v8086, Execute 16bit Code in Protected Mode

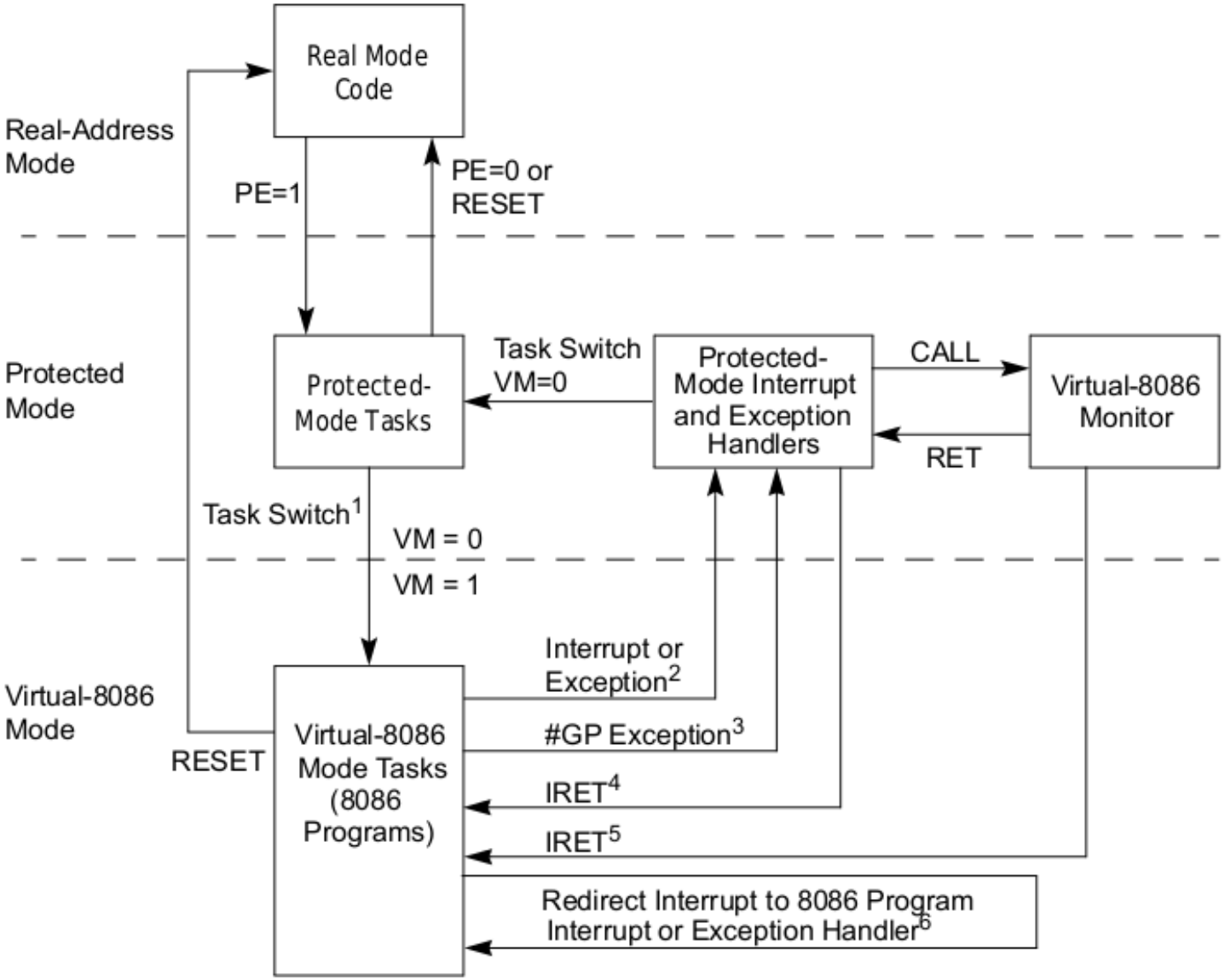
Corentin Derbois

corentin@lse.epita.fr
http://www.lse.epita.fr

July 17, 2013

Plan

Workflow


Real Mode Code
Real-Address Mode
PE=1
PE=0 or RESET
Protected Mode
Protected-Mode Tasks
Task Switch VM=0
Protected-Mode Interrupt and Exception Handlers
CALL
RET
Virtual-8086 Monitor
Task Switch[1]
VM = 0
VM = 1
Interrupt or Exception[2]
Virtual-8086 Mode
Virtual-8086 Mode Tasks (8086 Programs)
#GP Exception[3]
RESET
IRET[4]
IRET[5]
Redirect Interrupt to 8086 Program Interrupt or Exception Handler[6]

Rationale

- Easy video management
- 16 bit code execution
- BIOS data information access

Plan

Enable Virtual-8086 Mode

Needed value

- CS/SS/SP/IP
- Eflags
  - VM
  - IOCTLX
  - NT if iret is used

Man page: 20.2

The processor runs in virtual-8086 mode when the VM (virtual machine) flag in the EFLAGS register is set.

- A CALL or JMP instruction.
- An IRET instruction, where the NT flag in the EFLAGS image is set to 1.

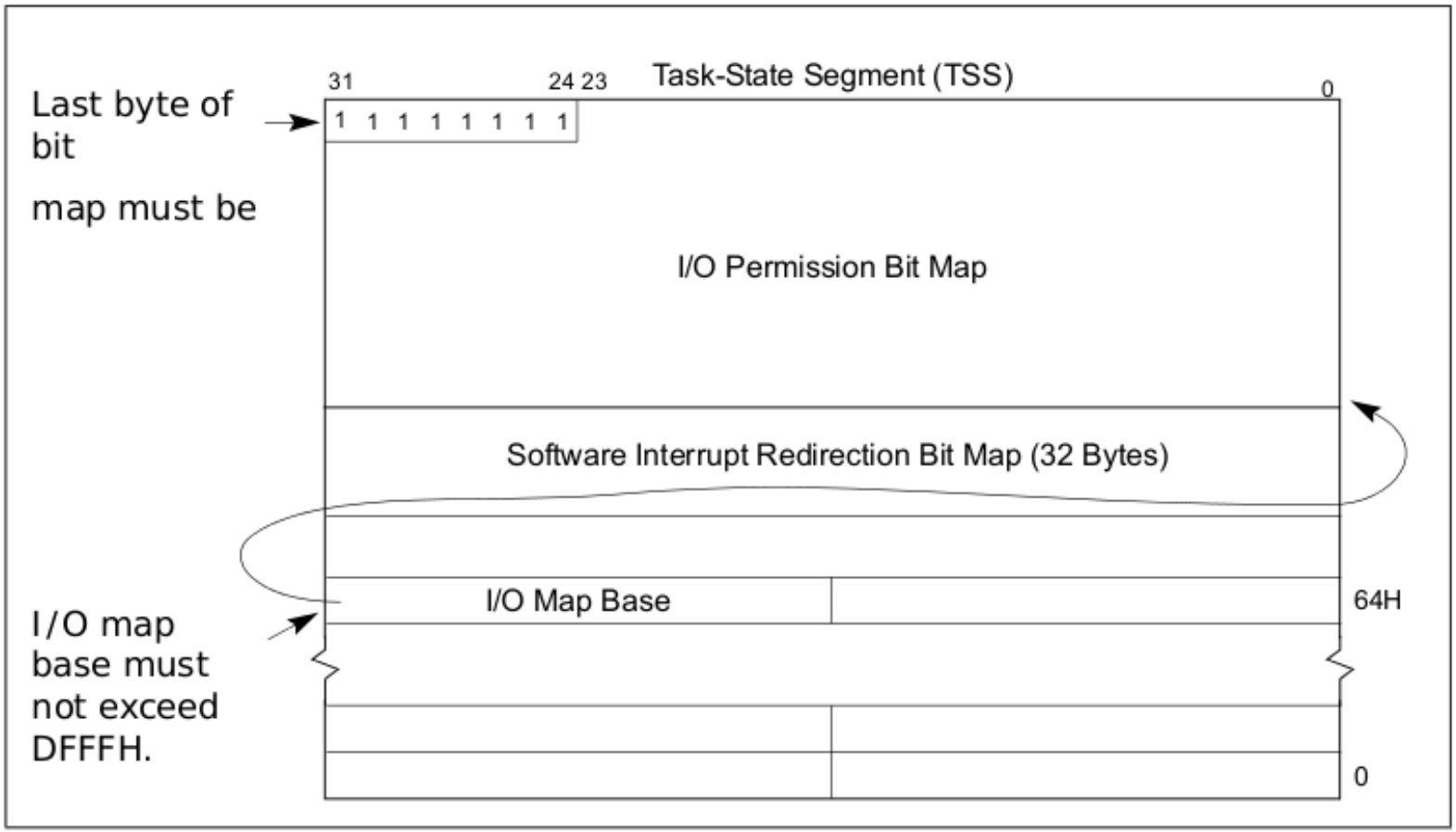


Figure 20-5. Software Interrupt Redirection Bit Map in TSS

Interruption mode

In v8086 all interruptions can be managed in two different way:

- Redirected in protected mode
- Managed by the 8086 virtual processor

IOPL table

Table 20-2. Software Interrupt Handling Methods While in Virtual-8086 Mode

Method	VME	IOPL	Bit in Redir. Bitmap*	Processor Action
1	0	3	X	Interrupt directed to a protected-mode interrupt handler: • Switches to privilege-level 0 stack • Pushes GS, FS, DS and ES onto privilege-level 0 stack • Pushes SS, ESP, EFLAGS, CS and EIP of interrupted task onto privilege-level 0 stack • Clears VM, RF, NT, and TF flags • If serviced through interrupt gate, clears IF flag • Clears GS, FS, DS and ES to 0 • Sets CS and EIP from interrupt gate
2	0	< 3	X	Interrupt directed to protected-mode general-protection exception (#GP) handler.
3	1	< 3	1	Interrupt directed to a protected-mode general-protection exception (#GP) handler; VIF and VIP flag support for handling class 2 maskable hardware interrupts.
4	1	3	1	Interrupt directed to protected-mode interrupt handler: (see method 1 processor action).
5	1	3	0	Interrupt redirected to 8086 program interrupt handler: • Pushes EFLAGS • Pushes CS and EIP (lower 16 bits only) • Clears IF flag • Clears TF flag • Loads CS and EIP (lower 16 bits only) from selected entry in the interrupt vector table of the current virtual-8086 task
6	1	< 3	0	Interrupt redirected to 8086 program interrupt handler; VIF and VIP flag support for handling class 2 maskable hardware interrupts: • Pushes EFLAGS with IOPL set to 3 and VIF copied to IF • Pushes CS and EIP (lower 16 bits only) • Clears the VIF flag • Clears TF flag • Loads CS and EIP (lower 16 bits only) from selected entry in the interrupt vector table of the current virtual-8086 task

NOTE

Real mode

- Real mode address
- Pagination is enabled
- Virtualized interruptions

RealMode address

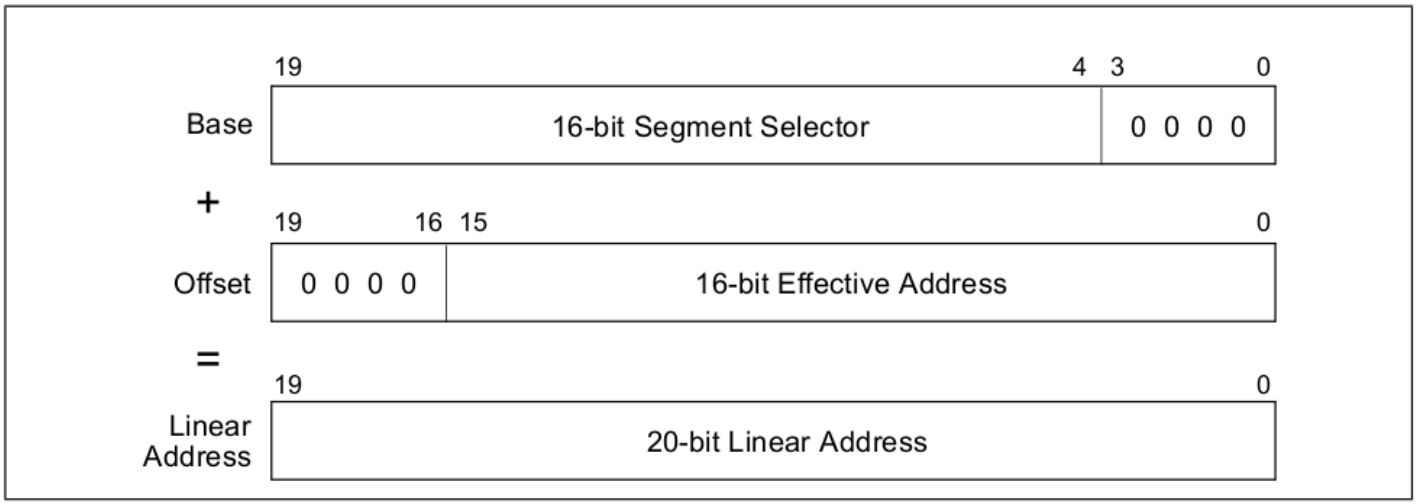


Figure 20-1. Real-Address Mode Address Translation

Exit

- Don’t exit, use task
- Use interruptions to comunicate

Issues

- Switch time
- Lower addresses in vitual address space
- BIOS at specific address
- Long mode

Solution

Emulation 8086

Emulation of 8086 gave the possibility to bypass most of the problems of v8086, like switching context.

Special case: Emulation of interruption

- IVT at 0x0
- transform interrupt to jump

Plan

3 Conclusion

Used by current system

- Windows
- Linux
- Bsd

Questions?