON 区域	176

3.2.2	Executive-VMCS 与 SMM-transfer VMCS	176
3.2.3	VMCS 区域格式	176
3.3	访问 VMCS 字段	177
3.3.1	字段 ID 格式	178
3.3.2	不同宽度的字段处理	179
3.4	字段 ID 值	181
3.4.1	16 位字段 ID	181
3.4.2	64 位字段 ID	182
3.4.3	32 位字段 ID	184
3.4.4	natural-width 字段 ID	185
3.5	VM-execution 控制类字段	187
3.5.1	Pin-based VM-execution control 字段	188
3.5.2	processor-based VM-execution control 字段	190
3.5.2.1	primary processor-based VM-execution control 字段	191
3.5.2.2	secondary processor-based VM-execution control 字段	195
3.5.3	exception bitmap 字段	200
3.5.4	PFEC_MASK 与 PFEC_MATCH 字段	200
3.5.5	I/O bitmap address 字段	202
3.5.6	TSC offset 字段	202
3.5.7	guest/host mask 与 read shadow 字段	202
3.5.8	CR3-target 字段	203
3.5.9	APIC-access address 字段	203
3.5.10	virtual-APIC address 字段	204
3.5.11	TPR threshold 字段	204
3.5.12	EOI-exit bitmap 字段	204
3.5.13	posted-interrupt notification vector 字段	205
3.5.14	posted-interrupt descriptor address 字段	205
3.5.15	MSR bitmap address 字段	205
3.5.16	executive-VMCS pointer	206
3.5.17	EPTP 字段	206
3.5.18	virtual-processor identifier 字段	207
3.5.19	PLE_Gap 与 PLE_Window 字段	207

3.5.20	VM-function control 字段	209
3.5.21	EPTP-list address 字段	210
3.6	VM-entry 控制类字段	210
3.6.1	VM-entry control 字段	211
3.6.2	VM-entry MSR-load 字段	214
3.6.3	事件注入控制字段	214
3.6.3.1	VM-entry interruption information 字段	215
3.6.3.2	VM-entry exception error code 字段	217
3.6.3.3	VM-entry instruction length 字段	217
3.7	VM-exit 控制类字段	218
3.7.1	VM-exit control 字段	218
3.7.2	VM-exit MSR-store 与 MSR-load 字段	220
3.8	guest-state 区域字段	221
3.8.1	段寄存器字段	224
3.8.1.1	access right 字段	224
3.8.2	GDTR 与 IDTR 字段	229
3.8.3	MSR 字段	229
3.8.4	SMBASE 字段	229
3.8.5	activity state 字段	230
3.8.6	interruptibility state 字段	232
3.8.7	pending debug exceptions 字段	235
3.8.7.1	#DB 异常的处理	237
3.8.8	VMCS link pointer 字段	243
3.8.9	VMX-preemption timer value 字段	243
3.8.10	PDPTes 字段	243
3.8.11	guest interrupt status 字段	244
3.9	host-state 区域字段	245
3.10	VM-exit 信息类字段	247
3.10.1	基本信息类字段	248
3.10.1.1	Exit reason 字段	248
3.10.1.2	VM-exit 原因	249

3.10.1.3	Exit qualification 字段	255
3.10.1.4	由某些指令引发的 VM-exit	256
3.10.1.5	由#DB 异常引发的 VM-exit	256
3.10.1.6	由#PF 异常引发的 VM-exit	257
3.10.1.7	由 SIPI 引发的 VM-exit	257
3.10.1.8	由 I/O SMI 引发的 VM-exit	257
3.10.1.9	由任务切换引发的 VM-exit	258
3.10.1.10	访问控制寄存器引发的 VM-exit	259
3.10.1.11	由 MOV-DR 指令引发的 VM-exit	260
3.10.1.12	由 I/O 指令引发的 VM-exit	260
3.10.1.13	由于访问 APIC-access page 引发的 VM-exit	261
3.10.1.14	由 EPT violation 引发的 VM-exit	262
3.10.1.15	由 EOI 虚拟化引发的 VM-exit	264
3.10.1.16	由 APIC-write 引发的 VM-exit	264
3.10.1.17	guest-linear address 字段	264
3.10.1.18	guest-physical address 字段	265
3.10.2	直接向量事件类信息字段	265
3.10.2.1	VM-exit interruption information 字段	265
3.10.2.2	VM-exit interruption error code 字段	267
3.10.3	间接向量事件类信息字段	267
3.10.3.1	IDT-vectoring information 字段	268
3.10.3.2	IDT-vectoring error code 字段	269
3.10.4	指令类信息字段	269
3.10.4.1	VM-exit instruction length 字段	269
3.10.4.2	VM-exit instruction information 字段	272
3.10.5	I/O SMI 信息类字段	280
3.10.6	指令错误类字段	280
3.11	VMM 初始化实例	280
3.11.1	VMCS 相关的数据结构	281
3.11.1.1	VMB 结构	281
3.11.1.2	VSB 结构	284
3.11.1.3	VMCS buffer 结构	287

3.11.2	初始化 VMXON 区域	288
3.11.3	初始化 VMCS 区域	289
3.11.3.1	分配 VMCS 区域	290
3.11.3.2	VMCS 初始化模式	291
3.11.3.3	VMCS buffer 初始化	293
3.11.4	例子 3-1	297
第 4 章	VM-entry 处理	301
4.1	发起 VM-entry 操作	302
4.2	VM-entry 执行流程	303
4.3	指令执行的基本检查	303
4.4	检查控制区域及 host-state 区域	305
4.4.1	VM-execution 控制区域检查	305
4.4.1.1	检查 pin-based VM-execution control 字段	306
4.4.1.2	检查 primary processor-based VM-execution control 字段	306
4.4.1.3	检查 secondary processor-based VM-execution control 字段	307
4.4.1.4	检查 CR3-target 字段	308
4.4.2	VM-exit 控制区域检查	308
4.4.2.1	VM-exit control 字段的检查	308
4.4.2.2	MSR-store 与 MSR-load 相关字段的检查	308
4.4.3	VM-entry 控制区域检查	309
4.4.3.1	VM-entry control 字段的检查	309
4.4.3.2	MSR-load 相关字段的检查	309
4.4.3.3	事件注入相关字段的检查	309
4.4.4	Host-state 区域的检查	310
4.4.4.1	Host 控制寄存器字段的检查	310
4.4.4.2	Host-RIP 的检查	310
4.4.4.3	段 selector 字段的检查	311
4.4.4.4	段基址字段的检查	311
4.4.4.5	MSR 字段的检查	311
4.5	检查 guest-state 区域	311
4.5.1	检查控制寄存器字段	312

4.5.2	检查 RIP 与 RFLAGS 字段	312
4.5.3	检查 DR7 与 IA32_DEBUGCTL 字段	313
4.5.4	检查段寄存器字段	313
4.5.4.1	virtual-8086 模式下的检查	314
4.5.4.2	unrestricted guest 位为 0 时的检查	315
4.5.4.3	unrestricted guest 位为 1 时的检查	318
4.5.5	检查 GDTR 与 IDTR 字段	320
4.5.6	检查 MSR 字段	320
4.5.7	检查 activity state 字段	321
4.5.8	检查 interruptibility state 字段	321
4.5.9	检查 pending debug exception 字段	322
4.5.10	检查 VMCS link pointer 字段	322
4.5.11	检查 PDPTE 字段	323
4.5.11.1	由加载 CR3 引发的 PDPTE 检查	323
4.6	检查 guest state 引起的 VM-entry 失败	324
4.7	加载 guest 环境信息	324
4.7.1	加载控制寄存器	325
4.7.2	加载 DR7 与 IA32_DEBUGCTL	325
4.7.3	加载 MSR	325
4.7.4	SMBASE 字段处理	326
4.7.5	加载段寄存器与描述符表寄存器	326
4.7.5.1	unusable 段寄存器	327
4.7.5.2	加载 GDTR 与 IDTR	327
4.7.6	加载 RIP、RSP 和 RFLAGS	327
4.7.7	加载 PDPTE 表项	327
4.8	刷新处理器 cache	328
4.9	更新 Vritual-APIC 状态	328
4.9.1	PPR 虚拟化	329
4.9.2	虚拟中断评估与 delivery	329
4.10	加载 MSR-load 列表	329
4.10.1	IA32_EFER 的加载处理	330

4.10.2 其他 MSR 字段的加载处理	331
4.11 由加载 guest state 引起的 VM-entry 失败	331
4.12 事件注入	332
4.12.1 注入事件的 delivery	335
4.12.1.1 保护模式下的事件注入	335
4.12.1.2 实模式下的事件注入	338
4.12.1.3 virtual-8086 模式下的事件注入	338
4.12.2 注入事件的间接 VM-exit	339
4.13 执行 pending debug exception	341
4.13.1 注入事件下的 #DB 异常 delivery	342
4.13.2 例子 4-1	346
4.13.3 非注入事件下的 #DB 异常 delivery	351
4.14 使用 MTF VM-exit 功能	354
4.14.1 注入事件下的 MTF VM-exit	354
4.14.2 非注入事件下的 MTF VM-exit	355
4.14.3 MTF VM-exit 与其他 VM-exit	355
4.14.4 MTF VM-exit 的优先级别	356
4.14.5 例子 4-2	356
4.15 VM-entry 后直接导致 VM-exit 的事件	362
4.15.1 VM-exit 事件的优先级别	362
4.15.2 TPR below threshold VM-exit	363
4.15.3 pending MTF VM-exit	364
4.15.4 由 pending debug exception 引发的 VM-exit	364
4.15.5 VMX-preemption timer	364
4.15.6 NMI-window exiting	366
4.15.7 interrupt-window exiting	367
4.16 处理器的可中断状态	367
4.16.1 中断的阻塞状态	367
4.16.2 阻塞状态的解除	368
4.16.3 中断的阻塞	369
4.16.4 VM-entry 后的可中断状态	370
4.17 处理器的活动状态	370