



比较器B

(Comparator_B)

本节内容

- 比较器工作原理
- Comp_B 概述
- Comp_B 结构
- Comp_B 功能使用
 - 模拟输入端
 - 模拟输入开关
 - 比较器
 - 参考电压发生器
 - 输出电路
- Comp_B 中断
- Comp_B 应用举例

比较器工作原理

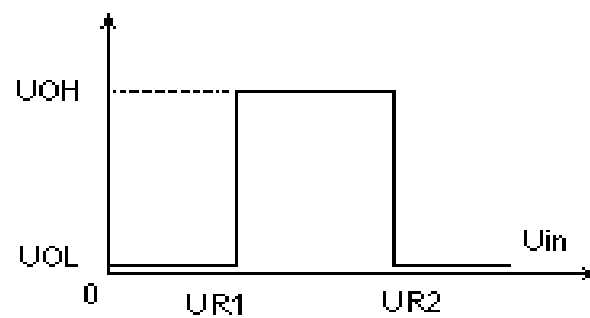
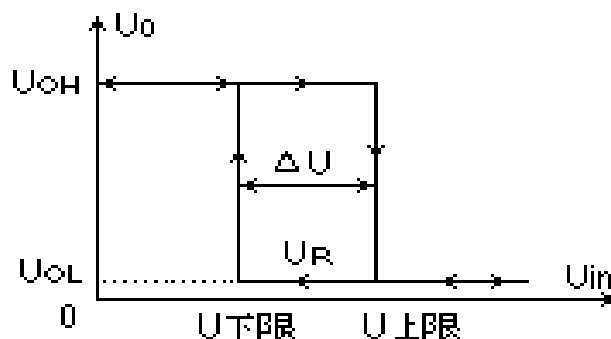
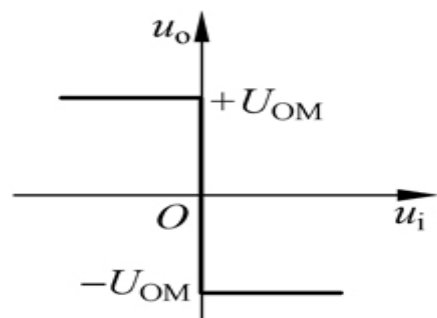
比较器被用于判断输入信号电位之间的相对大小，它至少有两个输入端及一个输出端。

比较方式不同比较器分为：

单限比较器

迟滞比较器

双限比较器



比较器B —— 概述 (1/1)

- 比较器B是为精确的比较测量而设计的，如电池电压监测、产生外部模拟信号、测量电流、电容和电阻，结合其他模块还可实现精确的A/D模数转换功能。
- 比较器B 是工业仪表、手持式仪表等产品设计中的理想选择。
- 5系列之前的MSP430单片机仅有比较器A，5/6系列430单片机升级为比较器B。
- Comp_B是一个模拟电压比较器，涵盖了多达16通道的通用比较器功能。

比较器B —— 概述 (2/2)

Comp_B模块主要特性有：

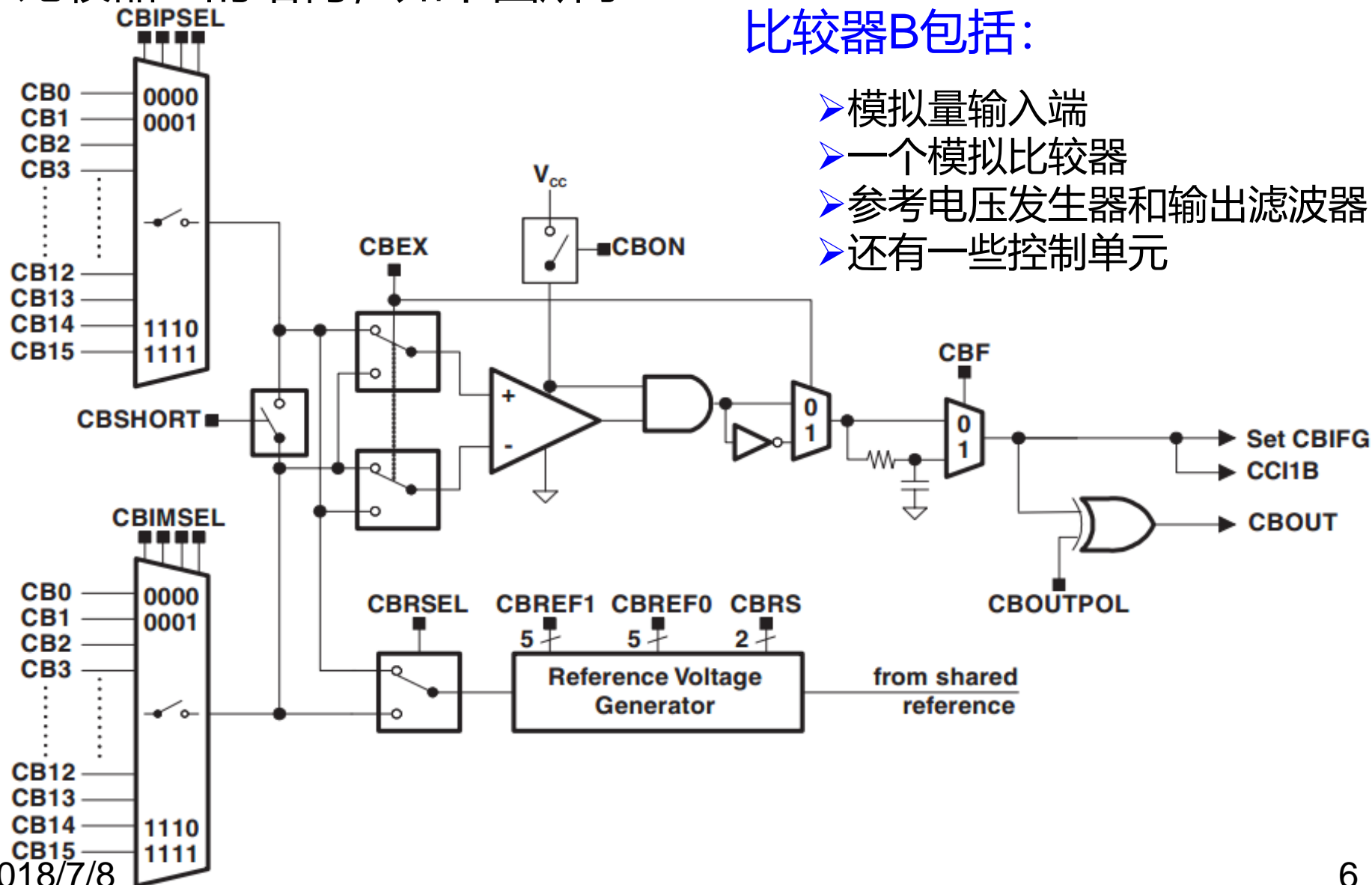
- 正向反向终端输入多路选择器
- 通过软件选择比较器输出的RC滤波
- 可输出到TA的捕获输入
- 软件控制端口输入缓冲
- 具有中断能力
- 可选的参考电压发生器、电压磁滞发生器
- 参考电压输入可选择共用参考电压
- 超低功耗的比较模式
- 低功耗模式支持中断驱动测量系统

比较器B —— 结构 (1/1)

比较器B 的结构，如下图所示：

比较器B包括:

- 模拟量输入端
- 一个模拟比较器
- 参考电压发生器和输出滤波器
- 还有一些控制单元



比较器B —— 功能使用 (1/5)

比较器B的主要功能：指出两个输入电压CB0和CB1的大小关系，然后设置输出信号CBOUT的值。如果 $CB0 > CB1$ 则：CBOUT = 1，否则CBOUT = 0。

◆模拟输入端

参与比较的两个模拟信号通过正、负两个输入电压端：CB0和CB1进入比较器B，输入电流极小。这两个输入端可由用户软件设置，最终能够选择6种信号（CB0、CB1、0.5VCC、0.25VCC、三极管阈值电压和外部参考源），而且能够进行多种组合比较。

硬件提供的比较组合如下：

- 两个外部输入比较
- 每个外部输入与0.5VCC或0.25VCC比较
- 每个外部输入与内部基准电压比较

比较器B —— 功能使用 (2/5)

◆模拟输入开关

➤通过**CBIPSELx**及**CBIMSELx**位，用于选择两个比较器输入终端与相应端口管脚之间连接还是断开。比较器的输入终端可以分别进行控制。

➤通过配置**CBIPSELx/CBIMSELx**位可以实现：

- ▲ 将外部信号连接到比较器的正端或负端
- ▲ 内部参考电压到相应输出端口管脚选择一个路径
- ▲ 将外部电流源应用到比较器的正端或负端
- ▲ 内部多路选择器的两个端口到外部的映射

➤**CBEX**位控制输入多路选择器，改变比较器正端或负端输入信号的顺序。另外，当比较器终端顺序发生改变时，比较器输出信号也发生反转，这使用户可以检测或补偿比较器输入端的偏置电压。

比较器B —— 功能使用 (3/5)

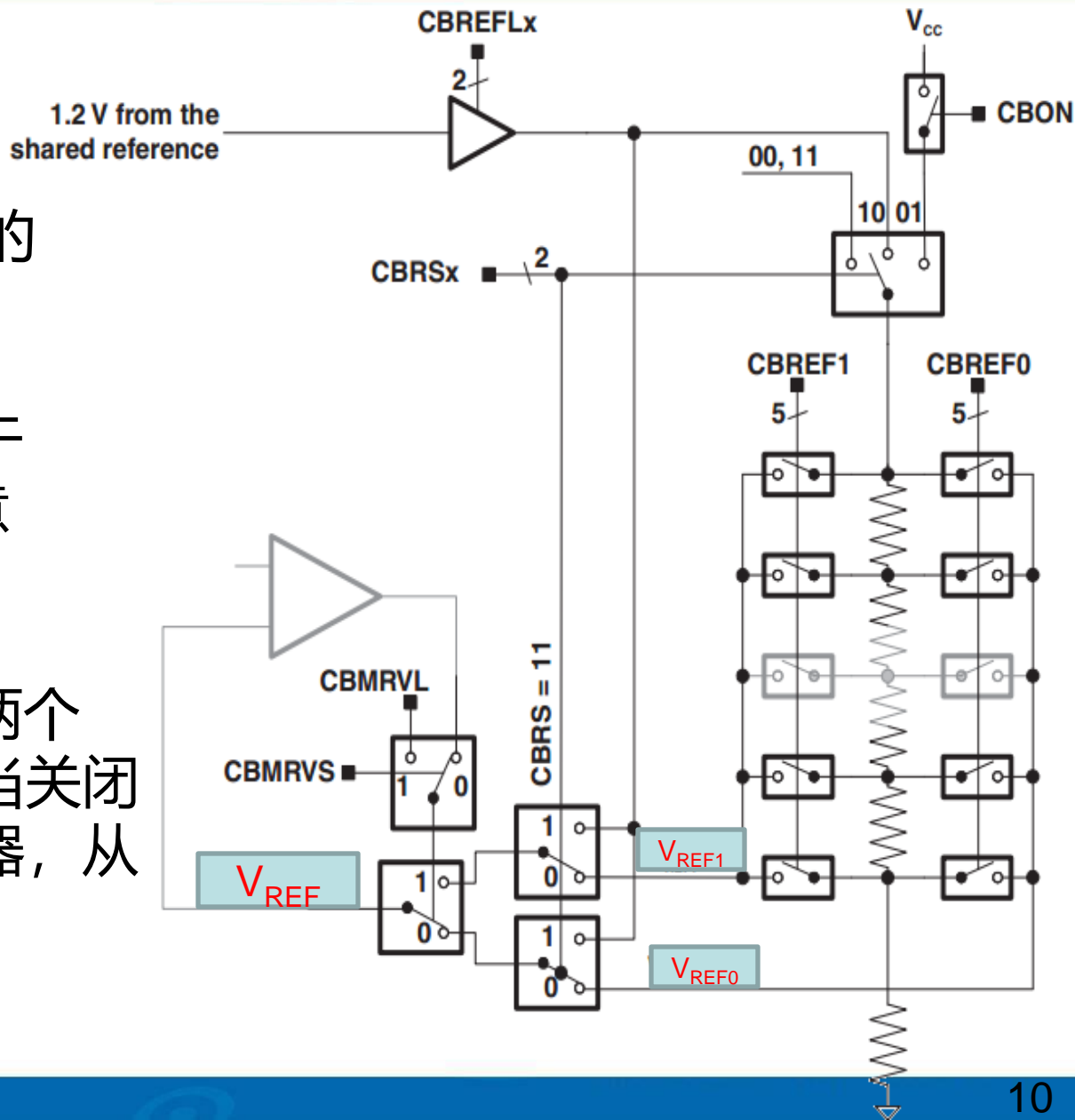
◆比较器

- ▶如比较器B 的结构图所示，比较器位于模拟信号输入和滤波输出之间的部分。
- ▶比较器对正和负输入终端的模拟信号进行比较。如果正端信号大于负端，则比较器输出CBOUT为高。
- ▶其中与门负责将比较输出信号进行整形。
- ▶可以通过CBON 位来关闭或打开比较器。
- ▶控制位CBEX选择正向或反向输出。

比较器B —— 功能使用 (4/5)

◆ 参考电压发生器

- ▶ COMP_B参考电压的框图，如右图所示：
- ▶ 参考电压发生器用于产生用于比较器任意输入端的VREF。
- ▶ 如果外部信号用于两个比较器输入端，应当关闭内部参考电压发生器，从而降低电流消耗。



比较器B —— 功能使用 (5/5)

◆ 输出电路

- 最终输出信号的上升沿或下降沿可以设置为具有中断能力。
- 如果不使用中断，可将输出信号送给内部其他模块，作为其他模块的一个输入信号；还可以由外部引脚引出。

◆ 另外，比较器B 可以适合于低功耗应用，该模块可通过软件打开/关闭，不用时关闭以便电流消耗最小。

比较器B —— 中断 (1/1)

- 中断标志CBIFG，在比较器输出的上升沿或下降沿时都会置位，上升沿或下降沿由CBIES位选择。
- 如果CBIE及GIE位都置位，CBIFG标志将产生中断请求。

比较器B —— 库函数 (1/3)

◆ 比较器 API分成3组函数：处理初始化和输出，处理中断，处理辅助功能。

◆ 处理初始化和输出的相关API:

➤ **Comp_B_init ()**

➤ **Comp_B_setReferenceVoltage ()**

➤ **Comp_B_enable ()**

➤ **Comp_B_disable ()**

➤ **Comp_B_outputValue()**

比较器B —— 库函数 (2/3)

◆ 处理中断的相关API

- **Comp_B_enableInterrupt ()**
- **Comp_B_disableInterrupt ()**
- **Comp_B_clearInterrupt ()**
- **Comp_B_getInterruptStatus ()**
- **Comp_B_interruptSetEdgeDirection ()**
- **Comp_B_interruptToggleEdgeDirection ()**

比较器B —— 库函数 (3/3)

◆ 处理辅助功能的相关API

- **Comp_B_enableShortOfInputs ()**
- **Comp_B_disableShortOfInputs ()**
- **Comp_B_disableInputBuffer ()**
- **Comp_B_enableInputBuffer ()**
- **Comp_B_IOSwap ()**

比较器B —— 应用举例1 (1/2)

◆例1、使用CompB，比较输入电压（用拨盘电位器取得）和内部参考电压大小，如果大于内部参考电压1.5V，则CBOUT输出高电平，否则输出低电平，用LED亮灭来标识比较结果。

相关程序如下：

```
#include <msp430.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW + WDTHOLD;    // 关闭看门狗定时器
```

```
    _____;                // P1.6输出方向
```

```
    _____;                // 选择P1.6/CBOUT引脚为外设功能
```

```
    _____;                // P8.1输出方向
```

```
    _____;                // P3.7输出方向
```

```
// 使能V+通道，输入通道CB5
```

```
CBCTL0 |= CBIPEN + CBIPSEL_5;
```

```
// 正常电源模式，非高速和低功耗模式
```

```
CBCTL1 |= CBPWRMD_1;
```


比较器B —— 应用举例1 (2/2)

```
// VREF应用到负端,使能内部参考电压选择
// (CBEX=0,不交换比较器输入输出反转顺序)
CBCTL2 |= CBRSEL;
// 选择1.5V作为复用参考电压输入 (CBREFL_1),
// 共用参考电压应用到VCREF,电阻负载关闭 (CBRS_1)
CBCTL2 |= CBRS_1+CBREFL_1;
CBCTL3 |= BIT5;           // 关闭输入缓冲P6.5/CB5
                           // 打开CompB
                           __delay_cycles(75); // 延时,用于comp判断电压大小
while(1)
{
    // 请继续.....
}
return 0;
```

比较器B —— 应用举例2 (1/2)

◆用库函数调用，实现应用举例1相同功能

相关伪代码如下（自己从网上下载库函数，路径在ppt尾页）

```
void main (void)
```

```
{
```

```
    // 关闭看门狗定时器
```

```
    WDT_hold(__MSP430_BASEADDRESS_WDT_A__);
```

```
    // 选择P1.6/CBOUT引脚为外设功能
```

```
    GPIO_setAsPeripheralModuleFunctionOutputPin  
        (__MSP430_BASEADDRESS_PORT3_R__,  
        GPIO_PORT_P1,  
        GPIO_PIN6 );
```

```
    //初始化比较器B
```

```
    // CB5连接到正端， // VREF应用到负端， // 正常电源模式
```

```
    Comp_B_init(__MSP430_BASEADDRESS_COMPB__,  
                COMPB_INPUT5,  
                COMPB_VREF,  
                COMPB_POWERMODE_NORMALMODE,  
                COMPB_FILTEROUTPUT_DLYLVL1,  
                COMPB_NORMALOUTPUTPOLARITY );
```

比较器B —— 应用举例2 (2/2)

// VREF应用到负端，选择参考电压为1.5V

// 参考电压为 1.5 V, // 上线 $1.5 \times (32/32) = 1.5V$,

// 下线 $1.5 \times (32/32) = 1.5V$

Comp_B_configureReferenceVoltage

**(__MSP430_BASEADDRESS_COMPB__,
COMPB_VREFBASE1_5V,
32,
32);**

// 使能COMP模块

Comp_B_enable(__MSP430_BASEADDRESS_COMPB__);

// 延时，用于comp判断电压大小

__delay_cycles(75);

while(1)






















{

// 请继续.....

}

比较器B

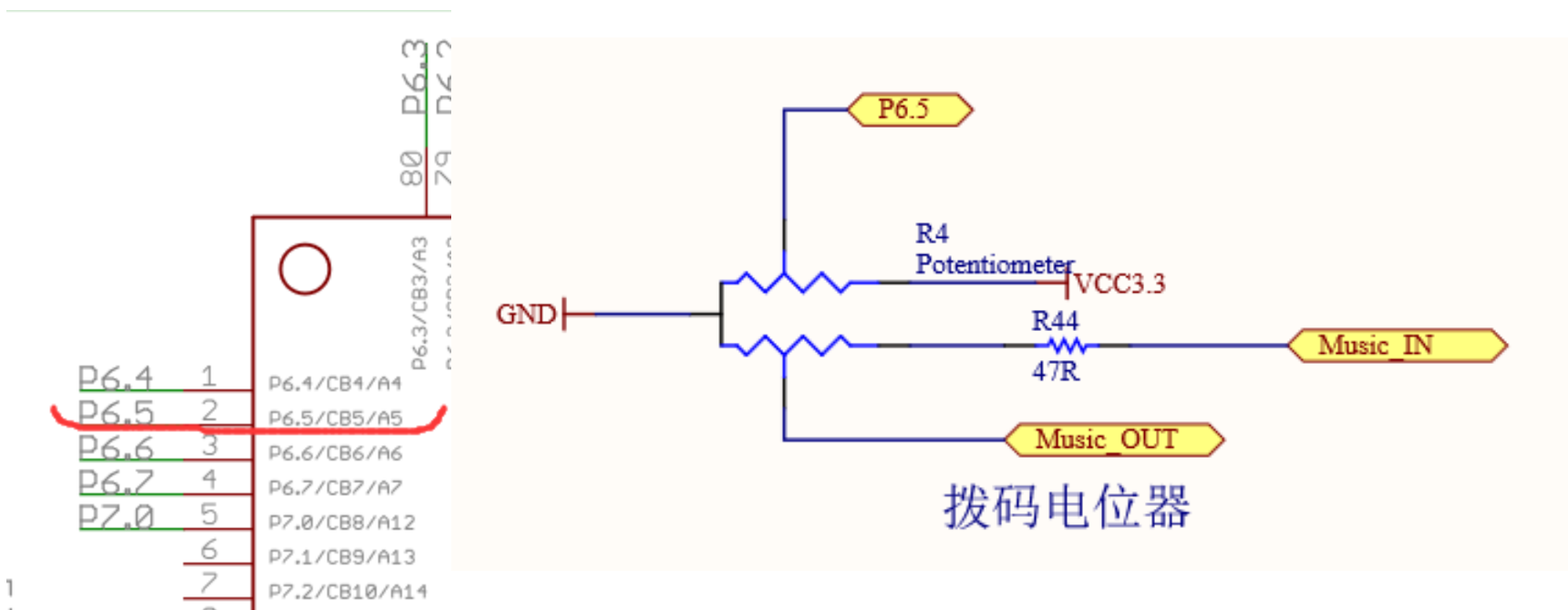
库函数 例程 下载路径

- ▲  MSP430Ware - v:3.80.03.07
 - ▶  Development Tools
 - ▶  Devices
 - ▲  Libraries
 -  MSP430 Function Code Examples
 - ▲  Driver Library
 -  Driver Library Release Notes
 - ▲  MSP430F5xx_6xx
 -  User's Guide
 -  API Programmer's Guide
 - ▲  Example Projects
 - ▶  00_EMPTYPROJECT
 - ▶  ADC10_A
 - ▶  ADC12_A
 - ▶  AES
 - ▶  BATTBAK
 - ▲  COMP_B
 -  comp_b_ex1_2VRef
 -  comp_b_ex2_1V5RefInt
 -  comp_b_ex3_avccRefULPM
 -  comp_b_ex4_1V5RefInt

比较器B实验任务 (1、2)

- 任务1:

参考**比较器B — 应用举例1**例程，学习代码，完成实验，通过调节拨盘电位器，获得3.3V以下需要的电压，与程序设计的1.5V参考电压作比较驱动LED灯亮灭。



- 任务2: 用调用库函数方式实现任务1功能。

比较器B实验作业

- 输入给定伪代码，试着调试通过，了解触摸电容按键功能。

本实验使用 MSP430F5529 触摸按键模块、LED 模块。利用 MSP430F5529 中的比较器B 来判断触摸按键的充放电，通过 CPU 时钟算出电容的振荡频率，再判断是否有按键被触摸，如果有按键触摸则通过相应的 LED 显示当前的状态。



比较器-触摸电容按键实验



模数转换器

(Analog-to-Digital Converter)

本节内容

- 介绍ADC工作原理
- 介绍MSP430 ADC模块的工作原理

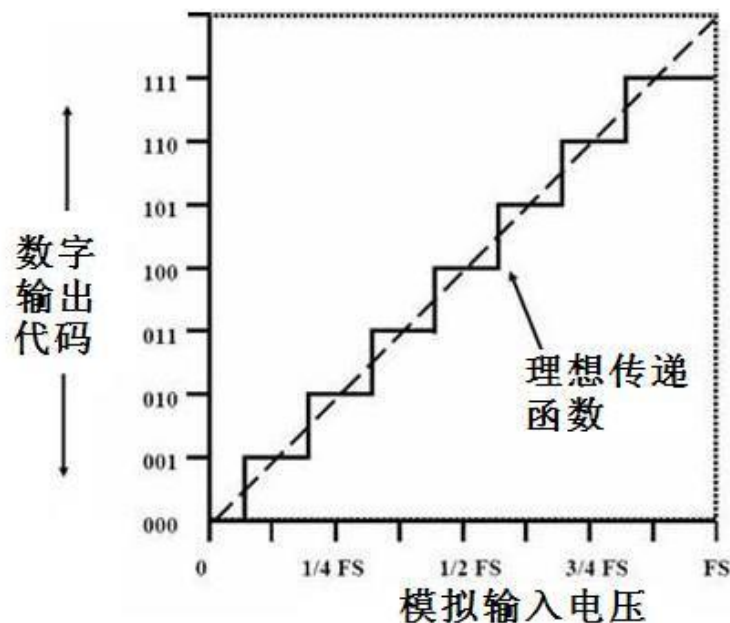
ADC工作原理

- 模数转换器(ADC)

Analog-to-Digital Converter

模拟量-->数字量的一种电子器件或电路

- 模数转换器（ADC）从信号系统中采集信号电压，经过信号处理之后，将其转换为等效的数字量



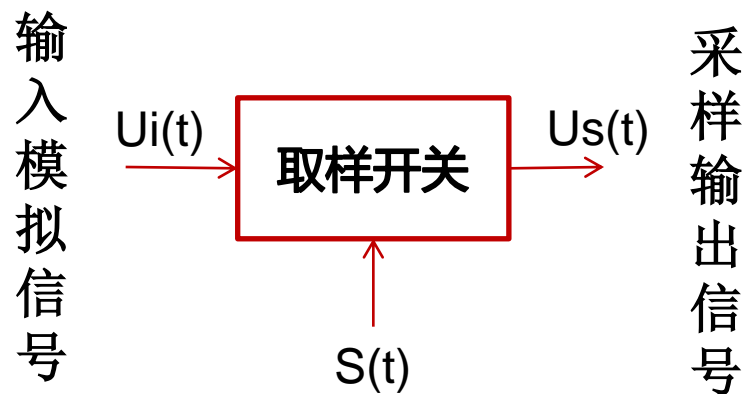
基本过程 (1/6)

- 模 / 数转换电路的作用是将输入连续变化的模拟信号变换为与其成正比的数字量信号输出。
- 在进行模 / 数(即A / D)转换时, 通常按**取样**、**保持**、**量化**、**编码**四个步骤进行。

基本过程 (2/6)

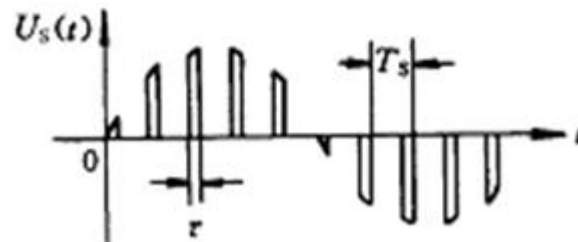
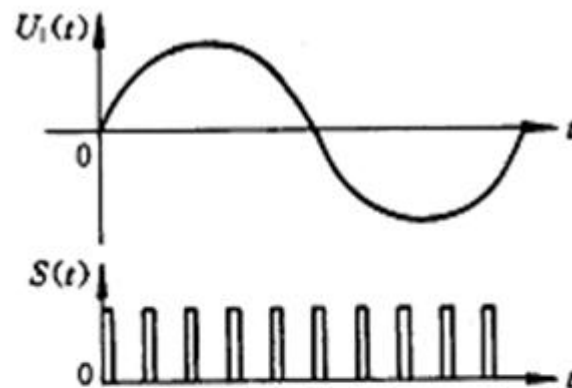
1、取样:

所谓取样, 就是对模拟信号 $U_i(t)$ 进行周期性抽取样值的过程。



采样脉冲

$$f_s \geq 2 f_{max}$$



基本过程(3/6)

$$f_s \geq 2 f_{\max}$$

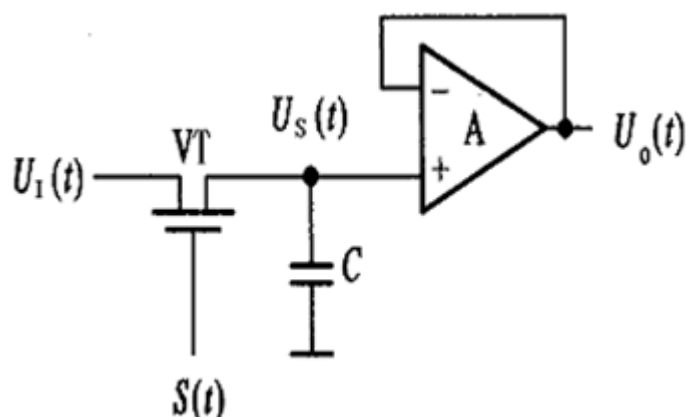
采样定理：在进行模拟/数字信号的转换过程中，当采样频率 f_s 大于信号中最高频率 f_{\max} 的2倍时 ($f_s > 2 f_{\max}$)，采样之后的数字信号完整地保留了原始信号中的信息，一般实际应用中保证采样频率为信号最高频率的2.56~4倍；采样定理又称奈奎斯特定理。

2、保持

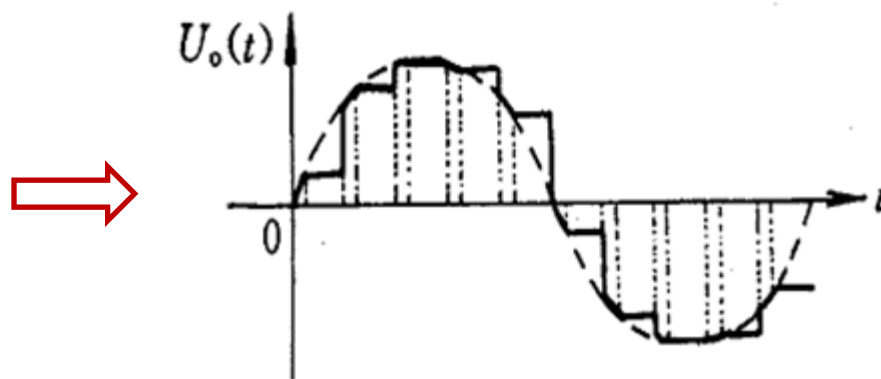
模拟信号经采样后，得到一系列样值脉冲。采样脉冲宽度 τ 一般是很短暂的，在下一个采样脉冲到来之前，应暂时保持所取得的样值脉冲幅度，以便进行转换。因此，在取样电路之后须加保持电路。

基本过程(4/6)

取样保持电路



输出波形



①在采样脉冲 $S(t)$ 到来的时间 τ 内，VT导通， $U_I(t)$ 向电容 C 充电，假定充电时间常数远小于 τ ，则有： $U_O(t) = U_S(t) = U_I(t)$ 。——采样

场效应管VT为采样门，电容 C 为保持电容，运算放大器为跟随器，起缓冲隔离作用。

②采样结束，VT截止，而电容 C 上电压保持充电电压 $U_I(t)$ 不变，直到下一个采样脉冲到来为止。——保持

基本过程(5/6)

取样保持后所得**阶梯波**仍是一个可以连续取值的**模拟量**，但 n 位数字量只能表示 2^n 个数值。因此，用数字量来表示连续变化的模拟量时就有类似于四舍五入的近似问题。

4、量化

用数字信号的最低位1 (LSB)所对应的模拟电压作为量化单位，用 Δ 表示，将样值电压变换为量化单位(Δ)电压整数倍的过程。

5、编码

量化后的离散量用相应的二进制码表示，称作**编码**。

数字量计算方法

ADC内核一般要使用两个参考电压VR+ 和VR-，一般这两个电压可以是用户接入或者是使用内部参考电压。 VR+ 是定义的转换最大值，VR-则是转换的最小值。

以12位分辨率为例。

- $V_{in} > V_{R+}$ ADC12_A输出满量程值0x0FFF;
- $V_{in} < V_{R-}$ ADC12_A输出0;
- $V_{R-} < V_{in} < V_{R+}$ ADC12_A的转换结果满足如下公式:

$$N_{ADC} = 4095 * (V_{in} - V_{R-}) / (V_{R+} - V_{R-})$$

主要性能指标(1/2)

分辨率 R :

- 可以转换成数字量的模拟电压量的最小值
- 最低有效位（LSB），即分辨率单位： $R = \frac{1}{2^n}$
- 分辨率只是规定了数字量输出的位数，而不是ADC的性能；
- MSP430系列大部分芯片中，都提供了一个高精度ADC：

斜率 ADC；

10，12或14位 ADC；

16位Sigma-Delta ($\Sigma - \Delta$) 型 ADC。

主要性能指标(2/2)

转换精度

表示模拟电压实际值与其对应数字量的相对误差。

转换时间：

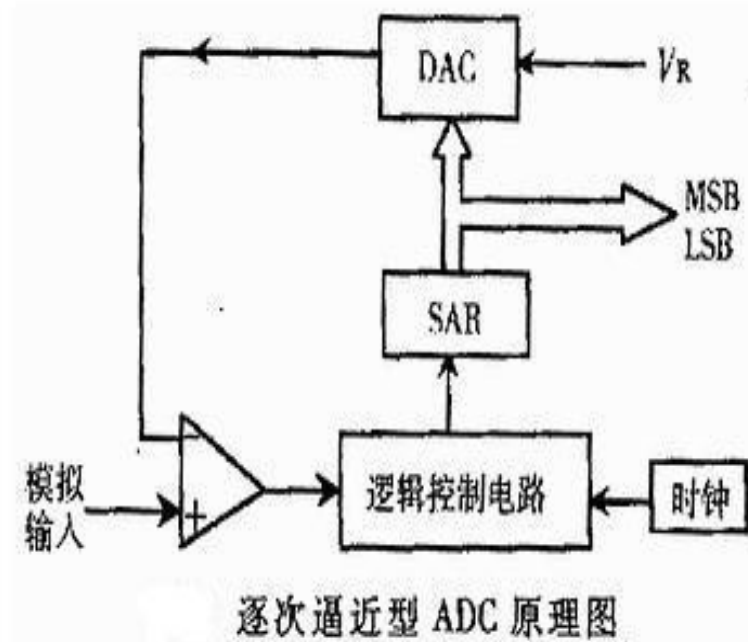
指**ADC**模块完成一次模拟数字转换所需要的时间，转换时间越短越能适应输入信号的变化。转换时间与ADC模块的结构和位数有关。

- **模数转换器(ADC)类别:**

- 逐次逼近型 (SAR) ;
- Sigma Delta型 (SD 或 $\Sigma-\Delta$) ;
- 斜率或双斜率型 (Slope 或 Dual Slope) ;
- 管道型 (Pipeline) ;
- 闪存型 (Flash) ...

逐次逼近型 (SAR)

- 逐次逼近型ADC是应用最为广泛
- 它由比较器、D/A转换器、比较寄存器SAR、时钟发生器以及控制逻辑电路组成，将采样输入信号与已知电压不断进行比较，然后转换成二进制数。



逐次逼近型 (SAR) --工作原理:

- 首先将DAC的最高有效位MSB保存到SAR;
- 接着将该值对应的电压与输入电压进行比较。比较器输出被反馈到DAC, 并在一次比较前对其进行修正;
- 在逻辑控制电路和时钟驱动下, SAR不断进行比较和移位操作, 直到完成LSB的转换。
- 此时所产生的 DAC输出逼近输入电压的 $\pm 1/2\text{LSB}$ 。当每一位都确定后, 转换结果被锁存到SAR并作为ADC输出。

逐次逼近型 (SAR) — 优缺点:

优点:

- 高速，采样速率可达 1MSPS；
- 与其它ADC相比，功耗相当低；
- 在分辨率低于12位时，价格较低。

缺点:

- 在高于14位分辨率情况下，价格较高；
- 传感器产生的信号在进行模/数转换之前需要进行调理，包括增益级和滤波，这样会明显增加成本。

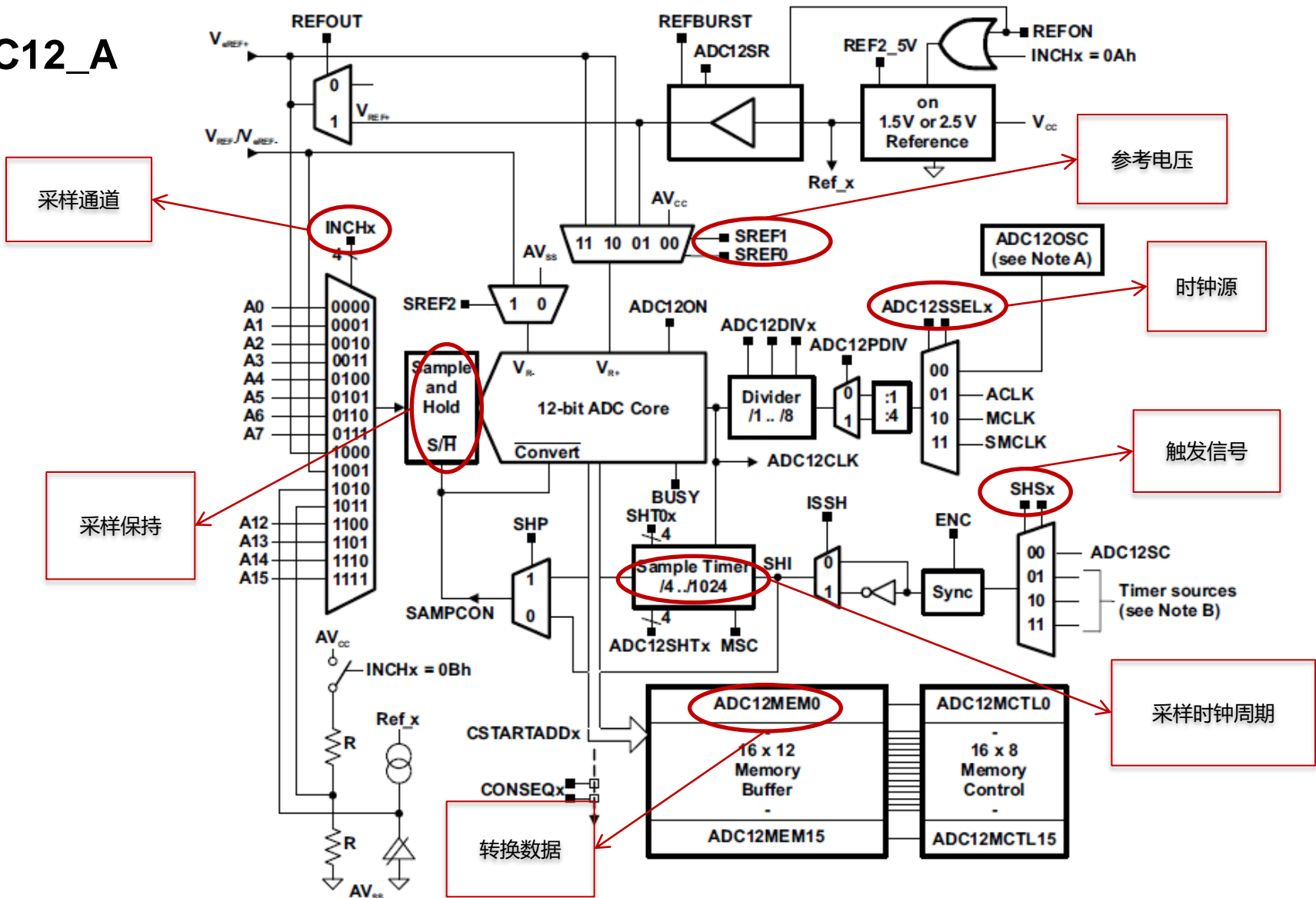
- MSP430F5529的**ADC12_A模块**支持快速**12位**模数转换
- 该模块包含:
 - 12位**ADC核**;
 - 采样选择控制;
 - 参考电流发生器。

Table 28-1. ADC12_A Conversion Result Formats

Analog Input Voltage	ADC12DF	ADC12RES	Ideal Conversion Results	ADC12MEMx
-V _{REF} to +V _{REF}	0	00	0 to 255	0000h - 00FFh
	0	01	0 to 1023	0000h - 03FFh
	0	10	0 to 4095	0000h - 0FFFh
	1	00	-128 to 127	8000h - 7F00h
	1	01	-512 to 511	8000h - 7FC0h
	1	10	-2048 to 2047	8000h - 7FF0h

ADC12_A框图

• ADC12_A 框图:



ADC12_A特征

- 大于**200 ksp/s**的最大转换速率；
- 无失码的**12位**单调转换器；
- 软件或定时器控制的可编程采样保持周期；
- 通过软件或定时器控制转换开始；
- 软件可选择的片上参考电压生成器（**1.5 V， 2.0 V， 2.5 V**）；
- 软件选择的内部或外部参考；
- 多达**12个**可单独配置的外部输入通道；
- 内部温度传感器的转换通道，**AVCC**，和外部参考；
- 可选择的转换时钟源；
- 单通道，重复单通道，序列（自动扫描），重复序列（重复自动扫描）转换模式；
- **ADC**内核和参考电压可单独实现掉电；
- 快速解码的**18位ADC**中断的中断向量寄存器；
- **16个**转换结果存储寄存器。

- **12位ADC 核**(使能位为**ADC12ON**位):
 - 将模拟输入转化为**12**位数字表示;
 - 在**ADC12MEMx**寄存器中存储结果;
 - 转换结果的上限和下限: V_{R+} ; V_{R-} ;
 - 数码输出(N_{ADC}) 结果:
 - 最大值: $N_{ADC} = 0FFFh$, 输入信号 $\geq V_{R+}$;
 - 零: $N_{ADC} = 0000h$, 输入信号 $\leq V_{R-}$;
 - 转换结果:
 - 二进制形式:
$$N_{ADC} = 4096 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$
 - 补码形式

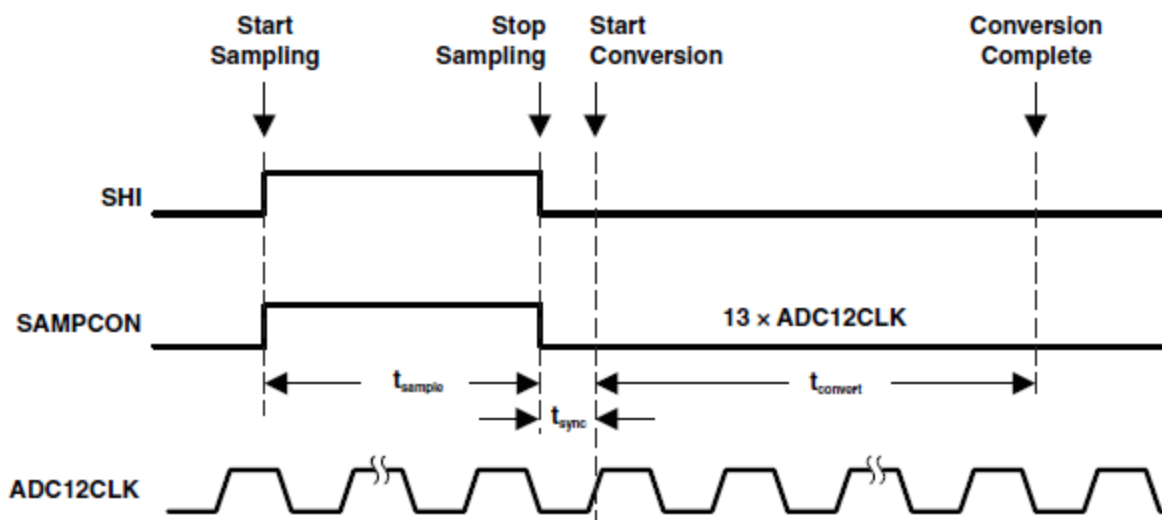
ADC12_A采样和转换时序(1/3)

- 数模转换由**SHI**信号（采样输入信号）的上升沿启动；
- **SHI (SHSx位选择)** 的来源可以是：
 - **00** **ADC12SC** 位；
 - **01** **Timer_A** 输出单元1；
 - **10** **Timer_B** 输出单元0；
 - **11** **Timer_B** 输出单元1。

• 采样时序方式:

– ADC12SHP = 0: 扩展采样模式

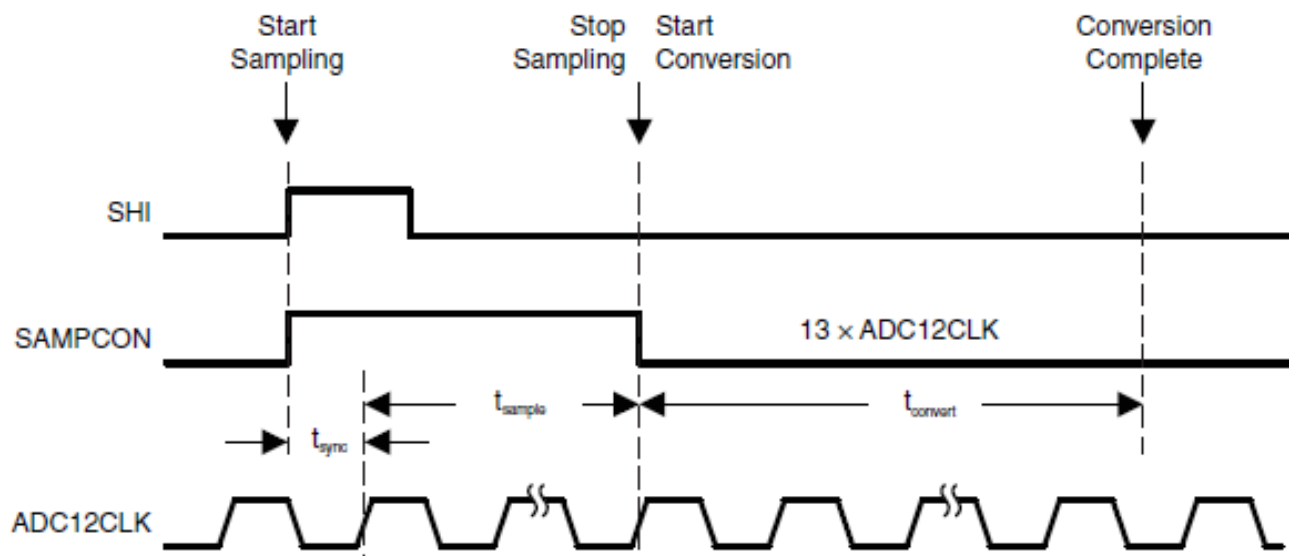
- SHI 信号直接控制 **SAMPCON**(采样保持脉冲模式信号);
- 定义采样周期 t_{sample} 的长度;
- **SAMPCON** = 1 \Rightarrow 采样处于活动状态;
- 高到低**SAMPCON** 转变 \Rightarrow 与**ADC12CLK**同步后开始转换.



• 采样时序方式:

– ADC12SHP = 1: 脉冲模式

- SHI 信号触发采样定时器;
- SHT0x 和 SHT1x 位 (ADC12CTL0) 定义SAMPCON采样周期, t_{sample} ;
- 与 AD12CLK 同步后, 采样定时器保持 **SAMPCON = 1**.



- 16个**ADC12MEMx** 转换存储寄存器(由相关的**ADC12MCTLx**控制寄存器配置)存储转换结果.
- **非连续转换**(单一或重复单一通道):
 - **ADC12CSTARTADDx** 定义转换的第一个和单个的**ADC12MCTLx**.
- **连续转换**(序列或重复序列通道):
 - 序列从**ADC12MCTLx** 寄存器中的值开始, 该值由 **ADC12CSTARTADDx**指定;
 - 指针自动递增到下一个**ADC12MCTLx**, 从而开启下一次转换;
 - **ADC12MCTL15** 之后的转换是 **ADC12MCTL0**;
 - 序列保持运行直至**ADC12EOS** 位信号出现, 此信号是实际运行序列的最后一次转换;
 - 16个 **ADC12MCTLx** 寄存器可以包含多个序列.

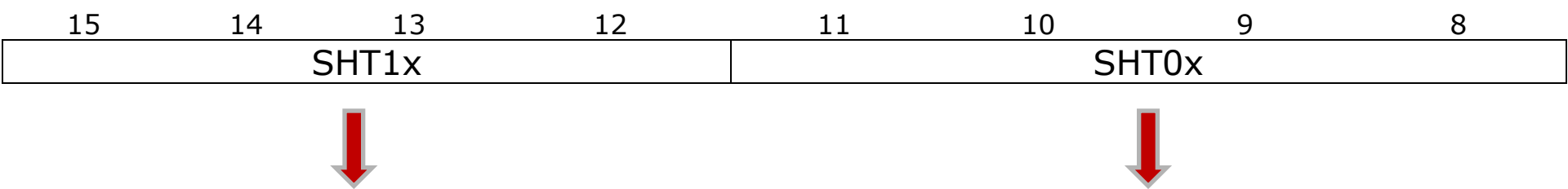
- **ADC12_A 包含 18 个中断源:**
 - **ADC12IFG0-ADC12IFG15:** 当ADC12MEMx存储寄存器加载转化结果时，其对应的ADC12IFGx位将会置位;
 - **ADC12OV, ADC12MEMx 溢出位:**在上一个转换结果未被读取之前，向任意一个ADC12MEMx写入新的转换结果时，ADC12OV位将会置位;
 - **ADC12TOV, ADC12_A 转换时间溢出位:**当前转换未完成就请求另一个采样和转换时，ADC12TOV位将会置位.
- 在单通道模式的转换 ,或序列通道模式完成后，可以触发**DMA**（高速数据传输）传输

ADC12_A中断向量发生器

- **中断向量寄存器ADC12IV** 用来确定是哪个ADC12_A 中断源发出了中断请求.
- **注意事项:**
 - 最高优先级的中断将其中断向量号存储到**ADC12IV**寄存器(计算或添加到程序计数器以自动调用正确的响应程序);
 - 任何对**ADC12IV**寄存器的访问、读或写操作, 都将**自动**重新设置**ADC12OV** 或 **ADC12TOV**的置位条件, 如果二者之一是最高优先级挂起中断;
 - **ADC12IFGx** 位可通过访问其对应的**ADC12MEMx**寄存器而被**自动**重置, 或者可由软件重置;
 - 如果在一个中断服务完成后有另一个中断挂起了, 那么将产生另一个中断.

ADC12_A寄存器 (1/7)

- **ADC12CTL0, ADC12_A 控制寄存器0 (高字节)**



ADC12SHT1x--采样保持定时器1--ADC12MEM8~ADC12MEM15 采样时钟周期 ($n * ADC12CLK$)

ADC12SHT0x--采样保持定时器0--ADC12MEM0~ADC12MEM7 采样时钟周期 ($n * ADC12CLK$)

ADC12SHITx	0	1	2	3	4	5	6	7	8	9	10	11	12~15
n	4	8	16	32	64	96	128	192	256	384	512	768	1024

ADC12_A寄存器 (2/7)

- ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
7	ADC12MSC	多次采样/转换位，只在序列通道或多次转换模式有效。

ADC12_A寄存器 (2/7)

- ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
6	ADC12REF2_5V	内部参考电压的电压值选择位，ADC12REFON也必须置位。 0 ⇒ 选择1.5V内部参考电压； 1 ⇒ 选择2.5V内部参考电压。

ADC12_A寄存器 (2/7)

- **ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
5	ADC12REFON	参考电压控制位。ADC12_A参考电压发生器控制位。对于有REF模块的设备，该位在REF模块的REFMSTR位置为0时唯一有效。对于F54xx设备，REF模块不可用。 0 ⇒ 内部参考电压发生器关闭 1⇒ 内部参考电压发生器打开

ADC12_A寄存器 (2/7)

- **ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
4	ADC12ON	ADC12_A内核控制位。 0 ⇒ 关闭ADC12_A内核 1 ⇒ 打开ADC12_A内核

ADC12_A寄存器 (2/7)

- ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
3	ADC12TOVIE	ADC12_A转换时间溢出中断允许位, GIE置位时有效。
2	ADC12OVIE	ADC12MEMx溢出中断允许位, GIE置位时有效。

ADC12_A寄存器 (2/7)

- ADC12CTL0, ADC12_A 控制寄存器0 (低字节)**

7	6	5	4	3	2	1	0
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC



Bit	Description	
1	ADC12ENC	转换允许位。 0 ⇒ ADC12_A为初始状态，不能启动A/D转换； 1 ⇒ 首次转换由SAMPCON上升沿启动
0	ADC12SC	开始转换

ADC12_A寄存器 (3/7)

- ADC12CTL1, ADC12_A 控制寄存**

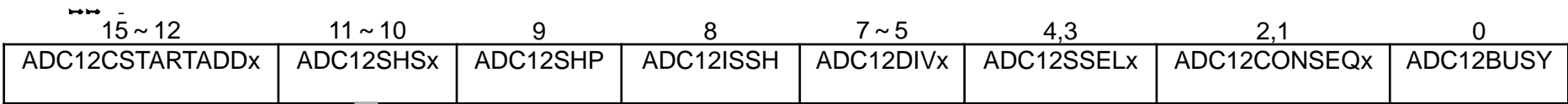
15 ~ 12	11 ~ 10	9	8	7 ~ 5	4,3	2,1	0
ADC12CSTARTADDx	ADC12SHSx	ADC12SHP	ADC12SSH	ADC12DIVx	ADC12SSELx	ADC12CONSEQx	ADC12BUSY



Bit	Description	
15~12	ADC12CSTARTADDx	ADC12_A转换存储器起始地址

ADC12_A寄存器 (3/7)

- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
11~10	ADC12SHSx	采样触发输入源选择位
		0 ⇒ ADC12SC
		1 ⇒ Timer_A.OUT1
		2 ⇒ Timer_B.OUT0
		3 ⇒ Timer_B.OUT1

ADC12_A寄存器 (3/7)

- ADC12CTL1, ADC12_A 控制寄存

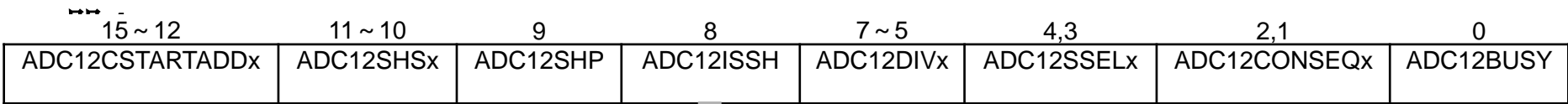
15 ~ 12	11 ~ 10	9	8	7 ~ 5	4,3	2,1	0
ADC12CSTARTADDx	ADC12SHSx	ADC12SHP	ADC12ISSH	ADC12DIVx	ADC12SSELx	ADC12CONSEQx	ADC12BUSY



Bit	Description	
9	ADC12SHP	采样信号（SAMPCON）选择控制位 0 ⇒ SAMPCON源自采样触发输入信号； 1 ⇒ SAMPCON源自采样定时器，由采样输入信号的上升沿触发采样定时器

ADC12_A寄存器 (3/7)

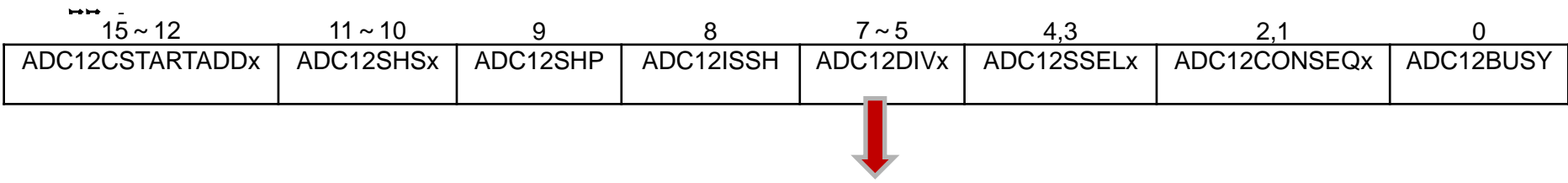
- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
8	ADC12ISSH	采样输入信号方向控制位。 0 ⇒ 采样输入信号为同向输入 1 ⇒ 采样输入信号为反向输入

ADC12_A寄存器 (3/7)

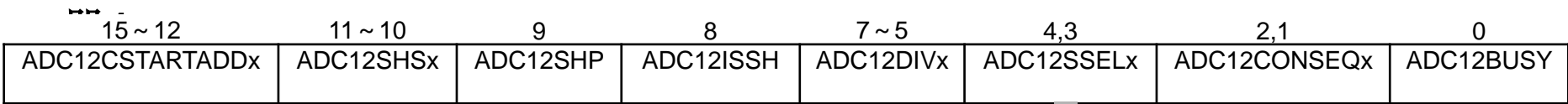
- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
7~5	ADC12DIVx	ADC12_A时钟源分频因子选择位。分频因子为该3位二进制数加1。

ADC12_A寄存器 (3/7)

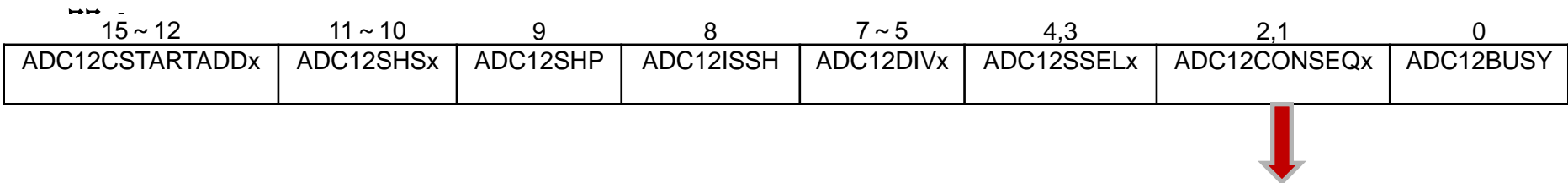
- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
4, 3	ADC12SSELx	ADC12_A内核时钟源选择。 0 ⇒ ADC12_A内部时钟源：ADC12OSC 1 ⇒ ACLK 2 ⇒ MCLK 3 ⇒ SMCLK

ADC12_A寄存器 (3/7)

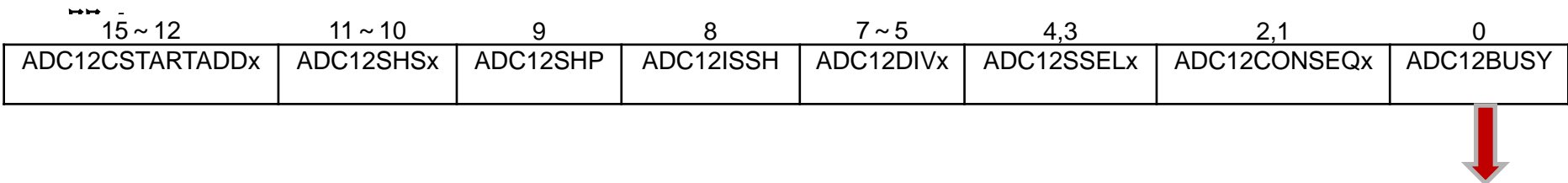
- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
2, 1	ADC12CONSEQx	转换模式选择位 0 ⇒ 单通道单次转换模式 1 ⇒ 序列通道单次转换模式 2 ⇒ 单通道多次转换模式 3 ⇒ 序列通道多次转换模式

ADC12_A寄存器 (3/7)

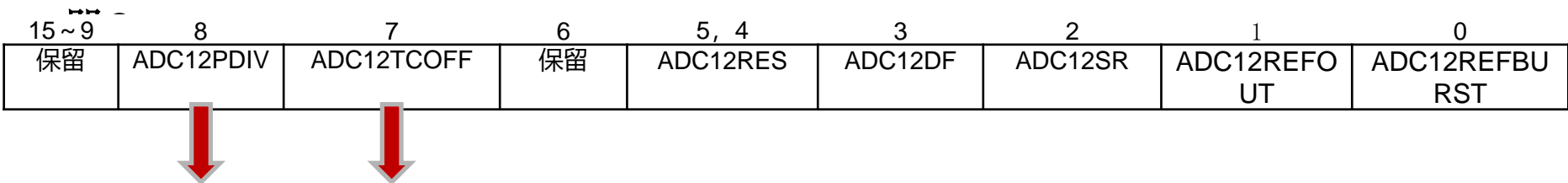
- ADC12CTL1, ADC12_A 控制寄存**



Bit	Description	
0	ADC12BUSY	ADC12_A忙标志位

ADC12_A寄存器 (4/7)

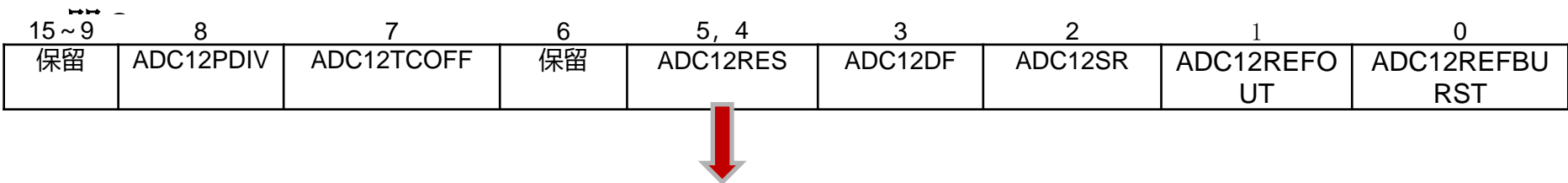
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
8	ADC12PDIV	ADC12_A预分频器。预分频选定的ADC12_A时钟源。 0 ⇒ 预分频/1 1 ⇒ 预分频/4
7	ADC12TCOFF	ADC12_A温度传感器关闭控制位。置位时温度传感器关闭，减小功率。

ADC12_A寄存器 (4/7)

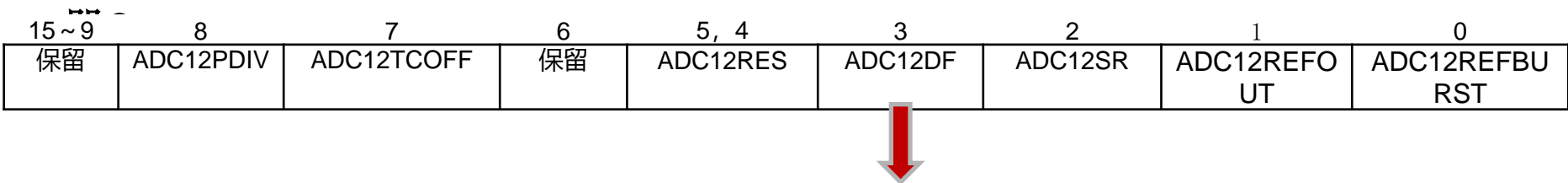
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
5, 4	ADC12RES	ADC12_A分辨率。该位定义转换结果的分辨率。 00 ⇒ 8位分辨率（9个时钟周期转换时间） 01 ⇒ 10位分辨率（11个时钟周期转换时间） 10 ⇒ 12位分辨率（13个时钟周期转换时间） 11 ⇒ 保留

ADC12_A寄存器 (4/7)

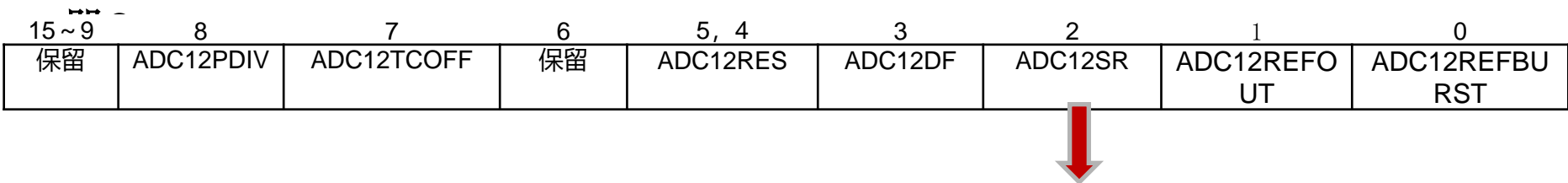
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
3	ADC12DF	ADC12_A数据读回格式。数据总是存储在无符号二进制格式。 0 ⇒ 无符号二进制。模拟输入电压-VREF为0000H，模拟输入电压+VREF为0FFFH。 1 ⇒ 有符号的二进制（2的补码），左对齐。模拟输入电压-VREF为8000h，模拟输入电压+VREF为7FF0h。

ADC12_A寄存器 (4/7)

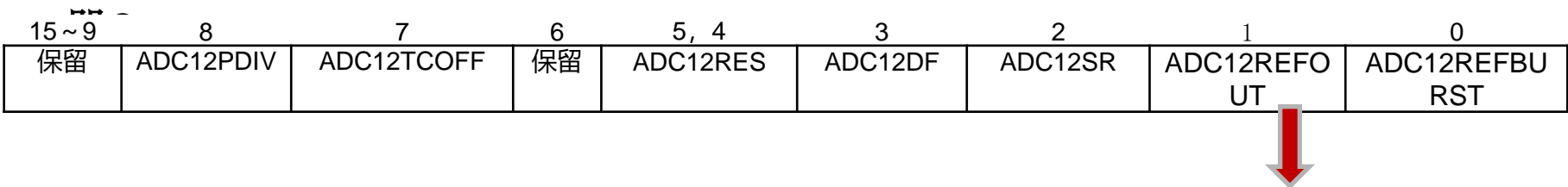
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
2	ADC12SR	ADC12_A采样率。该位选择参考电压缓冲器的最大采样率驱动力。可降低参考电压缓冲器的电流消耗。 0 ⇒ 参考电压缓冲器支持200ksps采样率。 1 ⇒ 参考电压缓冲器支持50ksps采样率。

ADC12_A寄存器 (4/7)

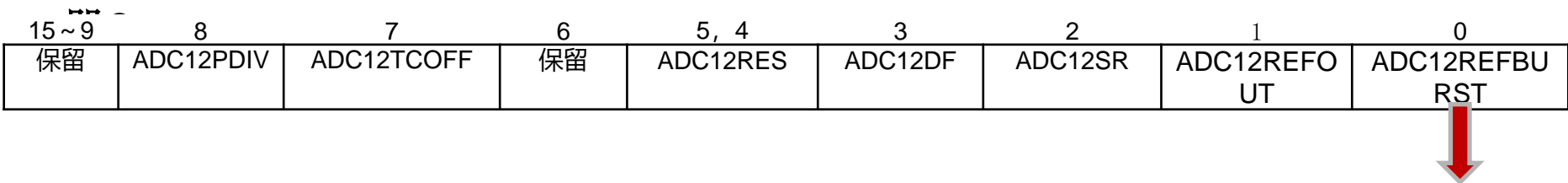
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
1	ADC12REFOUT	参考电压输出 0 ⇒ 参考电压输出关闭 1 ⇒ 参考电压输出打开

ADC12_A寄存器 (4/7)

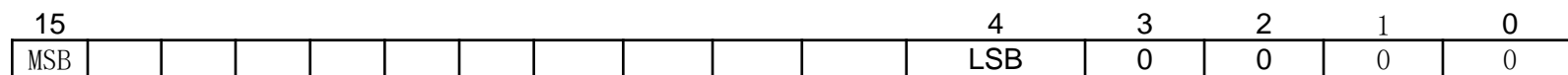
- ADC12CTL2, ADC12_A 控制寄存**



Bit	Description	
0	ADC12REFBURST	参考电压burst。 0 ⇒ 参考电压缓冲器一直打开 1 ⇒ 参考电压缓冲器，仅在采样和转换时打开

ADC12_A寄存器 (5/7)

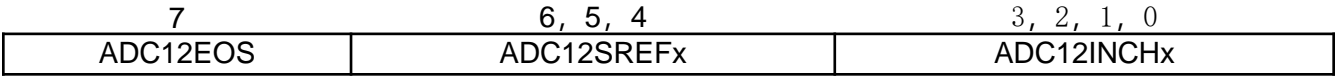
转换存储寄存器: **ADC12MEM0~ADC12MEM15**



- 12位转换结果左对齐，2进制补码格式。
- 15是MSB位，12位模式下3-0位为0，在10位模式下5-0位为0，8位模式下的7-0位是0。ADC12DF=1时使用这种数据格式。数据存储时右对齐格式，在读回时转换为左对齐2补码格式。

ADC12_A寄存器 (6/7)

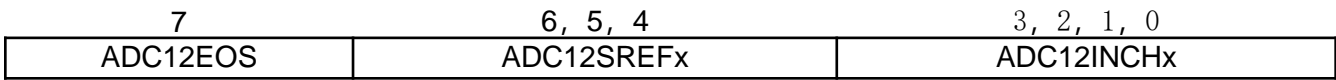
- **ADC12MCTLx** 转换存储器控制寄存器ADC12MCTL0~ADC12MCTL15



Bit	Description	
7	ADC12CSTARTADDx	ADC12_AEOS 序列结束控制位 0 ⇒ 序列没有结束 1 ⇒ 序列结束

ADC12_A寄存器 (6/7)

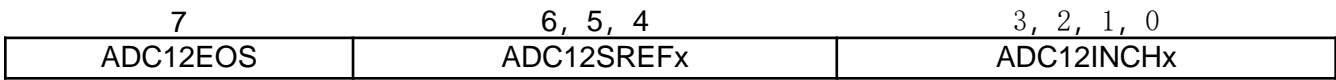
- **ADC12MCTLx** 转换存储器控制寄存器ADC12MCTL0~ADC12MCTL15



Bit	Description	
6, 5, 4	ADC12SREFx	采样触发输入源选择位 000 ⇒ VR+ = AVCC , VR- = AVSS 001 ⇒ VR+ = VREF+ , VR- = AVSS 010 ⇒ VR+ = VeREF+ , VR- = AVSS 011 ⇒ VR+ = VeREF+ , VR- = AVSS 100 ⇒ VR+ = AVCC , VR- = VREF-/ VeREF- 101 ⇒ VR+ = VREF+ , VR- = VREF-/ VeREF- 110 ⇒ VR+ = VeREF+ , VR- = VREF-/ VeREF- 111 ⇒ VR+ = VeREF+ , VR- = VREF-/ VeREF

ADC12_A寄存器 (6/7)

- **ADC12MCTLx** 转换存储器控制寄存器ADC12MCTL0~ADC12MCTL15



Bit	Description
3, 2, 1, 0	ADC12INCHx 选择模拟输入通道。该4位所表示的二进制数为所选的模拟输入通道
	0~7 ⇒ A0~A7;
	8 ⇒ VeREF+;
	9 ⇒ VREF-/VeREF-;
	10 ⇒ 片内温度传感器的输出;
	11 ⇒ (AVcc- AVss) /2;
	12~15 ⇒ A12~A15

ADC12_A寄存器 (7/7)

- **ADC12IE, ADC12_A中断使能寄存器**
 - 该16位寄存器开启($\text{ADC12IE}_x = 1$)或关闭 ($\text{ADC12IE}_x = 0$) ADC12IFG_x 位的中断请求.
- **ADC12IFG, ADC12_A中断标志寄存器**
 - 当把转换结果加载到寄存器 ADC12MEM_x 时, 置位该16位寄存器的相应位, 当软件访问寄存器 ADC12MEM_x 时, 复位该寄存器的相应位.
- **ADC12IV_x 中断向量寄存器**
 - ADC12_A 是一个多源中断: 有18个中断标志 ($\text{ADC12IFG}_0 \sim \text{ADC12IFG}_{15}$ 与 ADC12TOV , ADC12OV), 但只有一个中断向量。所以需要设置这18个标志的优先级顺序, 按照优先级来安排中断标志的响应, 高优先级的请求可以可以中断正在服务的低优先级。

ADC12_A框图

• **ADC12_A**
框图:

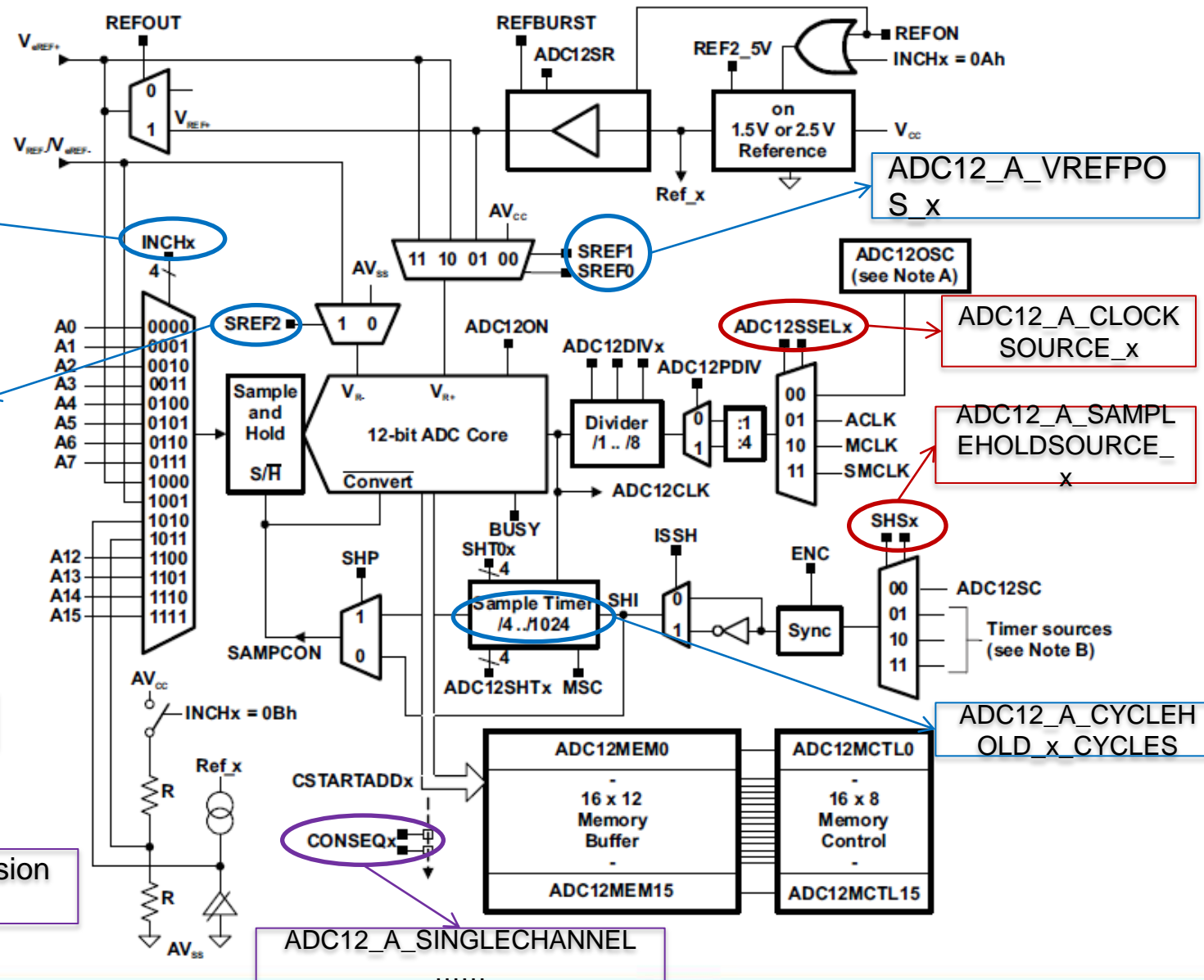
ADC12_A_INPUT_
X

ADC12_A_VREFNE
G_x

ADC12_A_init ()

ADC12_A_startConversion
()

ADC12_A_SINGLECHANNEL



ADC12_A库函数

ADC12_A_init(__MSP430_BASEADDRESS_ADC12_A_PLUS__,

.....

采样保持和转换信号

ADC12_A_SAMPLEHOLDSOURCE_SC [Default]	⇒	ADC12SC
ADC12_A_SAMPLEHOLDSOURCE_1	⇒	TimerA.OUT1
ADC12_A_SAMPLEHOLDSOURCE_2	⇒	TimerB.OUT0
ADC12_A_SAMPLEHOLDSOURCE_3	⇒	TimerB.OUT1

时钟源

ADC12_A_CLOCKSOURCE_ADC12_AOSC [Default]	⇒	MODOSC, 5 MHz晶振
ADC12_A_CLOCKSOURCE_ACLK	⇒	辅时钟
ADC12_A_CLOCKSOURCE_MCLK	⇒	主时钟
ADC12_A_CLOCKSOURCE_SMCLK	⇒	子时钟

时钟分频

ADC12_A_CLOCKDIVIDER_1 [Default]	⇒	1倍分频
ADC12_A_CLOCKDIVIDER_2	⇒	2倍分频
.....{3,4,5,6,7,8,12,16,20,24,28}.....		
ADC12_A_CLOCKDIVIDER_1	⇒	32倍分频

ADC12_A库函数

ADC12_A_enable(__MSP430_BASEADDRESS_ADC12_A_PLUS__); //使能ADC12_A

ADC12_A_setupSamplingTimer(__MSP430_BASEADDRESS_ADC12_A_PLUS__,

.....

采样保持时钟周期 (ADC12_AMEM0—ADC12_AMEM7)

ADC12_A_CYCLEHOLD_4_CYCLES [Default] ⇒ 4倍时钟周期

ADC12_A_CYCLEHOLD_8_CYCLES ⇒ 8倍时钟周期

.....{16,32,64,96,128,192,256,384,512,768}.....

ADC12_A_CYCLEHOLD_1024_CYCLES ⇒ 1024倍时钟周期

采样保持时钟周期 (ADC12_AMEM8—ADC12_AMEM15) (同上)

.....

时钟源

ADC12_A_MULTIPLESAMPLESDISABLE ⇒ 不允许多次采样

ADC12_A_MULTIPLESAMPLESENABLE ⇒ 允许多次采样

ADC12_A库函数

ADC12_A_startConversion(__MSP430_BASEADDRESS_ADC12_A_PLUS_
—,

.....

存储寄存器

ADC12_A_MEMORY_0 [Default]

⇒ ADC12MEM0

.....

采样转换模式

ADC12_A_SINGLECHANNEL [Default]

⇒ 单通道单次转换模式

ADC12_A_SEQOFCHANNELS

⇒ 序列通道单次转换模式

ADC12_A_REPEATED_SINGLECHANNEL

⇒ 单通道多次转换模式

ADC12_A_REPEATED_SEQOFCHANNELS

⇒ 序列通道多次转换模

式

ADC12_A_enableInterrupt(__MSP430_BASEADDRESS_ADC12_A_PLUS__,

.....

存储寄存器

ADC12_A_MEMORY_0 [Default]

⇒ ADC12MEM0

.....

课堂任务+作业

- 课上实验：
指导书中AD实验内容，完成下列功能。
要求（1）选择单通道循环采样转换（可称为单通道重复转换模式或单通道多次转换模式）；
（2）选择通道5，连接拨码电位器，控制LED灯。
（3）实现拨动拨码电位器，LED1~LED6顺序点亮的现象。
- 作业：
完成思考题：
（1）如何计算拨盘电位器分压和程序中实际转换的数据的关系？
（2）用ADC12MEM0寄存器，是否可用其他1-15寄存器，如果可以，该如何设置？请更改为使用ADC12MEM1并且实验验证。

谢谢