

FT62F08X

IO_INTERRUPT Application note

目录

1. GPIO	3
1.1. IO 端口相关寄存器汇总	6
1.2. I/O 配置.....	11
1.3. PORTx 功能及优先级	13
1.4. PORT 端口变化中断	14
1.5. 关于读端口 PORTx.....	14
2. 应用范例.....	17
联系信息	20

FT62F08x IO_INTERRUPT 应用

1. GPIO

本芯片共包含 30 个 GPIO。这些 IO 除了作为普通输入/输出端口以外还通常具备一些与内核周边电路通讯的功能。

每个端口有 8 个标准寄存器供其操作使用。这些寄存器包括：

- TRISx 寄存器 (数据方向寄存器)
- PORTx 寄存器 (用于读器件引脚上的电平)
- LATx 寄存器 (输出锁存器)
- WPUx 寄存器 (上拉控制)
- WPDx 寄存器 (下拉控制)
- PSRCx 寄存器 (源电流选择)
- PSINKx 寄存器 (灌电流选择)
- ITYPEx 寄存器 (中断类型选择)

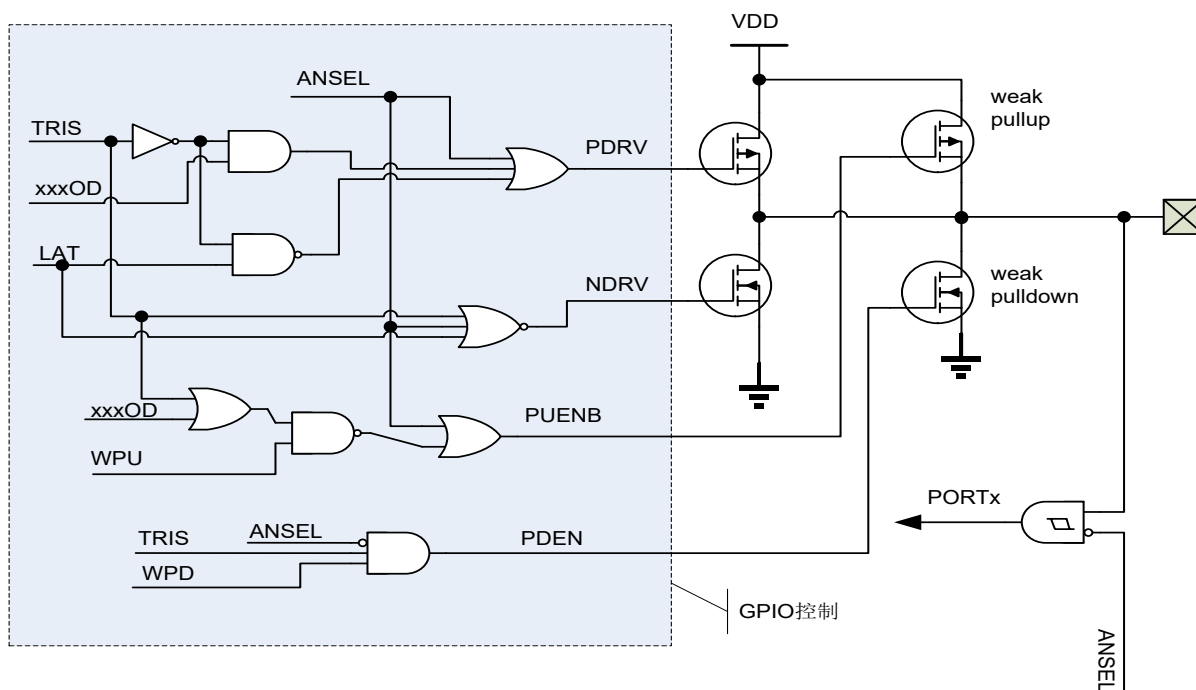


图 1-1 PORT 端口结构原理

某些端口可能有以下额外寄存器：

- ANSELx (模拟选择寄存器)

通常情况下，当某个端口上的外设使能时，其相关引脚可能不能用作通用输出引脚，但可读取该引脚。

数据锁存器 (LATx 寄存器) 用于对 I/O 引脚所驱动的值进行“读-修改-写”操作。对 LATx 寄存器的写操作与对相应 PORTx 寄存器的写操作具有相同的效果。读 LATx 寄存器将读取保存在 I/O 端口锁存器中的值，而读 PORTx 寄存器将读取实际的 I/O 引脚值。

支持模拟输入的端口具有相关的 ANSELx 寄存器。当 ANSELA 位置 1 时，禁止与该位相关的数字输入缓冲器。禁止输入缓冲器可防止逻辑输入电路产生短路电流。

所有 I/O 引脚均具有以下功能 (表 1-2, 表 1-3):

- 数字输出
- 数字输入
- 开漏 (SPI, I2C, USART 相应端口)
- 弱上拉
- 弱下拉

此外, 部分 I/O 具有以下特殊功能:

1. 烧录调试引脚 (ISP-Data, ISP-CLK), 硬件内部连接, 不需设置。
2. 通过 IDE 界面配置, 且在芯片初始化配置时加载的功能 (表 1-7):
 - 外部时钟/晶振输入 (OSC1, OSC2)
 - 系统外部复位 (/MCLR)
3. 通过指令对相应 I/O 引脚进行配置的其他功能, 可分为 4 类:
 - a. 数字输出
 - PWM
 - 内部时钟输出
 - b. 数字输入
 - PWM 故障刹车
 - Timer1 触发
 - GPIO 端口变化中断
 - ADC 触发 (ADC_ETR)
 - c. 模拟输入
 - LVD / BOR
 - ADC
 - TOUCH
 - V_{REF+}
 - V_{REF-}
 - d. 通信接口
 - SPI
 - I2C
 - USART

引脚名	ISP 调试	时钟	ADC	SPI	I2C	USART	中断	LVD	PWM	数字 I/O 上拉/下拉	开漏	源电流 (mA)	灌电流 (mA)
PA0				MOSI			√		PWM1	√	√	2, 4, 14, 26	53, 62
PA1				MISO			√		PWM2	√	√	2, 4, 14, 26	53, 62
PA2	CLK				[SCL]	[RX]	√			√	√	2, 4, 14, 26	53, 62
PA3			AN6				√		PWM2N	√		2, 4, 14, 26	53, 62
PA4			AN5/ Trigger				√		PWM6	√		2, 4, 14, 26	53, 62
PA5						CK	√		PWM5	√		2, 4, 14, 26	53, 62
PA6			AN4			TX	√			√	√	2, 4, 14, 26	53, 62
PA7			AN3			RX	√	ELVD0		√		2, 4, 14, 26	53, 62
PB0				SCK			√		PWM3N/ [PWM5]	√		2, 4, 14, 26	53, 62
PB1		输出	AN7				√		PWM4	√		2, 4, 14, 26	53, 62
PB2			Trigger		SCL		√			√	√	2, 4, 14, 26	53, 62
PB3					SDA		√		BKIN	√	√	2, 4, 14, 26	53, 62
PB4							√		PWM3	√		2, 4, 14, 26	53, 62
PB5			(V _{REF+})	NSS			√		PWM7	√		2, 4, 14, 26	53, 62
PB6	DATA		(V _{REF-})		[SDA]	[TX]	√			√	√	2, 4, 14, 26	53, 62
PB7		OSC-	AN0	[MOSI]			√	ELVD3		√	√	2, 4, 14, 26	53, 62
PC0			AN2				√	ELVD1	PWM1N	√		2, 4, 14, 26	53, 62
PC1		OSC+	AN1	[MISO]			√	ELVD2		√	√	2, 4, 14, 26	53, 62
PC2							√			√		2, 4, 14, 26	53, 62
PC3							√			√		2, 4, 14, 26	53, 62
PC4							√		TIM1_ETR	√		2, 4, 14, 26	53, 62
PC5							√		[PWM3N]	√		2, 4, 14, 26	53, 62
PC6							√		[PWM2N]	√		2, 4, 14, 26	53, 62
PC7							√		[PWM1N]	√		2, 4, 14, 26	53, 62
PD0				[NSS]			√		[PWM1]	√		2, 4, 14, 26	53, 62
PD1						[CK]	√			√		2, 4, 14, 26	53, 62
PD2							√		[PWM2]	√		2, 4, 14, 26	53, 62
PD3				[SCK]			√		[PWM3]	√		2, 4, 14, 26	53, 62
PD4		输出					√		[BKIN]	√		2, 4, 14, 26	53, 62
PD5							√		[PWM4]	√		2, 4, 14, 26	53, 62
注							/MCLR = PC0					V _{DD} =5, V _{DS} =0.5	

表 1-1 I/O 端口功能

注： 所有 IO 支持 4 档可配置源电流驱动能力 (参阅 “PSRCx”, 表 1-3), 和 2 档可配置灌电流驱动能力 (参阅 “PSINKx”, 表 1-3)。

1.1. IO 端口相关寄存器汇总

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	复位值
ANSELA	0x197	ANSELA[7:0]								0000 0000
TRISA	0x8C	TRISA[7:0], PORTA 方向控制								1111 1111
TRISB	0x8D	TRISB[7:0], PORTB 方向控制								1111 1111
TRISC	0x8E	TRISC[7:0], PORTC 方向控制								1111 1111
TRISD	0x8F	-	-	TRISD[5:0], PORTD 方向控制						--11 1111
PORTA	0x0C	PORTA 输出寄存器								xxxx xxxx
PORTB	0x0D	PORTB 输出寄存器								xxxx xxxx
PORTC	0x0E	PORTC 输出寄存器								xxxx xxxx
PORTD	0x0F	-	-	PORTD[5:0] 输出寄存器						--xx xxxx
LATA	0x10C	PORTA 数据锁存器								xxxx xxxx
LATB	0x10D	PORTB 数据锁存器								xxxx xxxx
LATC	0x10E	PORTC 数据锁存器								xxxx xxxx
LATD	0x10F	-	-	PORTD[5:0] 数据锁存器						--xx xxxx
WPUA	0x18C	PORTA 弱上拉								0000 0000
WPUB	0x18D	PORTB 弱上拉								0000 0000
WPUC	0x18E	PORTC 弱上拉								0000 0000
WPUD	0x18F	-		PORTD[5:0] 弱上拉						--00 0000
WPDA	0x20C	PORTA 弱下拉								0000 0000
WPDB	0x20D	PORTB 弱下拉								0000 0000
WPDC	0x20E	PORTC 弱下拉								0000 0000
WPDD	0x20F	-		PORTD[5:0] 弱下拉						--00 0000
ODCON0	0x21F	-	-	-	-	-	SPIOD	I2COD	UROD	---- -000
PSRC0	0x11A	PORTA[7:0], PORTB[7:0] 源电流设置								1111 1111
PSRCB1	0x11B	PORTC[7:0], PORTD[5:0] 源电流设置								1111 1111
PSINK0	0x19A	PORTA 灌电流设置								0000 0000
PSINK1	0x19B	PORTB 灌电流设置								0000 0000
PSINK2	0x19C	PORTC 灌电流设置								0000 0000
PSINK3	0x19D	-	-	PORTD 灌电流设置						--00 0000
ITYPE0	0x11E	PORTx[3:0] (x = A, B, C, D) 外部管脚中断类型设置								0000 0000
ITYPE1	0x11F	PORTD[5:4] 和 PORTx[7:4] (x = A, B, C) 外部管脚中断类型设置								0000 0000
AFP0	0x19E	管脚重映射寄存器 0								0000 0000
AFP1	0x19F	-	管脚重映射寄存器 1							-000 0000
AFP2	0x11D	-	-	-	管脚重映射寄存器 2					---0 0000
EPS0	0x118	外部中断 EINT3~0 管脚选择								0000 0000
EPS1	0x119	外部中断 EINT7~4 管脚选择								0000 0000
EPIE0	0x14	外部管脚中断使能位								0000 0000
EPIF0	0x94	外部管脚中断标志位								0000 0000

表 1-2 I/O 相关用户寄存器的地址和复位值

名称	状态		寄存器	地址	复位值	
TRISA	PORTA	PORT 端口数字输出 (方向控制) 1 = 关闭 0 = 使能 (关闭上拉/下拉)	TRISA[7:0]	0x8C	RW-1111 1111	
TRISB	PORTB		TRISB[7:0]	0x8D	RW-1111 1111	
TRISC	PORTC		TRISC[7:0]	0x8E	RW-1111 1111	
TRISD	PORTD		TRISD[5:0]	0x8F	RW-11 1111	
ANSELA	1 = 关闭上拉/下拉, 及数字输入 (仅适用于 8 个 ADC 通道) 0 = (无动作)		ANSELA[7:0]	0x197	RW-0000 0000	
WPUA	PORTA	弱上拉 1 = 使能 0 = 关闭	WPUA[7:0]	0x18C	RW-0000 0000	
WPUB	PORTB		WPUB[7:0]	0x18D	RW-0000 0000	
WPUC	PORTC		WPUC[7:0]	0x18E	RW-0000 0000	
WPUD	PORTD		WPUD[5:0]	0x18F	RW-00 0000	
WPDA	PORTA	弱下拉 1 = 使能 0 = 关闭	WPDA[7:0]	0x20C	RW-0000 0000	
WPDB	PORTB		WPDB[7:0]	0x20D	RW-0000 0000	
WPDC	PORTC		WPDC[7:0]	0x20E	RW-0000 0000	
WPDD	PORTD		WPDD[5:0]	0x20F	RW-00 0000	
PORTA	PORTA	数据输出寄存器 读: 返回 IO 引脚上的电平 写: 写入相应的 LATx 寄存器	PORTA[7:0]	0x0C	RW-xxxx xxxx	
PORTB	PORTB		PORTB[7:0]	0x0D	RW-xxxx xxxx	
PORTC	PORTC		PORTC[7:0]	0x0E	RW-xxxx xxxx	
PORTD	PORTD		PORTD[5:0]	0x0F	RW-xx xxxx	
LATA	PORTA	数据锁存器寄存器	LATA[7:0]	0x10C	RW-xxxx xxxx	
LATB	PORTB		LATB[7:0]	0x10D	RW-xxxx xxxx	
LATC	PORTC		LATC[7:0]	0x10E	RW-xxxx xxxx	
LATD	PORTD		LATD[5:0]	0x10F	RW-xx xxxx	
SPIOD	SPI_MISO, SPI_MOSI		开漏 1 = 使能 0 = 关闭	ODCON0[2]	0x21F RW-0	
I2COD	I2C_SDA, I2C_SCL			ODCON0[1]		RW-0
UROD	USART_TX			ODCON0[0]		RW-0
AFP0	USART_CK 复用管脚选择		1 = PD1 0 = PA5	AFP0[7]	0x19E RW-0	
	TIM1_CH1 复用管脚选择		1 = PD1 0 = PA0	AFP0[6]		RW-0
	SPI_NSS 复用管脚选择		1 = PD0 0 = PB5	AFP0[5]		RW-0
	TIM1_CH1N 复用管脚选择		1 = PC7 0 = PC0	AFP0[4]		RW-0
	TIM1_CH2N 复用管脚选择		1 = PC6 0 = PA3	AFP0[3]		RW-0
	TIM1_CH3N 复用管脚选择		1 = PC5 0 = PB0	AFP0[2]		RW-0
	ADC_ETR 复用管脚选择		1 = PB2 0 = PA4	AFP0[1]		RW-0
	I2C_SDA 复用管脚选择		1 = PB6 0 = PB3	AFP0[0]		RW-0
AFP1	CLKO 复用管脚选择		1 = PD4 0 = PB1	AFP1[6]	0x19F RW-0	
	TIM1_CH4 复用管脚选择		1 = PD5 0 = PB1	AFP1[5]		RW-0

名称	状态		寄存器	地址	复位值
	I2C_SCL 复用管脚选择	1 = PA2 0 = <u>PB2</u>	AFP1[4]		RW-0
	TIM1_BKIN 复用管脚选择	1 = PD4 0 = <u>PB3</u>	AFP1[3]		RW-0
	TIM2_CH1 复用管脚选择	1 = PB0 0 = <u>PA5</u>	AFP1[2]		RW-0
	TIM1_CH3 复用管脚选择	1 = PD3 0 = <u>PB4</u>	AFP1[1]		RW-0
	TIM1_CH2 复用管脚选择	1 = PD2 0 = <u>PA1</u>	AFP1[0]		RW-0
AFP2	SPI_SCK 复用管脚选择	1 = PD3 0 = <u>PB0</u>	AFP2[4]	0x11D	RW-0
	SPI_MOSI 复用管脚选择	1 = PB7 0 = <u>PA0</u>	AFP2[3]		RW-0
	SPI_MISO 复用管脚选择	1 = PC1 0 = <u>PA1</u>	AFP2[2]		RW-0
	USART_RX 复用管脚选择	1 = PA2 0 = <u>PA7</u>	AFP2[1]		RW-0
	USART_TX 复用管脚选择	1 = PB6 0 = <u>PA6</u>	AFP2[0]		RW-0
PSINK0	PA7-PA0	<u>灌电流 (mA)</u> 1 = 62 0 = <u>53</u>	PSINK0[7:0]	0x19A	RW-0000 0000
PSINK1	PB7-PB0		PSINK1[7:0]	0x19B	RW-0000 0000
PSINK2	PC7-PC0		PSINK2[7:0]	0x19C	RW-0000 0000
PSINK3	PD5-PD0		PSINK3[5:0]	0x19D	RW-00 0000
PSRCB[3:2]	PB7-PB4	<u>源电流 (mA)</u> (00) = 2 (01) = 4 (10) = 14 (11) = <u>26</u>	PSRC0[7:6]	0x11A	RW-11
PSRCB[1:0]	PB3-PB0		PSRC0[5:4]		RW-11
PSRCA[3:2]	PA7-PA4		PSRC0[3:2]		RW-11
PSRCA[1:0]	PA3-PA0		PSRC0[1:0]		RW-11
PSRCD[3:2]	PD5-PD4		PSRC1[7:6]	0x11B	RW-11
PSRCD[1:0]	PD3-PD0		PSRC1[5:4]		RW-11
PSRCC[3:2]	PC7-PC4		PSRC1[3:2]		RW-11
PSRCC[1:0]	PC3-PC0		PSRC1[1:0]		RW-11

表 1-3 I/O 相关用户寄存器

名称	状态		寄存器	地址	复位值
ITYPE0[1:0]	PORTx.0	<u>外部管脚中断触发类型</u> 00 = <u>低电平</u> 01 = 上升沿 10 = 下降沿 11 = 双边沿	ITYPE0[1:0]	0x11E	RW-00
ITYPE0[3:2]	PORTx.1		ITYPE0[3:2]		RW-00
ITYPE0[5:4]	PORTx.2		ITYPE0[5:4]		RW-00
ITYPE0[7:6]	PORTx.3		ITYPE0[7:6]		RW-00
ITYPE1[1:0]	PORTx.4		ITYPE1[1:0]	0x11F	RW-00
ITYPE1[3:2]	PORTx.5		ITYPE1[3:2]		RW-00
ITYPE1[5:4]	PORTy.6		ITYPE1[5:4]		RW-00
ITYPE1[7:6]	PORTy.7		ITYPE1[7:6]		RW-00

表 1-4 I/O 中断触发寄存器 (x = A, B, C, D; y = A, B, C)

名称	状态		寄存器	地址	复位值
EINT0	<u>EINT0 管脚选择</u>	00 = <u>PA0</u> 10 = PC0 01 = PB0 11 = PD0	EPS0[1:0]	0x118	RW-00
EINT1	<u>EINT1 管脚选择</u>	00 = <u>PA1</u> 10 = PC1 01 = PB1 11 = PD1	EPS0[3:2]		RW-00
EINT2	<u>EINT2 管脚选择</u>	00 = <u>PA2</u> 10 = PC2 01 = PB2 11 = PD2	EPS0[5:4]		RW-00
EINT3	<u>EINT3 管脚选择</u>	00 = <u>PA3</u> 10 = PC3 01 = PB3 11 = PD3	EPS0[7:6]		RW-00
EINT4	<u>EINT4 管脚选择</u>	00 = <u>PA4</u> 10 = PC4 01 = PB4 11 = PD4	EPS1[1:0]	0x119	RW-00
EINT5	<u>EINT5 管脚选择</u>	00 = <u>PA5</u> 10 = PC5 01 = PB5 11 = PD5	EPS1[3:2]		RW-00
EINT6	<u>EINT6 管脚选择</u>	00 = <u>PA6</u> 10 = PC6 01 = PB6 11 = 保留	EPS1[5:4]		RW-00
EINT7	<u>EINT7 管脚选择</u>	00 = <u>PA7</u> 10 = PC7 01 = PB7 11 = 保留	EPS1[7:6]		RW-00

表 1-5 外部中断管脚选择寄存器

名称	状态		寄存器	地址	复位值
EPIEx	<u>外部中断使能位</u>	1 = 使能 0 = 禁止	EPIE0[7:0]	0x94	RW-00000000
EPIF0x ¹	<u>外部中断标志位</u>	1 = Yes (锁存) 0 = No	EPIF0[7:0]	0x14	R_W1C-00000000

表 1-6 外部中断使能和中断标志寄存器

¹ 写 1 清 0，写 0 无效。建议只使用 STR、MOVWI 指令进行写操作，而不要用 BSR 或 IOR 指令。

名称	功能	默认
MCLRE	外部 I/O 复位	关闭
FOSC	<ul style="list-style-type: none"> LP: PC1 (+) 和 PB7 (-) 接外部低速晶振 XT: PC1 (+) 和 PB7 (-) 接外部高速晶振 EC: PC1 (+) 接外部时钟输入, PB7 为 I/O INTOSCIO: PC1 和 PB7 为 I/O 	INTOSCIO
I2CRMAP	<p><u>I2C 复用管脚选择</u></p> <p>[PB3, PB2]: (\geq I 版芯片适用)</p> <p>I2C_SDA = PB3, I2C_SCL = PB2, SPI_MOSI = PA0, SPI_MISO = PA1</p> <p>[PA0, PA1]:</p> <p>I2C_SDA = PA0, I2C_SCL = PA1, SPI_MOSI = PB3, SPI_MISO = PB2</p>	[PB3, PB2]

表 1-7 I/O 相关初始化配置寄存器

1.2. I/O 配置

每个 PORT 端口，均需根据其相应功能配置以下 4 个模块 (表 1-3):

- 数字输出
- 数字输入
- 开漏
- 弱上拉
- 弱下拉

功能	数字输入	上拉/下拉	数字输出	设置
ISP-DATA	On	Off	On	(硬件内置, 忽略指令)
ISP-CLK	On	Off	Off	(硬件内置, 忽略指令)
/MCLR	On	上拉	Off	(初始化配置, 忽略指令)
时钟输出	(忽略)	Off	On	(初始化配置, 忽略指令)
OSC+ (EC)	On	(可选)	Off	(初始化配置, 忽略指令)
OSC+ / OSC- (LP, XT)	Off	Off	Off	(初始化配置, 忽略指令)
ADC	Off	Off	Off	TRISx = 1; ANSELx = 1
TOUCH	Off	Off	Off	TRISx = 1
SPI 输出	On	Off	On	TRISx = 0
I2C 输出	On	Off	On	TRISx = 0
USART 输出	On	Off	On	TRISx = 0
LVD	Off ⁽⁵⁾	Off	Off	TRISx = 1; ANSELx = 1
V _{REF+} / V _{REF-}	Off	Off	Off	TRISx = 1
ADC 触发	On	(可选)	Off	TRISx = 1
SPI 输入	On	(可选)	Off	TRISx = 1
I2C 输入	On	(可选)	Off	TRISx = 1
USART 输入	On	(可选)	Off	TRISx = 1
端口变化中断	On	(可选)	Off	TRISx = 1
BKIN	On	(可选)	Off	TRISx = 1
数字输入	On	(可选)	Off	TRISx = 1
PWM	On	Off	On	TRISx = 0
数字输出	On	Off	On	TRISx = 0

表 1-8 I/O 配置标志和用户寄存器

注:

1. TRISx = 0: “数字输出” 使能, “上拉/下拉” 自动关闭 (忽略 WPDx, WPUx)。
2. TRISx = 1: “数字输出” 关闭。
3. ANSELx = 1: “上拉”、“下拉”、“数字输入” 自动关闭 (忽略 WPDx, WPUx)。
4. 可关闭 “数字输入” 的唯一指令为 “ANSELx = 1”。
5. 将 PORT 端口设置为 LVD 输入时, 其“数字输入”、“上拉”和“下拉”功能被自动关闭。当 LVD 输入需要在不同的通道之间切换使用时, 通过设置 “ANSELx = 1” 可关闭当前未被选择通道的“数字输入”。
6. /MCLR 使能: PC0 的弱上拉功能自动使能 (忽略 WPUC[0]); 读 PORTC[0] 的值为 “0”。

7. 对 PORTx 数据输出寄存器进行写操作，I/O 端口将输出相应的逻辑电平。每组多达 8 个 I/O 的数据寄存器共用相同的地址，写操作实际执行‘读-修改-写’的过程，即先读取该组 PORTx 端口锁存器值（输出或输入），然后修改，再写回 PORTx 数据寄存器。
8. 数字输出和数字输入功能可以共存，有些应用需要同时使能数字输出和数字输入。
9. 当 TRISx = 0 时，通过软件可读取 PORTx 输出或输入锁存器 LATx 的值。
10. ODCON0x = 1: “ SPI_MISO, SPI_MOSI ”, “ I2C_SCL, I2C_SDA ”, “ USART_TX ”管脚开漏输出。对于拥有重映射功能的 I2C，开漏设置只应用在对应的被映射的管脚上。管脚的开漏功能和内部上拉功能可以同时打开
11. 完全复位或系统复位时，PORTx 寄存器不会复位，但 TRISx 将被重置为“1”，从而关闭输出。
12. 部分管脚输入/输出支持重映射功能，AFPx 寄存器可在两个管脚之间选择。

1.3. PORTx 功能及优先级

每个 I/O 管脚均复用了多个功能，当某管脚复用的功能模块都使能输出的情况下，就存在优先级的问题。

因为输入是连到各个功能模块的，故输入不存在优先级问题，例如 PB0 作为 GPIO 输入功能时，同时也作为 TIM2 的捕捉输入。

管脚名称	功能优先级 0	功能优先级 1	功能优先级 2	功能优先级 3
PA0	PA0	SPI_MISO	TIM1_CH1	-
PA1	PA1	SPI_MOSI	TIM1_CH2	-
PA2	PA2	USART_RX	I2C_SCL	ISPCLK (调试模式)
PA3	PA3	TIM1_CH2N	-	-
PA4	PA4	TIM2_CH2	-	-
PA5	PA5	USART_CK	TIM2_CH1	-
PA6	PA6	USART_TX	-	-
PA7	PA7	-	-	-
PB0	PB0	SPI_SCK	TIM2_CH1	TIM1_CH3N
PB1	PB1	TIM1_CH4	CLKO	-
PB2	PB2	I2C_SCL	-	-
PB3	PB3	I2C_SDA	-	-
PB4	PB4	TIM1_CH3	-	-
PB5	PB5	SPI_NSS	TIM2_CH3	-
PB6	PB6	USART_TX	I2C_SDA	ISPDAT (调试模式)
PB7	PB7	SPI_MOSI	OSC2 (XT 模式)	-
PC0	PC0	TIM1_CH1N	MCLR (复位脚)	-
PC1	PC1	SPI_MISO	OSC1 (XT 模式)	-
PC2	PC2	-	-	-
PC3	PC3	-	-	-
PC4	PC4	-	-	-
PC5	PC5	TIM1_CH3N	-	-
PC6	PC6	TIM1_CH2N	-	-
PC7	PC7	TIM1_CH1N	-	-
PD0	PD0	SPI_NSS	-	-
PD1	PD1	USART_CK	TIM1_CH1	-
PD2	PD2	TIM1_CH2	-	-
PD3	PD3	SPI_SCK	TIM1_CH3	-
PD4	PD4	CLKO	-	-
PD5	PD5	TIM1_CH4	-	-

表 1-9 IO 功能优先级

1.4. PORT 端口变化中断

所有 I/O 都可选择为外部中断源 (通过 EPS0 和 EPS1 设置), 同一时刻最多只有 8 个 IO 可以作为外部中断管脚, 中断类型可通过寄存器 ITYPE0 和 ITYPE1 设置。它们具备以下特性:

- 上升沿中断
- 下降沿中断
- 双边沿中断
- 低电平中断

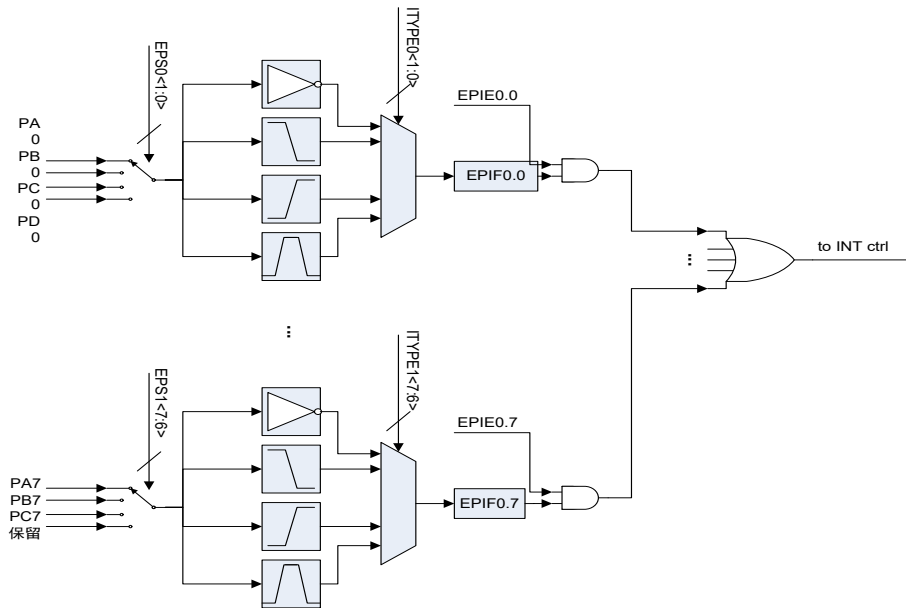


图 1-2 端口变化中断结构框图

1.5. 关于读端口 PORTx

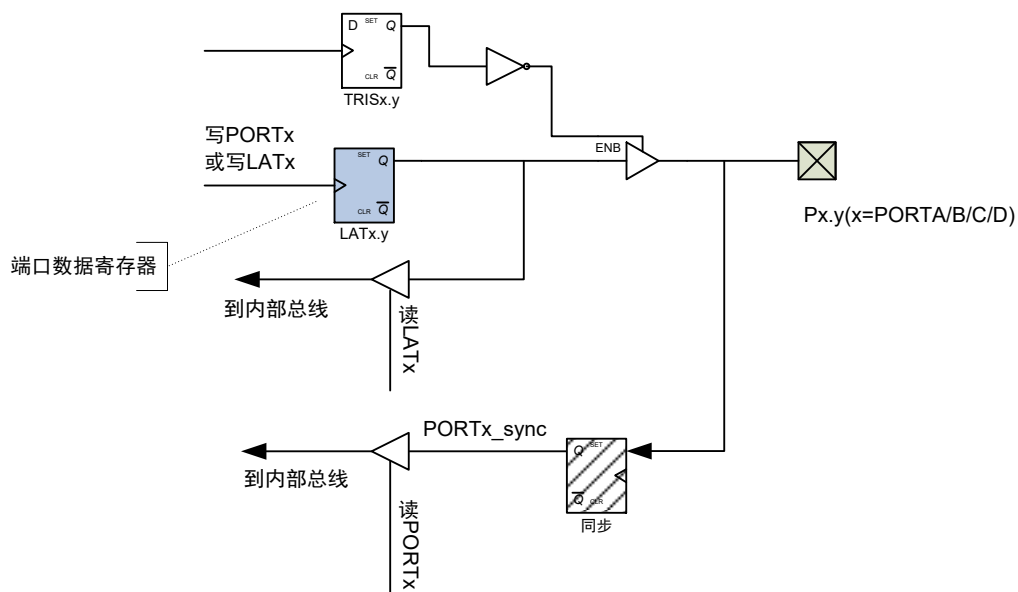


图 1-3 端口读操作原理框图

在 FT62F08x 系列芯片中，操作 GPIO 有两种方式：访问 PORTx 寄存器或者 LATx 寄存器，它们有不同的 SFR 地址。

对于读操作：“读 PORTx”返回的是管脚经过同步寄存器后的值，而“读 LATx”返回的是端口数据寄存器的值；换言之，软件对端口数据寄存器写操作之后，至少要经过一个系统时钟之后，才能通过“读 PORTx”的方式得到新值，而“读 LATx”则无需等待；

对于写操作：无论是写 PORTx 还是 LATx，都是对端口数据寄存器进行写；

由于以上特性，当软件使用“读-修改-写”指令对 PORTx 进行写操作时，需要特别注意以下情形：

```
BSR    PORTx, n    ; 对PORTx 第n 位置1
BSR    PORTx, m    ; 对PORTx 第m 位置1
...
```

软件期望的波形如下：

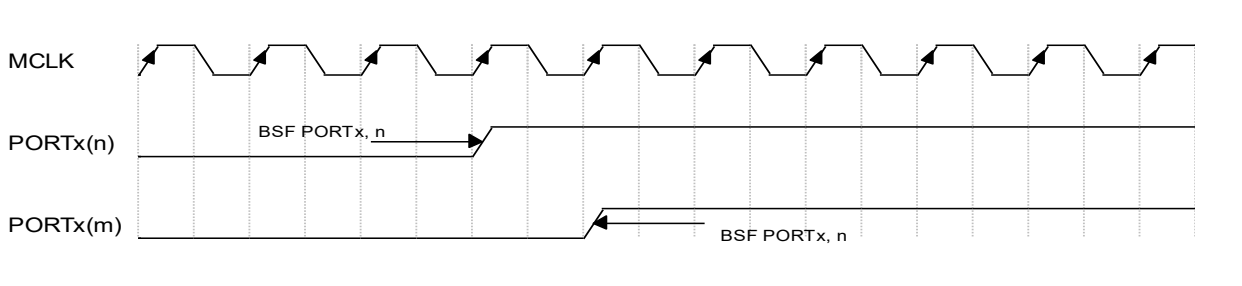


图 1-4 连续使用 RMW 指令对 PORTx 写操作的期望时序

实际输出波形如下：

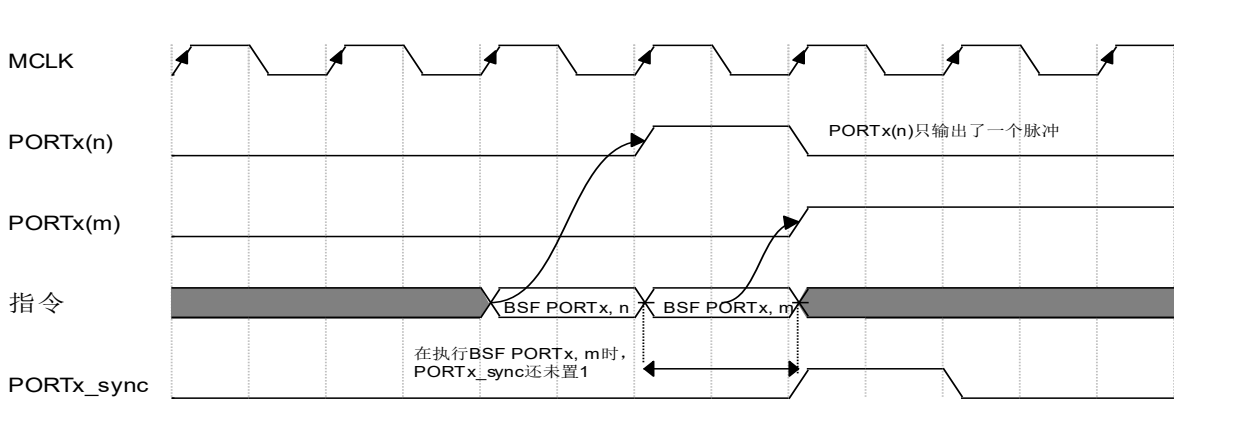


图 1-5 连续使用 RMW 指令对 PORTx 写操作的输出时序

出现这个现象的原因是在执行“BSR PORTx, m” (回顾一下 RMW 指令的执行流程：先读取 PORTx，修改数据，写 PORTx (LATx))时，由于同步的原因，PORTx_sync 还保持为 0，那么写回 PORTx 时刻，这一位的“0”又被写回到 LATx，导致管脚 PORTx.n 只有一个高脉冲。

有以下两种方式解决这一问题：

- a) 在 PORTx 连续写操作中间插入一个 NOP；

BSR PORTx, n ; 对 PORTx 第 n 位置 1

NOP ; 插入 NOP 等待

BSR PORTx, m ; 对 PORTx 第 m 位置 1

- b) 或者，写操作用 LATx 寄存器而不是 PORTx；

BSR LATx, n ; 直接操作端口数据寄存器 LATx

BSR LATx, m ; 直接操作端口数据寄存器 LATx

注意：只有 1T 速度模式下才有该现象，不存在于其它 2T/4T 模式，原因是处于 2T/4T 模式下，执行后续指令时，PORTx_sync 已经同步到最新的值。

但如果 I/O 直接驱动 LED 或三极管，或者其它会导致 I/O 被拉低的电路时，无论是在何种速度模式下，软件读到的 PORTx 值将是 0，这种情况下，对 I/O 数据寄存器只能使用 LATx，而不能是 PORTx。

2. 应用范例

```
//*****
/* 文件名: TEST_62F08x_IO_INTERRUPT.c
* 功能:    FT62F08x_IO_INTERRUPT 功能演示
* IC:      FT62F088 LQFP32
* 内部:    16M
* empno:   500
* 说明:    外部 RC3 引脚电平变化中断使 RB3 输出 0.75μS 脉冲
*
* 参考原理图 TEST_62F08x_sch.pdf
*/
//*****
#include "SYSCFG.h"
//*****宏定义*****
#define DemoPortOut  PB3
/*-----
* 函数名: interrupt ISR
* 功能:    中断处理函数
* 输入:    无
* 输出:    无
-----*/
void interrupt ISR(void)
{
    if(EPIF0 & 0x08)
    {
        EPIF0 = 0x08;          //写 1 清零标志位

        DemoPortOut=1;
        NOP();
        NOP();
        NOP();
        NOP();
        NOP();
        DemoPortOut=0;
    }
}
/*-----
* 函数名: POWER_INITIAL
* 功能:    上电系统初始化
* 输入:    无
* 输出:    无
-----*/
void POWER_INITIAL (void)
{
```

```

OSCCON = 0B01110001;    //IRCF=111=16MHz 1:1
INTCON = 0;              //暂禁止所有中断

PORTA = 0B00000000;
TRISA = 0B00000000;      //PA 输入输出 0-输出 1-输入
PORTB = 0B00000000;
TRISB = 0B00000000;      //PB 输入输出 0-输出 1-输入  PB3 输出
PORTC = 0B00000000;
TRISC = 0B00001000;      //PC 输入输出 0-输出 1-输入  PC3 输入
PORTD = 0B00000000;
TRISD = 0B00000000;      //PD 输入输出 0-输出 1-输入

WPUA = 0B00000000;      //PA 端口上拉控制 1-开上拉 0-关上拉
WPUB = 0B00000000;      //PB 端口上拉控制 1-开上拉 0-关上拉
WPUC = 0B00001000;      //PC 端口上拉控制 1-开上拉 0-关上拉
WPUD = 0B00000000;      //PD 端口上拉控制 1-开上拉 0-关上拉

WPDA = 0B00000000;      //PA 端口下拉控制 1-开下拉 0-关下拉
WPDB = 0B00000000;      //PB 端口下拉控制 1-开下拉 0-关下拉
WPDC = 0B00000000;      //PC 端口下拉控制 1-开下拉 0-关下拉
WPDD = 0B00000000;      //PD 端口下拉控制 1-开下拉 0-关下拉

PSRC0 = 0B11111111;      //PORTA,PORTB 源电流设置最大
PSRC1 = 0B11111111;      //PORTC,PORTD 源电流设置最大

PSINK0 = 0B11111111;     //PORTA 灌电流设置最大 0:最小, 1:最大
PSINK1 = 0B11111111;     //PORTB 灌电流设置最大 0:最小, 1:最大
PSINK2 = 0B11111111;     //PORTC 灌电流设置最大 0:最小, 1:最大
PSINK3 = 0B11111111;     //PORTD 灌电流设置最大 0:最小, 1:最大

ANSELA = 0B00000000;     //全为数字管脚
}
/*-----
* 函数名: IO_INT_INITIAL
* 功能:   初始化 IO 口中断
* 输入:   无
* 输出:   无
-----*/
void IO_INT_INITIAL(void)
{
    EPS0=0B10000000;       //选择 PC3 管脚中断
    //Bit[7:0]:Px[3:0]中断管脚选择
    EPS1=0B00000000;
    //Bit[7:0]:Px[7:4]中断管脚选择

```

```
    ITYPE0 = 0B11000000;    //双沿中断
    ITYPE1 = 0B00000000;

    EPIE0 = 0B00001000;    //使能中断 3

    INTCON = 0B11000000;
}
/*-----
 * 函数名: main
 * 功能:   主函数
 * 输入:   无
 * 输出:   无
-----*/
void main(void)
{
    POWER_INITIAL();    //系统初始化
    IO_INT_INITIAL();

    while(1)
    {
        NOP();
    }
}
```

联系信息

Fremont Micro Devices Corporation

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.