



教材：《32位数字信号控制器原理及应用》



# 《DSP原理及应用》

## Lecture4 CPU Timer

黄灿水  
2015.3





## Lecture4 CPU Timer

### 一、CPU Timer 模块概述

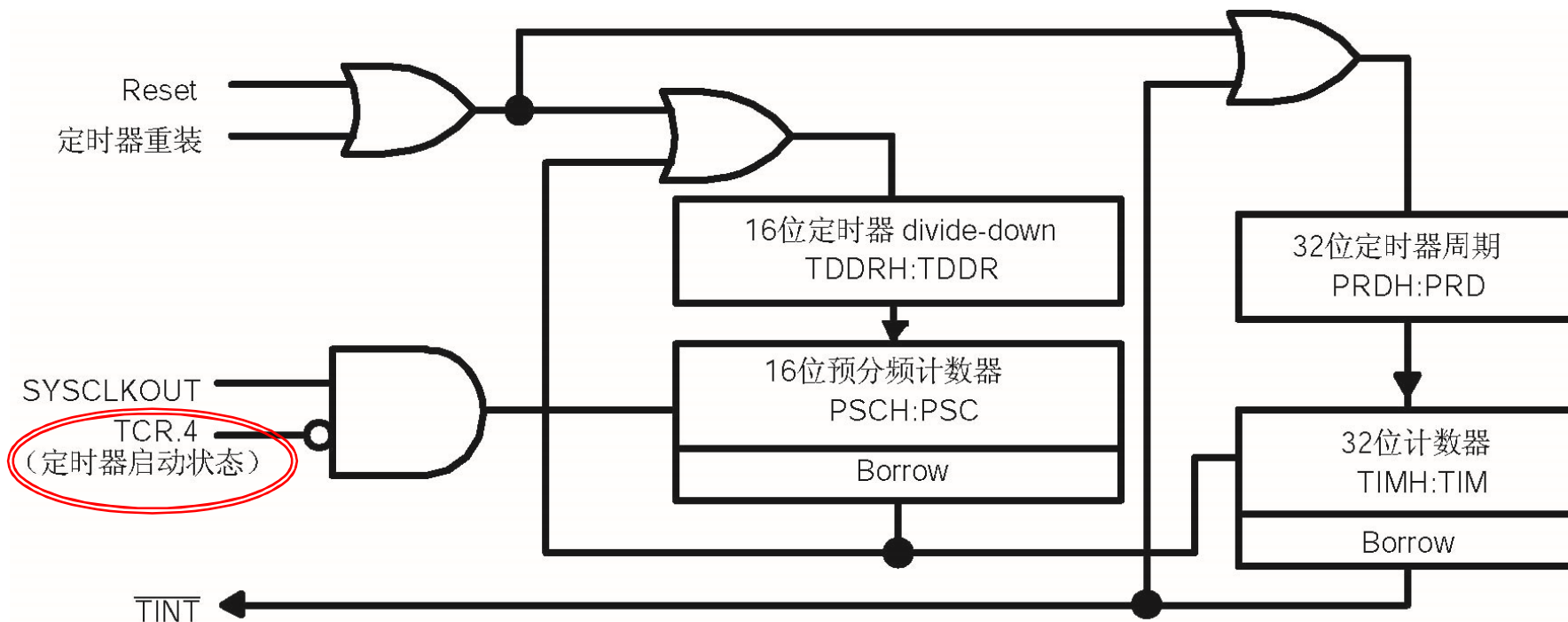
- 3个**32位**的CPU定时器 (Timer0/1/2)
- 可选**时基**: SYSCLKOUT、INTOSC1、INTOSC2、XCLKIN
- 16位时钟**前分频** (TPRH: TPR)
- 3个可预设、重载的**周期寄存器** (PRDH: PRD)
- 3个32位**减计数器** (TIMH: TIM)
- 3个CPU定时器**中断信号** (TINT0、TINT1、TINT2)
- CPU Timer1可为BIOS提供时基
- CPU Timer2可为实时操作提供时基





# Lecture4 CPUS timer

## 一、CPU timer 模块概述



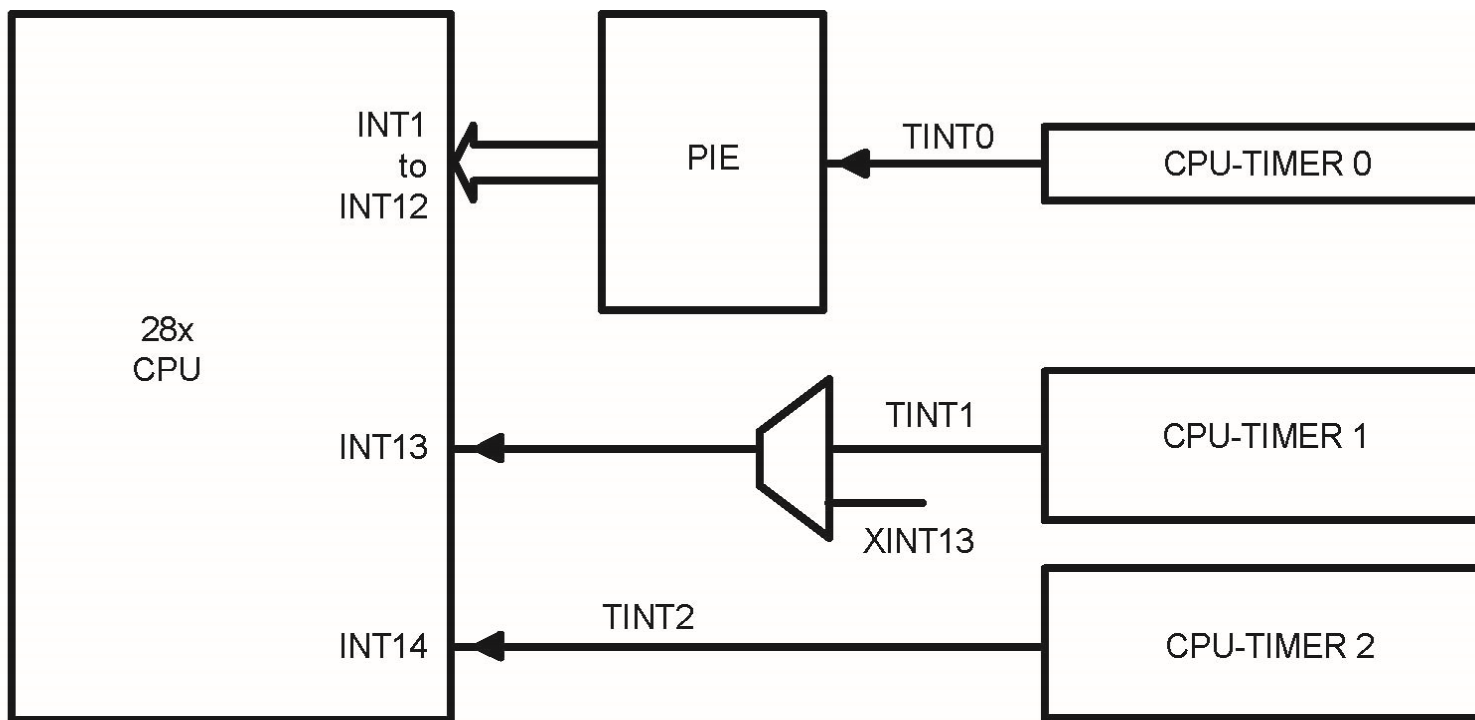
CPU定时器





# Lecture4 CPUMTimer

## 一、CPUMTimer模块概述



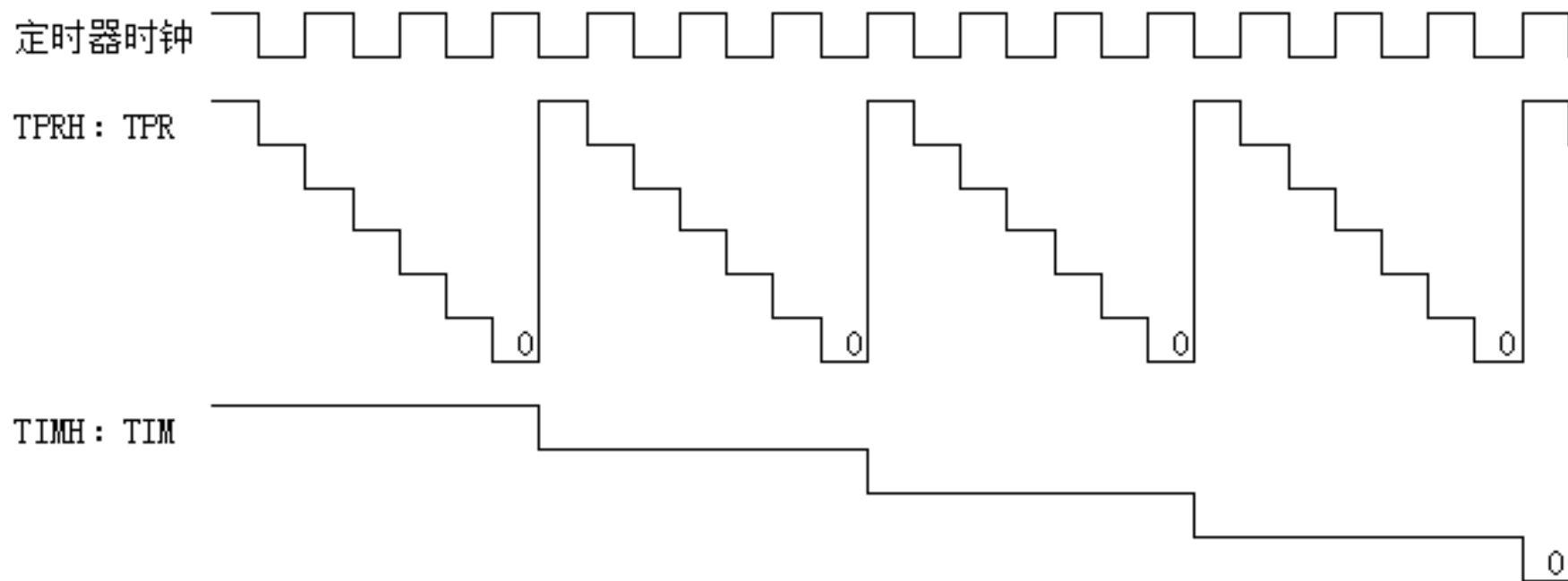
**CPU定时器中断信号和输出信号**





## Lecture4 CPUS timer

### 二、CPUS timer工作原理



$$\text{定时时间} T = (\text{TIMH:TIM} + 1) * (\text{TPRH:TPR} + 1) * \text{定时器时钟}$$







## Lecture4 CPU Timer

### 三、CPU Timer寄存器

名称	地址	大小(W)	描述
TIMER0TIM	0x0C00	1	CPU定时器0, 计数器寄存器
TIMER0TIMH	0x0C01	1	CPU定时器0, 计数器寄存器高16位字
TIMER0PRD	0x0C02	1	CPU定时器0, 周期寄存器
TIMER0PRDH	0x0C03	1	CPU定时器0, 周期寄存器高16位字
TIMER0TCR	0x0C04	1	CPU定时器0, 控制寄存器
TIMER0TPR	0x0C06	1	CPU定时器0, 预分频寄存器
TIMER0TPRH	0x0C07	1	CPU定时器0, 预分频寄存器高16位字
TIMER1TIM	0x0C08	1	CPU定时器1, 计数器定时器
TIMER1TIMH	0x0C09	1	CPU定时器1, 计数器定时器高16位字
TIMER1PRD	0x0C0A	1	CPU定时器1, 周期寄存器
TIMER1PRDH	0x0C0B	1	CPU定时器1, 周期寄存器高16位字
TIMER1TCR	0x0C0C	1	CPU定时器1, 控制寄存器
TIMER1TPR	0x0C0E	1	CPU定时器1, 预分频寄存器
TIMER1TPRH	0x0C0F	1	CPU定时器1, 预分频寄存器高16位字



## Lecture4 CPU Timer

### 三、CPU Timer寄存器

(续)

名称	地址	大小(W)	描述
TIMER2TIM	0x0C10	1	CPU定时器2, 计数器定时器
TIMER2TIMH	0x0C11	1	CPU定时器2, 计数器定时器高16位字
TIMER2PRD	0x0C12	1	CPU定时器2, 周期寄存器
TIMER2PRDH	0x0C13	1	CPU定时器2, 周期寄存器高16位字
TIMER2TCR	0x0C14	1	CPU定时器2, 控制寄存器
TIMER2TPR	0x0C16	1	CPU定时器2, 预分频寄存器
TIMER2TPRH	0x0C17	1	CPU定时器2, 预分频寄存器高16位字

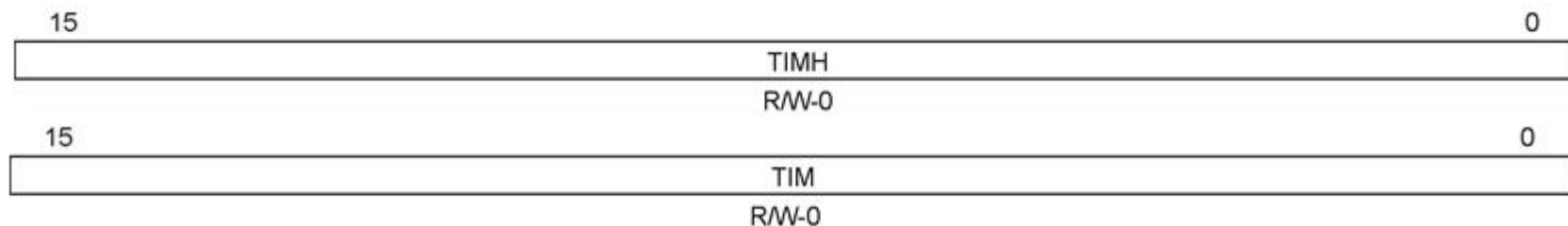




## Lecture4 CPU Timer

### 三、CPU Timer寄存器

#### 1、计数器寄存器 - $TIMER_xTIMH:TIM$ ( $x=0,1,2$ )



计数器寄存器（ $TIMH:TIM$ ）：每个时钟周期递减一次，当（ $TIMH:TIM$ ）递减到零时，（ $TIMH:TIM$ ）寄存器自动装入（ $PRDH:PRD$ ）寄存器的周期值，并产生定时器中断（ $TINT_x$ ）信号。



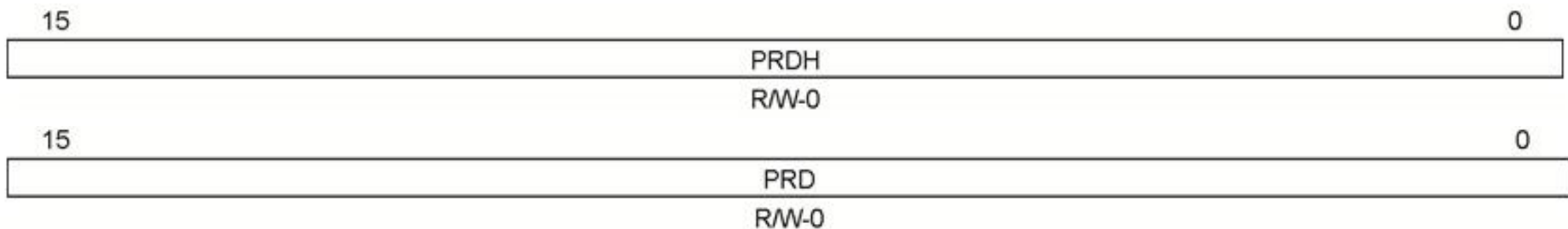




## Lecture4 CPU Timer

### 三、CPU Timer寄存器

#### 2、周期寄存器 - $TIMER_xPRDH:PRD$ ( $x=0,1,2$ )



周期寄存器（ $PRDH:PRD$ ）：当( $TIMH:TIM$ )递减到零时，在下一个定时器输入时钟周期开始时( $TIMH:TIM$ )寄存器装入( $PRDH:PRD$ )寄存器包含的周期值。当定时器控制寄存器（ $TCR$ ）的定时器重装位（ $TRB$ ）置位时，( $PRDH:PRD$ )的内容也被装入到( $TIMH:TIM$ )中。





## Lecture4 CPUTimer

### 三、CPUTimer寄存器

#### 3、控制寄存器 - TIMERxTCR (x=0,1,2)

15	14	13	12	11	10	9	8
TIF	TIE	保留		FREE	SOFT	保留	
RW-0	R/W-0	R-0		RW-0	R/W-0	R-0	
7	6	5	4	3			0
保留	TRB	TSS			保留		
R-0	R/W-0	R/W-0			R-0		

- **BIT15: 中断标志位。** 写0无效，写1清零。
- **BIT14: 中断使能位。** 写0禁止，写1使能。
- **BIT11-10: 仿真模式。** 00下个时钟停，01下个周期停  
10/11自由运行





## Lecture4 CPUMTimer

### 三、CPUMTimer寄存器

#### 3、控制寄存器 - **TIMERxTCR** (x=0,1,2)

15	14	13	12	11	10	9	8
TIF	TIE	保留	FREE	SOFT	保留		
RW-0	R/W-0	R-0	RW-0	R/W-0	R-0		
7	6	5	4	3			0
保留	TRB	TSS	保留				
R-0	RW-0	R/W-0	R-0				

➤ **BIT5: 重载位。**

写0无效。写1时，(TIMH:TIM)装入(PRDH:PRD)的值，并且，  
预分频器计数器(PSCH:PSC)装入(TDDRH:TDDR)的值。

读出总为0。





## Lecture4 CPUTimer

### 三、CPUTimer寄存器

#### 3、控制寄存器 - TIMERxTCR (x=0,1,2)

15	14	13	12	11	10	9	8
TIF	TIE	保留	FREE	SOFT	保留		
RW-0	R/W-0	R-0	RW-0	R/W-0	R-0		
7	6	5	4	3			0
保留	TRB	TSS	保留				
R-0	RW-0	R/W-0	R-0				

➤ **BIT4: 停止状态**位。

读出**0**表示**TIMER**正在运行，读出**1**表示**TIMER**被停止。

写**0**启动或重启**TIMER**，复位时被清零，**TIMER**立刻启动。

写**1**停止**TIMER**。

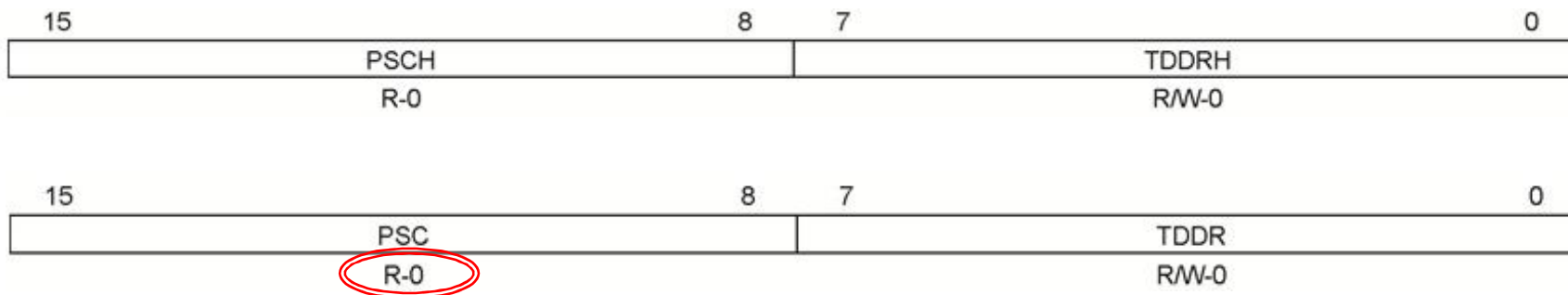




## Lecture4 CPU Timer

### 三、CPU Timer寄存器

#### 4、预分频寄存器 - $TIMER_xTPRH:TPR$ ( $x=1, 2, 3$ )



- **PSCH:PSC - 预分频计数器**。每个定时器时钟(**PSCH:PSC**)减1，到达0后下一个定时器时钟，(**PSCH:PSC**)装入(**TDDR:TDDR**)的值，且(**TIMH:TIM**)减1。置位(**TRB**)时**PSCH:PSC**也被重装。复位**PSCH:PSC**被清0。





## Lecture4 CPU Timer

### 四、编程实现

#### 1、DSP2802x\_CpuTimers.h 和 DSP2802x\_CpuTimers.c

**A、寄存器定义：** CpuTimer0Regs.寄存器名.bit.位名

CpuTimer1Regs.寄存器名.bit.位名

CpuTimer2Regs.寄存器名.bit.位名

寄存器名：TIM(32位)、PRD (32位)、TCR (16位)、

TPRH (16位)、TPR (16位)

例如： CpuTimer0Regs.TCR.bit.TSS = 0;

CpuTimer1Regs.TCR.all = 0x4001;

CpuTimer1Regs.PRD.all = 0x00000000;







## Lecture4 CPU Timer

### 四、编程实现

#### 1、DSP2802x\_CpuTimers.h 和 DSP2802x\_CpuTimers.c

#### B、定时器常用函数

```
void InitCpuTimers();
```

```
void ConfigCpuTimer(struct CPUTIMER_VARS *Timer,  
float Freq, float Period);
```

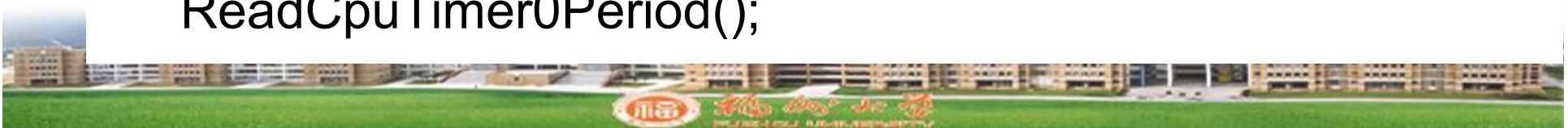
```
StartCpuTimer0();
```

```
StopCpuTimer0();
```

```
ReloadCpuTimer0();
```

```
ReadCpuTimer0Counter();
```

```
ReadCpuTimer0Period();
```





## Lecture4 CPUTimer

### 四、编程实现

#### 1、DSP2802x\_CpuTimers.h 和 DSP2802x\_CpuTimers.c

#### B、定时器常用函数

```
void ConfigCpuTimer(struct CPUTIMER_VARS *Timer,  
float Freq, float Period);
```

例如: `ConfigCpuTimer(&CpuTimer0, 60, 1000000);`

上述语句定义Cputimer0时钟是60MHz, 周期是1s





## Lecture4 CPUTimer

### 五、Timer配置步骤

步骤1: 根据定时时间**计算PRD和TPR数值**, 及时钟**分频值**

步骤2: 外设定时器时钟使能**PCLKCR3 (10-8位)**

步骤3: CPUTimer2时钟源选择、预分频**CLKCTL (7-3位)**

步骤4: 相应的**中断**设置 (参考PIE)

步骤5: 引脚**复用**设置 (参考GPIO)

步骤6: **CPUTimer初始化**

步骤7: **启动CPUTimer**

步骤8: 编写CPUTimer**中断服务子程序**





**Thank You & Question?**

