第8章 通用输入/输出多路复用器GPIO

本章内容

8.1 GPIO的寄存器

8.2 用GPIO引脚控制LED闪烁程序的编写

本章重点

1、了解GPIO各个寄存器；

2、掌握I/O口的复用功能；

3、掌握如何正确编写一个LED闪烁程序；

通用输入/输出多路复用器GPIO介绍

F28035芯片提供了45个多功能引脚，为了节省资源，这些引脚是复用的，它们既可以作为通用的I/O接口也可以复用为其他功能来进行操作。下面将详细介绍由些引脚所组成的输入/输出多路复用器GPIO的工作原理及相关寄存器详细介绍。

8.1 GPIO的寄存器

引脚复用：指引脚既可以作为DSP片内外设，又可以作为通用I/O接口来使用。引脚是否复用取决于相关寄存器。

GPIO0-GPIO31引脚复用原理图如图1所示，或见教材图8-1所示。

GPIO32-GPIO33 引脚复用原理图如图2所示，或见教材图8-2所示。

JTAG多功能复用原理图如图3所示，或见教材图8-3所示。

模拟量输入I/O多功能复用原理图如图4所示，或见教材图8-4所示。

8.1 GPIO的寄存器

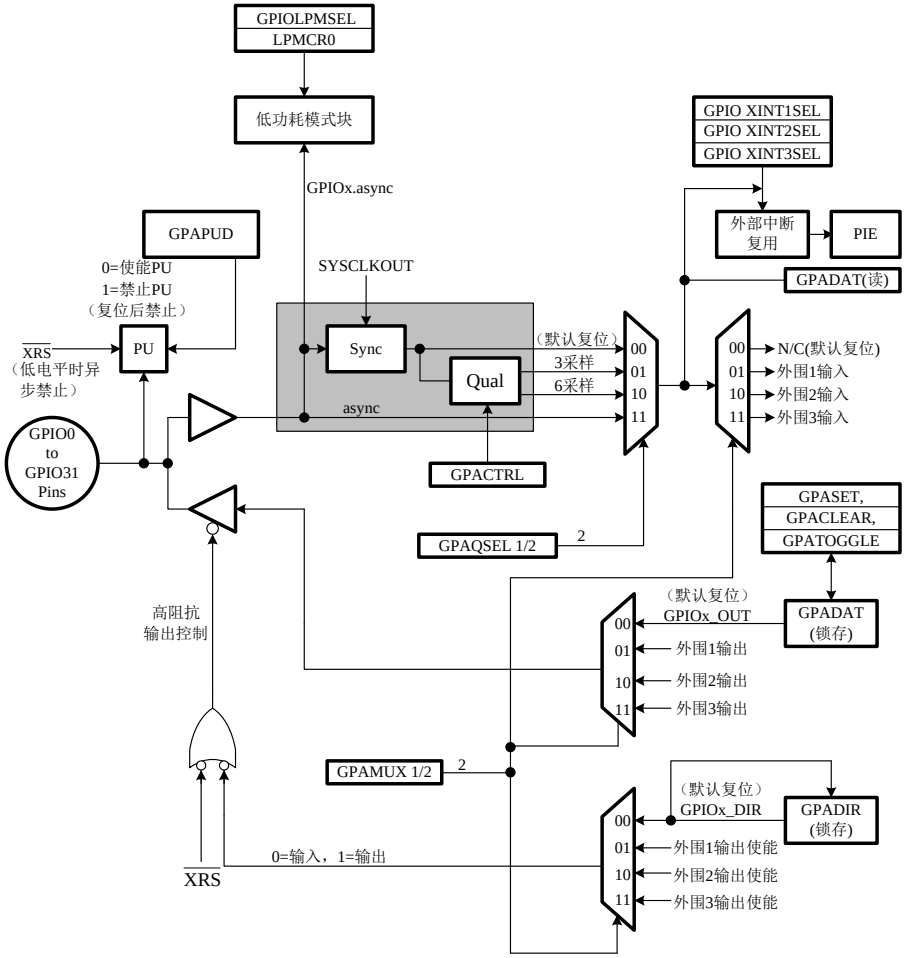

图1 GPIO0-GPIO31引脚复用原理图

8.1 GPIO的寄存器

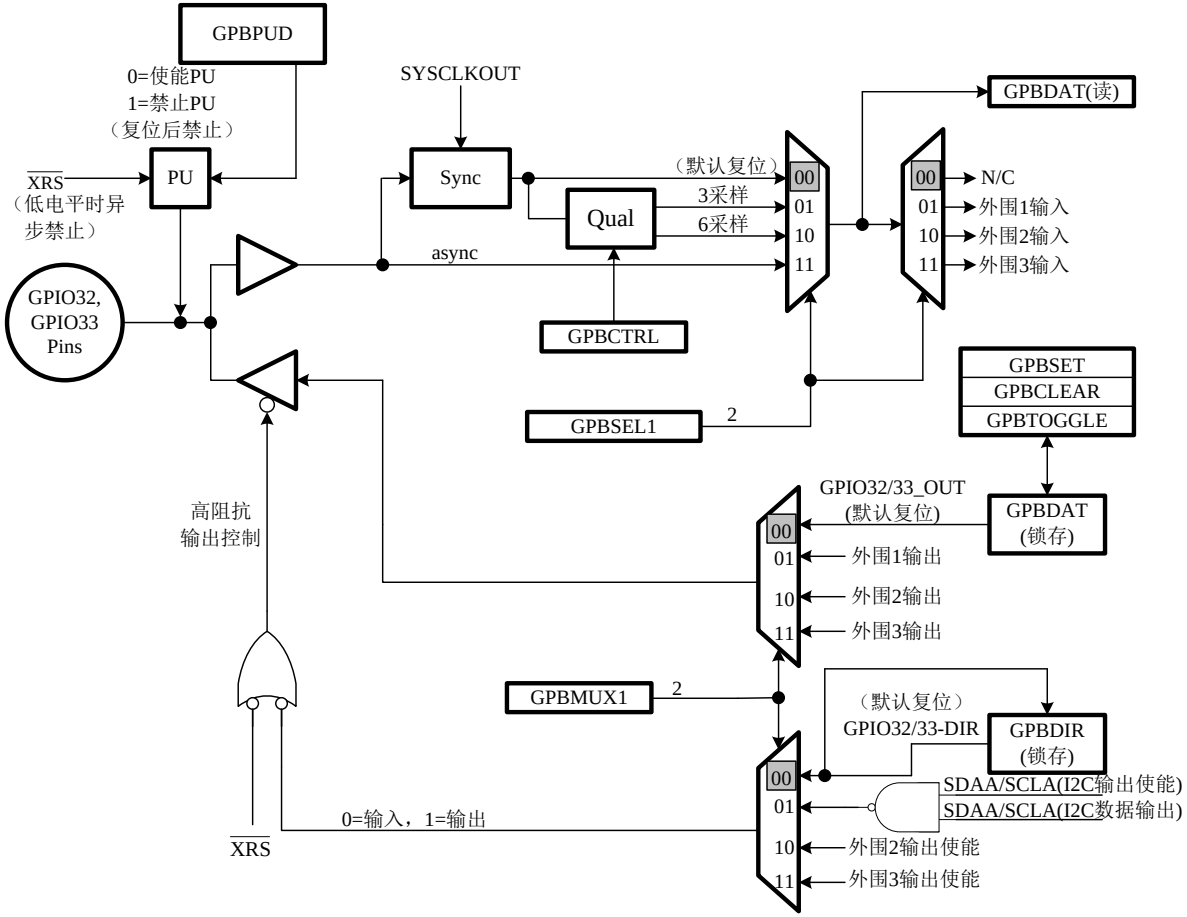

图2 GPIO32-GPIO33 引脚复用原理图

8.1 GPIO的寄存器

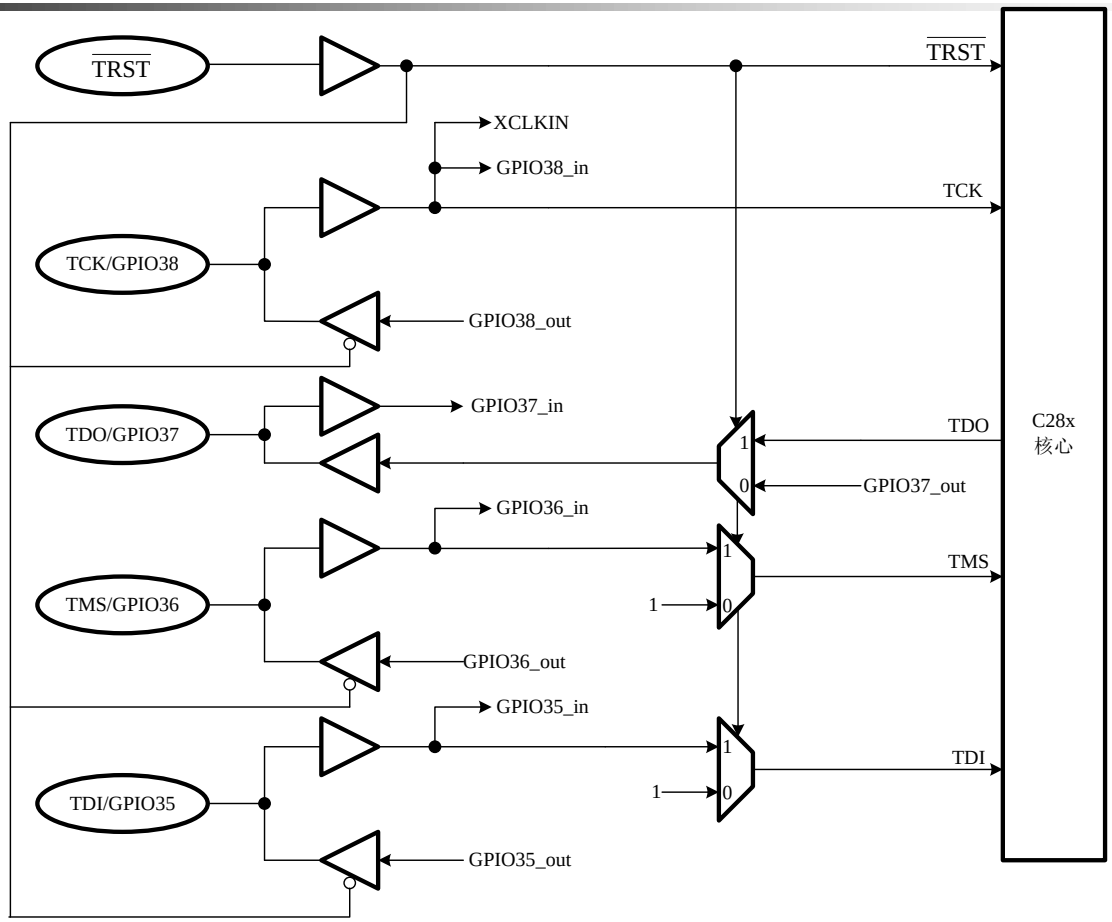

图3 JTAG多功能复用原理图

8.1 GPIO的寄存器

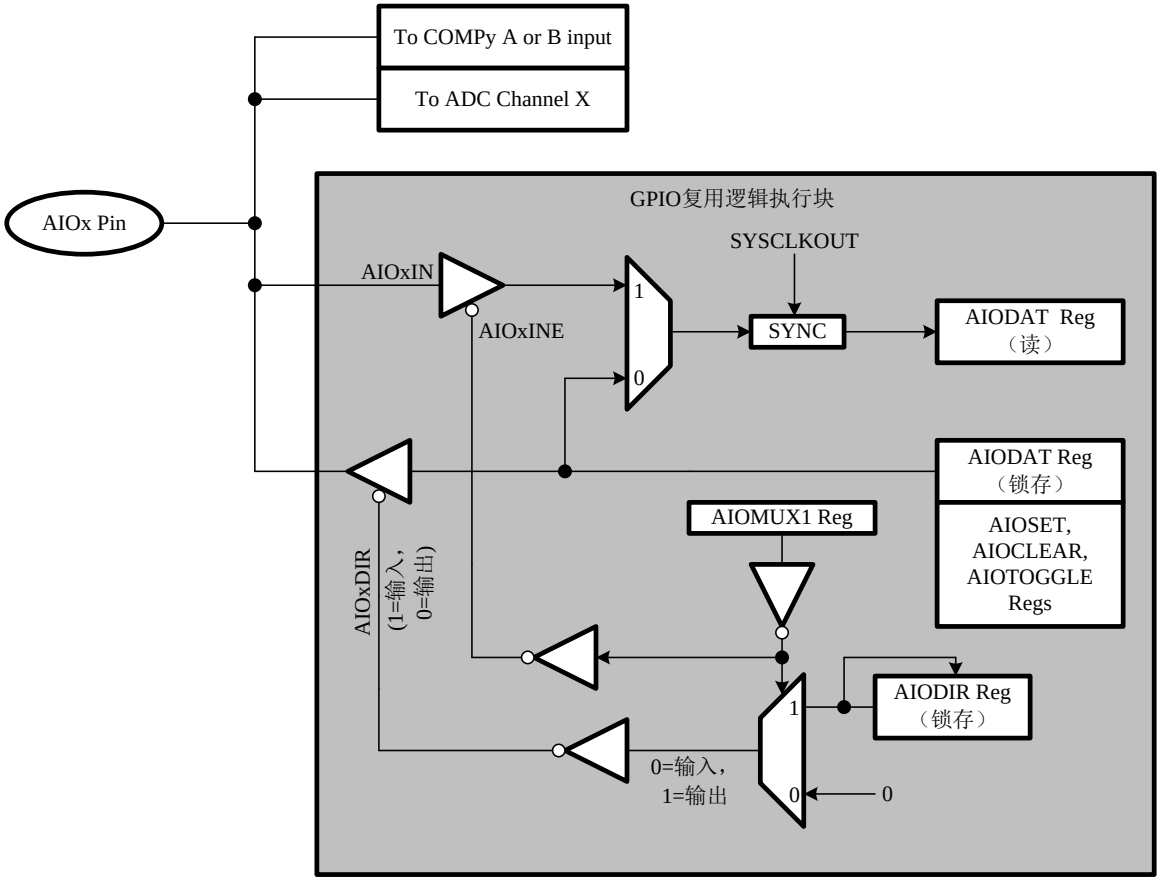

图4 模拟量输入I/O多功能复用原理图

8.1 GPIO的寄存器

- GPIO寄存器的分类
  - 控制类寄存器
    - GPIO功能选择控制寄存器
    - GPIO方向控制寄存器
    - GPIO内部上拉寄存器
    - GPIO输入限定控制寄存器
  - 数据类寄存器
    - GPIO数据寄存器
    - GPIO置位寄存器
    - GPIO复位寄存器
    - GPIO电平翻转寄存器

8.1.1 GPIO功能选择控制寄存器

GPIO控制寄存器的说明如表1所示，或见教材表8-1所示。

表1 GPIO控制寄存器

名称	地址	大小（×16）	寄存器说明
GPACTRL	0x6F80	2	GPIOA控制寄存器（GPIO0-GPIO31）
GPAQSEL1	0x6F82	2	GPIOA限定选择寄存器1（GPIO0-GPIO15）
GPAQSEL2	0x6F84	2	GPIOA限定选择寄存器2（GPIO16-GPIO31）
GPAMUX1	0x6F86	2	GPIOA功能选择寄存器1（GPIO0-GPIO15）
GPAMUX2	0x6F88	2	GPIOA功能选择寄存器2（GPIO16-GPIO31）
GPADIR	0x6F8A	2	GPIOA方向寄存器（GPIO0-GPIO31）
GPAPUD	0x6F8C	2	GPIOA上拉关闭寄存器（GPIO0-GPIO）
GPBCTRL	0x6F90	2	GPIOB控制寄存器（GPIO32-GPIO44）
GPBQSEL1	0x6F92	2	GPIOB限定选择寄存器1（GPIO32-GPIO44）
GPBMUX1	0x6F96	2	GPIOB功能选择寄存器1（GPIO32-GPIO44）
GPBDIR	0x6F9A	2	GPIOB方向寄存器（GPIO32-GPIO44）
GPBPUD	0x6F9C	2	GPIOB上拉关闭寄存器（GPIO32-GPIO44）
AIOMUX1	0x6FB6	2	模拟I/O功能选择寄存器1（AIO0-AIO15）
AIODIR	0x6FBA	2	模拟I/O方向寄存器（AIO0-AIO15）

8.1.1 GPIO功能选择控制寄存器

GPIOA功能选择寄存器1 （GPAMUX1 ）各位的情况以及各位的说明见教材图8-5和表8-2所示。

下面通过程序来解释如何配置GPAMUX1的选择功能。

```
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;//GPIO0作为普通IO
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1;//GPIO0作为复用IO，功能为输出pwm波
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 2;//GPIO0保留，即无功能
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 3;//GPIO0保留，即无功能
```

8.1.1 GPIO功能选择控制寄存器

GPIOA功能选择寄存器2（GPAMUX2）各位的情况及说明见教材图8-6和表8-3所示。

下面通过程序来解释如何配置GPAMUX2的选择功能。

```
GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 0;//GPIO16作为普通IO
GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 1;//GPIO16作为复用IO，功能为SPI，从站输入，主站输出
GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 2;//GPIO16即无功能
GpioCtrlRegs.GPAMUX2.bit.GPIO16 = 3;//GPIO16作为复用IO，功能为错误区域控制
```

8.1.1 GPIO功能选择控制寄存器

GPIOB功能选择寄存器1的各位情况及各位的说明见教材图8-7和表8-4所示。

下面通过程序来解释如何配置GPBMUX1的选择功能。

```
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 0;//GPIO32作为普通IO
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 1;//GPIO32作为复用IO,功能为I2C
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 2;//GPIO32作为复用IO,功能为ePWM同步脉冲输入
GpioCtrlRegs.GPBMUX1.bit.GPIO32 = 3;//GPIO32作为复用IO,功能为ADC转换开始
```

8.1.2 GPIO方向控制寄存器

GPIOA方向寄存器 （GPADIR ）的各位情况及各位的说明见叫教材图8-8所示和表8-5所示。

下面通过程序来解释如何配置GPIOA方向寄存器。

```
GpioCtrlRegs.GPADIR.bit.GPIO0=0;  //GPIO0方向为输入
GpioCtrlRegs.GPADIR.bit.GPIO0=1;  //GPIO0方向为输出
```

8.1.2 GPIO方向控制寄存器

GPIOB方向寄存器 （GPBDIR ）的各位情况及各位的说明见教材图8-9所示和表8-6所示。

下面通过程序来解释如何配置GPIOB方向寄存器。

```
GpioCtrlRegs.GPBDIR.bit.GPI32=0;  //GPIO32方向为输入
GpioCtrlRegs.GPBDIR.bit.GPI32=1   //GPIO32方向为输出
```

8.1.3 GPIO内部上拉寄存器

GPIOA上拉关闭寄存器（GPAPUD ）的各位情况及各位的说明见图8-10和表8-7所示。

下面通过程序来解释如何配置GPIOA上拉关闭寄存器。

```
GpioCtrlRegs.GPAPUD.bit.GPIO0 = 0; //开启GPIO0内部上拉
GpioCtrlRegs.GPAPUD.bit.GPIO0 = 1; //关闭GPIO0内部上拉
```

8.1.3 GPIO内部上拉寄存器

GPIOB上拉关闭寄存器（GPBPUD ）的各位情况及各位的说明见教材图8-11和表8-8所示。

下面通过程序来解释如何配置GPIOB上拉关闭寄存器。

```
GpioCtrlRegs.GPBPUD.bit.GPIO32 = 0;//开启GPIO32内部上拉
GpioCtrlRegs.GPBPUD.bit.GPIO32 = 1;//关闭GPIO32内部上拉
```

8.1.4 GPIO输入限定控制寄存器

GPIOA限定控制（GPACTRL）寄存器的各位情况及各位的说明见教材图8-12和表8-9所示。

GPIOB限定控制（GPBCTRL）寄存器的各位情况及各位的说明见教材图8-13和表8-10所示。

GPIOA限定选择1（GPAQSEL1）寄存器的各位情况及各位的说明见教材图8-14和表8-11所示。

GPIOA限定选择2（GPAQSEL2）寄存器的各位情况及各位的说明见教材图8-15和表8-12所示。

8.1.4 GPIO输入限定控制寄存器

GPIOB限定选择1（GPBQSEL1）寄存器的各位情况及各位的说明见教材图8-16和表8-13所示。

输入限定功能介绍：将GPIO引脚设置为通用的I/O口，当输入相应的信号时，在实际中输入信号往往会受到干扰，如果不采取任何的措施，DSP将作出错误的决策，所以，在其内部添加了输入限定功能。

8.1.4 GPIO输入限定控制寄存器

输入限定的几种情况介绍：

（1）与系统时钟SYSCLKOUT同步

所有引脚在复位时默认采用此种限定方式。在这种方式下，输入信号仅被限定到与系统时钟SYSCLKOUT同步，因为输入到引脚的信号是异步的，为了使输入到DSP内部与SYSCLKOUT同步的过程中，会产生一个SYSCLKOUT的延迟。

8.1.4 GPIO输入限定控制寄存器

（2）异步输入

工作在外设I/O口模式下的引脚不需要同步输入信号或其自身就具有信号同步的功能。如果引脚被用作通用的数字输入引脚，异步输入的选择就无效。如果该引脚配置为通用的输入引脚却又选择了异步输入，其输入限制将默认为与系统时钟SYSCLKOUT同步。

（3）通过采样窗限定

在该模式下，外部引脚的输入信号首先与系统时钟SYSCLKOUT同步，然后在输入信号允许改变之前被指定数量周期所限定。

8.1.4 GPIO输入限定控制寄存器

采样窗介绍：如图5和图6表示了如何去除噪声信号的，其中需压迫介绍采样周期和采样数量的概念。

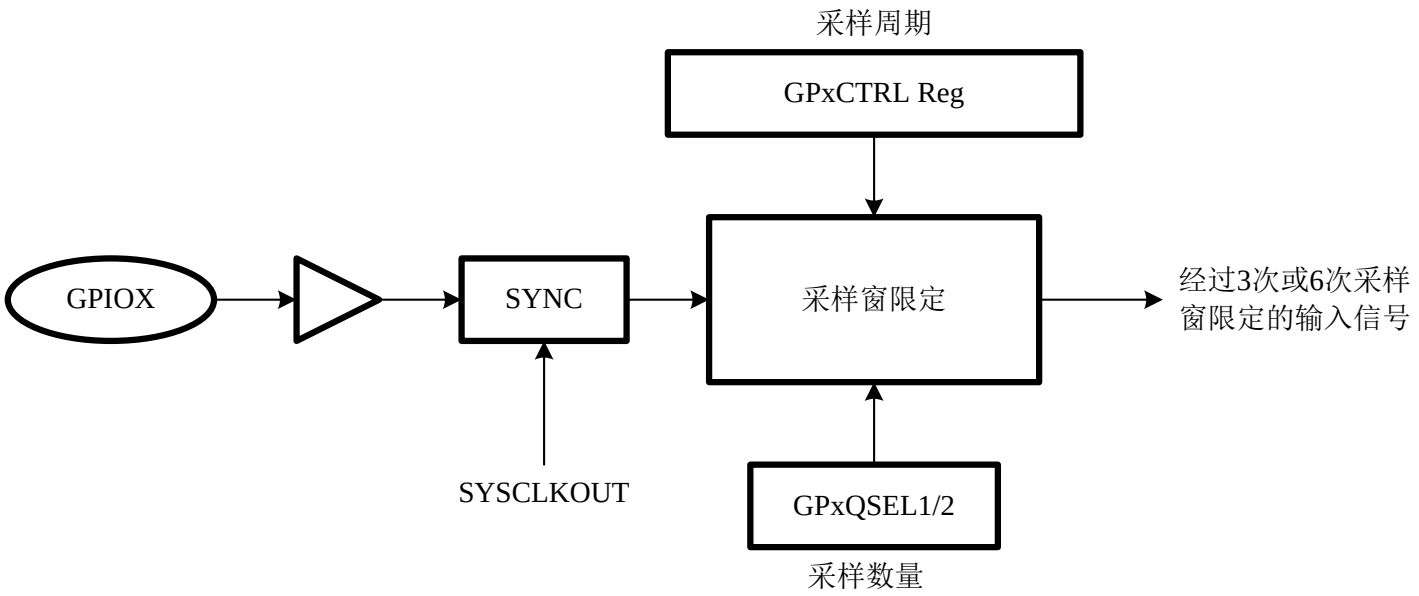

图5 采样窗输入限定

8.1.4 GPIO输入限定控制寄存器

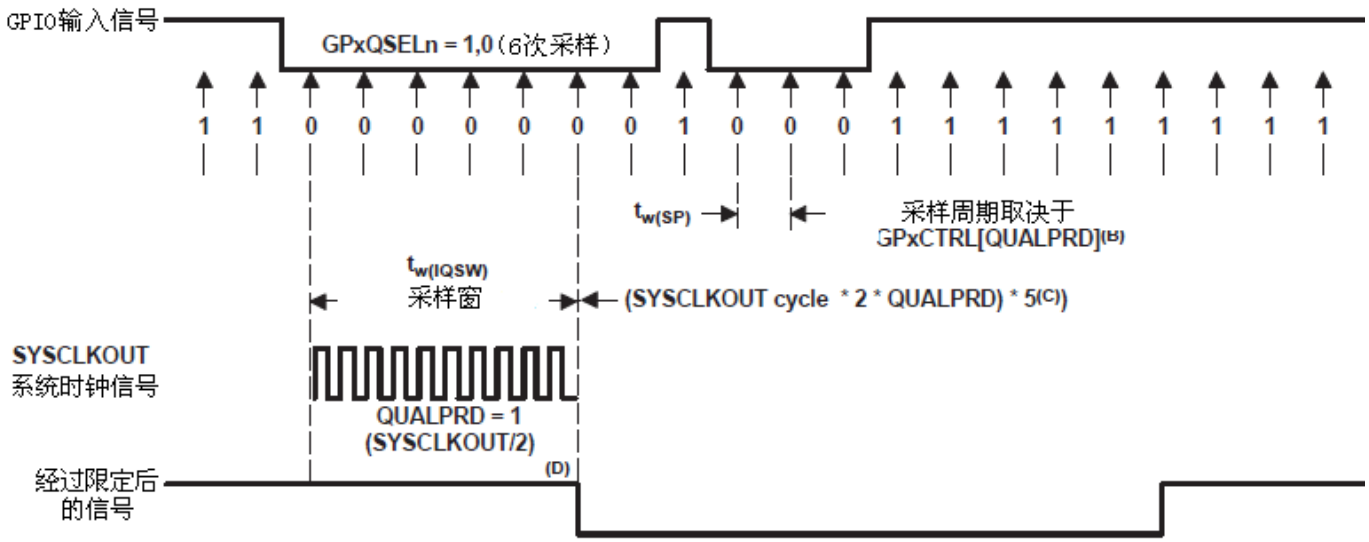

图6 输入限定模式下的时钟周期

8.1.4 GPIO输入限定控制寄存器

（1）采样周期

为了对输入信号进行限定，需要间隔一定的周期对输入信号进行采样。采样周期由用户设定，并以CPU系统时钟SYSCLKOUT为基本单位。采样周期由寄存器GPxCTRL中的QUALPRDn决定，并且采样周期在配置过程中是以8个输入信号为一组。教材中的表8-13和表8-14给出了采样周期和采样频率与GPxCTRL[QUALPRDn]之间的关系。

8.1.4 GPIO输入限定控制寄存器

（2）采样数量

在限定选择寄存器（GPAQSEL1、GPAQSEL2、GPBQSEL1和GPBQSEL2）可以指定信号的采样次数（3次和6次），当输入信号在3次和6次采样内保持不变，信号可通过采样窗送入到DSP内部。教材中表8-15和表8-16列出了如何根据GPxCTRL[QUALPRDn]和采样次数来计算采样窗的宽度。

8.1.5 GPIO数据寄存器

GPIO数据寄存器的说明见教材表8-17所示。

GPIOA数据寄存器各位的情况及各位的说明见教材图8-19和表8-18所示。

下面通过程序来解释如何配置GPIOA数据寄存器。

```
GpioDataRegs.GPADAT.bit.GPIO0 = 0;  //GPIO0输出低电平
GpioDataRegs.GPADAT.bit.GPIO0 = 1;  //GPIO0输出高电平
```

8.1.5 GPIO数据寄存器

GPIOB数据寄存器各位的情况及各位的说明见教材图8-20和表8-19所示。

下面通过程序来解释如何配置GPIOB数据寄存器。

```
GpioDataRegs.GPBDAT.bit.GPIO32 = 0; //GPIO32输出低电平
GpioDataRegs.GPBDAT.bit.GPIO32 = 1; //GPIO32输出高电平
```

8.1.5 GPIO数据寄存器

GPIOA置位寄存器、GPIOA复位寄存器、GPIOA电平翻转寄存器各位情况的描述及各位说明见教材图8-21和表8-20、表8-21、表8-22所示。

下面通过程序来解释如何配置GPIOA置位寄存器、GPIOA复位寄存器、GPIOA电平翻转寄存器。

```
GpioDataRegs.GPASET.bit.GPIO0 = 1;//Set置位则输出高电平，即置位
GpioDataRegs.GPACLEAR.bit.GPIO0=1;//CLEAR置位则输出低电平，即复位
GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;//GPIO0端口电平翻转一次
```

8.1.5 GPIO数据寄存器

GPIOB置位寄存器、GPIOB复位寄存器、GPIOB电平翻转寄存器各位情况的描述及各位说明见教材图8-22和表8-23、表8-24、表8-25所示。

下面通过程序来解释如何配置GPIOB置位寄存器、GPIOB复位寄存器、GPIOB电平翻转寄存器。

```
GpioDataRegs.GPBSET.bit.GPIO32= 1;//Set置位则输出高电平，即置位
GpioDataRegs.GPBCLEAR.bit.GPIO32=1; //CLEAR置位则输出低电平，即复位
GpioDataRegs.GPBTOGGLE.bit.GPIO32 = 1;//GPIO32端口电平翻转一次
```

8.2 用GPIO引脚控制LED闪烁程序的编写

要求：实现开发板上LED D3灯的闪烁。

（1）添加文件，所需添加的文件如图7所示。

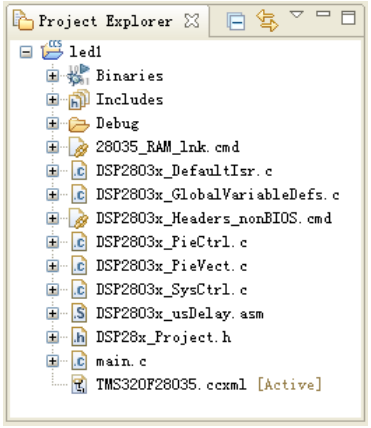

图7 工程中需要添加的文件

8.2 用GPIO引脚控制LED闪烁程序的编写

（2）初始化系统函数

```
void InitSysCtrl(void)
{
   DisableDog();     //关闭看门狗
   EALLOW;
   SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 1; // 使能ADC 外设时钟
   (*Device_cal)();
   SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 0; // 返回ADC 时钟到原始状态
   EDIS;
   IntOsc1Sel();
   InitPll(DSP28_PLLCR,DSP28_DIVSEL);  //初始化 PLL 控制: PLLCR 和
CLKINDIV
   InitPeripheralClocks();   // 初始化外部时钟
}
```

8.2 用GPIO引脚控制LED闪烁程序的编写

（3）初始化GPIO

```
void initGPIO(void)
{
    EALLOW;
    GpioCtrlRegs.GPBMUX1.bit.GPIO40=0;    //将GPIO40配置为通用I/O口
    GpioCtrlRegs.GPBDIR.bit.GPIO40=1;     //将GPIO40方向变为输出
    EDIS;
}
```

8.2 用GPIO引脚控制LED闪烁程序的编写

主函数程序编写如下：

```
#include "DSP28x_Project.h"
void main(void)
{
      InitSysCtrl();      //初始化系统函数
      DINT;               //禁止中断
      InitPieCtrl();      //初始化PIE控制寄存器
      IER=0x0000;         //清除CPU中断
      IFR=0x0000;         //清除CPU中断标志
      InitPieVectTable(); //初始化PIE中断向量表
      initGPIO();         //初始化LED的GPIO
```

8.2 用GPIO引脚控制LED闪烁程序的编写

```
while(1)
{
    GpioDataRegs.GPBDAT.bit.GPIO40=0;//将GPIO40置为低电平,即LED亮
    delay();    //LED亮持续一段时间
    GpioDataRegs.GPBDAT.bit.GPIO40=1;//将GPIO40置为高电平,即LED灭
    delay(); //LED灭持续一段时间
     }
}
```

8.2 用GPIO引脚控制LED闪烁程序的编写

```
void delay(void)      //延时函数
{
        int i,j;
        for(i=0;i<2000;i++)
        {
                for(j=0;j<100;j++);
        }
}
```