Application Notes
ZHCAA08 – 5.2020

基于 C2000 Piccolo 系列的脉冲计数方法

JOHNSON CHEN/ EP FAE

摘要

该文档详细描述一种脉冲计数方法。这个方法基于 C2000 Piccolo 系列，使用 CPU Timer2 来实现脉冲计数功能。

脉冲计数功能广泛应用于伺服驱动器，步进驱动器等产品， CNC 或者 PLC 通过发脉冲数给伺服驱动器/步进驱动器发送命令信息(转动位置，速度等)。

这篇文档将阐述此功能的具体实现原理，包含软件代码及相关注意事项。

通过此方法客户可以在低成本的 C2000 产品如 TMS320F28027 上实现脉冲计数功能。

Contents

Figures

1 介绍

对于 C2000 产品来说，通常实现脉冲计数的方法有下面两种：

1. 使用 eQEP 模块实现脉冲计数功能。
2. 使用 CLB 编程来实现脉冲计数功能，较新的 C2000 产品有 CLB 模块。

在使用 C2000 来实现脉冲计数功能的时候，我们可能会面临以下挑战，从而不知道如何实现脉冲计数功能。

- 因为低成本要求，想使用低端的 F2802x 等系列不含 eQEP 和 CLB 模块的芯片。
- 系统要使用 eQEP 来实现增量编码器解码，但使用的 C2000 产品只有一个 QEP 比如 F2803x/F2805x，也没有 CLB 模块。

本文主要介绍除了 eQEP 模块和 CLB 以外，如何使用 CPU Timer2 来实现脉冲计数的方法 。

2 实现原理

2.1 原理分析

下图为 F2802x 的时钟选项框图，

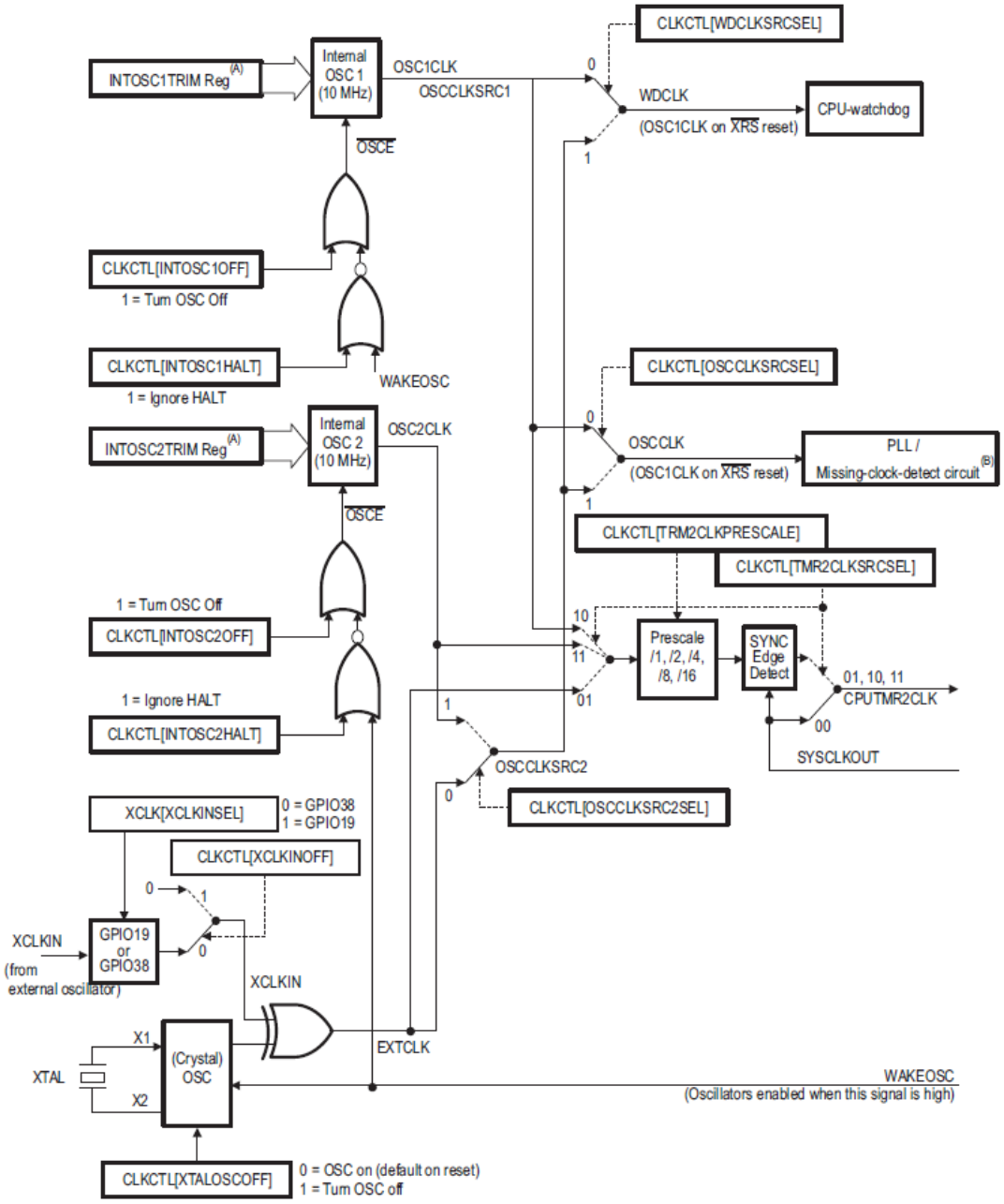

Figure 1. 时钟选项

从上图我们可以看到对于外部时钟来说，有 XTAL 和 XCLKIN。

通常 XTAL 为无源晶振，接到芯片的 X1 和 X2 管脚， 而 XCLKIN 是作为有源时钟输入。

对于 C2000 来说有 3 个 32 位的定时器 CPU Timer0, CPU Time1 和 CPU Timer2，通常我们是把它们当做定时器来使用，而在 CPU Timer 手册里也并未提到除了 SYSCLK 外可以用其他时钟源。

但通过分析上面时钟框图并结合 TMS320F2802x Technical Reference Manual 里面 CLKCTL[TMR2CLKSRCSEL]的描述, 我们可以发现 CPU Timer0 和 CPU Time1 的时钟源只能是 SYSCLK，但 CPUTIMER2 的时钟源有四种选择：

00：SYSCLK(不支持分频） 01: 外部时钟（EXTCLK）10: OSC1 11: OSC2

EXTCLK 可以是 XTAL 或者 XCLKIN。

因此 XCLKIN 可以作为 CPU Timer2 的时钟源。

2.2 实现方法

通过时钟框图的分析，我们知道 XCLKIN 可以作为 CPU Timer2 的时钟源，那么我们可以将上位机（如 PLC 或者 CNC）的脉冲输入连接到 XCLKIN 管脚，然后配置 CPU Timer2 选择 EXTCLK 为时钟源，这样 CPU Timer2 能实现了对输入信号的脉冲计数功能。而 CPU Timer2 的计数值（TIMH:TIM）便代表实际的输入脉冲个数。

这时候我们需要注意到因为 EXTCLK 是 XCLKIN 和 XTAL 异或的输出，因此 XTAL 功能将不能使用。下图红色线为 XCLKIN 连接到 CPU Timer2 的路径。

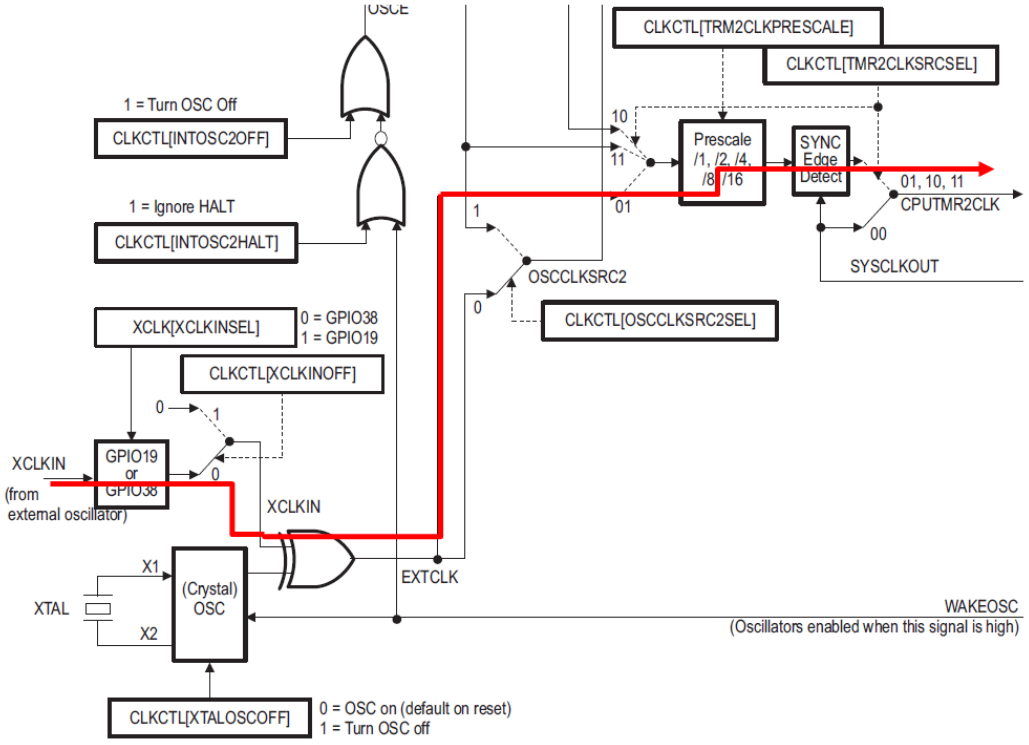

Figure 2. XCLKIN 到 CPU Timer2 路径

具体实现：

假设脉冲信号输入接到 GPIO19，每个脉冲计数一次，CPU 时钟选择 INTOSC1。

配置如下：

XCLK[XCLKINSEL] =1，即选择 GPIO19,

CLKCTL[XCLKINOFF] = 0,即使能外部时钟

CLKCTL[XTALOSCOFF] = 1,即禁止芯片内部振荡器

CLKCTL[TMR2CLKSRCSEL] = 01,CPUTIMER2 时钟源选择 EXTCLK 即 XCLKIN

CLKCTL[TMR2CLKPRESCALE] = 0,即不分频

2.3 软件实现

下面为具软件代码：

```
void IntOsc1Sel(void)
{
 EALLOW;
       SysCtrlRegs.CLKCTL.bit.INTOSC1OFF  = 0;    //Turn on INTOSC1
 SysCtrlRegs.CLKCTL.bit.OSCCLKSRCSEL=0;           // Clk Src = INTOSC1
 SysCtrlRegs.XCLK.bit.XCLKINSEL = 1;        //设置GPIO19为XCLK输入
 SysCtrlRegs.CLKCTL.bit.XCLKINOFF=0;        // Turn on XCLKIN
 SysCtrlRegs.CLKCTL.bit.TMR2CLKPRESCALE=0; //XCLKIN Pass
 SysCtrlRegs.CLKCTL.bit.TMR2CLKSRCSEL=1;//CPUTIMER2 CLOCK=External Clock
 SysCtrlRegs.CLKCTL.bit.XTALOSCOFF=1;       // Turn off XTALOSC
       SysCtrlRegs.CLKCTL.bit.INTOSC2OFF=1;            // Turn off INTOSC2

 EDIS;
}


void InitCpuTimer2(void)
{
      // CPU Timer 2
 // Initialize timer period to maximum:
 CpuTimer2Regs.PRD.all  = 0xFFFFFFFF;
 CpuTimer2Regs.TPR.all  = 0;
 CpuTimer2Regs.TPRH.all = 0;
 CpuTimer2Regs.TCR.bit.TSS = 1; // CPUTimer Stop Status bit1=stop ;
 // Reload all counter register with period value:
 CpuTimer2Regs.TCR.bit.TRB = 1; //CPU-Timer Reload bit;
 // Reset interrupt counters:
 CpuTimer2.InterruptCount = 0;
}
```

```
  Uint32  Pulse_Number_New;          //即为接收到的脉冲个数

  void main(void)
  {
      // Initialize System Control:
InitSysCtrl();    //里面需要调用 IntOsc1Sel()函数

      // Initalize GPIO:

      // Disable CPU interrupts
DINT;

      //Copy time critical code and Flash setup code from Flash to RAM
MemCopy(&RamfuncsLoadStart, &RamfuncsLoadEnd, &RamfuncsRunStart);

// Call Flash Initialization to setup flash waitstates

InitFlash();

      // Initialize the PIE control registers
InitPieCtrl();

      // Disable CPU interrupts and clear all CPU interrupt flags:
IER = 0x0000;
IFR = 0x0000;

      // Initialize the PIE vector table
InitPieVectTable();

      // initialize the Cpu Timer2
InitCpuTimer2 ();

      // Start Cpu Timer2
      CpuTimer2Regs.TCR.all = 0x4001;


EINT;   // Enable Global interrupt INTM
ERTM;   // Enable Global realtime interrupt DBGM

while (1)
      {

          Pulse_Number_New = 0xffffffff - CpuTimer2Regs.TIM.all;

      }

}
```

3 注意事项

此实现方法适用于所有有内部时钟源（INTOCS1）的 C2000 Piccolo 产品。

使用此实现方法软件上注意事项如下：

1. 在时钟初始化时，调用上面的 IntOsc1Sel()函数。
2. 请确保 CPU Timer2 时钟已使能。

```
SysCtrlRegs.PCLKCR3.bit.CPUTIMER2ENCLK = 1;   //Enable CPUTimer2
```

3. 在 main 函数初始化时调用上面的 InitCpuTimer2()函数
4. CPU Timer2 的计数器是递减计数的。

此实现方法的限制条件：

- 仅限 C2000 Piccolo 有内部时钟源 INTOSC1 的产品。
- CPU 时钟(OSCCLK)只能使用内部时钟 INTOSC1，不可使用外部时钟 XTAL 或者 XCLKIN。
- 仅限 CPU Timer2 能实现此功能，CPU Timer0 和 CPU Timer1 无法实现此功能。

4 参考文献

1. TMS320F28027 Datasheet (SPRS523M)
2. TMS320F28027 Technical Reference Manual (SPRUI09)

重要声明和免责声明

TI 均以“原样”提供技术性及可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用TI 产品进行设计使用。您将对以下行为独自承担全部责任：(1) 针对您的应用选择合适的TI 产品；(2) 设计、验证并测试您的应用；(3) 确保您的应用满足相应标准以及任何其他安全、安保或其他要求。所述资源如有变更，恕不另行通知。TI 对您使用所述资源的授权仅限于开发资源所涉及TI 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它TI或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，TI对此概不负责，并且您须赔偿由此对TI 及其代表造成的损害。

TI 所提供产品均受TI 的销售条款 (http://www.ti.com.cn/zh-cn/legal/termsofsale.html) 以及ti.com.cn上或随附TI产品提供的其他可适用条款的约束。TI提供所述资源并不扩展或以其他方式更改TI 针对TI 产品所发布的可适用的担保范围或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122