



# 从零开始

# 自制操作系统

# OPERATING SYSTEM FROM SCRATCH

# O Brave New Kernel

## 外中断应用：APIC 计时器与 RTC

EP 8-3

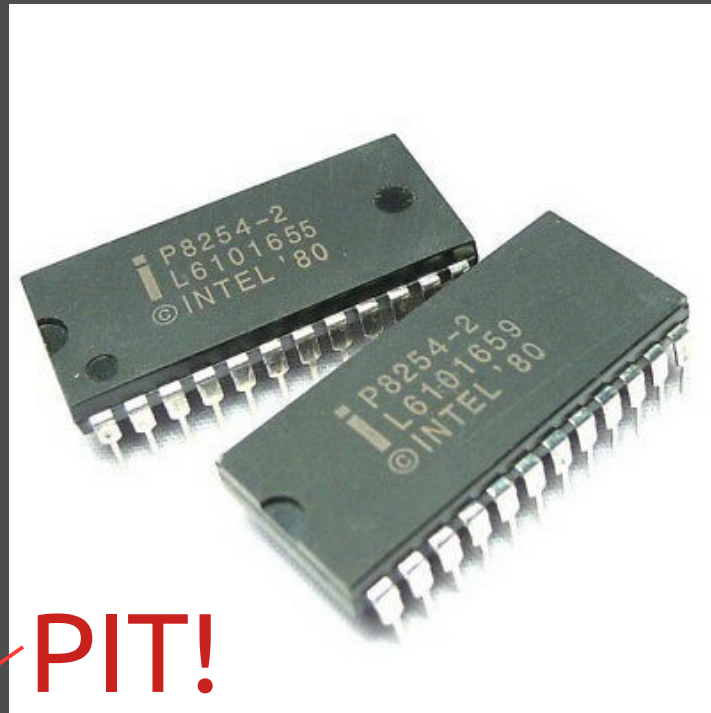




# 计时器？

向 CPU 发送固定间隔的、周期性的中断

1. 告诉 CPU 什么时候该进行任务调度（schedule）
2. 实现更加精确地等待与延时
3. And more....



**PIT!**

可编程间隔计时器（Programmable Interval Timer）



# LAPIC Timer



更加高的精度

可针对每个 CPU 进行独立的设置

初始计数器 ( Current Counter Register )

```
if (ICR 发生写入事件) {  
    do {  
        CCR = ICR;  
        while (CCR--);  
        触发中断;  
    } while (Periodic_Mode);  
}
```

当前计数器 ( Current Counter Register )

是否为周期模式



## 两个计数器

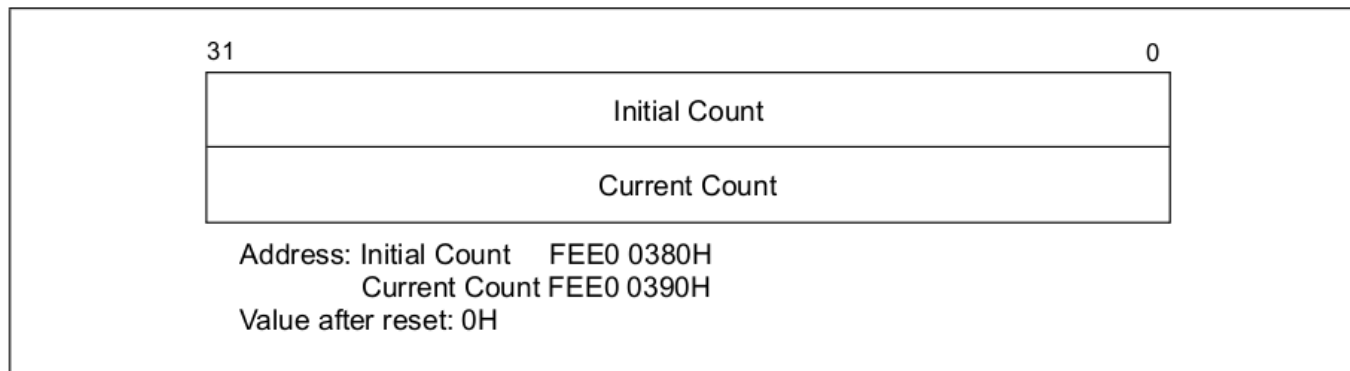
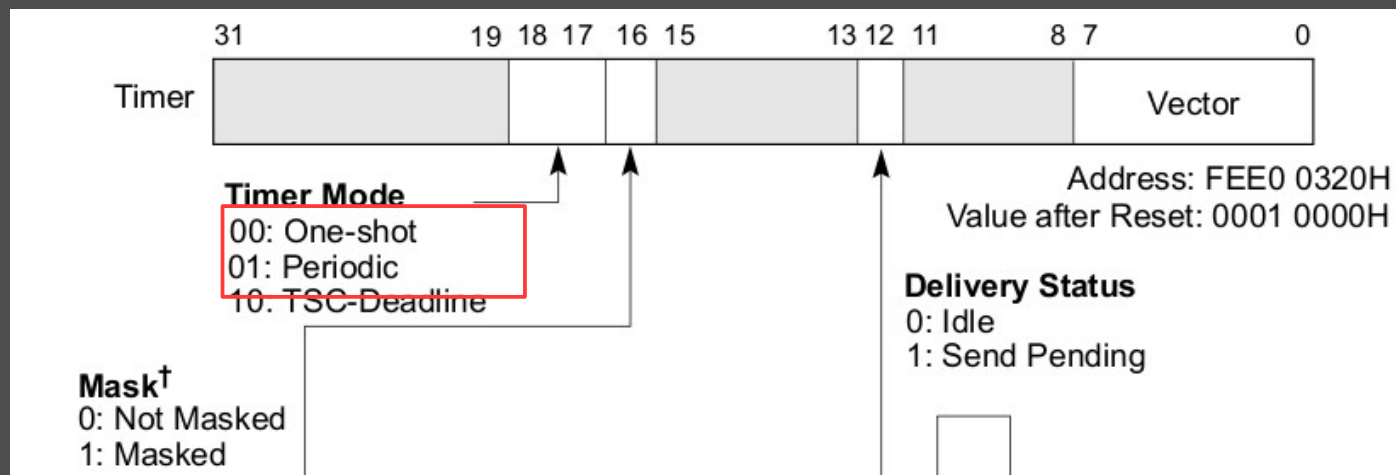


Figure 10-11. Initial Count and Current Count Registers

```
if (ICR 发生写入事件) {  
    do {  
        CCR = ICR;  
        while (CCR--);  
        触发中断;  
    } while (Periodic_Mode);  
}
```



```
if (ICR 发生写入事件) {  
    do {  
        CCR = ICR;  
        while (CCR--);  
        触发中断;  
    } while (Periodic_Mode);  
}
```



## 这玩意儿的频率？

```
do {  
    CCR = ICR;  
    while (CCR--);  
    触发中断;  
} while (Periodic_Mode);
```

How fast?

$$F_{\text{timer}} = \frac{1}{k} F_{\text{CPU}}$$

CPU 时钟频率!



Address: FEE0 03E0H  
Value after reset: 0H

Divide Value (bits 0, 1 and 3)

- 000: Divide by 2
- 001: Divide by 4
- 010: Divide by 8
- 011: Divide by 16
- 100: Divide by 32
- 101: Divide by 64
- 110: Divide by 128
- 111: Divide by 1

Figure 10-10. Divide Configuration Register



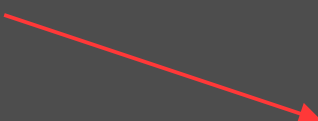
# 我们怎么知道 CPU 时钟频率？

手动测量！

使用一个固定且已知频率时钟作为参考

LAPIC 计时器发生中断时，经过了多少个固定间隔的 ticks.

$$\frac{ICR}{F_{\text{timer}}} = \frac{t}{F_{\text{RTC}}}$$



我们将使用主板上的  
CMOS 来作为时间参照



基本上都是与 MC146818A 兼容



MC146818A

## Advance Information

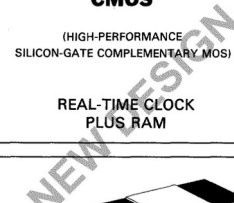
### REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818A Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818A uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power. The Real-Time Clock plus RAM has two distinct uses. First, it is

### CMOS

(HIGH-PERFORMANCE SILICON-GATE COMPLEMENTARY MOS)

REAL-TIME CLOCK PLUS RAM



## REGISTER A (\$0A)

MSB				LSB				Read/Write Register except UIP
b7	b6	b5	b4	b3	b2	b1	b0	
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0	

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes	—	N = 0
1.048576 MHz	0	0	1	Yes	—	N = 2
32.768 kHz	0	1	0	Yes	—	N = 7
Any	1	1	0	No	Yes	—
Any	1	1	1	No	Yes	—

Note: Other combinations of divider bits are used for test purposes only.

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Select Bits Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
				Periodic Interrupt Rate t <sub>PI</sub>	SQW Output Frequency	Periodic Interrupt Rate t <sub>PI</sub>	SQW Output Frequency
RS3	RS2	RS1	RS0				
0	0	0	0	None	None	None	None
0	0	0	1	30.517 μs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 μs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 μs	8.192 kHz	122.070 μs	8.192 kHz
0	1	0	0	244.141 μs	4.096 kHz	244.141 μs	4.096 kHz
0	1	0	1	488.281 μs	2.048 kHz	488.281 μs	2.048 kHz
0	1	1	0	976.562 μs	1.024 kHz	976.562 μs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

## REGISTER B (\$0B)

MSB				LSB			
b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

Read/Write  
Register

IRQ 0  
IRQ 1  
IRQ 2

Timer Output 0  
Keyboard (Output Buffer Full)  
Interrupt from CTLR 2  
**Realtime Clock Interrupt**  
Software Redirected to INT 0AH (IRQ 2)  
Reserved  
Reserved  
Reserved  
Coprorocessor  
Fixed Disk Controller  
Reserved  
Serial Port 2  
Serial Port 1  
Parallel Port 2  
Diskette Controller  
Parallel Port 1

IRQ 8  
IRQ 9  
IRQ 10  
IRQ 11  
IRQ 12  
IRQ 13  
IRQ 14  
IRQ 15

IRQ 3  
IRQ 4  
IRQ 5  
IRQ 6  
IRQ 7

**PIE** — The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the  $\overline{\text{IRQ}}$  pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks  $\overline{\text{IRQ}}$  from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal MC146818A functions, but is cleared to "0" by a  $\overline{\text{RESET}}$ .



## 31 Real Time Clock (RTC)

---

### 31.1 RTC Indexed Registers Summary

The RTC contains two sets of indexed registers that are accessed using the two separate **Index and Target registers (70h/71h or 72h/73h)**, as shown in the following table:

RTC (Standard) RAM Bank

Index	Name
-------	------

00h	Seconds
-----	---------

01h	Seconds Alarm
-----	---------------

02h	Minutes
-----	---------

03h	Minutes Alarm
-----	---------------

04h	Hours
-----	-------

05h	Hours Alarm
-----	-------------

06h	Day of Week
-----	-------------

07h	Day of Month
-----	--------------

08h	Month
-----	-------

09h	Year
-----	------

0Ah	Register A
-----	------------

0Bh	Register B
-----	------------

0Ch	Register D
-----	------------

0Dh	Register D
-----	------------


0Bh-7Fh	114 Bytes of User RAM
---------	-----------------------



## 所以我们的步骤如下

- #1. 安装 RTC 与 LAPIC Timer 的临时中断服务例程
- #2. 配置 RTC 的频率（我们使用 1024Hz）
- #3. 打开中断（sti）
- #4. 写入 ICR（随便找个很大的值）
- #5. PIE 置位
- #6. 阻塞
- #7. 当 LAPIC Timer 触发后，计算频率并关闭 RTC
- #8. 清除掉临时的例程

RTC 例程：每次调用 `t++`


$$\frac{ICR}{F_{timer}} = \frac{t}{F_{RTC}}$$

1024



MC146818A Datasheet

Intel ® 500 Series Chipset Family Platform Controller Hub (PCH) Datasheet  
Vol. 2, Section 31.1

Intel Manual, Vol. 3.A, Section 10.5.4

IBM PC/AT Technical Reference, Section 1.10

# Code Time