Sungine

GC7810A

技术手册

2020

Sungine Science Limited

Ver 1.0

八位多功能微控制器

1 概述

GC7810A 是一种多功能八位微控制器，其指令系统与 PIC16F87X 兼容，内部集成 4K 字节现场可编程的程序空间，256 字节的 SRAM。接口部分包括 SPI、UART 和 GPIO 之外，内部还集成了八位 AD，两个模拟比较器，一个跨阻放大器，以及可用于温度监控的 R2FC 模块。

1.1 特点

◆高性能的 RISC 单片机（与 PIC16F87X 兼容）
◆仅 35 个单字指令
◆16bit*4K 的 MTP，无需外部编程电源
◆256 字节的 SRAM
◆8 级深层硬件堆栈
◆直接，间接和相对寻址模式
◆看门狗定时器，具有可选择性 8 位预分频器（WDT）
◆Timer0：8 位定时器，带自动重载和 8 位预分频器
◆Timer1：16 位定时器，带 3 位预分频器
◆UART 接口（通用异步接收/发送）
◆高速 SPI 接口（Master），支持 3 线/4 线模式
◆波特率可通过编程的方式选择（1200bps ～256000bps）
◆8 位通用的 I/O 端口（A,B）
◆通过 4 个引脚进行在线编程、调试和测试
◆功率消耗：

500uA(1MHz) ~ 6mA(16MHz)
15 uA 睡眠电流
1 uA 看门狗
100nA 深度睡眠模式电流

◆可编程检测阈值电压选择（0.5～2.0V）
◆可编程选择 0.2MHz~4MHz 的内部 RC 谐振器，低频和高频振荡器/谐振器，外部时钟输入（最大=16MHz）
◆8 位 SAR A/D 转换器，三通道可编程外部输入
◆两个模拟比较器，可编程选择内部/外部参考电压（0.5~2.0V），可选择比较器输入输出
◆集成 TIA（跨阻放大器）可用于单片烟雾探测器
◆集成内部温度监测器（-40 - 85℃）
◆集成 R2FC 功能用于温度，湿度测量

◆24 脚封装

1.2 应用范围

◆烟雾报警器

◆温度检测模块

◆KEELOQ 发射机和接收机

◆其他一些简单的应用

1.3 电气特性

参数	符号	测试条件	最小	典型	最大	单位
工作电压	VDD	正常工作	+2.5	+3.0	+3.6	V
MTP 编程电压	VDD	编程模式	-+2.5	+3.0	-+3.6	V
振荡频率	XOSC	晶体	-	4	16	MHz
工作电流	I_{DD}	工作频率 4MHz	-	1.25	2.0	mA
休眠电流	I_{SLEEP}	4MHz @3V，内部 RC	-	40	60	uA
休眠电流	I_{SLEEP}	4MHz @3V，外部晶振	-	15	20	uA
储藏温度	T_A		-55	25	+125	℃
工作温度	T_M		-40	-	+80	℃

1.4 管脚图

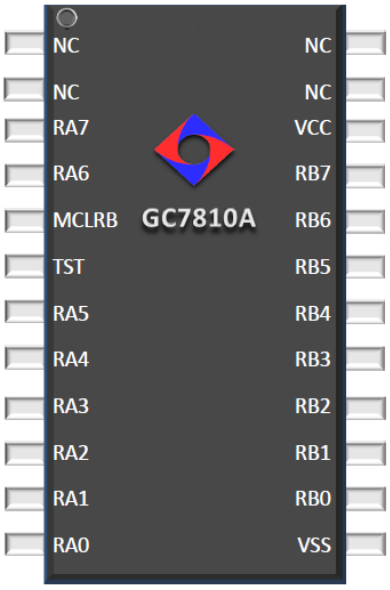


图 1：GC7810A 管脚图

1.5 管脚说明

表 1：GC7810A 管脚描述

管脚	名称	输入/输出	描述
3	**RA7**	I/O	Port A
	(XI)	I/O	晶振输入
4	**RA6**	I/O	Port A
	(XO)	/O	晶振输出
5	**MCLRB**	I	复位输入（低电平有效）
6	**TST**	I	测试模式设置输入（低电平有效）
7	**RA5**	I/O	Port A
	(CIN)	I/O	电容器连接端口用于火灾报警
	(AIN2)	I	模拟输入
8	**RA4**	I/O	Port A
	(IRED)	I/O	红外二极管连接端口用于火灾报警
	(AIN1)	I	模拟输入
9	**RA3**	I/O	Port A
	(AIN0)	I	模拟输入
10	**RA2**	I/O	Port A
	(TSM)	I	测试模式的模式设置输入
	(XREFH)	I	参考高电平设置输入
11	**RA1**	I/O	Port A
	(TSK)	I	测试模式串行时钟输入
	(XREFL)	I	参考低电平设置输入
12	**RA0**	I/O	Port A
	(TDIO)	I/O	测试模式串行数据输入
1,2 23,24	**NC**		
13	**VSS**	I	地
14	**RB0**	I/O	Port B
	(TX)	O	UART 接口的发送输出
15	**RB1**	I/O	Port B
	(RX)	I	UART 接口的接收输入
16	**RB2**	I/O	Port B
	(CSN)	O	SPI 接口芯片选择输出
	(RTO)	O	温度电阻器通道产生的输出
17	**RB3**	I/O	Port B
	(SCK)	O	SPI 接口的串行时钟输出
	(RFO)	O	参考电阻器通道产生的输出
18	**RB4**	I/O	Port B
	(SDO)	O	SPI 接口的串行数据输出
19	**RB5**	I/O	Port B

	(SDI)	I	SPI 接口的串行数据输入
	(RHO)	O	湿度电阻器通道产生的输出
20	RB6	I/O	Port B
	(CXO)	O	RC 振荡器输出用于 R2FC
21	RB7	I/O	Port B
	(CXI)	I	RC 振荡器输入用于 R2FC
22	VCC	I	3V 电源引脚

1.6 系统框图

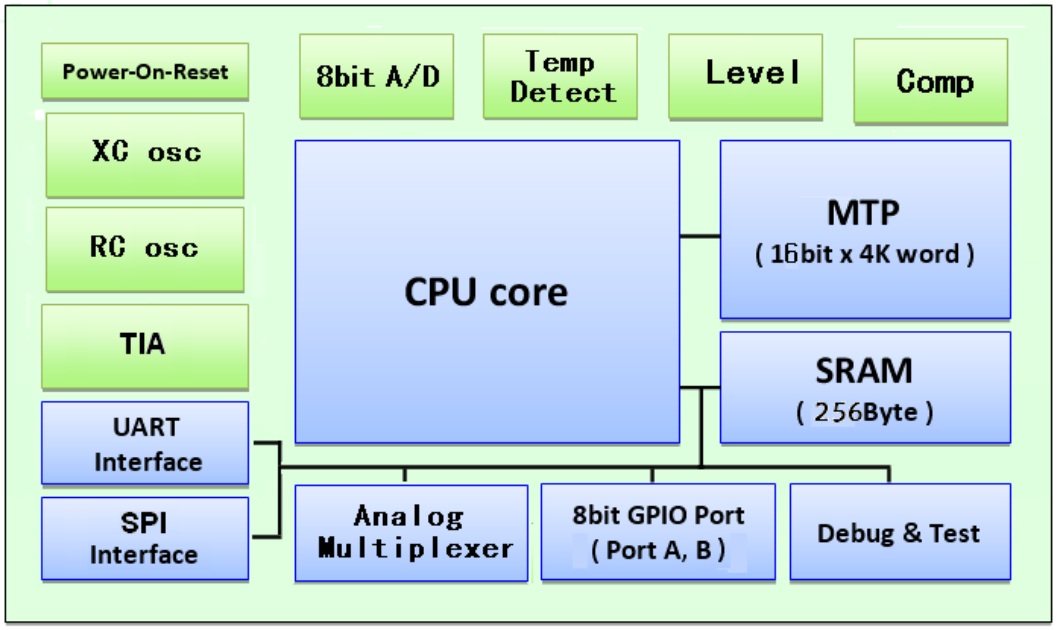


图 2：GC7810A 系统架构

1.7 CPU 结构框图

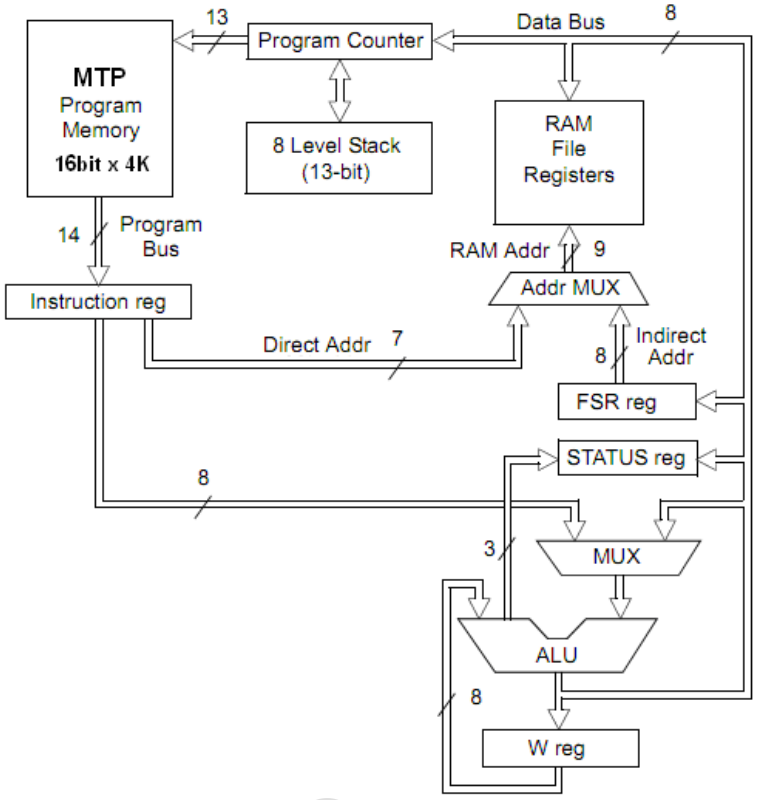


图 3：GC7810A 的 CPU 框图

2 存储器组织架构

GC7810A 包括程序存储器（MTP）和数据存储器两部分，由于采用了两条单独的总线，可以同时读取数据和指令。

2.1 程序存储器架构

GC7810A 有一个 13 位的程序计数器，可以寻址 4K×16 的程序存储器空间和 8 级深层硬件堆栈。复位地址为 0000H，中断向量地址在 0004h。

程序存储器和堆栈架构图：

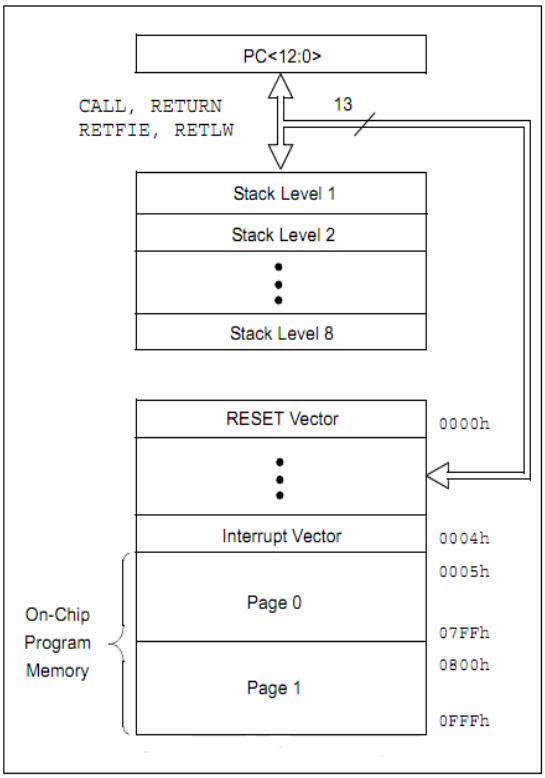


图 4：程序存储器和堆栈架构图

2.2 数据存储器架构

数据存储器分为四块，通过 RP1（STATUS,6）和 RP0(STATUS, 5)两位来选择。每块都有 128 个字节，都有普通功能寄存器和特殊功能寄存器。每个块的较低的位置是特殊功能寄存器，然后是通用寄存器，通用存储器可以用作静态 RAM。一些经常使用的特殊功能寄存器可以从一个块可以映射到另一个块，以利于减少代码和更快的访问。

RP1 : RP0	Bank
0 0	Bank0
0 1	Bank1
1 0	Bank2
11	Bank3

2.2.1 通用功能寄存器

通用功能寄存器可以进行直接或间接的访问，使用寄存器 FSR 进行选择。

寄存器名称	地址	寄存器名称	地址	寄存器名称	地址	寄存器名称	地址
INDF	000h	INDF	080h	INDF	100h	INDF	180h
Timer0	001h	OPTION	081h	Timer0	101h	OPTION	181h
PCL	002h	PCL	082h	PCL	102h	PCL	182h
STATUS	003h	STATUS	083h	STATUS	103h	STATUS	183h
FSR	004h	FSR	084h	FSR	104h	FSR	184h
PORTA	005h	TRISA	085h	PULLA	105h	PORTAE	185h
PORTB	006h	TRISB	086h	PULLB	106h		186h
ADRES	007h	CALOSC	087h	OSCCTR	107h	XCONF	187h
ANACON	008h	ANACON1	088h	NET	108h	ANACON2	188h
PERCON	009h	CALREF	089h	E2CON	109h	ANACON3	189h
PCLATH	00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh
PIF1	00Ch	PIE1	08Ch	E2AL	10Ch	USBUF	18Ch
SPID	00Dh	REFR	08Dh	E2AH	10Dh	UBRL	18Dh
SPIC	00Eh	TMR1H	08Eh	E2DL	10Eh	UBRM	18Eh
TMR1L	00Fh	TMR1C	08Fh	E2DH	10Fh	UCTR	18Fh
R2FC	010h	通用寄存器 96 字节	090h	通用寄存器 48 字节	110h	直接访问 000h-00Fh 080h-08Fh 0B0h-0BFh	190h
R2DIVL	011h						
R2DIVH	012h						
R2CNTSL	013h						
R2CNTSH	014h						
通用寄存器 107 字节	015h				13Fh		1BFh
				直接访问 040h-06Fh	140h	直接访问 0C0h-0EFh	1C0h
			0EFh		16Fh		1EFh
		Accesses 070h-07Fh	0F0h	Accesses 070h-07Fh	170h	Accesses 070h-07Fh	1F0h
	07Fh		0FFh		17Fh		1FFh
BANK0		BANK1		BANK2		BANK3	

图 5：GC7810A 寄存器图表

2.2.2 特殊功能寄存器

特殊功能寄存器是控制 CPU 和外围模块所使用的寄存器，这些寄存器被实现为静态的 RAM，分为核心和外设两部分，特殊功能寄存器列表如下：

地址	寄存器名称	位								默认值
		bit7	bit6	bit5	Bit4	bit3	bit2	bit1	Bit0	
000, 080, 100, 180	INDF	Indirect Address Register (Non physical)								xxxxxxxx
001h, 0101h	TMR0	Timer0 Register (Write Only)								00000000
002, 082, 102, 182	PCL	Program Counter Low Byte								00000000
003, 083, 103, 183	STATUS	IRP	RP1	RP0	TON	PDN	Z	DC	C	00011000
004, 084, 104, 184	FSR	Indirect Address Register Pointer								xxxxxxxx
005h	PORTA	Port A Data Register								
006h	PORTB	Port B Data Register								
007h	ADREG	AD Catch Register								
008h	ANACON	REFLE	REFHE	REFGE	OPSMP	OPE	CMP2E	CMP1E	ADCE	00000000
009h	PERCON	CMP2Y	CMP1Y	ADRDY	R2EN	LVDE	-SPIEN	T0E	UEN	00000000
00A, 08A, 10A, 18A	PCLATH	--	--	--	--	PC upper 4 bits Write Buffer				
00B, 08B, 10B, 18B	INTCON	GIE	PEIE	T0IE	RTCIE	RPIE	T0IF	RTCIF	RPIF	00000xxx
00Ch	PIF1	---	R2FIF	CMP2IF	CMP1IF	ADCIF	T1IF	TXIF	RXIF	
00Dh	SPIBUF	SPI Interface(Master) Data Register (Buffer)								
00Eh	SPICON			RDY	CSN	CKPH	W3MD	RCMD	WCMD	xx011000
00Fh	TMR1L	Timer1 Register Low Byte (Write Only)								
010h	R2FC	MEAS_EN	MEAS_MODE	MEAS_DONE	-	-	-	SENCHN_SEL		000xxx00
011h	R2DIVL/ R2CNTRL	R2F Counter Set Register Low Byte (when Wirte) / Internal 32KHz Reference Clock Counter Low Byte (when Read)								00000000
012h	R2DIVH/	R2F Conter Set Register High Byte (when Write) /								00000000

	R2CNTRH	Internal 32KHz Reference Clock Counter High Byte (when Read)								
013h	R2CNTSL	Temperature Sensor Clock Counter Low Byte (when Read)								xxxxxxxx
014h	R2CNTSH	Temperature Sensor Clock Counter High Byte (when Read)								xxxxxxxx
081h, 181h	OPTION	WDTE	ATRLD	FOUTE	---	PSA	PS2	PS1	PS0	01001111
085h	TRISA	Port A Direction Control Register								11111111
086h	TRISB	Port B Direction Control Register								11111111
087h	CALOSC	Internal RC OSC Calibration Register								00000000
088h	ANACON1	ATEST	TEMPE	CMP2OE	CMP1OE	CMP2POL	CMP1POL	ADCRATE[1:0]		10110000
089h	CALREF	Analog Reference Calibration Register								00000000
08Ch	PIE1	---	R2FIE	CMP2IE	CMP1IE	ADCIE	T1IE	TXIE	RXIE	00000000
08Dh	REFR	Analog Reference Regulation Register								10010111
08Eh	TMR1H	Timer1 Register High Byte (Write Only)								00000000
08Fh	T1CON	---	---	FSEL1	FSEL0	T1E	T1CS2	T1CS1	T1CS0	Xxx00000
105h	PULLA	Port A Pullup Control Register								00000000
106h	PULLB	Port B Pullup Control Register								00000000
107h	OSCCTR	External Crystal Control Register								10000000
108h	NET	AnalogFE Switch Network Setup Register								10100000
109h	E2CON	--	--	--	--	E2WERR	E2WE	E2WR	E2RD	xxxx0000
10Ch	E2AL	E2PROM R/W Address Low Byte Register								
10Dh	E2AH	E2PROM R/W Address High Byte Register								
10Eh	E2DL	E2PROM R/W Data Low Byte Register								
10Fh	E2DH	E2PROM R/W Data High Byte Register								
185h	PORTAE	Port A Analog Enable Register								00111111
186h										
187h	XCONF	RXSEL	EXTN	XCEN	RCENB	XCYC1	XCYC0	XCTERN	DPSLP	00100000
188h	ANACON2	DSL PDY	SLP DY1	SLP DY0	XKSEL1	XKSEL0	ADCK S2	ADCK1	ADCK0	00000000
189h	ANACON3							CMPCK1	CMPC0	00000000
18Ch	UABUF	UART Data Register								
18Dh	UBRL	UART Baud Rate Set Register Low Byte (Write Only)								01101101
18Eh	UBRM	UART Baud Rate Control Register (Write Only)								00000011
18Fh	UCTR	---	---	---	TXRDY			UBRH1	UBRH0	xxxxxx00

- STATUS 寄存器

(地址 03h, 83h, 103h, 183h)复位值 : 0001_1000

状态寄存器包含 ALU、RESET 和数据存储器的存储区选择位，状态寄存器可以是任何指令的目的地址，与其他寄存器一样。如果指令影响到了状态寄存器的 Z、DC 或 C 状态位，则对这三位的写入将被禁用，这些位被设置或清除，是根据设备本身的逻辑。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IRP	RP1	RP0	TON	PDN	Z	DC	C
w/r	w/r	w/r	/r	/r	w/r	w/r	w/r

IRP： 存储区选择位（用于间接寻址）

0： bank0，1（00H—FFH）（默认）

1： bank2（100 H－17fh）

RP1, RP0： 存储区选择位（用于直接寻址），每一块是 128 个字节。

00： bank0 (00 – 7Fh) (默认)

01： bank1 (80 – FFh)

10： bank2 (100 – 17Fh)

11： bank3 (180 – 1FFh)

TON： 超时状态位

0： WDT 超时发生

1： 上电后，执行了 CLRWDT 指令或睡眠指令（默认）

PDN： 断电状态位

0： 执行睡眠指令

1： 上电或执行 CLRWDT 指令

Z :零状态位

0： 算术或逻辑运算的结果不是零

1： 算术或逻辑运算的结果为零

DC： 数字进位/ 借位状态位（ADDWF, ADDLW, SUBWF, SUBLW 指令）（对借位来说，极性是相反的）

0： 没有执行从第四低阶位的结果

1： 执行从第四低阶位发生的结果

C： 进位/ 借位状态位（ADDWF, ADDLW, SUBWF, SUBLW 指令）

0： 没有进行从最有效的位发生的结果

1： 执行从最有效的位发生的结果

- OPTION 寄存器

(地址 081h,0181h)复位值 ：0100_1111

OPTION 寄存器是可读写的，其中包含用于配置 TMR0 预分频器/WDT 后分频器的各种控制位 WatchDog 定时器使能的时钟频率选择。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
WDTE	ATRLD	FOUTE		PSA	PS2	PS1	PS0

w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

WDTE :看门狗定时器使能位

0 :禁止 (默认)

1 :使能

ATRLD :定时器 0 自动重载使能位

0 :禁止

1 :使能(默认)

FOUTE :FCLK 输出使能位

0 :禁止 (默认)

1 :使能

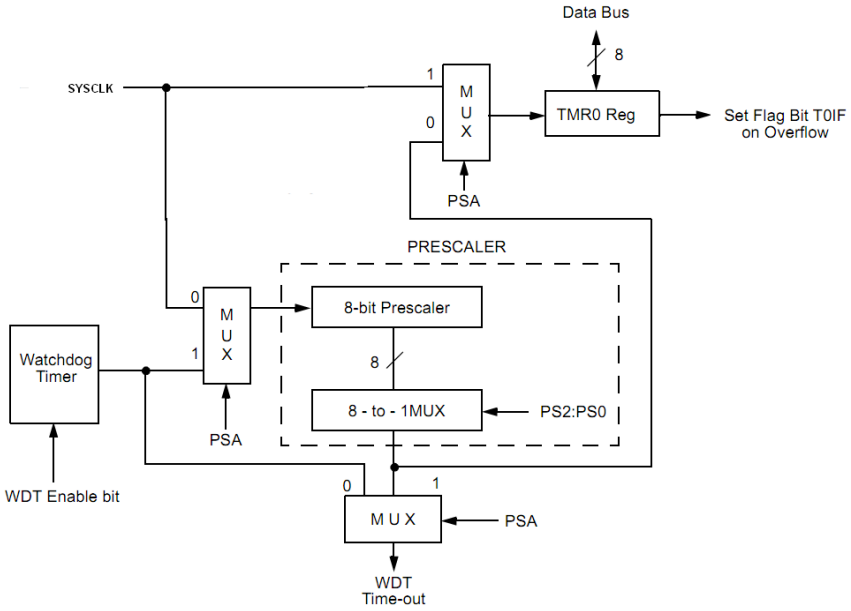
PSA :分频器配置位

0 :分频器分配给定时器 0

1： 分频器分配给看门狗 (默认)

PS2, PS1, PS0 :分频器分频比配置位

PS2, PS1, PS0	TMR0	WDT
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111 (默认)	1 : 256	1 : 128



- INTCON 寄存器

(地址 0Bh, 8Bh, 10Bh, 18Bh)复位值 ：0000_0000

INTCON 寄存器是一个可读写的寄存器，其中包含各种使能和标志位用于 TMR0 寄存器溢出位，RTC,RB 的更改。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
GIE	PEIE	T0IE	RTCIE	RBIE	T0IF	RTCIF	RBIF
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

GIE： 所有中断使能引脚

0：禁用所有的中断（默认）

1：使能所有非屏蔽中断

PEIE： 外围中断允许位

0：禁用所有外围中断（默认）

1：使能所有非屏蔽外围中断

T0IE：TMR0 溢出中断使能位

0： 禁用 TMR0 中断（默认）

1：允许

RTCIE：1 秒的中断使能位

0：禁用(默认)

1：允许

RBIE：RB 端口变化中断使能位

0：禁用 RB 端口变化中断（默认）

1：启用 RB 端口变化中断

T0IF：TMR0 溢出中断标志位

0： TMR0 寄存器没有溢出

1： TMR0 寄存器溢出(必须由软件清零)

RTCIF： 1 秒的中断标志位

0 :RTC 中断没有发生

1 :RTC 中断发生了

RBIF : RB 端口变化中断标志位

0 :RB 端口没有发生中断变化

1 : RB 端口发生了电平变化中断

- PIE1 寄存器

(地址 08Ch)复位值: x000_0000

外围中断使能寄存器。注意：INTCON 寄存器中的第 6 位 PEIE 必须置 1，外围中断控制位才能起作用。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
---	R2FIE	CMP2IE	CMP1IE	ADCIE	T1IE	TXIE	RXIE
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

R2FIE： R2FC 使能位

0： 禁用（默认）

1： 使能

CMP2IE :comparator2 使能位

0 :禁用（默认）

1 :使能

CMP1IE :comparator1 使能位

0 :禁用（默认）

1 :使能

ADCIE :ADC 模块使能位

0 :禁用（默认）

1 :使能

T1IE :定时器 1 中断使能位

0 :禁用（默认）

1 :使能

TXIE :UART 发送中断允许位

0 :禁用 UART TX 中断（默认）

1 :使能 UART TX 中断

RXIE :UART 接收中断使能位

0 :禁用 UART 的 RX 中断（默认）

1 :使能 UART 接收中断

-PIF1 寄存器

(地址 0Ch)复位值: 0000_0000

PIF1 寄存器包含外围模块的中断标志。

注意：只要中断条件发生，中断标志位就会置位，跟其相应的中断使能位和全局中断使能位的状态无关。在启用中断之前，用户软件应该确保适当的中断标志位清零。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
---	R2FIF	CMP2IF	CMP1IF	ADCIF	T1IF	TXIF	RXIF
---	w/r	w/r	w/r	w/r	w/r	w/r	w/r

R2FIF：R2FC 检测中断标志位

0：没有检测到事件发生（默认）

1：检测到事件发生

CMP2IF :comparator2 检测中断标志位

0 :没有检测到事件发生 (默认)

1 :检测到事件发生

CMP1IF :comparator1 检测中断标志位

0 :没有检测到事件发生 (默认)

1 :检测到事件发生

ADCIF :ADC 模块使能中断标志位

0 :无中断（默认）

1 :中断发生

T1IF :TMR1 溢出中断标志位

0 :TMR1 寄存器不溢出（默认）

1 :TMR1 寄存器已经溢出

TXIF :UART 发送中断标志位

0 :UART TX 中断没有发生（默认）

1 :UART 的 TX 中断发生

RXIF :UART 接收中断标志位

0 :UART 的 RX 中断没有发生（默认）

1 :UART 接收中断发生

- PERCON 寄存器

(地址 009h)复位值: xxx0_0000

PERCON 寄存器包含端口 A，端口 C，TMR0，USB，LCD，R2FC，UART 模块使能位

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
C2RDY	C1RDY	ADCRDY	R2EN	LVDE	SPIEN	T0N	UEN
r	r	r	w/r	w/r	w/r	w/r	w/r

C2RDY :comparator2 状态位

0 :加载中

1 :完成

C1RDY :comparator1 状态位

0 :加载中

1 :完成

ADCRDY :ADC 状态位

0 :加载中

1 :完成

R2EN：R2FC 模块使能位

0：禁止（默认）

1：使能

LVDE :低电压检测模块使能位

0 :禁用（默认）

1 :使能

SPIEN :SPI 模块使能位

0 :禁用（默认）

1 :使能

T0E :Timer0 使能位

0 :禁用（默认）

1 :使能

UEN : UART 模块使能位

0 :禁用（默认）

1 :使能

- ANACON 寄存器

(地址 08h)复位值: 0000_0000

ANACON 寄存器是可读写的寄存器，它包含模拟块的各种控制位。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
REFLE	REFHE	REFGE	OPSMP	OPE	CMP2E	CMP1E	ADCE
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

REFLE :参考电压低端缓冲器的使能位

0 :禁用（默认）

1 :使能

REFHE :基准电压高端缓冲使能位

0 :禁用（默认）

1 :使能

REFGE:参考电流产生使能位

0 :禁用（默认）

1 :使能

OPSMP :TIA 采样使能位

0 :禁用（默认）

1 :使能

OPE:TIA 接收使能位

0 : 禁用（默认）

1 :使能

CMP1.2E :comparator1,2 使能位

0 : 禁用（默认）

1 :使能

ADCE:ADC 使能位

0 :禁用（默认）

1 :使能

-ANACON1 寄存器

(地址 088h)复位值: 1011_0000

ANACON1 寄存器是可读写的寄存器，它包含模拟块的各种控制位。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ATEST	TEMPE	CMP2OE	CMP1OE	CMP2POL	CMP1POL	ADRATE1	ADRATE0
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

ATEST :SAR-ADC 逻辑电路检查位

0 : 连接

1 : 断开（默认）

TEMPE :温度传感器电路使能位

0 : 禁用（默认）

1 : 使能

CMP1,2OE:比较器结果输出状态位

0 :可用 (默认)

1 :不可用

CMP1,2POL :比较器相位翻转位

0 :反相 (默认)

1 :不反相

ADRATE1,0 :AD 转换的采样率选择位

00 :fs=fadclk/128

01 :fs=fadclk/ 64

10: fs=fadclk/ 32

11: fs=fadclk/ 16

-ANACON2 寄存器

(地址 188h)复位值: 0000_0000

ANACON2 寄存器是可读写的寄存器，它包含模拟块的各种控制位。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DSLPDY	SLPDLY1	SLPDLY0	XRSEL1	XRSEL0	ADCKS2	ADCKS1	ADCKS0
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

DSLPDY：RC OSC 深度睡眠 wake_up 时间选择位

0：64Hz

1：32KHz

SLPDLY[1:0]：睡眠 wake_up 时间选择位

00：2KHz

01：256Hz

10：32KHz

11：32KHz

XRSEL[1:0]：晶振选择位

00：4MHz

01：1MHz

10：8MHz

11：12~16MHz

ADCKS[2:0]:AD 转换时钟选择位

00 :2MHz(RC:512k)

01 :1MHz(256k)

10: 32KHz(8k)

11: 4MHz(1M)

-ANACON3 寄存器

（地址 189h）复位值：xxxx_xx00

ANACON2 寄存器是可读写的寄存器，它包含模拟块的各种控制位。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	-	-	-	-	CMPCK1	CMPCK0
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

CMPCK[1:0]： 比较器时钟选择位

00 :32KHz

01 :2KHz

10: 64Hz

11: 1MHz

- NET 寄存器

(地址 108h)复位值: 1010_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
NET							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

Bit7:IREF 输出允许位

0 : 使能

1 : 禁用（默认）

Bit<6:5>:比较器 2 输入选择位

00 :AIN0 连接到比较器

01 :AIN1 连接到比较器

10:低电压检测连接到比较器

11:XREFH 连接到比较器

Bit4:comparator2 阈值/ADC 低电平选择位

0 :内部低门槛电压

1 :外部电压

Bit3: Comparator1 阈值/ADC 高电平选择位

0 :选择内部高门槛电压

1 :外部电压

Bit<2:0>: Comparator1 和 ADC 的输入选择位

000 :选择 AIN0

001 :选择 AIN1

010： 选择 AIN2

011:选择 VDD

100 :选择 NTAT 传感器输出信号.

101 :选择 TIA 输出信号.

110： 选择 REFL

111:选择 REFH

- REFR 寄存器

(地址 08Dh)复位值: 1001_0111

REFR 是一个可读写的寄存器，它包含参考电压选项寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
REFR							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

Bit<7:4>:VREFH 选择位 **Bit<3:0>:**VREFL 选择位

0000 :1.062V	**0000** :0.562V
0001 :1.125V	**0001** :0.625V
0010 : 1.187V	**0010** : 0.687V
0011 : 1.250V	**0011** : 0.750V
0100 :1.312V	**0100** :0.812V
0101 :1.375V	**0101** :0.875V
0110 : 1.437V	**0110** : 0.937V
0111 : 1.500V	**0111** : 1.000V
1000 :1.062V	**1000** :1.062V
1001 :1.625V	**1001** :1.125V
1010 : 1.687V	**1010** : 1.187V
1011 : 1.750V	**1011** : 1.250V
1100 :1.812V	**1100** : 1.312V
1101 :1.875V	**1101** : 1.375V
1110 : 1.937V	**1110** : 1.437V
1111: 2.000V	**1111** : 1.500V

- CALR 寄存器

(地址 089h)复位值: 0000_0000

CALR 是一个可读写的寄存器，它包含参考电压生成选项寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CALR							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

Bit<6:3>:ITAT 带隙基准电压校准位 **Bit<2:0>:**ITAT 电流校准位

VOUT/VBG	**VREFT/IPNP(KOhm)**
0000 :0.3856	**000** :482.9
0001 :0.3799	**001** :442.1
0010 : 0.3742	**010** : 401.3
0011 : 0.3685	**011** : 360.5
0100 :0.3629	**100** :523.7
0101 :0.3572	**101** :564.5

0110 : 0.3515
0111: 0.3459
1000 :0.3912
1001 :0.3969
1010 : 0.4026
1011 : 0.4082
1100 :0.4139
1101 :0.4192
1110 : 0.4252
1111: 0.4309

110 : 605.2
111: 645.6

-XCN 寄存器

（地址 187h）复位值：0010_0000

XCN 寄存器是一个可读写的寄存器，它包含 XOSC 控制寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RXSEL	EXTN	XCEN	RCENB	XCYCS1	XCYCS0	XCTREN	DPSLP
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

Bit7:RXSEL：RC/XC 选择位

0：RC（内部 RC 振荡器）

1：XC（外部晶振）

Bit6: EXTN：外部时钟模式选择位

0：选择内部时钟（默认）

1：选择外部时钟

Bit5: XCEN：XOSC 使能位

0：禁用

1：使能（默认）

Bit4: RCENB：RCOSC 使能位

0：使能（默认）

1：禁用

Bit3,2: XCYCS1,XCYCS0：XOSC 输出停止时间选择位

00：4ms（默认）

01：16ms

10：125ms

11：1s

Bit1: XCTREN ： XOSC 输出停止时间使能位

0：禁用（默认）

1：使能

Bit0: DPSLP：深度睡眠使能位

0：禁用（默认）

1：使能

-OSCCTR 寄存器

(地址 107h) 复位值：1000_0000

OSCCTR 寄存器是一个可读写的寄存器，其中包含 XOSC 电流控制选项寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OSCCTR							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

注册值=80h,40h,20h,10h,08h,04h,02h,01h,00h.

I_{max}=80h,→I_{min}=00h

-CALOSC 寄存器

（地址 087h）复位值：0000_0000

CALOSC 寄存器是一个可读写的寄存器，它内部包含 RCOSC 频率控制校准器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CALOSC							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

00h=最高频率，7Fh=最低频率

-R2FC 寄存器

（地址 010h）复位值：000x_xx00

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MEAS_EN	MEAS_MODE	MEAS_DONE				SENCHN_SEL	
w/r	w/r	/r				w/r	w/r

Bit7:MEAS_EN：温度测量使能位

0：未开始（默认）

1：开始

Bit6:MEAS_MODE：温度测量模式设置位

0：模式 0（默认）

1：模式 1

Bit5:MEAS_DONE：温度测量完成检测位

0：温度测量未完成（默认）

1：温度测量完成

Bit1,0:SENCHN_SEL：温度传感器通道选择位

00：10K 参考电阻 RF（默认）

01：温度传感器 RT

10：温度电阻 RH

11：没有影响

-R2DIVL 寄存器

（地址 011h）复位值：0000_0000

R2DIVL 是一个只可写寄存器，包含计数低字节值设置寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2DIVL							
w	w	w	w	w	w	w	w

-R2DIVH 寄存器

（地址 012h）复位值：0000_0000

R2DIVH 是一个只可写寄存器，包含计数高字节值设置寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2DIVH							
w	w	w	w	w	w	w	w

-R2CNTRL 寄存器

（地址 011h）复位值：0000_0000

R2DIVL 是一个只读寄存器，包含 32KHz 参考频率读取寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2CNTRL							
r	r	r	r	r	r	r	r

-R2CNTRH 寄存器

（地址 011h）复位值：0000_0000

R2DIVL 是一个只读寄存器，包含 32KHz 参考频率读取寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2CNTRH							
r	r	r	r	r	r	r	r

-R2CNTSL 寄存器

（地址 013h）复位值：0000_0000

R2DIVL 是一个只读寄存器，包含传感器频率计数低字节值寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2CNTSL							
r	r	r	r	r	r	r	r

-R2CNTSH 寄存器

（地址 013h）复位值：0000_0000

R2DIVL 是一个只读寄存器，包含传感器频率计数高字节值寄存器。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
R2CNTSH							
r	r	r	r	r	r	r	r

-USBUF 寄存器

（地址 018Ch）复位值：0000_0000

USBUF 寄存器是可读写的日期缓冲区，其中包含在接收和发送期间要发送和接收的数据。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
USBUF							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

-UBRL 寄存器

（地址 018Dh）复位值：0110_1101

UBRL 寄存器用于设置波特率的积分分割因子的低字节（只写）。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
UBRL							
w/	w/	w/	w/	w/	w/	w/	w/

-UBRM 寄存器

（地址 018Eh）复位值：0000_0011

UBRM 寄存器用于设置波特率的分数因子（只写）。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
UBRM							
w/	w/	w/	w/	w/	w/	w/	w/

-UCTR 寄存器

（地址 018Fh）复位值：xxx0_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
---	---	---	TXRDY	---	---	UBRH1	UBRH0
---	---	---	/r	w/r	w/r	w/r	w/r

Bit4:TXRDY：UART 发送就绪位（只读）

0：发送未就绪（默认）

1：发送就绪

Bit1,0:UBRH1,UBRH0：整体分频系数高字节波特率

-TMR1C 寄存器

（地址 08Fh）复位值：xx00_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

---	---	FSEL1	FSEL0	T1E	T1CS2	T1CS1	T1CS0
---	---	w/r	/r	w/r	w/r	w/r	w/r

Bit5,4:FSEL1,FSEL0：FOUT 输出频率选择位

00：1Hz

01：8Hz

10：2KHz

11：64Hz

Bit3:T1E：定时器 1 使能位

0：禁用（默认）

1：使能

Bit2,1,0:T1CS2,T1CS1,T1CS0：定时器 1 时钟频率选择位

000：1Hz

001：8Hz

010：64Hz

011：256Hz

100：2KHz

101：32KHz

110：1MHz

111：2MHz

-TIMR1H 寄存器

（地址 08Eh）复位值：0000_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TIM1H							
w/	w/	w/	w/	w/	w/	w/	w/

-TIMER1L 寄存器

（地址 00Fh）复位值：0000_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TIM1L							
w/	w/	w/	w/	w/	w/	w/	w/

TIMER1 是 16 位计数器，时钟源为 4MHz，没有自动加载功能。

-TIMER0 寄存器

（地址 001h，101h）复位值：0000_0000

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TIMER0							

w/	w/	w/	w/	w/	w/	w/	w/

-PORTAE 寄存器

（地址 185h）复位值：0011_1111

TRISA 寄存器控制 PORTA 引脚的方向。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PORTAE							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

PORTAE：0：模拟引脚使能

1：GPIO 使能（默认）

-TRISA 寄存器

（地址 085h）复位值：1111_1111

TRISA 寄存器控制 PORTA 引脚的方向。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRISA							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

TRISA：0：输出使能

1：输入使能（默认）

-PULLA 寄存器

（地址 105h）复位值：0000_0000

PULLA 寄存器控制 PORTA 引脚的上拉电阻连接。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRISA							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

PULLA： 0：Pull-Up 使能（默认）

1：Pull-Up 失能

-TRISB 寄存器

（地址 085h）复位值：1111_1111

TRISB 寄存器控制 PORTA 引脚的方向。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRISA							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

TRISB：0：输出使能

1：输入使能（默认）

-PULLB 寄存器

（地址 106h）复位值：0000_0000

PULLB 寄存器控制 PORTB 引脚的上拉电阻连接。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TRISA							
w/r	w/r	w/r	w/r	w/r	w/r	w/r	w/r

PULLB： 0：Pull-Up 使能（默认）

1：Pull-Up 失能

-SPID 寄存器

（地址 00Dh）复位值：0000_0000

SPID 寄存器是 SPI 接口的读取/写入数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SPID							
w	w	w	w	w	w	w	w

SPID：SPI 读/写数据

-SPIC 寄存器

（地址 00Ch）复位值：xx01_1000

SPID 寄存器是 SPI 接口的读取/写入数据寄存器

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
---	---	RDY	CSN	CKPH	W3MODE	RCMD	WCMD
---	---	/r	w/r	w/r	w/r	w/r	w/r

Bit5:RDY：SPI 数据读写就绪位（只读）

0：未就绪（默认）

1：就绪

Bit4:CSN：SPI 芯片选择位

0：选择

1：不选择（默认）

Bit3:CKPH：时钟相位选择位

0：选择 CKPH0 模式

1：选择 CKPH1 模式（默认）

Bit2:W3_MODE：3 线/4 线模式选择位

0：选择 3 线模式（默认）

1：选择 4 线模式

Bit1:RCMD：SPI 读数据使能端

0：失能（默认）

1：使能

Bit0:WCMD：SPI 写数据使能端

0：失能（默认）

1：使能

2.3 PCL 和 PCLATH

该程序计数器（PC）是 13 位宽。低字节来自 PCL 寄存器，这是一个可读和可写寄存器。高位（PC < 12:8 >）不可读，但可以间接通过 PCLATH 寄存器写入，只要复位，程序计数器（PC）的高位将被清除。图 6 显示了加载 PC 的两种情况，图中上面的例子表明了 PC 写入 PCL 的时候是怎样加载的；下面的例子表明了 PC 在执行 CALL 或 GOTO 指令的时候是怎样加载的。

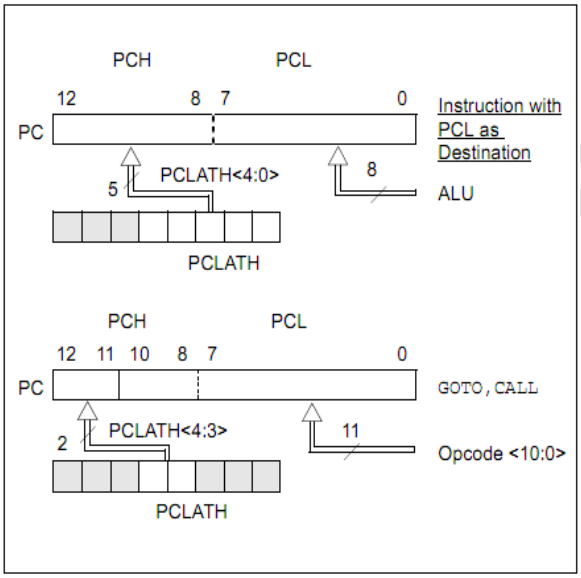


图 6：在不同情况下加载 PC

2.3.1 GOTO

计算 GOTO 指令是通过在程序计数器的基础上添加一个偏移量来完成（ADDWF PCL）。当使用 GOTO 指令来读取数据表的时候，一定要注意程序计数器有没有跨界（每一块是 256 个字节）。如果跨越了 PCL 内存边界，则应该进行处理。

2.3.2 STACK

GC7810A 有一个 8 级深，13 位宽的硬件堆栈。堆栈空间不是程序或数据空间的一部分，堆栈指针不可读写。当执行一个 CALL 指令或中断程序的时候，PC 会被送到堆栈里。当执行 RETURN, RETLW 或者 RETFIE 指令的时候，再将堆栈里的 PC 出栈。PCLATH 不会影响入栈和出栈。

堆栈是一个循环的缓冲区，这意味着在连续八次入栈之后，第九次入栈将会覆盖第一次的入栈值。第十次入栈将会覆盖第二次的入栈值（以此类推）。

注意：

1 :没有指示堆栈的状态指示位（溢出或不溢出）

2 :没有 PUSH 和 POP 指令，这些都是发生在执行 CALL, RETURN, RETLW 和 RETFIE 指令，或者进入中断的时候执行的操作。

2.4 程序存储器跨页访问

GC7810A 能够寻址连续的 4K 字节的程序存储器。 CALL 和 GOTO 指令仅提供 11 位地址，以允许在任何 2K 程序存储器页面中进行分支。当执行 CALL 或 GOTO 指令时，地址的高 2 位由 PCLATH <4：3>提供。

当执行 CALL 或 GOTO 指令时，用户必须确保页面选择位被编程，以便寻址所需的程序存储器页面。如果执行 CALL 指令（或中断）的返回，则整个 13 位 PC 从堆栈中弹出。因此，对于返回指令（从堆栈中弹出地址）不需要操纵 PCLATH <4：3>位。示例 1 显示了在程序存储器的第 1 页中调用子程序。此示例假定 PCLATH 由中断服务程序保存并恢复（如果使用了中断）。

注意：执行 RETURN 或 RETFIE 指令后，PCLATH 寄存器的内容不变。当执行子程序或 GOTO 指令时，用户必须重写 PCLATH 寄存器中的内容。

示例 1：

```
        ORG 0x500
        BCF PCLATH,4
        BSF PCLATH,3   ;Select page 1
                       ;(800h-FFFh)
        CALL SUB1_P1   ;Call subroutine in
        :              ;page 1 (800h-FFFh)
        :
        ORG 0x900      ;page 1 (800h-FFFh)
SUB1_P1
        :              ;called subroutine
                       ;page 1 (800h-FFFh)
        :
        RETURN         ;return to
                       ;Call subroutine
                       ;in page 0
                       ;(000h-7FFh)
```

例 1：从第 0 页调用第 1 页的子程序

2.5 间接寻址

INDF 和 FSR 寄存器

INDF (000h, 080h, 100h, 180h)

FSR (004h, 084h, 104h, 184h)

(FSR： 文件选择寄存器)

INDF 寄存器不是物理寄存器，通过使用 INDF 寄存器可以进行间接寻址。任何使用 INDF 寄存器的指令实际访问的是由文件选择寄存器（FSR）指向的寄存器。读取 INDF 寄存器本身，就是间接的读取（FSR ='0'），读取 00h。写入 INDF 寄存器是无操作（尽管可能会影响状态位）。通过连接 8 位 FSR 寄存器和 IRP 位（STATUS <7>）可获得有效的 9 位地址，如图 7 所示。

示例 2 中显示了使用间接寻址清除 RAM 地址 20h-2Fh 的简单程序。

```
        MOVLW  0x20    ;initialize pointer
        MOVWF  FSR     ;to RAM
NEXT    CLRF   INDF    ;clear INDF register
        INCF   FSR,F   ;inc pointer
        BTFSS  FSR,4   ;all done?
        GOTO   NEXT    ;no clear next
CONTINUE
        :              ;yes continue
```

例 2：间接寻址

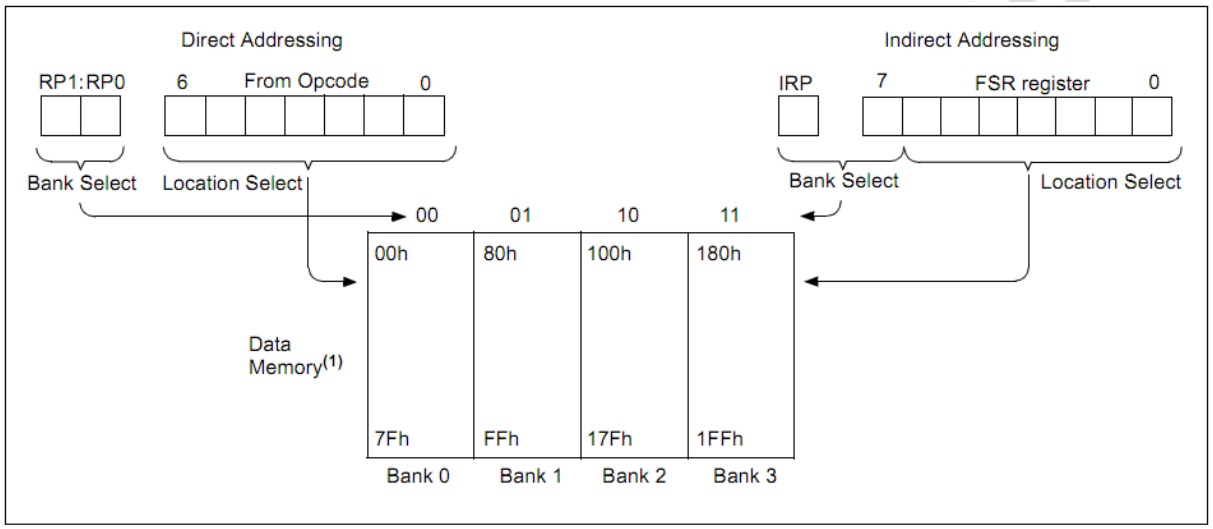


图 7 直接/间接寻址

3I/O 口

这些 I / O 端口的一些引脚与设备外围设备的备用功能复用。通常，当外设启用时，该引脚可能不会用作通用 I / O 引脚。

3.1 PORTA、TRISA 和 PULLA 寄存器

PORTA 是一个 8 位宽的双向端口。相应的数据方向寄存器为 TRISA。设置 TRISA 位（= 1）将使相应的 PORTA 引脚设置为输入（即将相应的输出驱动器置于高阻模式）。清除 TRISA 位（= 0）将使相应的 PORTA 引脚设置输出（即将输出锁存器的内容放在所选引脚上）。

读 PORTA 寄存器读取引脚的状态，而写入它将写入端口锁存器。所有写入操作都是读 - 修改 - 写操作。因此，写入端口意味着端口引脚被读取，该值被修改，然后写入端口数据锁存器。

当 PORA 引脚用作输入引脚时，PULLA 寄存器控制 PORTA 引脚的上拉电阻连接。当使用 PORTA 引脚作为输出或测试模式输入时，相应的上拉电阻自动断开。

对于输入 PORTA 引脚，设置 PULLA 位（= 1）将断开相应的上拉电阻。清除 TRISA 位（= 0）将连接相应的上拉电阻。

表 4TRISA 寄存器控制 PORTA 引脚的方向

寄存器名称	位	值	描述
TRISA (85h)	bit0~bit7	0	PORTA 的对应管脚设置为输出
		1	PORTA 的对应管脚设置为输入(默认)

表 5PULLA 寄存器控制 PORTA 引脚的上拉电阻

寄存器名称	位	值	描述
PULLA (105h)	bit0~bit7	0	连接到上拉电阻（默认）
		1	断开与上拉电阻的连接

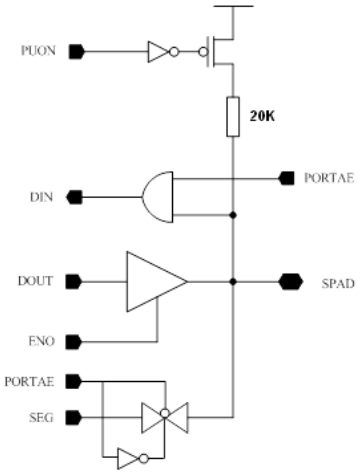


图 8 PORTA 管脚的输出驱动电路

3.2 PORTB、TRISB 和 PULLB 寄存器

PORTB 是一个 8 位宽的双向端口。相应的数据方向寄存器为 TRISB。设置 TRISB 位（= 1）将使相应的 PORTB 引脚设置为输入（即将相应的输出驱动器置于高阻模式）。清除 TRISB 位（= 0）将使相应的 PORTB 引脚设置为输出（即将输出锁存器的内容置于所选引脚上）。

读取 PORTB 寄存器读取引脚的状态，而写入它将写入端口锁存器。所有写入操作都是读 - 修改 - 写操作。因此，写入端口意味着端口引脚被读取，该值被修改，然后写入端口数据锁存器。

表 6 PORTB

名称	位							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PORTB (06h, 106h)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			SDI	SDO	SCK	CSN	RX	TX
	CXO	CXI	RHO		RFO	RTO		

如表 6 所示，PORTB 引脚与 UART 和 INPUT 输出复用。

TRISB 寄存器控制 PORTB 引脚的方向，即使它们被用作 UART 输出或 IROUT 输出。当使用它们作为 UART 输出或 IROUT 输出时，用户必须确保 TRISB 寄存器中的位保持置 1。

当 PORTB 引脚用作输入引脚时，PULLB 寄存器控制 PORTB 引脚的上拉电阻连接。当使用 PORTB 引脚作为输出或 UART 输出或 IROUT 输出时，相应的上拉电阻自动断开。

对于输入 PORTB 引脚，设置 PULLB 位（= 1）将断开相应的上拉电阻。清零 TRISB 位（= 0）将连接相应的上拉电阻

表 7　Direction Control of PORTB Pins by TRISB Register

名称	位	值	描述
TRISB (86h)	bit0~bit7	0	PORTB 的对应管脚设置为输出
		1	PORTB 的对应管脚设置为输入(默认)

表 8Pull-up Control of PORTB Pins by PULLB Register

名称	位	值	描述
PULLB (106h)	bit0~bit7	0	连接到上拉电阻（默认）
		1	断开与上拉电阻的连接

4 Timer0 模块

Timer0 模块具有以下特点：

- 8 位定时器
- 可读写
- 8 位软件可编程预分频器
- FFh 溢出到 00h 时产生中断
- 定时器的自动重载功能

图 10 是 Timer0 模块与 WDT 共享的预分频框图，Timer0 模块的时钟频率为 4MHz，通过设置/清除控制位 T0E（PERCON<4>）可以使能/禁止 Timer0 模块，预分频器在 Timer0 和 WDT 之间相互独立的共享，预分频器不可读写。

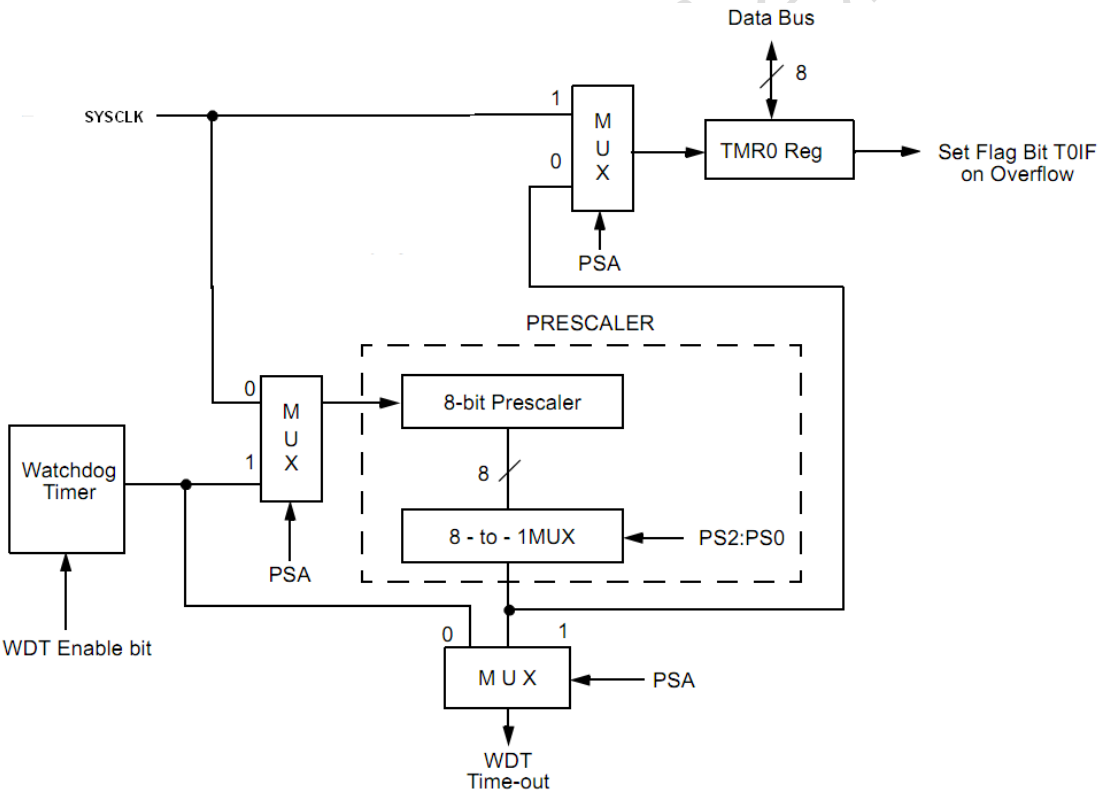


图 10　Timer0 / WDT 预分频器结构图

4.1 Timer0 中断

当 TMR0 寄存器从 FFh 溢出到 00h 时，产生 TMR0 中断。该溢出将置位 T0IF（INTCON <2>）。可以通过清零 T0IE（INTCON <5>）来屏蔽中断。在重新使能该中断之前，Timer0 模块中断服务程序必须通过软件清零位 T0IF。由于定时器在休眠期间关闭，因此 TMR0 中断无法唤醒处理器。

4.2 预分频器

只有一个预分频器可用，它们在 Timer0 模块和看门狗定时器之间相互独立地共享。把预分频器指定给 Timer0 模块则意味着看门狗定时器没有预分频器，反之亦然。

该预分频器不可读写（见图 10）。

PSA 和 PS2：PS0 位（OPTION_REG <3：0>）决定预分频器分配和预分频比。

当分配给 Timer0 模块时，写入 TMR0 寄存器（例如 CLRF 1，MOVWF 1，BSF 1，x ....等）的所有指令将清零预分频器。当分配给 WDT 时，CLRWDT 指令将与看门狗定时器一起清除预分频器。预分频器不可读写。

注意：当预分频器分配给 Timer0 时，写入 TMR0 时，将清零预分频器计数，但不会更改预分频器分配。

4.3 Timer1 模块

Timer1 模块是一个 16 位定时器，由两个可读写的 8 位寄存器（TMR1H 和 TMR1L）组成。 TMR1 寄存器对（TMR1H，TMR1L）从 0000h 递增到 FFFFh，并转到 0000h。如果使能 TMR1 中断，则会在溢出时产生，该溢出被锁存在中断标志位 T1IF（PIF1 <4>）中。

通过设置/清除 TMR1 中断允许位 T1IE（PIE1 <4>）可以使能/禁止该中断。

Timer1 模块的源时钟频率为 4MHz。通过设置/清除控制位 T1E（TMR1C <2>）可以使能/禁止 Timer1 模块。 Timer1 模块中没有自动重新加载功能。

Timer1 预分频器计数器在写入 TMR1H 或 TMR1L 寄存器时清零。

5 UART 模块

UART（通用异步接收/发送）模块通过 RX 引脚和 TX 引脚将 GC7810A 连接到外部系统（RB7 引脚，RB6 引脚）

每个字符的时序基于 UART 的选定波特率。发送与接收功能使用相同的波特率频率。

UART 模块包括以下功能：

- 具有非奇偶校验的 8 位数据
- 支持使用一个移位寄存器进行发送和接收
- 支持使用一个缓冲寄存器进行发送和接收
- LSB 优先数据发送和接收
- 可编程波特率，支持波特率小波变换
- 接收和发送的独立中断能力

5.1 字符格式

通常，如图 11 所示，UART 字符格式由起始位，七位或八位数据位，偶数/奇数/无奇偶校验位，地址位和一位或两位停止位组成。位周期由选定的时钟源和波特率寄存器的设置来定义。

GC7810A 支持的字符格式和规格，如图 12 所示。

- 波特率：1200～250000（默认为 9600）
- 数据位数：8 位
- 校验位：无
- 停止位：1 位

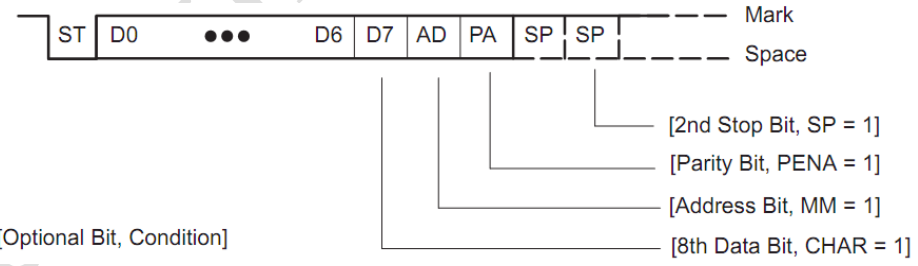


图 11 字符格式

图 12 GC7810A 的字符格式

5.2 UART 波特率生成

USART 波特率发生器能够从非标准源频率产生标准波特率。波特率发生器使用一个预分频器/分频器和调制器，如图 13 所示。该组合支持波特率生成的分数因子。

UART 的最大波特率是 UART 源时钟频率 BRCLK 的三分之一。

每个位的时序如图 14 所示。对于接收的每个位，采用多数表决来确定位值。这些样本出现在 N / 2-1，N / 2 和 N / 2 + 1 BRCLK 周期，其中 N 是每个 BITCLK 的 BRCLK 数。

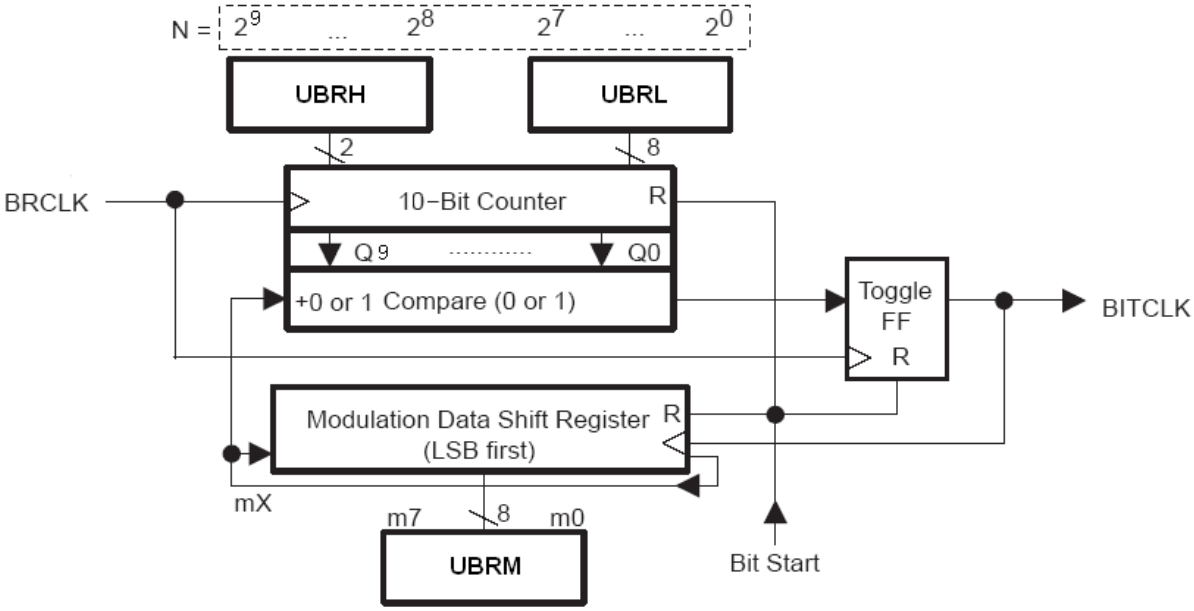


图 13 GC7810A 的波特率产生器

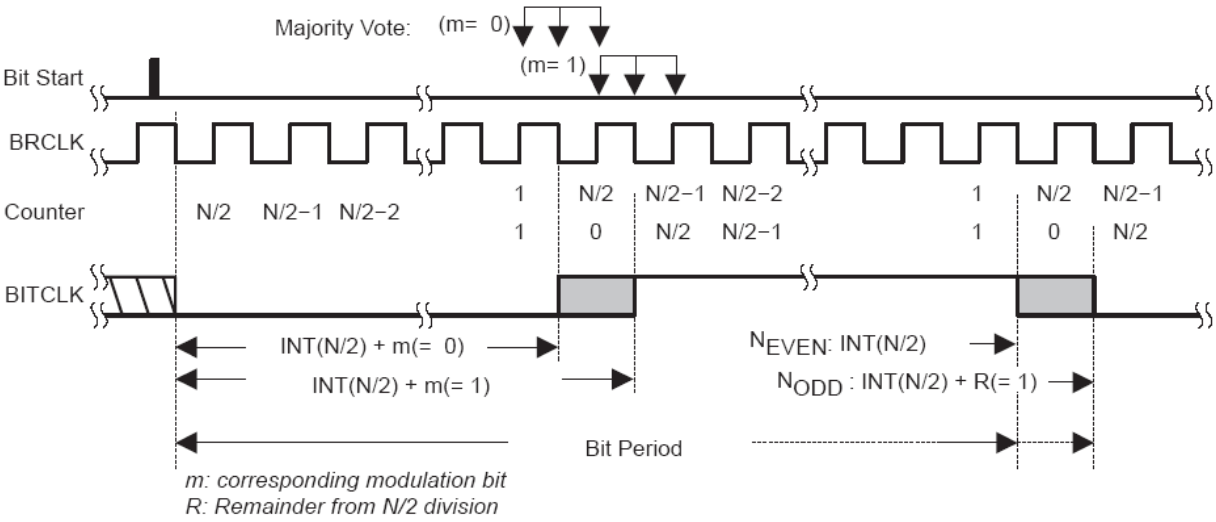


图 14 BITCLK 波特率时序

5.2.1 波特率时序

波特率发生器的第一级是 16 位计数器和比较器。在发送或接收的每个位的开始，计数器加载有 INT（N / 2），其中 N 是存储在 UBRH 和 UBRL 的组合中的值。计数器为每个位周期的半周期重新加载 INT（N / 2），给出 N 个 BRCLK 的总位周期。对于给定的 BRCLK 时钟源，使用的波特率决定了所需的分频因子 N：

$$N = \frac{BRCLK}{baud\ rate}$$

分频因子 N 通常是非整数值，其整数部分可以由预分频器/分频器实现。波特率发生器（调制器）的第二阶段用于尽可能接近分数部分。因子 N 被定义为：

$$N = UxBR + \frac{1}{n}\sum_{i=0}^{n-1} m_i$$

这里：

N：目标分解因子

UxBR：寄存器 UBRH 和 UBRL 的 16 位表示形式

i：字符中的位位置

n：字符中的总位数

mi：每个对应的调制位（1 或 0）的数据

$$Baud\ rate = \frac{BRCLK}{N} = \frac{BRCLK}{UxBR + \frac{1}{n}\sum_{i=0}^{n-1} m_i}$$

当需要非整数的除数时，可以使用调制器从位到位调整 BITCLK 以满足时序要求。如果调制器的 mi 置位，则每个位的定时被扩展一个 BITCLK 时钟周期。每次接收或者发送一个位时，调制控制寄存器中的下一位决定该位的时序。设置调制位将除法系数提高 1，而清除调制位则保持由 UxBR 给出的除法系数。

起始位的定时由 UxBR 加 m0 确定，下一位由 UxBR 加 m1 确定，以此类推。调制序列从 LSB 开始，当字符大于 8 位时，调制序列以 m0 重新开始，并继续，直到所有位都被处理。

5.2.2 确定调制值

确定调制值是一个交互过程。使用提供的定时误差公式，从起始位开始，通过相应的调节器位置 1 和清零来计算各个位错误。调制位设置具有较低的误差

并计算下一位误差。继续该过程直到所有位错误被最小化。当字符包含超过 8 位时，重复调制位。例如，字符的第 9 位使用调制位 0。

5.2.3 发送位时序

每个字符的定时是各个位定时的总和。通过调制每个位，累积位错误减少。单个位错误可以通过以下方式计算：

$$Error\,[\%] = \left\{\frac{baud\ rate}{BRCLK} \times \left[(j+1) \times UxBR + \sum_{i=0}^{j} m_i\right] - (j+1)\right\} \times 100\%$$

这里：

baud rate：所需波特率

BRCLK：输入频率-UCLK（1,048,576 Hz）

j：位位置 - 起始位为 0，数据位为 D0 为 1，依此类推

UxBR：寄存器 UBRH 和 UBRL 中的分频系数

例如，计算以下条件的传输误差：

波特率= 2400

BRCLK = 32,768Hz（ACLK）

UxBR = 13，因为理想的除数为 13.65

UBRM = 6Bh：m7 = 0，m6 = 1，m5 = 1，m4 = 0，m3 = 1，m2 = 0，m1 = 1，m0 = 1。

首先使用 UBRM 的 LSB。

$$\text{Start bit Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((0 + 1) \times \text{UxBR} + 1) - 1\right) \times 100\% = 2.54\%$$

$$\text{Data bit D0 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((1 + 1) \times \text{UxBR} + 2) - 2\right) \times 100\% = 5.08\%$$

$$\text{Data bit D1 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((2 + 1) \times \text{UxBR} + 2) - 3\right) \times 100\% = 0.29\%$$

$$\text{Data bit D2 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((3 + 1) \times \text{UxBR} + 3) - 4\right) \times 100\% = 2.83\%$$

$$\text{Data bit D3 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((4 + 1) \times \text{UxBR} + 3) - 5\right) \times 100\% = -1.95\%$$

$$\text{Data bit D4 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((5 + 1) \times \text{UxBR} + 4) - 6\right) \times 100\% = 0.59\%$$

$$\text{Data bit D5 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((6 + 1) \times \text{UxBR} + 5) - 7\right) \times 100\% = 3.13\%$$

$$\text{Data bit D6 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((7 + 1) \times \text{UxBR} + 5) - 8\right) \times 100\% = -1.66\%$$

$$\text{Data bit D7 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((8 + 1) \times \text{UxBR} + 6) - 9\right) \times 100\% = 0.88\%$$

$$\text{Parity bit Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((9 + 1) \times \text{UxBR} + 7) - 10\right) \times 100\% = 3.42\%$$

$$\text{Stop bit 1 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times ((10 + 1) \times \text{UxBR} + 7) - 11\right) \times 100\% = -1.37\%$$

结果显示最大每位误差为 BITCLK 周期的 5.08%。

5.2.4 接收位时序

接收时序由两个误差源组成。第一个是比特定时误差。第二个是发生启动边沿和起始边沿被 USART 接收的误差。图 15 显示了 RX 引脚上的数据和内部波特率时钟之间的异步定时误差。

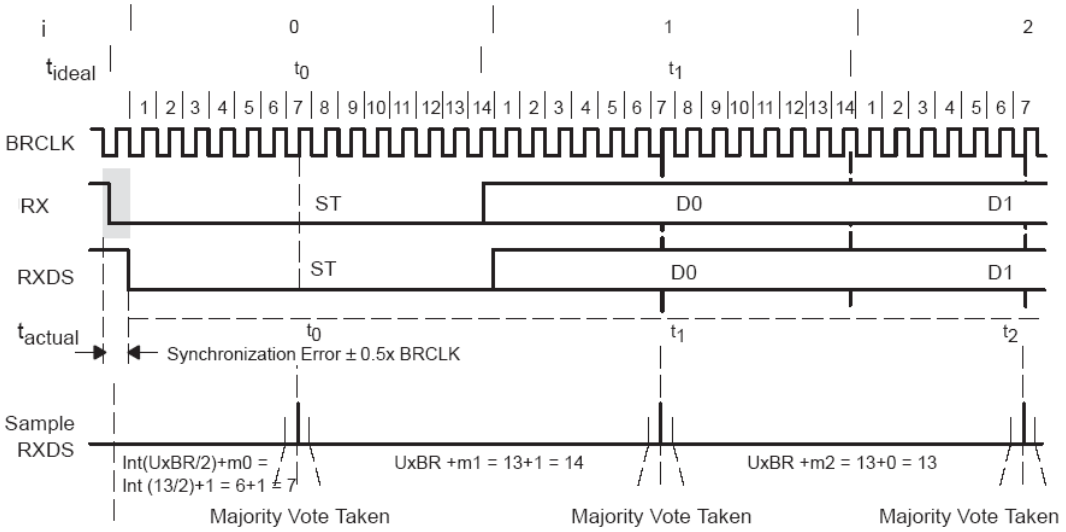


图 15 接收误差

理想的起始位时序 $t_{ideal(0)}$ 是波特率时序 $t_{baud\ rate}$ 的一半，因为该位在其周期的中间进行了测试。其余字符位的理想波特率时序是波特率时序 $t_{baud\ rate}$。单个位的误差可以通过以下方式计算：

$$Error\,[\%] = \left(\frac{baud\ rate}{BRCLK} \times \left\{2 \times \left[m0 + int\left(\frac{UxBR}{2}\right)\right] + \left(i \times UxBR + \sum_{j=1}^{j} m_j\right)\right\} - 1 - j\right) \times 100\%$$

这里：

baud rate :：所需波特率

BRCLK：输入频率-UCLK（1,048,576Hz）

j：位位置 - 起始位为 0，数据位 D0 为 1，依此类推

UxBR：寄存器 UBRH 和 UBRL 中的分频系数

例如，计算以下条件的接收误差：

波特率= 2400

BRCLK = 32,768Hz（ACLK）

UxBR = 13，因为理想的除数为 13.65

UxMCTL = 6B：m7 = 0，m6 = 1，m5 = 1，m4 = 0，m3 = 1，m2 = 0，m1 = 1，m0 = 1

首先使用 UxMCTL 的 LSB。

$$\text{Start bit Error } [\%] = \left(\frac{baud\ rate}{BRCLK} \times [2x(1+6) + (0 \times UxBR + 0)] - 1 - 0\right) \times 100\% = 2.54\%$$

$$\text{Data bit D0 Error } [\%] = \left(\frac{baud\ rate}{BRCLK} \times [2x(1+6) + (1 \times UxBR + 1)] - 1 - 1\right) \times 100\% = 5.08\%$$

$$\text{Data bit D1 Error } [\%] = \left(\frac{baud\ rate}{BRCLK} \times [2x(1+6) + (2 \times UxBR + 1)] - 1 - 2\right) \times 100\% = 0.29\%$$

$$\text{Data bit D2 Error } [\%] = \left(\frac{baud\ rate}{BRCLK} \times [2x(1+6) + (3 \times UxBR + 2)] - 1 - 3\right) \times 100\% = 2.83\%$$

$$\text{Data bit D3 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (4 \times \text{UxBR} + 2)] - 1 - 4\right) \times 100\% = -1.95\%$$

$$\text{Data bit D4 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (5 \times \text{UxBR} + 3)] - 1 - 5\right) \times 100\% = 0.59\%$$

$$\text{Data bit D5 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (6 \times \text{UxBR} + 4)] - 1 - 6\right) \times 100\% = 3.13\%$$

$$\text{Data bit D6 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (7 \times \text{UxBR} + 4)] - 1 - 7\right) \times 100\% = -1.66\%$$

$$\text{Data bit D7 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (8 \times \text{UxBR} + 5)] - 1 - 8\right) \times 100\% = 0.88\%$$

$$\text{Parity bit Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (9 \times \text{UxBR} + 6)] - 1 - 9\right) \times 100\% = 3.42\%$$

$$\text{Stop bit 1 Error [\%]} = \left(\frac{\text{baud rate}}{\text{BRCLK}} \times [2x(1 + 6) + (10 \times \text{UxBR} + 6)] - 1 - 10\right) \times 100\% = -1.37\%$$

结果显示最大每位误差为 BITCLK 周期的 5.08%

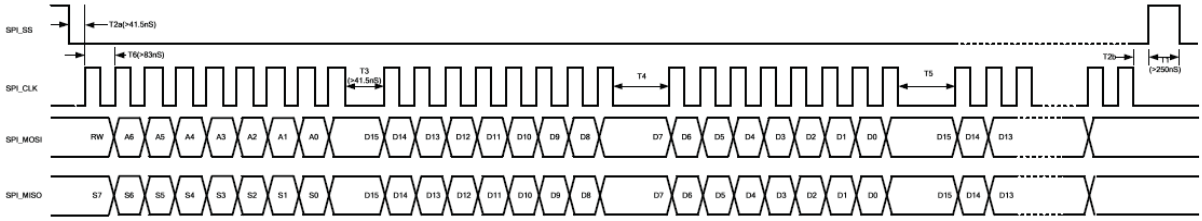
6 附加模块

6.1 SPI 模块

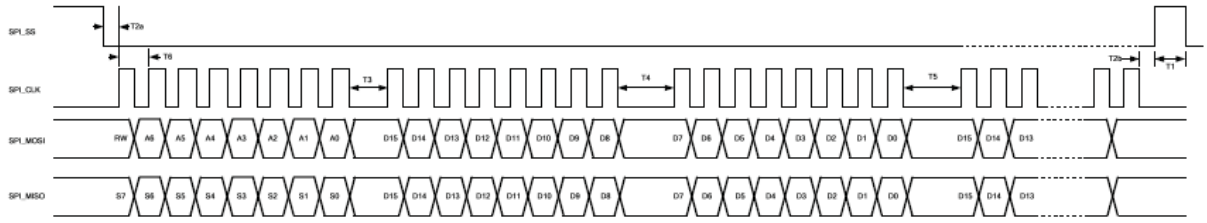
GC7810 具有 SPI 主机，支持与外部芯片（RFchip(LT8900)）1Mbps 的接口速率。该模块可以支持 3/4 线模式，CKPH0/CKPH1 模式。

片选信号（CSN）→RB2,（SCK）→RB3,（SDO）→RB4,（SDI）→RB5。在 4 线模式下仅使用 CSN,SCK,SDO。此外，如果使能 CSN,SPI 可以与许多其他器件（LT8900,ARF2496K,具有 SPI 接口的高速闪存等）连接。

SPI 模块的时序图如下：



CKPH=1 的 SPI 主接口时序图



CKPH=0 的 SPI 主接口时序图

SPI 模块以一个字节传输，当 SPI 接口启动时，CPU 必须格式化 CSN 位。为了提高 SPI 传输速率，GC7810A 设计为 WCMD 只能读取，一旦传输日期加载到 SPID 寄存器中，WCMD 自动为“1”，GC7810 使用 mSDO 端口发送数据，当 1Byte 数据传输完成时，WCMD 自动变为“0”。发送日期在 SPID 寄存器中装载后，SPI 模块需要三个“nop”指令进行发送。详细的示例代码如下：

```
bcf     CSN   //chip select signal
movlw   24h   //   transmit data
movwf   SPID //24h→SPID
nop;
nop;
nop;
```

由于 SPI 传输速率与 CPU 时钟同步，SPI 平均速率为 4Mbps。当 CPU 在监测 SPIC 寄存器的 RDY 位时，SPI 模块并没有中断。在读取数据时，RCMD 设置为 1，CPU 监测 RDY 位。如果 RDY 设置为 1，则 CPU 读取 SPID 寄存器。RCMD 可以自动选择，也可以写为 0。

6.2 EEPROM 功能

在 FSR（E2AH/E2AL,E2DH/E2DL）和 E2CON 寄存器的 WR 引脚上的写入地址和数据设置为 1 时，WR 位由 MTPROM 置位，数据被写入 MTPROM 的指定地址。读取数据时，如果分配的 FSR（E2AH/E2AL）和 E2CON 寄存器的 RD 引脚写入的地址设置为 1，则 RD 指令必须置位后才能把数据写入到（E2DH/E2DL）。写入和读取是在 16 位字长单位中实现的，当 WR/RD 设置完成后，必须跟两个 nop.

6.3 编程键锁功能

一旦 0bcdh 在 0fffh 中置位，TST 模式将会被锁定，RD/WR 功能也会被禁止，但 CPU 仍然工作在正常模式。此时，程序员必须保证“0fffh”地址内没有任何程序。如果禁止内部 EEPROM 功能，在 TST 模式下的读写功能也会被禁止。芯片设计师可以将其解除，使 TST 模式反转。

7 Debug&TestScan 模块（ICD&ICSP）

GC7810 可以通过 3 个引脚（RA2，RA1，RA0）进行读写。 RA0 引脚可以输入和输出信号，当 CPU 断点执行 TSEG 的时候，可以通过监视 TDIO（RA）引脚来实现。不仅可以读取 MTP ROM、RAM，还可以读取状态寄存器和堆栈指针。此功能对开发人员开发设备非常有用。在测试模式下使用的引脚及其功能如下：

TDIO：TestScan 移位寄存器串行数据输入/输出端（RA0）

TSCK：TestScan 移位寄存器串行时钟输入端（RA1）

TSM：TestScan 移位寄存器模式设置输入端（RA2）

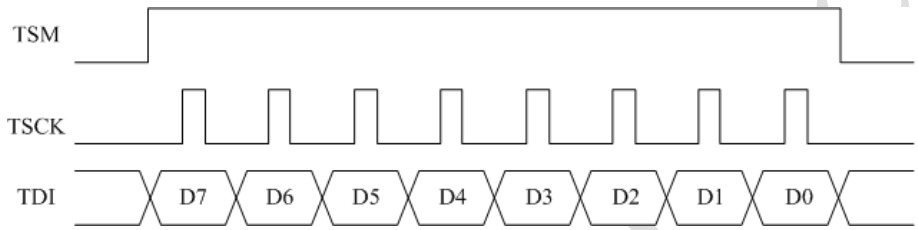


图 23： 测试模式控制信号时序图

8. CPU 指令集汇总

GC7810A 单片机指令是一个 16 位字，划分为一个操作码，它指定指令类型和一个或多个操作数，进一步指定指令的操作。在表 12 中，GC7810 的指令集简要的列出了面向字节的，面向位的、文字和控制的操作。表 11 显示了 opcode 字段的描述。面向字节的指令，“f”代表一个文件注册指示器和“d”代表目标指示器。文件寄存器指示符指定该指令使用哪个文件寄存器。

目标指示符指定要放置操作结果的位置。如果‘d’为 0，结果将放在 W 寄存器中。如果‘d’为 1，结果将放在指令指定的文件寄存器中。

对于面向位的指令，‘b’表示一个位字段指示符，它的选择受操作位数的影响，而‘f’表示位所在的文件的地址。

对于文字和控制操作，k 代表一个 8 或 11 位常量或文字值。

表 11：操作码字段描述

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

指令集高度正交，分为三个基本类别：

- 面向字节的操作
- 面向位的操作
- 文字和控制操作

所有指令都在一个指令周期内执行，除非条件测试为真，或程序计数器由于指令而改变。在这种情况下，执行需要两个指令周期，第二个周期作为 NOP 执行。一个指令周期由四个振荡周期组成。因此，对于振荡频率为 4MHz 的正常指令执行时间为 0.5 μs。如果条件测试为真，或程序计数器由于指令而改变，则指令执行时间为 1 μs。表 12 列出了 MPASMTM 汇编程序识别的指令。

图 25 显示了指令的通用格式，所有示例使用一下格式来表示十六进制数：0xhh 其中 h 表示十六进制数字。

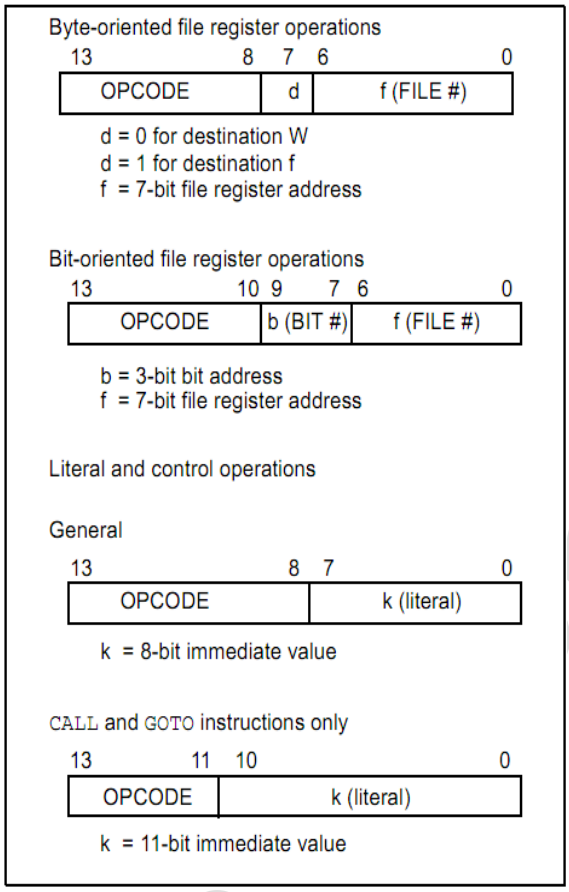


图 25：指令的通用格式

9. 模拟功能说明

9.1 内部 RC 振荡器

10KHz RC 振荡器

RC 振荡器是一种低功耗振荡器，在深度睡眠模式下，通过设计而达到了最小的电流消耗。这种基于电流模式的振荡器的工作原理是弛张振荡器，电路原理如下。

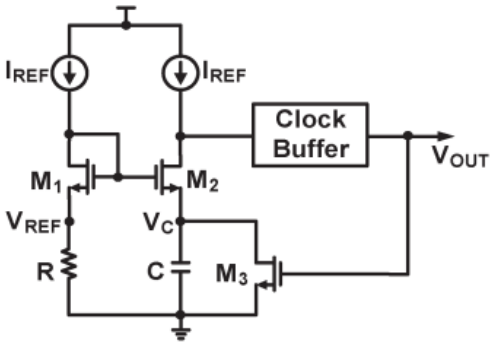


图 26　弛张振荡器原理

2MHz RC 振荡器，具有频率选择性。

频率可调范围：0.2MHz∽4MHz。

CAL_OSC 寄存器值控制 RC 振荡器频率。

00h=最大频率（4MHz），7Fh=最小频率（0.2MHz）

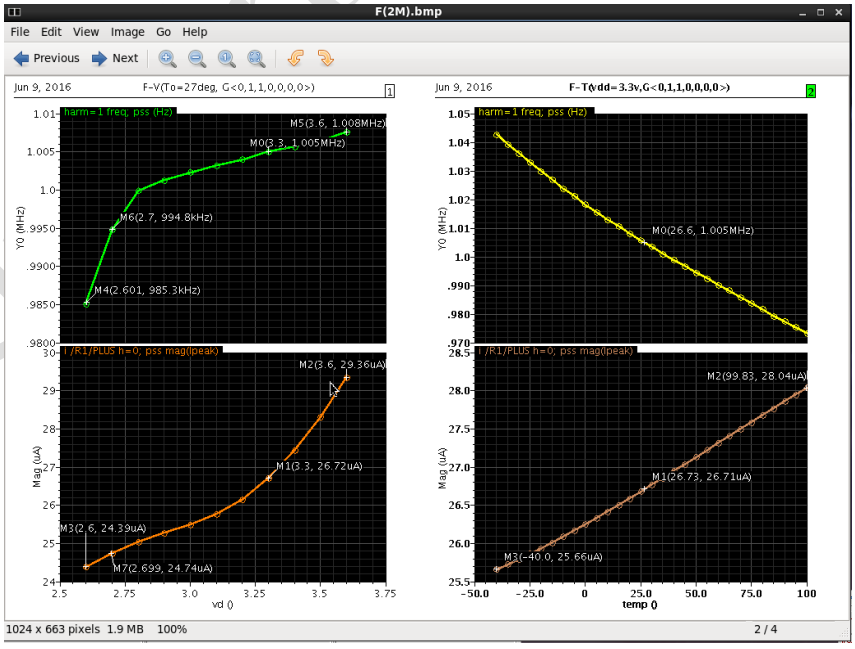


图 27　F-V，F-T，F-I，Vdd-I 特性

频率可以近似为

$$f0 = \alpha * \frac{1}{R * C}$$

控制系数 - 电阻 trimming 模式

控制位 - 7bit

表 11：电阻选项表默认代码=“h00”

<7>	<6>	<5>	<4>	<3>	<2>		<1>	
0	0	0	0	0	0		0	
					11	10	01	00
64k	32k	16k	8k	4k	1.6k	0.8k	0.4k	0.2k

9.2 温度传感器特性

传感器输出电压 - NTAT 特性

温度分辨率 - ±0.5℃

测量范围 - 40 - +85℃

图 28 显示了温度传感器电路的输出电压。

其中灰线为测量值，红线为拟合曲线。

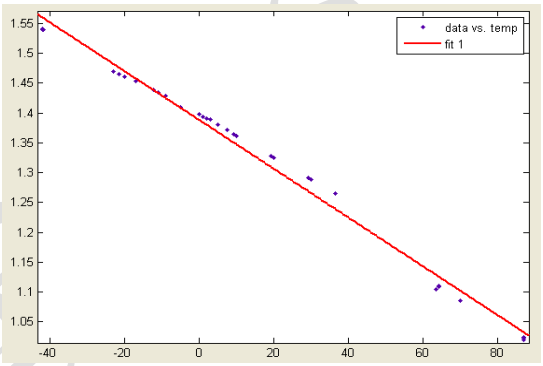


图 28. ITAT 输出特性

使用方法：

当 REFGE(ANACON[5])=1 时，REFHE(ANACON[6])=1，REFLE(ANACON[7])=1，ADCE(ANACON[0])=1，TEMPE(ANACON1[6])=1 设置“1”带隙电路，ADC 模块，Temp 传感器模块使能。NET[4:3]设置“00”，AD Vref 为使能。此时 REFR[7：0]是控制 Vref 选项寄存器，然后 NET[2:0]设置为“100”，传感器输出连接 AD 输入。ANACON1[1：0]和 ANACON2[2：0]是控制 AD 采样率。然后，GIE,PEIE,ADCIE 设置为“1”，ADCI 标志位置 1 后，ADRES[7：0]寄存器读取。

9.3 A / D 转换器

GC7810A 具有 8 位 SAR（逐次逼近）A / D 转换器。

图 29 显示了 SAR A / D 转换器的动态响应特性（fin = 10kHz，fs = 125k）

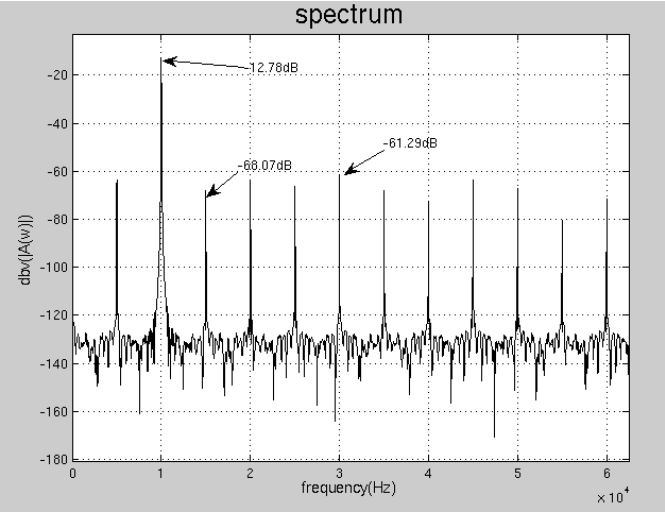


图 29. A / D 动态响应特性

A / D 转换器输入信号可分别为外部信号，TIA 输出，内部 ITAT 输出，参考输出电压采用内部寄存器控制。

-使用方法

•使用内部基准源

设置 PORTAE [5：3] = 000，使 RA <5：3>端口作为模拟端口，然后设置 REFGE（ANACON [5]）= 1，REFHE（ANACON [6]）= 1，REFLE（ANACON [7]）= 1，ADCE（ANACON [0]）= 1，运行带隙模块，内部参考模块，ADC 模块，接着设置 NET [4：3] = 00，将 AD 高/低参考连接到内部参考模块，再设置 REFR [7：0]，设置高/低参考值，接下来设置 NET [2：0]，所以选择 AD 输入端口（RA <5：3>端口之一，NET [2：0] = 0 作为 RA <3>端口，1 作为 RA <5>端口，2 作为 RA <4>端口)），下一个设置 ANACON1 [1：0]和 ANACON2 [2：0]，配置 AD 采样率，完成基本寄存器设置。

接下来设置 GIE，PEIE，ADCIE，启用 ADC 中断，当生成 ADCI 时，读取 ADRES [7：0]，可以测量 AD 值。

•使用外部基准源

设置 PORTAE [5：1] = 00000，使 RA <5：1>端口作为模拟端口，设置 ADCE（ANACON [0]）= 1，运行 ADC 模块，然后设置 NET [4：3] = 11 ，因此将 AD 高/低引用连接到外部引用（RA <2>作为高引用，RA <1>为低引用）然后设置 NET [2：0]，选择 AD 输入端口（RA <5：3 之一>端口），下一次设置 ANACON1 [1：0]和 ANACON2 [2：0]，配置 AD 采样率，完成基本寄存器设置。

接下来设置 GIE，PEIE，ADCIE，启用 ADC 中断，当生成 ADCI 时，读取 ADRES [7：0]，可以测量 AD 值。

9.4 比较器功能

比较器参考电压可以使用内部寄存器控制转换为内部/外部模式，在内部模式下，参考电压有多种电压可供选择。

比较器输入信号可以是外部信号，TIA 输出，内部 ITAT 输出，参考输出电压采用内部寄存器控制。

-使用方法

•使用比较器独立模式

比较器输入引脚可以与 PORTA 连接。 PORTAE [5：3] = 0000，RA <5：3>设置模拟端口。

REFGE（ANACON [5]）= 1，REFHE（ANACON [6]）= 1，REFLE（ANACON [7]）= 1，CMP1E（ANACON [1]）= 1，CMP2E（ANACON [2]）= 1 带隙内部 V ref 生成块，比较器块设置“使能”，NET [4：3]设置为“00”，比较器 1,2 V ref 与内部 V ref 生成块 REF REF [7：0]控制内部 1， 2Vref 值，NET [2：0] NET [6：5]是两个比较器 1,2 输入连接 RA <5：3> ANACON1 [5：4] = 00 - RA <4>的寄存器<0>。

NET [2：0] = 0 组 RA <3> NET [2：0] = 1 组 RA <5> NET [2：0] = 2 组 RA <4>

如果外部 V ref 电压使用 1,2 V ref 值，则将 PORTAE [5：0] =“000000”设置 RA <5：0>更改模拟端口，CMP1E（ANACON [1]）= 1，CMP2E（ANACON [2]）= 1 组，NET [4：3]设置“11”比较器输入引脚可以与 PORTA 连接; 设置“1”ANACON1 [5：4]设置“00”，两个比较器被使能，RA <2>，NET [2：0]，NET [2：0]

•使用比较器中断

过程与上述相同，不同的是设置 ANACON1 [5：4] = 11，所以将比较器 1,2 输出连接到 CPU，然后设置 GIE，PEIE，CMP1IE，CMP2IE，因此当产生中断可以测量时，使能比较器中断。

9.5 TIA 模块

GC7810A 设计可以使用带开关电容放大器的红外报警模块

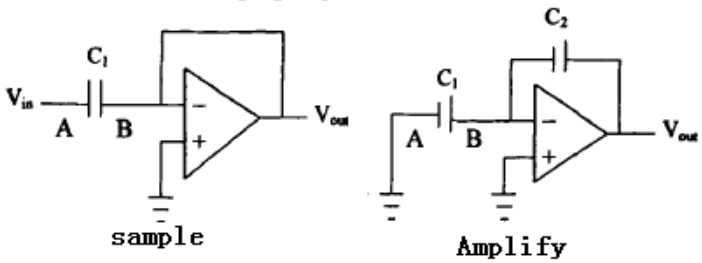


图 30 红外报警模块内部应用电路

增益=C1/C2,输出电压可以由 A/D 比较器来检测。

图 30 显示了内部使用的运算放大器，图 31 显示了外部电路的应用

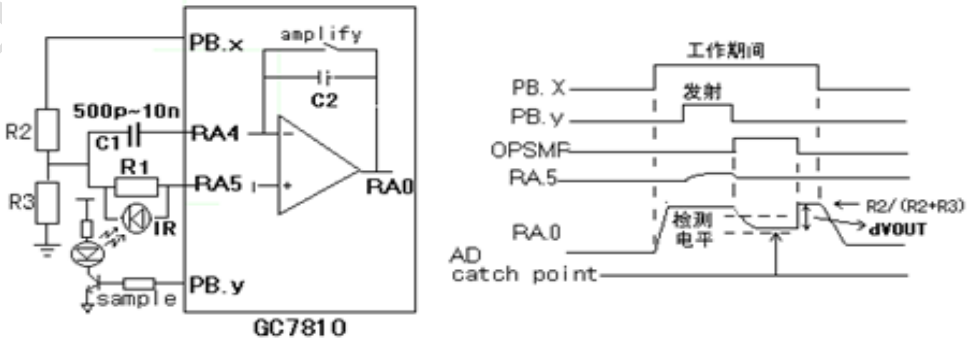


图 31 红外报警模块外部应用电路

-使用方法：

设置 PORTAE [5：4]，[0] = 000，采用 RA <5：4><0>引脚作为模拟端口，其次设置 OPE（ANACON [3]）= 1，运行 TIA 模块，然后设置 NET [2：0] = 5，连接 TIA 输出到 AD 输入，再设置 OPSMP（ANACON [4]）如上图所示，如上所述 AD 测量方法可以测量 TIA 输出值。

9.6 LVD 模块设计（低电压检测电路）

LVD（低电压检测）电路，带隙参考输出电压与分压传感器电压相比较时，检测电压可以使用 cpu 控制（REFR [3：0]）来选择不同的电压电平。 CPU 设置“LVDE = 1”，并读取施密特比较器 2 的输出电压。在正常电压情况下，CMP2POL 设置“L / H”，在低电压情况下，CMP2POL 设置“H / L”。如果 LVDE 置“0”，CMP2POL 置“L / H”。 VDD = 3.0V 时 LDV 工作电流为 60uA。

用户可以使用 LVD 中断来处理很多事件，GC7810A 具有两个峰值比较器，用来约束低功率输出电压摆幅噪声。

GC7810 仅使用 REFR [3：0] = 6,5（2.5,2.7V）

0011 : -----

0100 : -----

0101 :2.5V

0110 :2.7V

111 -----

9.7 模拟多路复用器结构框图

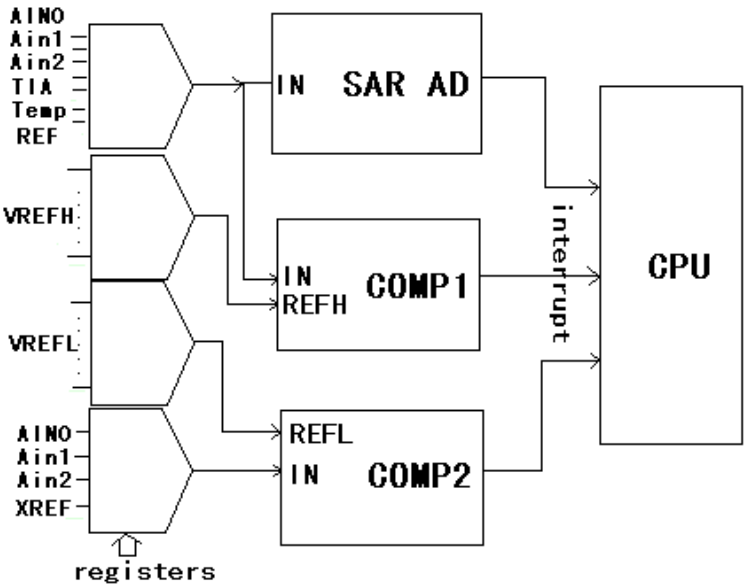


图 32 模拟多路复用器

模拟复用器可以通过使用内部注册表控制来连接 A / D 和比较器的输入通道中的不同信号。

9.8 R2FC

温度测量是以测量 RC 振荡器的频率改变的方式进行的，用于温度传感器的电阻的阻值变化而引起 RC 振荡器的频率改变。

用于温度测量的 RC 振荡器结构如下图所示

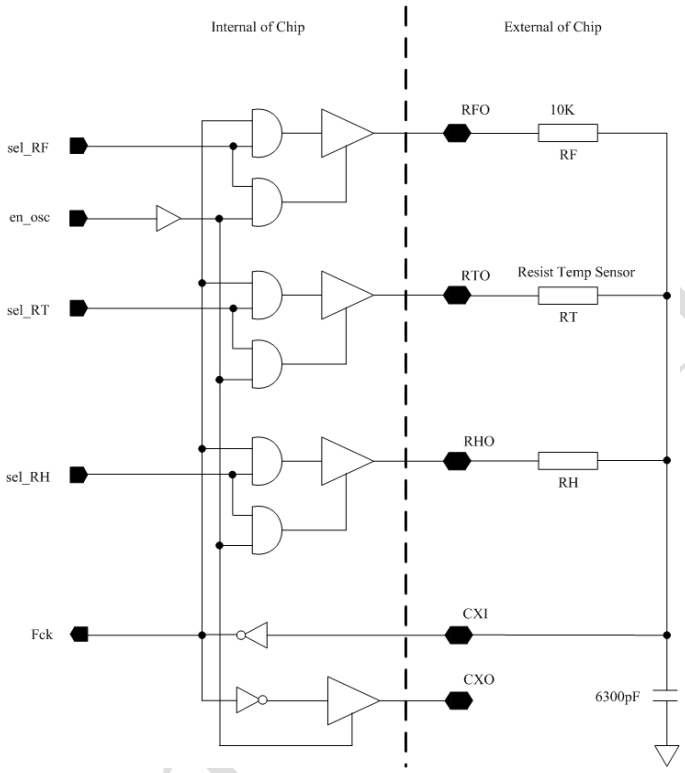


图 33.连接图

有两个 16 位计数器，其中一个 16 位计数器（R2CNTRH，R2CNTRL）用于计数内部 32768Hz 时钟，另一个 16 位计数器（R2CNTSH，R2CNTSL）用于计数接温度传感器的 RC 振荡器进入的时钟。

还有用于比较计数器值的 16 位设置寄存器（R2DIVH，R2DIVL）。

R2DIVH，R2DIVL 是只写寄存器，R2CNTRH，R2CNTRL 是只读寄存器，它们使用相同的 RAM 地址空间（$ 011，$ 012）

R2CNTSH，R2CNTSL 是只读寄存器。地址是（$ 013，$ 014）

在 R2FC（$ 00D）中，SENCHN_SEL 是一个选择传感器通道的位。

SENCHN_SEL = 00 : 10K standard resistance RF

= 01 :temperature resistance RT(NTC) 502NTC, 103NTC

= 10 :humidity resistance RH(HS1101)

= 11 :no action

MEAS_EN 是一个用于开始温度测量的 bit。 MEAS_DONE 是一个用于指示温度测量完成的 bit。

完成温度测量后， R2FI 中断就会产生。 CPU 将 MEAS_EN 位设置为“1”，开始温度测量，然后进入休眠状态，通过 R2FI 中断唤醒和读取计数器值。

MEAS_MODE 是一个用于设置温度测量模式的 bit。如果 MEAS_MODE = 0，是 Mode0 模式，MEAS_MODE = 1，是 Mode1 模式。

在 Mode0 模式下，通过比较 16 位设置寄存器（R2DIVH，R2DIVL）和 RC 振荡器计数器（R2CNTSH，R2CNTSL），继续进行直到它们相等，最终得到 32768Hz 计数器（R2CNTRH，R2CNTRL）的值。

在模式 1 模式下，通过比较 16 位设置寄存器（R2DIVH，R2DIVL）和 32768Hz 计数器（R2CNTRH，R2CNTRL），继续进行直到它们相等，最终得到标准 RC 振荡器计数器（R2CNTSH，R2CNTSL）的值。

$f_T = 1/(R_T*C_0)$ ： 使用温度电阻 RT 时 RC 振荡频率

$f_F = 1/(R_F*C_0)$ ： 使用 10K 标准电阻 RF 时 RC 振荡频率

Mode0 模式: $(R_T*C_0)*N_{DIV} = T_{32K} * N_T$

Mode1 模式: $(R_F*C_0)*N_F = T_{32K} * N_T$

由上面两个等式相等得到 $R_T = R_F * (N_F/N_{DIV})$

R_F = 10K,then $R_T = 10 * (N_F/ N_{DIV})$

两边乘以 250 得到 $250 * R_T = 2500 * (N_F/ N_{DIV}) = N_F$

在上表达式中，左侧 250 * RT 对应于温度表，通过 Mode0，Mode1 模式可快速的获得 250 * RT，而温度测量 CPU 保持睡眠状态，因此可以大大降低消耗电流。

表 12：GC7810A CPU 指令集

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into standby mode	1	00 0000 0110 0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

CPU 指令说明

ADDLW	Add Literal and W
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

ANDLW	AND Literal with W
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2TCY instruction.

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2TCY instruction.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2TCY instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

IORLW	Inclusive OR Literal with W
Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (destination)
Status Affected:	Z
Description:	The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register, since status flag Z is affected.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W)
Status Affected:	None
Description:	The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

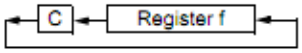
MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	0 ≤ f ≤ 127
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.

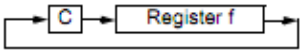


SLEEP	
Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT prescaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

SUBLW	Subtract W from Literal
Syntax:	[*label*] SUBLW k
Operands:	0 ≤ k ≤ 255
Operation:	k - (W) → (W)
Status Affected:	C, DC, Z
Description:	The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

RRF	Rotate Right f through Carry
Syntax:	[*label*] RRF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - (W) → (destination)
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed in register 'f'.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.

XORLW	Exclusive OR Literal with W
Syntax:	[*label*] XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.

10. 订货信息

产品型号	供货方式
GC7810A	SSOP24 封装片

11. 文档修改记录

版本	更改内容（每行一项）	更改日期&更改者（简写）
V10	发布	2017-10-27by lyy
V10	更改一些寄存器的值和电气参数	2017-11-06 by wyq
V10	更改一些电气参数值，主要是工作电流工作频率	2020-12-25 by wyq

如有任何改动，以更新版本为准。