

## 第7章 通用输入输出GPIO

- 7.1 GPIO模块概述
- 7.2 GPIO模块的使用方法
- 7.3 相关寄存器说明
- 7.4 应用举例

## 7.1 GPIO模块概述

- GPIO是微机系统最基本的外设之一
- 现在的微机系统中的GPIO往往功能复用
- 掌握GPIO功能的选择方法
- 掌握IO作为GPIO的操作方法和注意事项



- **GPIO多路复用（MUX）寄存器用来选择共享引脚的操作。**
- **这些共享引脚用它们的通用I/O名称来命名（即，GPIO0 – GPIO38）。**
- **它们可以单独用作数字I/O、被称为GPIO，或者连接到3个外设I/O信号中的一个（通过GPxMUXn寄存器）。**
- **如果引脚被选择用作数字I/O模式，可以使用寄存器来配置引脚的方向（通过GPxDIR寄存器）。**

除了单个引脚位-I/O功能之外，每个GPIO使能的引脚还可以复用为多达3个独立的外设信号。

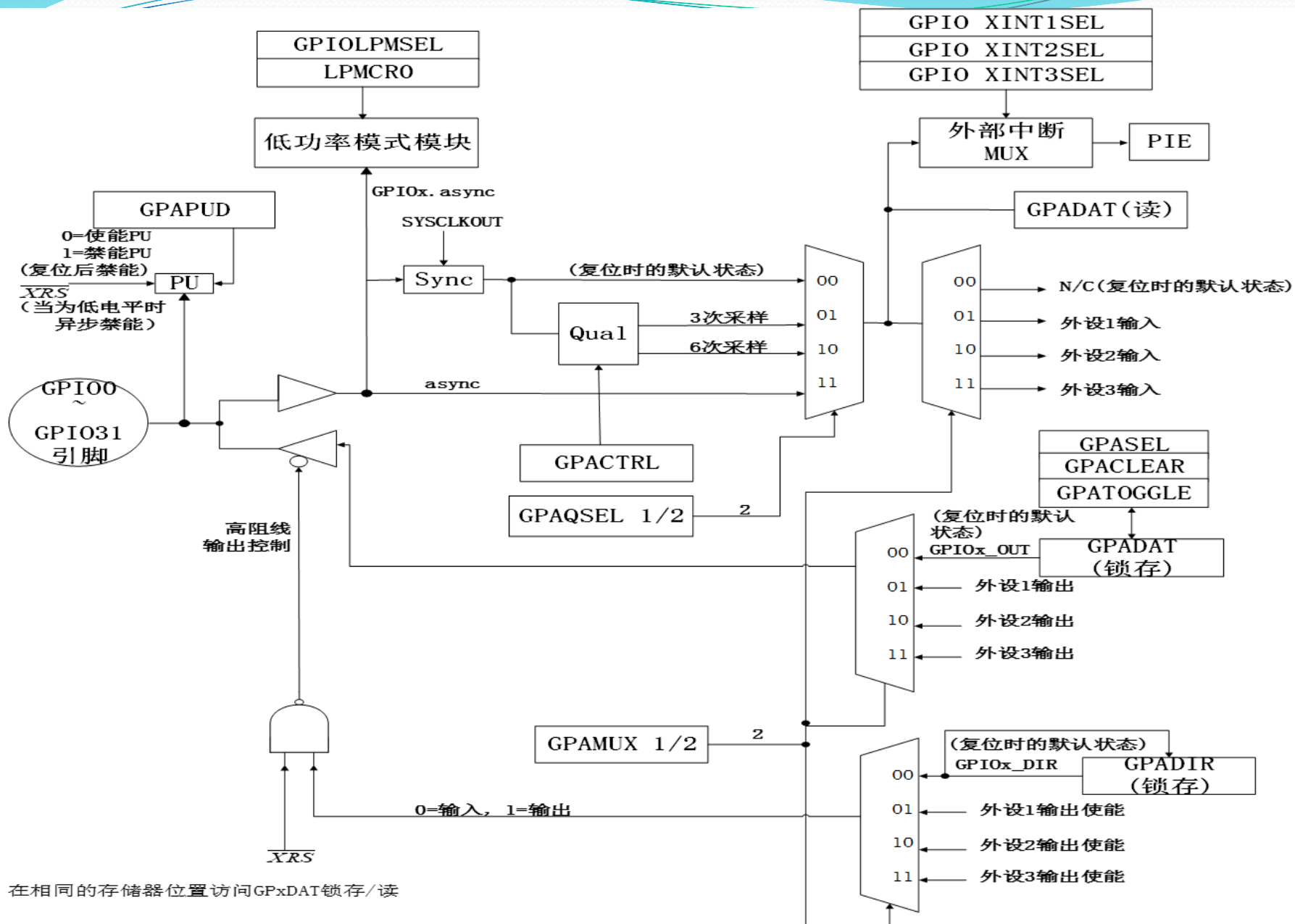
有3个I/O端口：

Port A由GPIO0 – GPIO31组成，

Port B由GPIO32 – GPIO38组成，

模拟端口包括AIO0 – AIO15。

# GPIO0~GPIO31多路复用图



A 在相同的存储器位置访问GPxDAT锁存/读



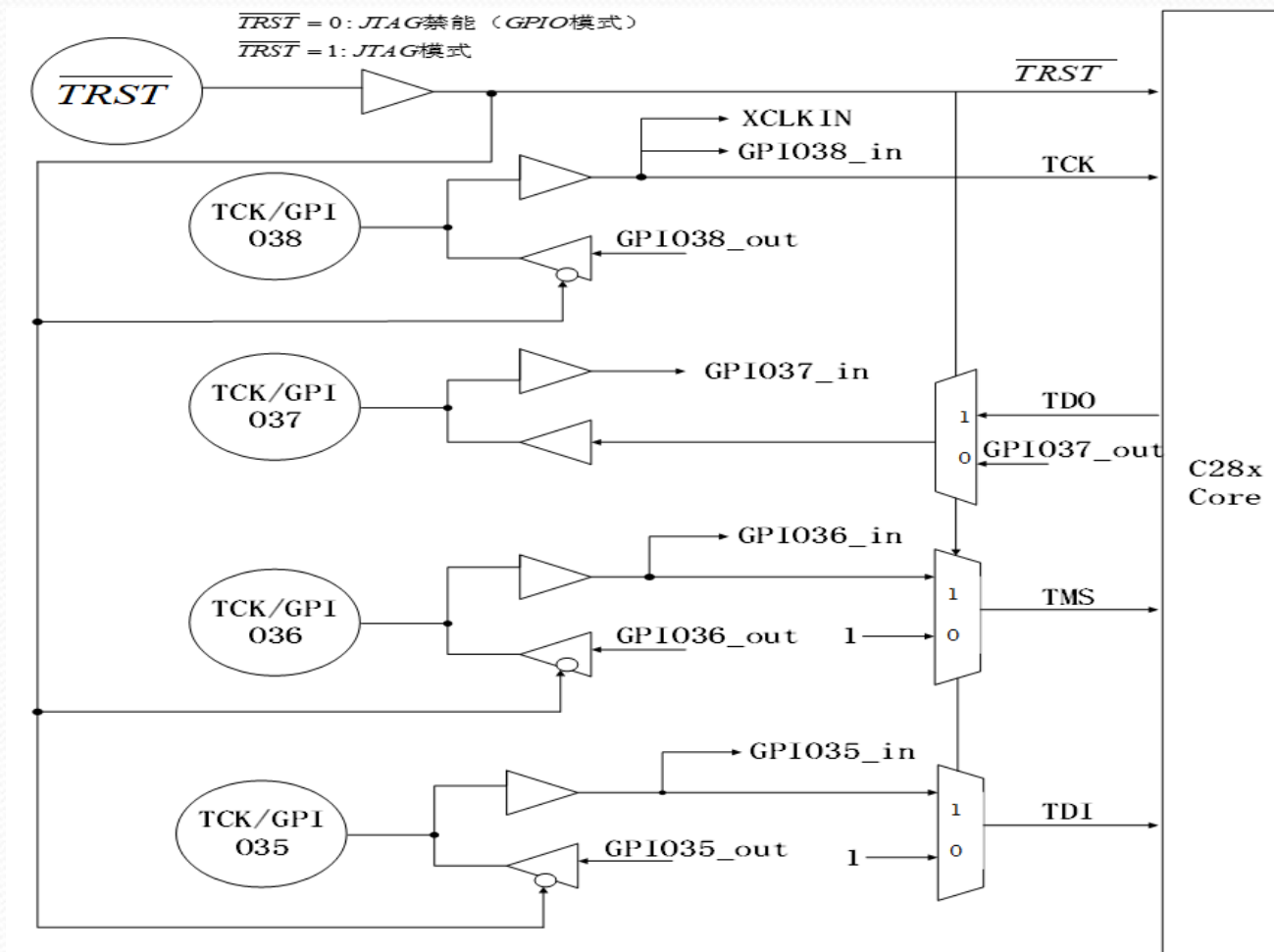


# JTAG PORT PINS :

在2802x器件上，JTAG端口的引脚减少到5个

TCK、TDI、TMS和TDO引脚也是GPIO引脚

$\overline{TRST}$  信号选择引脚用作JTAG工作模式还是GPIO工作模式







## 7.2 GPIO的使用方法

### 1) GPIO的配置

GPIO的配置由GPIO配置控制寄存器控制

Step 1. Plan the device pin-out

-- 安排器件引脚输出

Step 2. Enable or disable internal pull-up resistor

-- 使能或禁能内部上拉电阻

Step 3. Select input qualification

-- 选择输入鉴定

Step 4. Select the pin function

-- 选择引脚功能

Step 5. For digital general purpose I/O , select the direction of pin

-- 为数字通用I/O选择引脚的方向 输入或者输出

Step 6. Select low power mode wake-up sources

-- 选择低功率模式唤醒源(是否作为唤醒源)

Step 7. Select external interrupt sources

-- 选择外部中断源

## 2) 数字通用I/O操作

名称	地址	大小 (x16)	寄存器描述
<b>GPADAT</b>	0x6FC0	2	GPIO A数据寄存器 (GPIO0 – GPIO31)
<b>GPASET</b>	0x6FC2	2	GPIO A设置寄存器 (GPIO0 – GPIO31)
<b>GPACLEAR</b>	0x6FC4	2	GPIO A清除寄存器 (GPIO0 – GPIO31)
<b>GPATOGGLE</b>	0x6FC6	2	GPIO A翻转寄存器 (GPIO0 – GPIO31)
<b>GPBDAT</b>	0x6FC8	2	GPIO B数据寄存器 (GPIO32 – GPIO44)
<b>GPBSET</b>	0x6FCA	2	GPIO B设置寄存器 (GPIO32 – GPIO44)
<b>GPBCLEAR</b>	0x6FCC	2	GPIO B清除寄存器 (GPIO32 – GPIO44)
<b>GPBTOGGLE</b>	0x6FCE	2	GPIO B翻转寄存器 (GPIO32 – GPIO44)
<b>AIODAT</b>	0x6FD8	2	模拟I/O数据寄存器 (AIO0 – AIO15)
<b>AIOSET</b>	0x6FDA	2	模拟I/O设置寄存器 (AIO0 – AIO15)
<b>AIOCLEAR</b>	0x6FDC	2	模拟I/O清除寄存器 (AIO0 – AIO15)
<b>AIOTOGGLE</b>	0x6FDE	2	模拟I/O翻转寄存器 (AIO0 – AIO15)



## (1) GPxDAT/AIODAT寄存器

每个I/O端口有一个数据寄存器，该寄存器的每一位对应一个GPIO引脚。

不管引脚如何配置（配置成GPIO或外设功能）数据寄存器中相应的位都反映了鉴定后引脚的当前状态（不包括AIOx引脚）。

写GPxDATA/AIODAT寄存器清除或设置相应的输出锁存器：如果引脚使能用作一个通用输出（GPIO输出），引脚也将被驱动为高或低；如果引脚配置成外设功能，那么值将被锁存但引脚将不被驱动，只要引脚稍后会配置用作一个GPIO输出时，锁存的值才能驱动到引脚上。

不建议使用GPxDAT寄存器来改变一个引脚的输出，因为很容易一不小心将其它引脚的电平改变了。要改变一个引脚的输出，建议使用设置、清除和翻转寄存器。

(2) GPxSET/AIOSET寄存器

(3) GPxCLEAR/AIOCLEAR寄存器

(4) GPxTOGGLE/AIOTOGGLE寄存器

配置用作GPIO输出时：

写“1”，起作用，写“0”无影响。

如果引脚不配置用作GPIO输出时：

操作的值将被锁存，但不驱动引脚。

只有引脚稍后将配置用作GPIO输出时，锁存的值才被反应到引脚上。



### 3) 输入鉴定

输入鉴定机制设计得非常灵活，主要是用来消除不希望的噪声。

当不使用这个功能时，采用复位时的默认值即可。



## 4) 低功率模式唤醒

可以选择GPIOA的一个或者多个GPIO引脚将器件从停机和待机低功率模式中唤醒。

低功率模式唤醒选择寄存器（GPIO\_LPMSEL）的第0位至第31位分别对应为GPIO0至GPIO31，

如果某位被清零，相应引脚上的信号将对停机和待机低功率模式没有影响；

如果某位被设置为1，相应引脚上的信号就能将器件从停机和待机低功率模式中唤醒。

## 7.3 相关寄存器说明

- (1) 多路复用 (MUX) 寄存器
- (2) 方向寄存器
- (3) 上拉禁能寄存器 (GPxPUD)
- (4) 数据寄存器
- (5) XINTn中断选择寄存器 (GPIOXINTnSEL)
- (6) 输入鉴定寄存器
- (7) GPIO头文件中寄存器结构体说明

## ● (1) 多路复用 (MUX) 寄存器

名称	地址	大小 (x16)	寄存器描述
GPAMUX1	0x6F86	2	GPIO A MUX1 寄存器 (GPIO0 – GPIO15)
GPAMUX2	0x6F88	2	GPIO A MUX2 寄存器 (GPIO16 – GPIO31)
GPBMUX1	0x6F96	2	GPIO B MUX1 寄存器 (GPIO32 – GPIO38)
AIOMUX1	0x6FB6	2	模拟, I/O MUX1 寄存器 (AIO0 – AIO15)



# ● 多路复用 MUX

	复位时为默认状态	外设选择	外设选择2	外设选择3
GPAMUX1寄存器位	(GPAMUX1位=00)	(GPAMUX1位=01)	(GPAMUX1位=10)	(GPAMUX1位=11)
1-0	GPIO0	EPWM1A(O)	保留	保留
3-2	GPIO1	EPWM1B(O)	保留	COMP1OUT(O)
5-4	GPIO2	EPWM2A(O)	保留	保留
7-6	GPIO3	EPWM2B(O)	保留	COMP2OUT(O)
9-8	GPIO4	EPWM3A(O)	保留	保留
11-10	GPIO5	EPWM3B(O)	保留	ECAP1(I/O)
13-12	GPIO6	EPWM4A(O)	EPWMSYNCI(I)	EPWMSYNCO(O)
15-14	GPIO7	EPWM4B(O)	SCIRXDA(I)	保留
17-16	保留	保留	保留	保留
19-18	保留	保留	保留	保留
21-20	保留	保留	保留	保留
23-22	保留	保留	保留	保留
25-24	GPIO12	TZ1 (I)	SCITXDA(O)	保留
27-26	保留	保留	保留	保留
29-28	保留	保留	保留	保留
31-30	保留	保留	保留	保留

# ● 多路复用 MUX

	复位时为默认状态	外设选择	外设选择2	外设选择3
GPAMUX2寄存器位	(GPAMUX2位=00)	(GPAMUX2位=01)	(GPAMUX2位=10)	(GPAMUX2位=11)
1-0	GPIO16	SPISIMOA(I/O)	保留	$\overline{TZ2}(I)$
3-2	GPIO17	SPISOMIA(I/O)	保留	$\overline{TZ3}(I)$
5-4	GPIO18	SPICLK(A/I/O)	SCITXDA(O)	XCLKOUT(O)
7-6	GPIO19/XCLKIN	$\overline{SPISTE}A(I/O)$	SCIRXDA(I)	ECAP1(I/O)
9-8	保留	保留	保留	保留
11-10	保留	保留	保留	保留
13-12	保留	保留	保留	保留
15-14	保留	保留	保留	保留
17-16	保留	保留	保留	保留
19-18	保留	保留	保留	保留
21-20	保留	保留	保留	保留
23-22	保留	保留	保留	保留
25-24	GPIO28	SCIRXDA(I)	SDAA(I/OC)	$\overline{TZ2}(O)$
27-26	GPIO29	SCITXDA(O)	SCLA(I/OC)	$\overline{TZ3}(O)$
29-28	保留	保留	保留	保留
31-30	保留	保留	保留	保留

# ● 多路复用 MUX

	复位时为默认状态	外设选择	外设选择2	外设选择3
GPBMUX1寄存器位	(GPBMUX1位=00)	(GPBMUX1位=01)	(GPBMUX1位=10)	(GPBMUX1位=11)
1-0	GPIO32	SDAA(I/OC)	EPWMSYNCI(I)	$\overline{\text{ADCSOCAO}}$ (O)
3-2	GPIO33	SCLA(I/OC)	EPWMSYNCO(O)	$\overline{\text{ADCSOCBO}}$ (O)
5-4	GPIO34	COMP2OUT(O)	保留	保留
7-6	GPIO35(TDI)	保留	保留	保留
9-8	GPIO36(TMS)	保留	保留	保留
11-10	GPIO37(TDO)	保留	保留	保留
13-12	GPIO38/XCLKIN(TCK)	保留	保留	保留
15-14	保留	保留	保留	保留
17-16	保留	保留	保留	保留
19-18	保留	保留	保留	保留
21-20	保留	保留	保留	保留
23-22	保留	保留	保留	保留
25-24	保留	保留	保留	保留
27-26	保留	保留	保留	保留
29-28	保留	保留	保留	保留
31-30	保留	保留	保留	保留



# ● 多路复用 MUX

表7.5 2802x 模拟MUX

AIOMUX1寄存器位	AIOx和外设选择1 AIOMUX1位=0,x	复位时为默认状态 外设选择2和外设选择3 AIOMUX1位=1,x
1-0	A0(I)	A0 (I)
3-2	A1(I)	A1(I)
5-4	AIO2(I/O)	A2(I),COMP A1(I)
7-6	A3(I)	A3(I)
9-8	AIO4(I/O)	A4(I),COMP A2(I)
11-10	A5(I)	A5(I)
13-12	AIO6(I/O)	A6(I)
15-14	A7(I)	A7(I)
17-16	B0(I)	B0(I)
19-18	B1(I)	B1(I)
21-20	AIO10(I/O)	B2(I),COMP B1(I)
23-22	B3(I)	B3(I)
25-24	AIO12(I/O)	B4(I),COMP B2(I)
27-26	B5(I)	B5(I)
29-28	AIO14(I/O)	B6(I)
31-30	B7(I)	B7(I)

## GPIO MUX配置举例 (1):

GPAMUX1[13:12] Bit Setting	Pin Functionality Selected
If GPAMUX1[13:12] = 0,0	Pin configured as GPIO6
If GPAMUX1[13:12] = 0,1	Pin configured as EPWM4A (O)
If GPAMUX1[13:12] = 1,0	Pin configured as EPWMSYNCI (I)
If GPAMUX1[13:12] = 1,1	Pin configured as EPWMSYNCO (O)

## GPIO MUX配置举例 (2):

Pin Assigned to SPISIMOB	MUX Configuration
Choice 1 - GPIO12	GPAMUX[25:24] = 1,1
or Choice 2 - GPIO24	GPAMUX2[17:16] = 1,1

## (2) 方向寄存器

名称	地址	大小 (x16)	寄存器描述
<b>GPADIR</b>	0x6F8A	2	GPIO A方向寄存器 (GPIO0 – GPIO31)
<b>GPBDIR</b>	0x6F9A	2	GPIO B方向寄存器 (GPIO32 – GPIO38)
<b>AIODIR</b>	0x6FBA	2	模拟, I/O方向寄存器 (AIO0 – AIO15)

### GPADIR

位	域	值	描述 <sup>(1)</sup>
31-0	GPIO31- GPIO0	0 1	当指定的引脚在相应的 GPAMUX1 或 GPAMUX2寄存器中被配置用作一个GPIO, 这个域的值就控制着GPIO Port A引脚的方向。 将GPIO引脚配置作为输入。(默认) 将GPIO引脚配置作为输出。 GPADAT输出锁寄存器的当前值被驱动到引脚上。在将引脚从输入变成输出之前, 使用 GPASET、GPACLEAR 和 GPATOGGLE 寄存器初始化GPADAT锁寄存器。



# GPBDIR

位	域	值	描述 <sup>(1)</sup>
31-7	保留		保留
6-0	GPIO38- GPIO32	0 1	当选择GPIO模式时该域控制GPIO引脚的方向。读寄存器返回寄存器设置的当前值。 将GPIO配置用作输入。（默认） 将GPIO配置用作输出。

# AIODIR

位	域	值	描述 <sup>(1)</sup>
31- 15	保留		保留
14-0	AION	0 1	当选择AIO模式时该域控制可用AIO引脚的方向。读寄存器返回寄存器设置的当前值。 将AIO引脚配置用作输入。（默认） 将AIO引脚配置用作输出。

### (3) 上拉禁能寄存器 (GPxPUD)

#### GPAPUD

位	域	值	描述
31-0	GPIO31-GPIO0	0	配置所选GPIO Port A引脚上的内部上拉电阻。每个GPIO引脚对应该寄存器中的一个位。 <b>使能</b> 指定引脚上的内部上拉。(GPIO12 – GPIO31的默认状态)
		1	<b>禁能</b> 指定引脚上的内部上拉。(GPIO0 – GPIO11的默认状态)

#### GPBPUD

位	域	值	描述 <sup>(1)</sup>
31-7	保留		
6-0	GPIO38-GPIO32	0	配置所选GPIO Port B引脚上的内部上拉电阻。每个GPIO引脚对应该寄存器中的一个位。 <b>使能</b> 指定引脚上的内部上拉。(默认)
		1	<b>禁能</b> 指定引脚上的内部上拉。

## (4) 数据寄存器

表 7.2 GPIO数据寄存器

名称	地址	大小 (x16)	寄存器描述
GPADAT	0x6FC0	2	GPIO A数据寄存器 (GPIO0 – GPIO31)
GPASET	0x6FC2	2	GPIO A设置寄存器 (GPIO0 – GPIO31)
GPACLEAR	0x6FC4	2	GPIO A清除寄存器 (GPIO0 – GPIO31)
GPATOGGLE	0x6FC6	2	GPIO A翻转寄存器 (GPIO0 – GPIO31)
GPBDAT	0x6FC8	2	GPIO B数据寄存器 (GPIO32 – GPIO44)
GPBSET	0x6FCA	2	GPIO B设置寄存器 (GPIO32 – GPIO44)
GPBCLEAR	0x6FCC	2	GPIO B清除寄存器 (GPIO32 – GPIO44)
GPBTOGGLE	0x6FCE	2	GPIO B翻转寄存器 (GPIO32 – GPIO44)
AIODAT	0x6FD8	2	模拟I/O数据寄存器 (AIO0 – AIO15)
AIOSET	0x6FDA	2	模拟I/O设置寄存器 (AIO0 – AIO15)
AIOCLEAR	0x6FDC	2	模拟I/O清除寄存器 (AIO0 – AIO15)
AIOTOGGLE	0x6FDE	2	模拟I/O翻转寄存器 (AIO0 – AIO15)



## (5) XINTn中断选择寄存器 (GPIOXINTnSEL)

Bits	Field	Value	Description <sup>(2)</sup>
15-5	Reserved		Reserved
4-0	GPIOXINTnSEL	<p>Select the port A GPIO signal (GPIO0 - GPIO31) that will be used as the XINT1, XINT2, or XINT3 interrupt source. In addition, you can configure the interrupt in the XINT1CR, XINT2CR, or XINT3CR registers described in <a href="#">Section 6.6</a>. To use XINT2 as ADC start of conversion, enable it in the desired ADCSOCxCTL register. The ADCSOC signal is always rising edge sensitive.</p> <p>00000 Select the GPIO0 pin as the XINTn interrupt source (default)</p> <p>00001 Select the GPIO1 pin as the XINTn interrupt source</p> <p>... ..</p> <p>11110 Select the GPIO30 pin as the XINTn interrupt source</p> <p>11111 Select the GPIO31 pin as the XINTn interrupt source</p>	

n	Interrupt	Interrupt Select Register	Configuration Register
1	XINT1	GPIOXINT1SEL	XINT1CR
2	XINT2	GPIOXINT2SEL	XINT2CR
3	XINT3	GPIOXINT3SEL	XINT3CR

需要说明三点：

- 1) 可选作中断源的只能是Port A GPIO（GPIO0 – GPIO31）；
- 2) XINT1、XINT2或XINT3中断在XINT1CR、XINT2CR或XINT3CR寄存器中进行配置；
- 3) 如果要将XINT2用作ADC转换的启动信号，要在ADCSOCxCTL寄存器中使能它。

## (6) 输入鉴定寄存器

名称	地址	大小 (x16)	寄存器描述
<b>GPACTRL</b>	0x6F80	2	GPIO A控制寄存器（GPIO0 – GPIO31）
<b>GPAQSEL1</b>	0x6F82	2	GPIO A限定器选择1寄存器（GPIO0 – GPIO15）
<b>GPAQSEL2</b>	0x6F84	2	GPIO A限定器选择2寄存器（GPIO16 – GPIO31）
<b>GPBCTRL</b>	0x6F90	2	GPIO B控制寄存器（GPIO32 – GPIO38）
<b>GPBQSEL1</b>	0x6F92	2	GPIO B限定器选择1寄存器（GPIO32 – GPIO38）
<b>GPIOLPMS EL</b>	0x6FE8	1	LPM唤醒源选择寄存器（GPIO0 – GPIO31）

当不使用这个功能时，采用复位时的默认值即可。



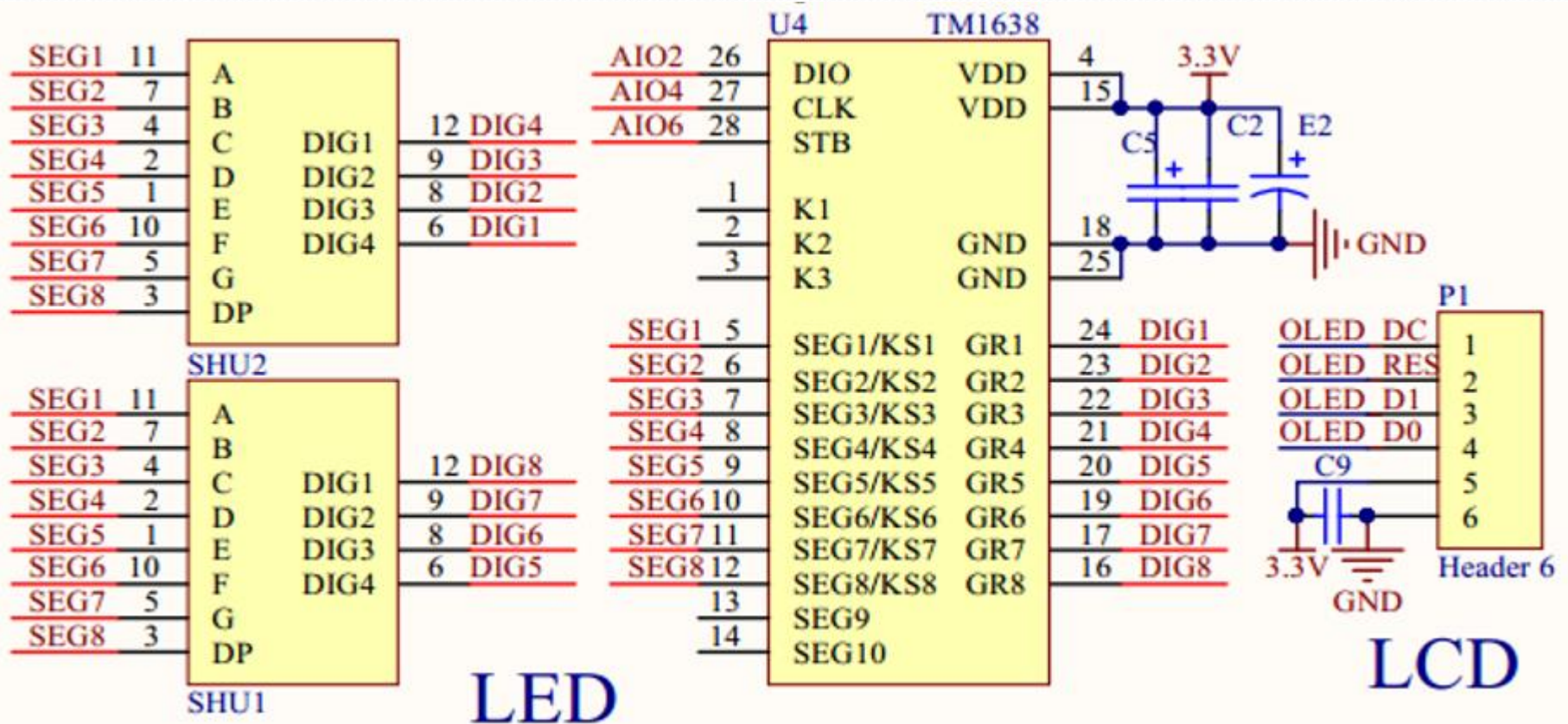
## (7) GPIO头文件中寄存器结构体说明

头文件中包括三个寄存器结构体：

控制寄存器结构体GPIO\_CTRL\_REGS、  
数据寄存器结构体GPIO\_DATA\_REGS、  
外部中断相关寄存器结构体GPIO\_INT\_REGS。

这些其中寄存器结构体中的位定义与前面介绍的寄存器位定义相同。

## 7.4 应用举例





## ● 程序举例：

```
#define TM_STB_L() (GpioDataRegs.AIOCLEAR.bit.AIO6 = 1)
#define TM_STB_H() (GpioDataRegs.AIOSET.bit.AIO6 = 1)
#define TM_CLK_L() (GpioDataRegs.AIOCLEAR.bit.AIO4 = 1)
#define TM_CLK_H() (GpioDataRegs.AIOSET.bit.AIO4 = 1)
#define TM_DIO_L() (GpioDataRegs.AIOCLEAR.bit.AIO2 = 1)
#define TM_DIO_H() (GpioDataRegs.AIOSET.bit.AIO2 = 1)
```

```
void InitLEDGpio(void) {
    EALLOW;
    GpioCtrlRegs.AIOMUX1.bit.AIO2 = 0;
    GpioCtrlRegs.AIOMUX1.bit.AIO4 = 0;
    GpioCtrlRegs.AIOMUX1.bit.AIO6 = 0;
    GpioCtrlRegs.AIODIR.bit.AIO2 = 1;
    GpioCtrlRegs.AIODIR.bit.AIO4 = 1;
    GpioCtrlRegs.AIODIR.bit.AIO6 = 1;
    EDIS;
}
```