第12章 中断

罗文坚
中国科大 计算机学院

http://staff.ustc.edu.cn/~wjluo/mcps/

1

本章内容

- 基本中断处理
- 硬件中断
- 扩展中断结构
- 8259A可编程中断控制器
- 中断实例

2

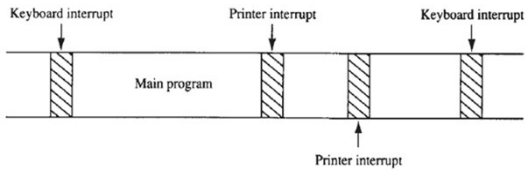
基本中断处理

- 中断的目的
- 中断（中断引脚、中断向量和专用中断）
- 中断指令
- 实模式中断操作
- 保护模式中断操作
- 中断标志位
- 将一个中断向量存入向量表

3

中断的目的

- 在与低速I/O设备进行数据传输时，中断特别有用。
- 例，一个典型系统中表明中断使用情况的时间线。



4

中断引脚

- 整个Intel系列微处理器的中断包括：
  - 2个申请中断的硬件引脚：INTR和NMI
  - 1个相应INTR中断申请的硬件引脚：INTA
- 注意：除了这些引脚外，微处理器还有：
  - 软件中断指令INT、INTO、INT3和BOUND；
  - 标志位IF（Interrupt Flag）和TF（Trap Flag）；
  - 中断返回指令IRET（或在80386~Pentium4中的IRETD）。

5

中断向量

- 中断向量表（IVT，Interrupt Vector Table）为于存储器的最低1024字节，地址为0000H~03FFH。
  - 包含256个不同的4字节中断向量。
- 中断向量包含中断服务程序的地址（段和偏移）。

3	Segment (high)
2	Segment (low)
1	Offset (high)
0	Offset (low)

中断向量的内容

6

中断向量

- 在256个中断向量中，Intel保留前32个中断向量为Intel各种微处理器系列成员专用，后224个向量可用作用户中断向量。
- 在前32个中断向量中：
  - 前5个中断向量在所有Intel系列微处理器中都是相同的。
  - 其他中断向量存在于80286及向上兼容的80386~Core2中，但不向下兼容8086或8088。

7

中断向量

- 类型0：除法出错
- 类型1：单步或陷阱
- 类型2：不可屏蔽硬件中断
- 类型3：断点中断
- 类型4：溢出中断
- 类型5：边界
  - 边界指令将寄存器与存储器中的边界值相比较。如果寄存器的内容大于或等于存储器中的第一个字，并小于或等于第二个字，则不发生中断，因为寄存器的内容在边界之内。如果寄存器的内容超出边界，则发生类型5中断。

8

中断向量

- 类型6：无效操作码
  - 一旦在程序中遇到未定义的操作码时发生此中断。
- 类型7：协处理器不存在
  - 如果执行了ESC或WAIT指令且没有找到协处理器，则发生此中断。
- 类型8：双故障中断
  - 在同一指令期间发生2个独立的中断时激活此中断。

9

中断向量

- 类型9：协处理器段超限
  - 实模式下，若ESC指令（协处理器操作码）的存储器操作数超出偏移地址FFFFH，则发生该中断。
- 类型10：无效任务状态段
  - 在保护模式下，由于段限区域不是002BH或更高，则TSS无效，此时发生该中断。
- 类型11：段不存在
  - 当保护模式描述符中的P位（P=0）指示段不存在或无效时，发生该中断。

10

中断向量

- 类型12：堆栈段超限
  - 保护模式下，如果堆栈段不存在（P=0）或堆栈段超限，则发生该中断。
- 类型13：一般性保护
  - 在80286~Core2保护模式下，大多数保护机制冲突都会引起该中断。这些错误在Windows中表现为一般性保护错。
  - 这些保护违规包括：描述符表边界超限；违反特权规则；装入了无效的描述符段类型；对被保护的代码段执行写操作；从只能执行的代码段读数据；对只读数据段的写操作；段边界超限；当执行CLTS、HLT、LGDT、LIDT、LLDT、LMSW或LTR时，CPL!=0；当执行CLI、IN、INS、LOCK、OUT、OUTS和STI时，CPL>IOPL。

11

中断向量

- 类型14：页面出错
  - 在80386、80486和Pentium~Core2中，页故障将产生该中断。
- 类型16：协处理器出错
  - 当80386、80486和Pentium~Core2的ESC或WAIT指令发生协处理器错误时，发生该中断。
- 类型17：对齐检查
  - 当对齐检查允许时，指示处理器检测到一个未对齐的存储器操作数。
- 类型18：机器检查
  - 在Pentium~Core2中，指示处理器检测到一个内部机器错误或一个总线错误，或外部Agent检测到一个总线错误。

12

中断指令

- 微处理器现有5条中断指令：
  - INT，INT3
  - INTO，BOUND
  - IRET
- INT n：调用入口地址存于向量号n的中断服务程序。
- INT3：相当于INT 3。
- INTO：相当于INT 4。
  - INTO指令检查OF标志。如果OF=1，则INTO指令调用类型号为4号的中断服务程序。

13

中断指令

- BOUND reg, src：比较寄存器和2个字存储器数据。
  - 例，BOUND AX, DATA
- IRET：中断返回
  - 用于软件和硬件中断的返回。
  - 如果工作在实模式下，则使用IRET指令。
  - 在80386~Core2保护模式下，用IRETD指令，因为这些微处理器将EIP/CS/EFLAG压入堆栈。
  - 在Pentium 4的64位模式下，使用IRETQ指令。

14

实模式中断操作

- 当微处理器执行完当前指令后，按下列顺序判断是否有中断发生：
  1. 指令执行情况；
  2. 单步中断；
  3. NMI；
  4. 协处理器段超限；
  5. INTR；
  6. INT 指令。

15

实模式中断操作

- 如果有中断发生，则按下列顺序处理：
  - 将标志寄存器入栈；
  - 清除IF、TF标志；
  - CS入栈；
  - 指令指针IP入栈；
  - 取出中断向量内容，送入IP和CS。

16

实模式中断操作

- 关于返回地址：
  - 有时，返回地址为程序中的下一条指令。
  - 有时指向程序中发生中断的地方。
- 中断类型0、5、6、7、8、10、11、12和13压入堆栈的返回地址是指向错误指令，而不是下一条指令。
  - 使得中断服务程序在某些错误情况下有可能重新执行该指令。
- 一些保护模式中断（类型8、10、11、12和13）将错误代码紧跟返回地址压入堆栈。错误代码识别引起中断的选择符（Selector）。如果不包括选择符，则错误代码为0。

17

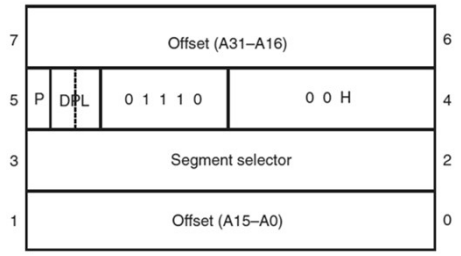
保护模式中断操作

- 保护模式下的中断与实模式几乎完全相同，但中断向量表不同。
- 保护模式使用一组存储在中断描述符表（IDT）中的256个中断描述符取得中断向量。每个描述符占8个字节。
  - 中断描述符表占256×8=2KB字节。

18

保护模式中断操作

- 中断描述符
  - 中断服务程序的地址：Segment Selector和Offset
  - 描述符是在内存中：P
  - 描述符特权级：DPL



19

保护模式中断操作

- 除了IDT和中断描述符，保护模式中断像实模式中断一样发挥功能。
- 用IRET或IRETD指令从两种中断返回。
  - 64位模式下，用IRETQ从中断返回。
- 实模式的中断向量可以转换成保护模式中断。
  - 复制中断向量表中的中断服务程序地址，并将其转换成存储于中断描述符中的32位偏移地址。
  - 全局描述符表将存储器的前1MB标识为中断段，相应的选择符和段描述符可放在全局描述符表中。

20

中断标志位

- IF：中断允许标志
  - STI、CLI指令
- TF：陷阱标志
  - 当TF=1，它在每条指令执行之后产生一个陷阱中断（类型1）。这也是我们常成陷阱中断为单步中断的原因。
  - 当TF=0，程序正常执行。
  - 没有特殊的指令来置位和清除陷阱标志。
    - PUSHF/PUSHFD，POPF/POPFD

21

置位TF

- 例，置位TF的一个中断服务程序。

```
TRON  PROC  FAR  USES  AX  BP
      MOV  BP, SP
      MOV  AX, [BP+8]
      OR    AH, 1
      MOV [BP+8], AX
      IRET
TRON EDNP
```

22

清除TF

- 例，清除TF的一个中断服务程序。

```
TRON  PROC  FAR  USES  AX  BP
      MOV  BP, SP
      MOV  AX, [BP+8]
      AND  AH, 0FEH
      MOV [BP+8], AX
      IRET
TRON EDNP
```

23

跟踪过程

- 假定INT 40H指令访问TRON，INT 41H访问TROFF。下面的程序用于跟踪一个紧跟在INT 40H指令后的程序，对应于中断类型1或陷阱中断。

```
REGS    DD   8 DUP(?)          ;space for registers

TRACE   PROC FAR USES EBX

        MOV  EBX,OFFSET REGS
        MOV  [EBX],EAX         ;save EAX
        POP  EAX
        PUSH EAX
        MOV  [EBX+4],EAX       ;save EBX
        MOV  [EBX+8],ECX       ;save ECX
        MOV  [EBX+12],EDX      ;save EDX
        MOV  [EBX+16],ESP      ;save ESP
        MOV  [EBX+20],EBP      ;save EBP
        MOV  [EBX+24],ESI      ;save ESI
        MOV  [EBX+28],EDI      ;save EDI
        IRET

TRACE   ENDP
```

24

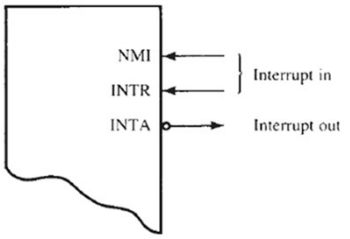
本章内容

- 基本中断处理
- 硬件中断
- 扩展中断结构
- 8259A可编程中断控制器
- 中断实例

25

硬件中断

- 微处理器有2个硬件中断输入：
  - NMI、INTR



所有型号的Intel微处理器上的中断引脚

26

硬件中断

- NMI：一旦激活NMI输入，就发生类型2中断。
- INTR：INTR的输入必须外部译码，以选择一个中断向量。
  - INTR引脚可以选择任何中断向量，但通常只使用20H~FFH之间的中断向量。
  - Intel保留00H~1FH之间的中断。
- INTA#：用于响应INTR输入的一个输出引脚，将向量类型号加载到数据总线D7~D0上。

27

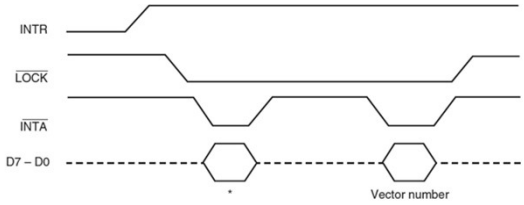
硬件中断

- NMI：边沿触发，在上升沿申请中断。
  - 在上升沿之后，NMI引脚必须保持逻辑1，直到微处理器识别它。
  - 在上升沿被识别之前，NMI引脚必须保持逻辑1至少2个时钟周期。
- NMI常用于奇偶校验错误和其他主要系统故障（如掉电）。
  - 响应这种类型的中断时，微处理器将所有内部寄存器存于使用电池的备份存储器或EEPROM中。

28

硬件中断

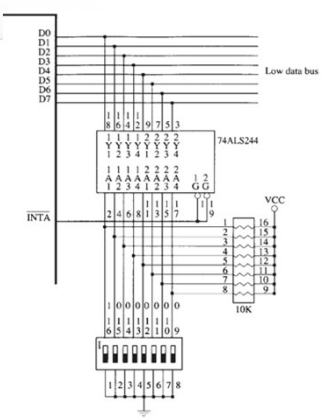
- INTR是电平敏感的，必须保持在逻辑1电平直到被识别为止。
- INTR和INTA的时序图



29

硬件中断

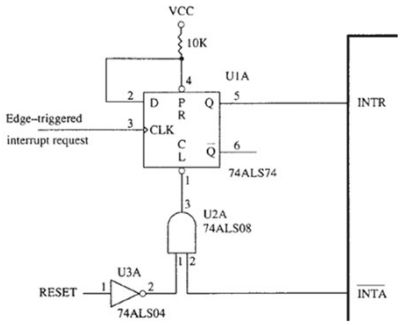
- INTA#的响应



30

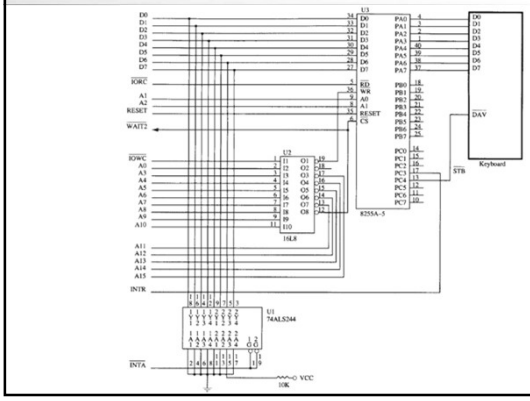
硬件中断

- 将INTR转换为边沿触发



31

82C55键盘中断



32

82C55键盘中断

- 读取按键的中断服务程序

```
PORTA   EQU   500H
CNTR    EQU   506H

FIFO    DB    256 DUP(?)          ;queue

INP     DD    FIFO                ;input pointer
OUTP    DD    FIFO                ;output pointer

KEY     PROC  FAR USES EAX EBX EDX EDI

        MOV   EBX,CS:INP          ;get pointers
        MOV   EDI,CS:OUTP

        INC   BL
        .IF BX == DI              ;if full
              MOV   AL,8
              MOV    DX,CNTR
              OUT   DX,AL         ;disable 82C55 interrupt
        .ELSE                     ;if not full
              DEC  BL
              MOV  DX,PORTA
              IN   AL,DX          ;read key code
              MOV  CS:[BX]        ;save in queue
              INC  BYTE PTR CS:INP
        .ENDIF
        IRET

KEY     ENDP
```

33

82C55键盘中断

- 从FIFO队列读出数据的过程

```
READQ   PROC  FAR USES EBX EDI EDX

        .REPEAT
              MOV  EBX,CS:INP          ;get pointers
              NOV  EDI,CS:OUTP
        .UNTIL EBX == EDI              ;while empty

        MOV   AH,CS:[EDI]              ;get data
        MOV   AL,9
        MOV   DX,CNTR
        OUT   DX,AL                    ;enable 52C55 interrupt
        INC   BYTE PTR CS:OUTP
        RET

READQ   ENDP
```

34

本章小结

- 基本中断处理
  - 中断向量，标志位，实模式中断，保护模式中断
- 硬件中断
  - 引脚，特点

35

作业

- 习题13，习题27

36