Lecture 1

How to Boot a PC

Andrei Pitiș

Operating Systems Practical

October 8, 2014

Table of Contents

Outline

Hardware

- Processor
  - Protected mode
- Memory
  - Segmentation
- I/O subsystems
  - Interrupt controller (8259)
  - Timer (8253/8254)
  - Keyboard
  - Display

Outline

Processor - x86

- Registers:
  - AX, BX, CX, DX
  - SI, DI
  - CS, DS, ES, SS
  - SP, BP, IP
  - Flags
  - GDTR, LDTR, IDTR

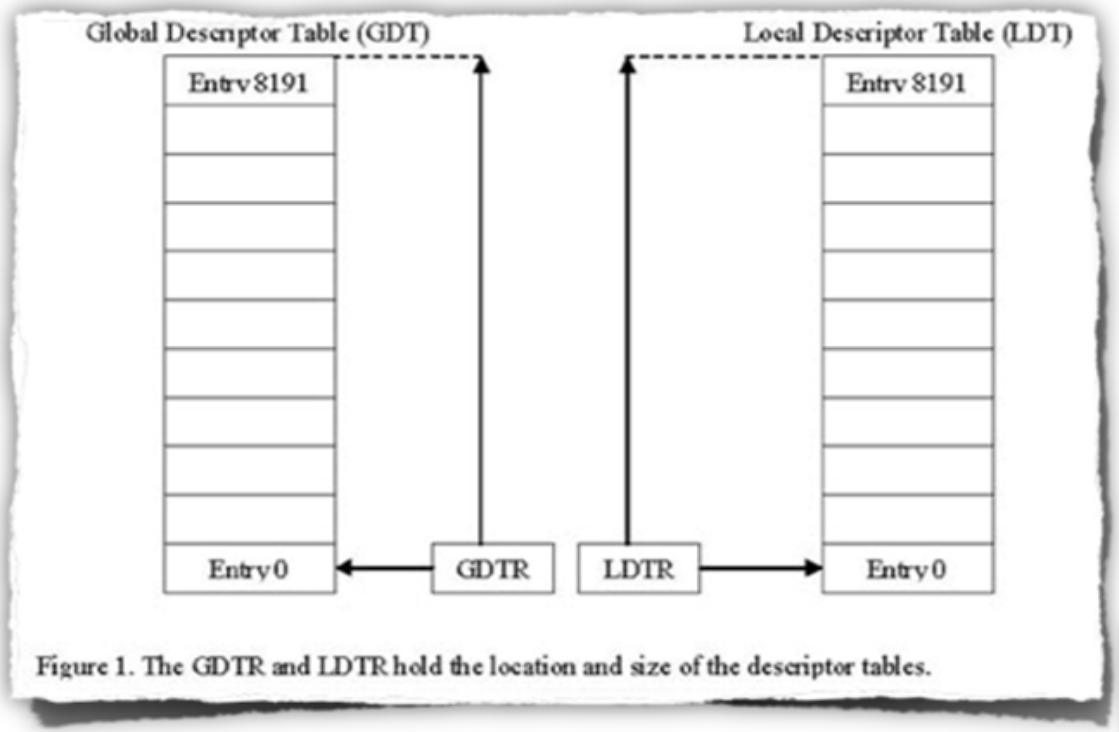
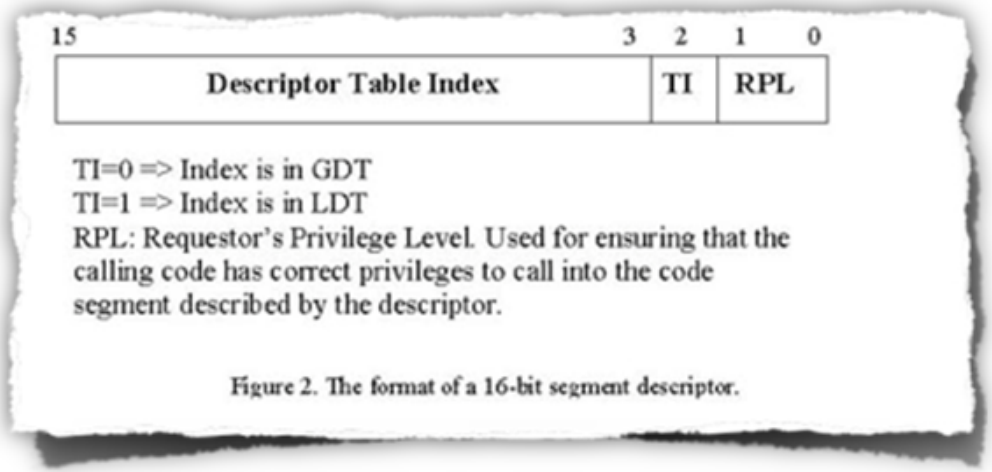


Figure 1. The GDTR and LDTR hold the location and size of the descriptor tables.

```
descriptor     struc

limit             dw        0
loww              dw        0
highb             db        0
rights            db        0
                  dw        0

descriptor     ends
```

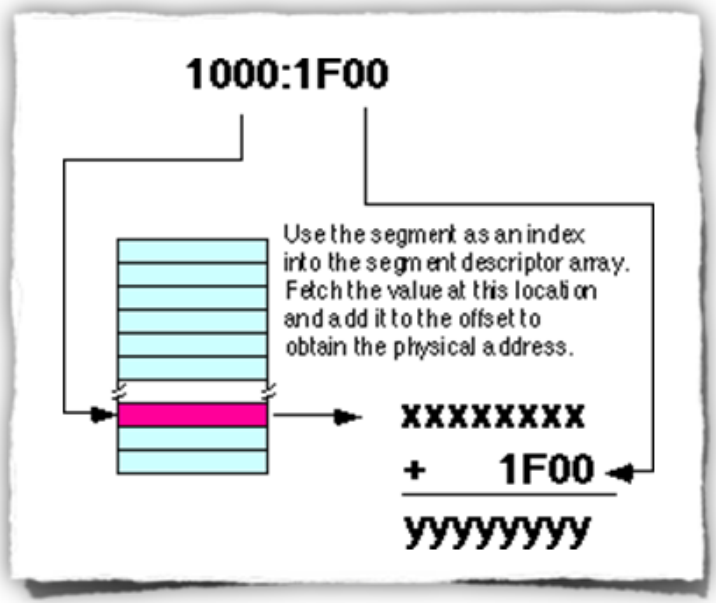
Outline



15 ... 3	2	1	0
Descriptor Table Index	TI	RPL	

TI=0 => Index is in GDT
TI=1 => Index is in LDT
RPL: Requestor's Privilege Level. Used for ensuring that the calling code has correct privileges to call into the code segment described by the descriptor.

Figure 2. The format of a 16-bit segment descriptor.

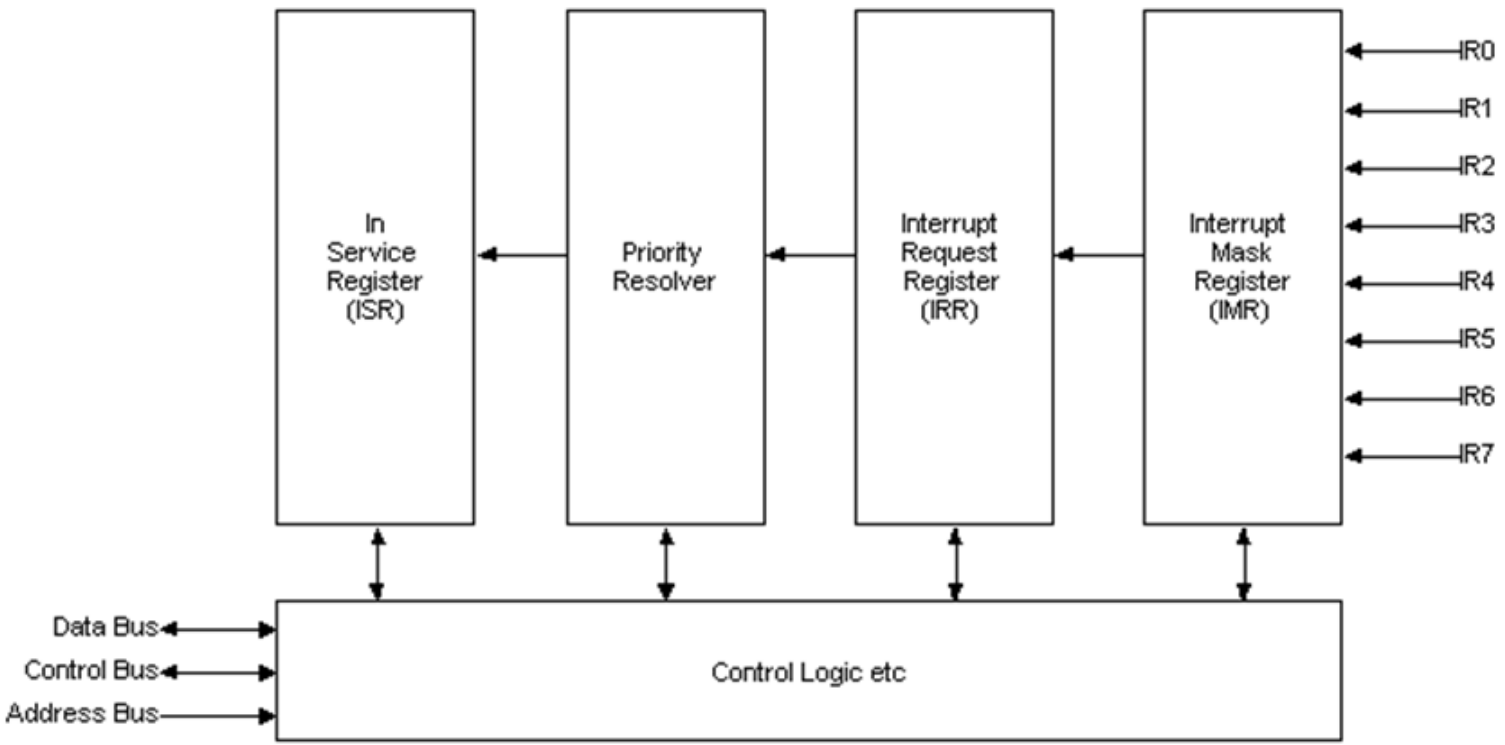


LDT/GDT

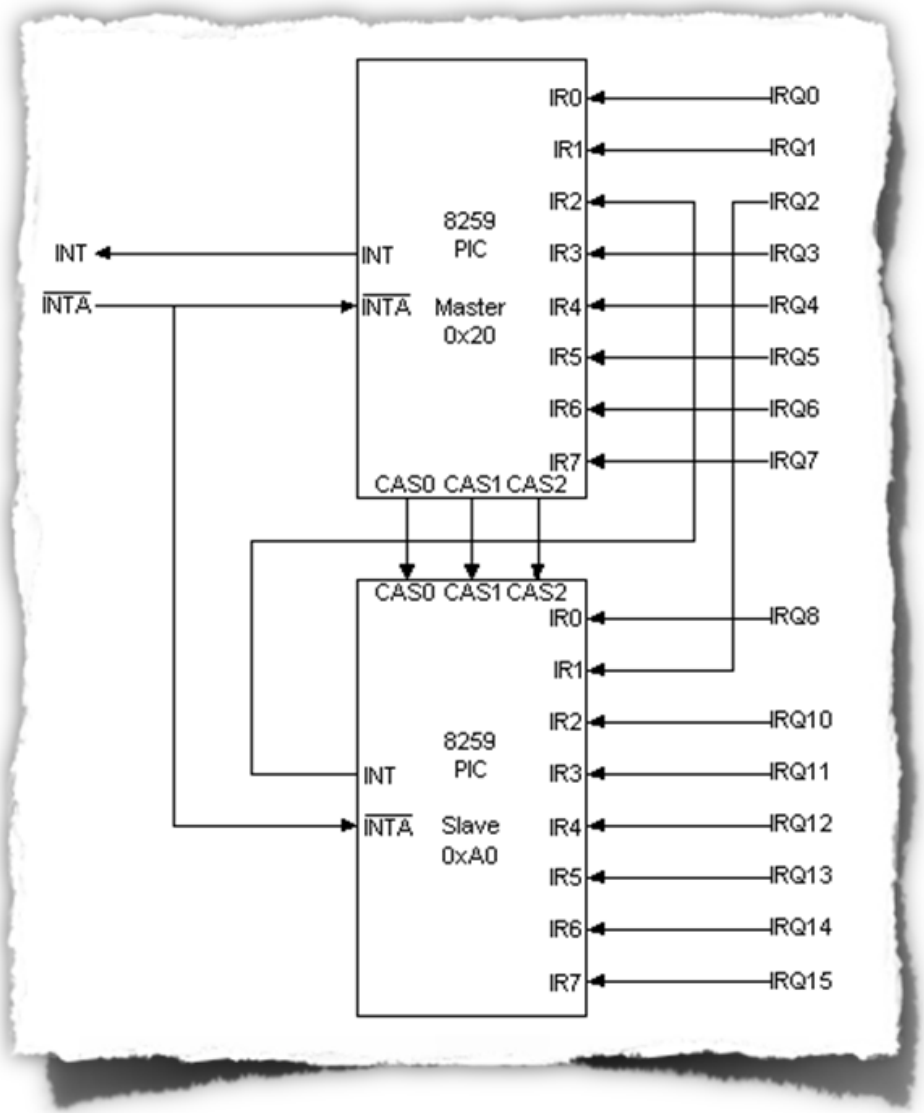
```
LDT1        descriptor <       ,                ,   ,                > ;dummy   07h
            descriptor <01000h, offset code1,    , 11111010b>          ;code    0Fh
            descriptor <01000h, offset data1,    , 11110010b>          ;data    17h
            descriptor <       ,                , 2, 11110110b>       ;stack   1Fh
            descriptor <00001h, offset shdata,   , 11110010b>          ;shared data 27h

GDT         descriptor <       ,      ,     ,                >        ;dummy   0h
            descriptor <0FFFFh,       ,     , 10011010b>              ;code    8h
            descriptor <0FFFFh,       ,     , 10010010b>              ;data    10h
            descriptor <       ,      ,     , 10010110b>              ;stack   18h
            descriptor <0FFFFh, 8000h, 0Bh, 11110010b>                ;video   23h

            descriptor <00007h, offset LDT0,    , 10000010b>          ;LDT0    28h
            descriptor <0002Bh, offset TSS0,    , 10000001b>          ;TSS0    30h
```

Outline

PIC - 8259


In Service Register (ISR)
Priority Resolver
Interrupt Request Register (IRR)
Interrupt Mask Register (IMR)
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
Data Bus
Control Bus
Address Bus
Control Logic etc

- The original 8086 had only one PIC
- Needed more devices - cascaded a second PIC



Interrupts - real mode

- 00-01 - Exception handlers
- 02 - NMI
- 03-07 - Exception handlers
- 08-0F - IRQ0 - IRQ7
- 10-6F - Software interrupts
- 70-77 - IRQ8 - IRQ15
- 78-FF - Software interrupts

- IDT - Interrupt Descriptor Table

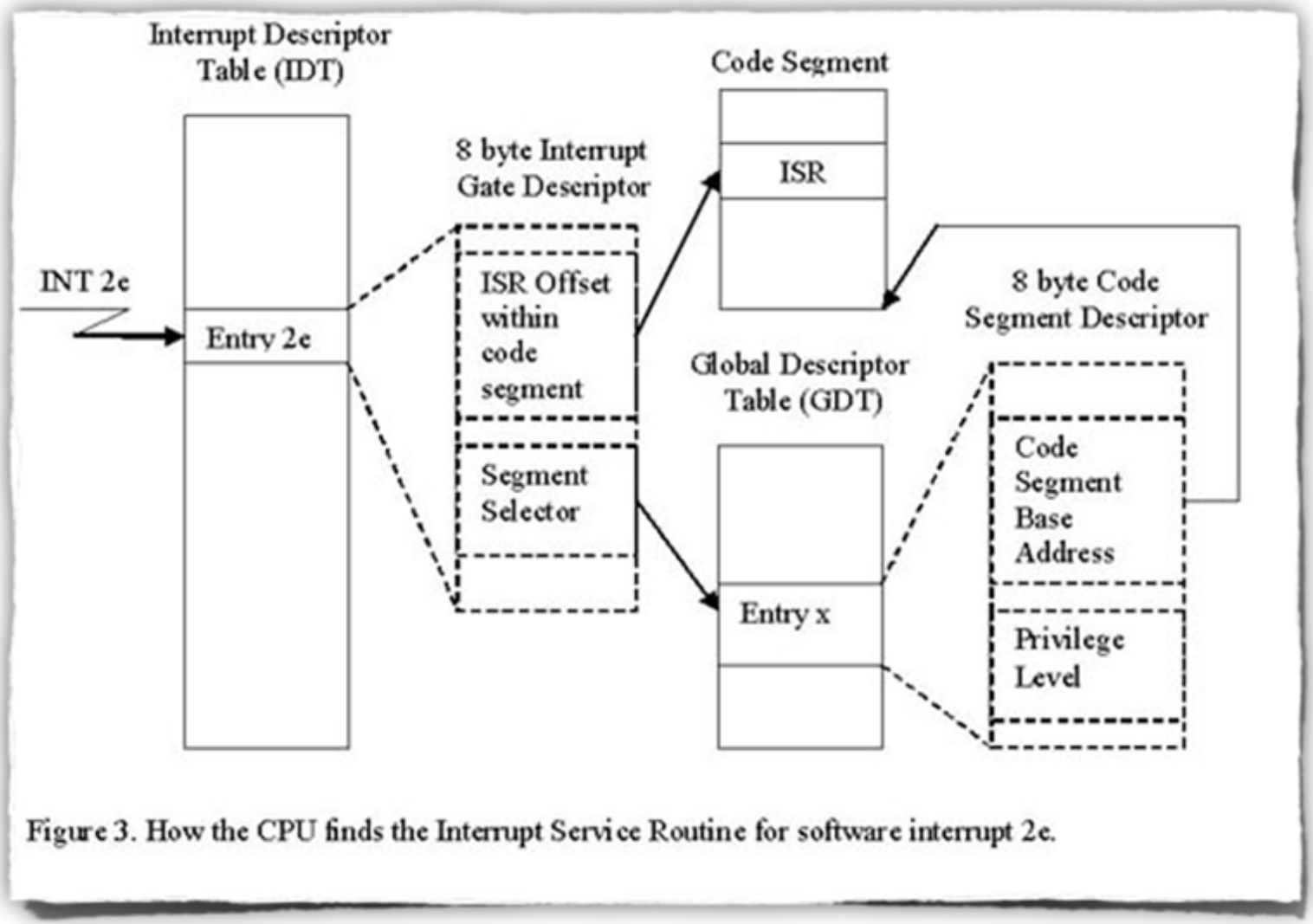
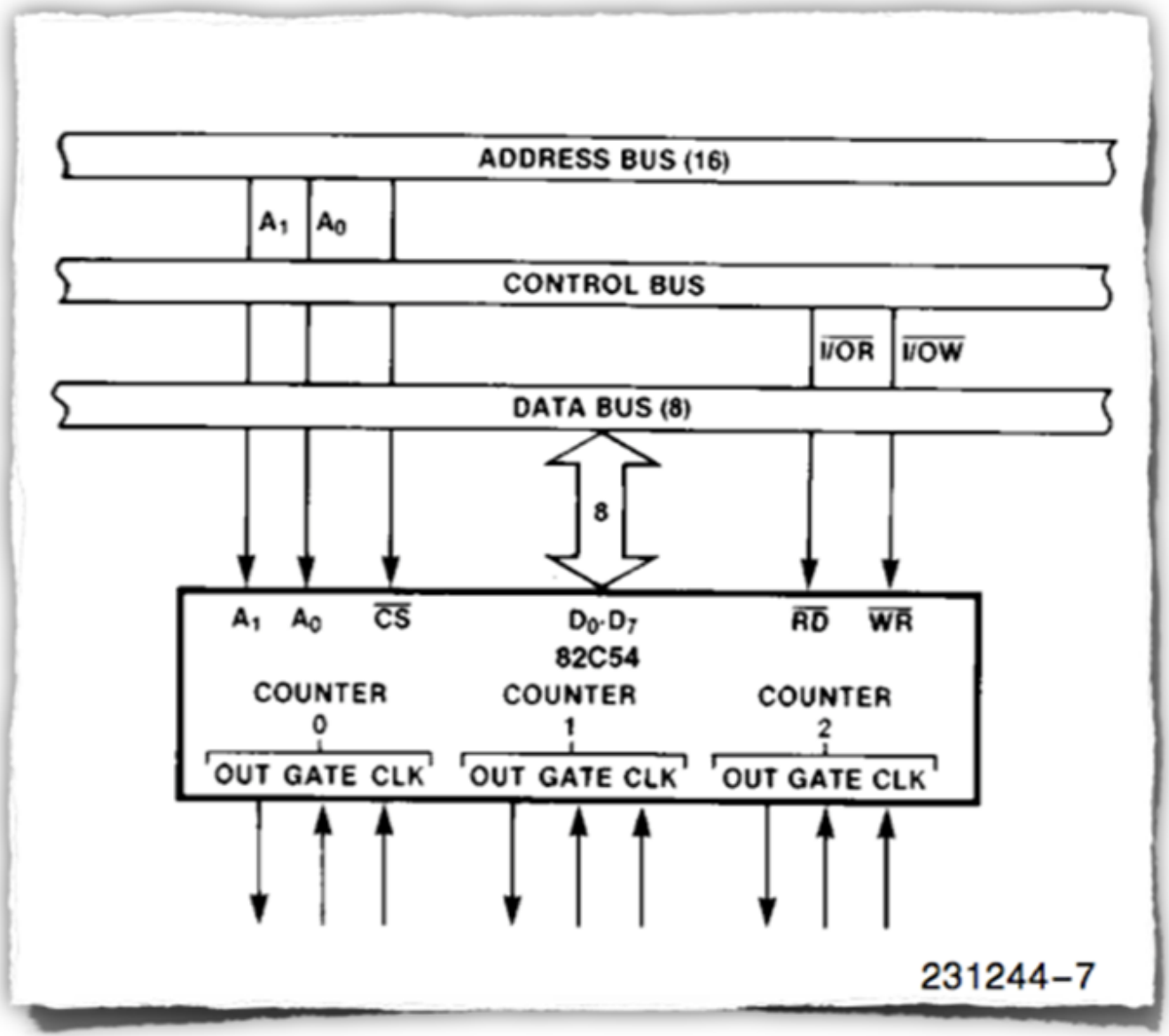


Figure 3. How the CPU finds the Interrupt Service Routine for software interrupt 2e.

init_8259

```
init_8259 proc near

        mov al,11h      ; ICW1: edge, call address interval 8, cascaded, sed ICW4
        out 20h,al
        out 0A0h,al

        mov al,20h      ; ICW2: first 8 IRQs start at 20h after the reserved exceptions
        out 21h,al

        mov al,28h      ; ICW2: next 8 IRQS are immediately after
        out 0A1h,al

        mov al,4        ; ICW3: IRQ2 is connected to a slave
        out 21h,al

        mov al,2
        out 0A1h,al     ; ICW3: Slave ID 2

        mov al,1        ; ICW4: 8086 Mode
        out 021h,al
        out 0A1h,al

        mov al,0FFh
        out 0A1h,al     ; OCW1: Mask all ints on PIC2
        mov al,0FCh     ; OCW1: mask all ints but timer and keyboard on PIC1
        out 021h,al

        ret

        endp
```


ADDRESS BUS (16)
A1
A0
CONTROL BUS
I/OR
I/OW
DATA BUS (8)
8
A1
A0
CS
D0-D7
RD
WR
82C54
COUNTER 0
COUNTER 1
COUNTER 2
OUT GATE CLK
OUT GATE CLK
OUT GATE CLK
231244-7

Control Word Format

$A_1, A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW — Read/Write:

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Timer - 8253

```
init_8253 proc near

        mov al,36h      ; LSB then MSB, mode 011
        out 43h,al
        mov ax,11932    ; clock freq is 1193181.8181..Hz, 1/3 NTSC color subcarrier freq
        jmp $+2
        jmp $+2
        out 40h,al
        jmp $+2
        jmp $+2
        mov al,ah
        out 40h,al
        rets

        endp
```

Scheduler

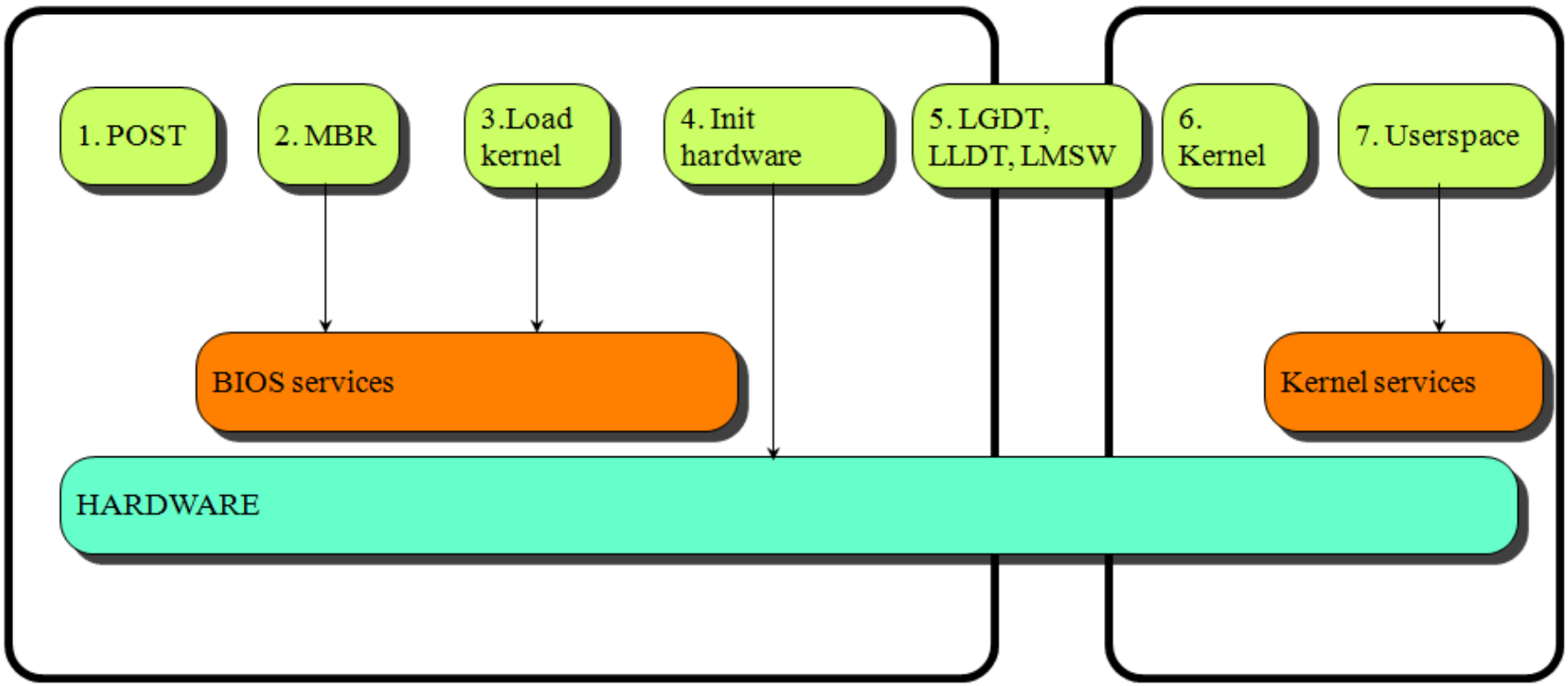
```
clk proc far              ; Task scheduler (100 Hz)

        push ax
        push bx
        push ds
        mov ax,10h
        mov ds,ax

        add task,18h
        cmp task,88h
        jne jmpfar
        mov task,40h
        jmp jmpfar

jmpfar: mov bx,task
        and GDT[bx].rights,0FDh

        mov al,20h
        out 20h,al

        pop ds
        pop bx
        pop ax

        db 0EAh           ; jmp TSSi (i = 1, 2, 3)
        dw 0
task dw 40h

        iret
```

Outline

Boot process

- POST
- Search boot device: floppy, HD, etc
- Load MBR at 0000:7C00
- Jump at that address
- Usually the code in the MBR looks for the active partition, loads its first sector also at 7C00 and jumps there
- We will instead load sectors from the disk and jump

Boot sequence



Bootstrap

```
start:  cli

        xor ax,ax
        mov ss,ax
        mov sp,7C00h
        mov si,sp
        mov ds,ax
        mov es,ax
        sti
        cld
        mov di,600h
        mov cx,100h
        repnz movsw
        db 0EAh
        dw 61Dh,0          ; jmp far ptr 0000:061D

real_start:
```

Outline

Keywords

- processor
- memory
- i/o subsystems
- boot process
- bootstrap
- 0x7c00
- interrupts
- timer
- 8259
- 8253

Outline

Resources

- Use the source, Luke!

Outline

Questions

?