OTP-Based 8-Bit Microcontroller Series

FC1601

SZFC

2021/7/7

功能特色：

*双时钟模式IRC_RTC (IRC for Fcpu/LF for T0)，省外部电容

*32级电压比较器，可比较2个IO或电源电压（通过软件控制）

*二路独立控制定时器（T0\T1）

*三路独立控制PWM输出（PB2\PB1\PB0）,100mA大驱动

*PB3口软件可控上拉，PB3可做标准IO口

*4级硬件控制端口强弱驱动电流

*内置64*8bit的EEPROM

注意事项：

1. 使用EEP时，BANK2中的10-1F地址的寄存器，不能使用。如果用了EEP，需要用到RAM的10-1F地址，需要切BANK。

2. 由于工艺的差异性，烧录软件上配置的**LVDT**复位电压，与芯片实际的复位电压，有至少**±0.2V**以上的误差。

修正记录

版本号	时间	内容
V10	2020年12月	初版
V11	2021年1月	增加10~1F切BANK注意事项，修改E2PCON寄存器
V12	2021年7月	增加注意事项2

OTP-Based 8-Bit Microcontroller Series

- FC1601 : OTP device

功能特性

- 只有42个单字指令
- 除跳转指令为两个周期指令以外其余为单周期指令
- 13-bit指令宽度
- GOTO指令能跳转到所有的ROM/EPROM地址空间
- 子程序能返回到所有的ROM/EPROM地址空间
- 能处理8位数据
- 5级硬件堆栈
- 运行速度: DC-20 MHz 工作频率
  DC-100 ns 指令周期

型号	管脚#	I/O#	EPROM/ROM空间 (Byte)	RAM (Byte)
FC1601	8	6	1K	61

- 支持直接与间接数据寻址方式
- 一个带8位预置器的8位定时/计数器（Timer0）），一个带8位自动重载功能的8位定时/计数器（T1）
- 内部上电复位
- 内含一个低电压检测电路供掉电复位使用，一个高精度电压比较器可检测VDD电压，2个IO口的电压
- 上电复位计数器（PWRT）和振荡启动计数器（Oscillator Start-up Timer OST）
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作
- 1类双向输入输出I/O口 IOB
- 三路独立控制可编程PWM/输出
- 通过编程控制I/O端口的上拉/下拉、开漏等状态
- 两个内部计数中断源；两个外部中断源: INT管脚，PortB的输入改变
- 通过INT管脚或者PortB的输入改变来实现睡眠唤醒
- 省电睡眠模式
- 内部有16MHz，8MHz， 4MHz， 1MHz， 和 455KHz RC振荡器
- 有可靠的保证使得程序代码不被读出。
- 内部RC振荡器
- 提供以下振荡源的选择:
  - ERC: External Resistor/Capacitor Oscillator（外部的RC振荡器）
  - IRC/ERIC: Internal or External Resistor/Internal Capacitor Oscillator –（内部电阻内部的电容RC振荡器或外部的电阻内部的电容RC振荡器）
  - IRC_RTC: Internal or External Resistor/ Low Frequency Crystal Oscillator For T0 RTC
  - HF: High Frequency Crystal/Resonator Oscillator（高频率的晶体振荡器）
  - LF: Low Frequency Crystal Oscillator（低频率的晶体振荡器）
  - XT: Crystal/Resonator Oscillator (晶体/陶瓷振荡器)
- 工作电压范围:2.0V - 5.5V
  -≤4MHZ: 2.4V - 5.5V
  -≤8MHZ : 2.6V- 5.5V
  -≤16MHZ : 3.6V- 5.5V

概叙

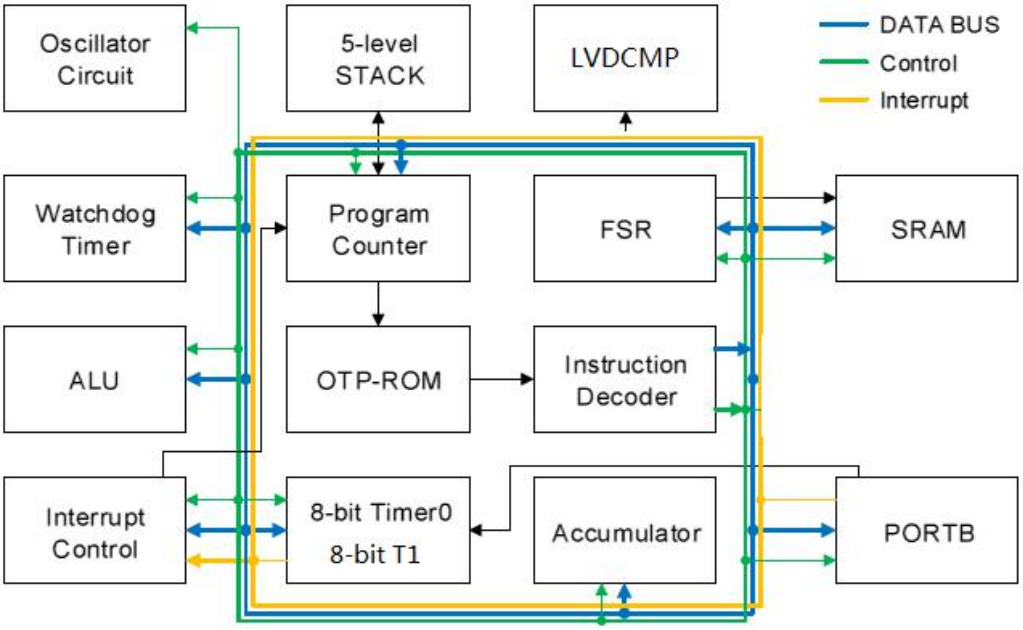
FC1601是一款低功耗，高速，高噪声容限，EPROM/ROM基于8位CMOS工艺制造的单片机，采用RISC指令集，共有42条指令，除分支指令为两个周期指令以外其余为单周期指令。这种易用、易记的指令集大大缩短了开发时间。

FC1601包含了上电复位(Power-on Reset POR)，掉电复位(Brown-out Reset BOR)， 上电复位计数器（Power-up Reset Timer PWRT），振荡启动计数器 (Oscillator Start-up Timer OST)， 看门狗定时器（Watchdog Timer)， EPROM/ROM， SRAM，双向三态I/O口，（可以设置为上拉/下拉、开漏）， 省电睡眠模式， 一个带8位预置器的8位定时/计数器，一个带8位自动重载的8位定时器/计数器，一个高精度电压比较器，独立中断，睡眠唤醒模式和可靠的代码保护，有两个振荡源可供用户配置选择，包含省电振荡源和低功耗振荡器。

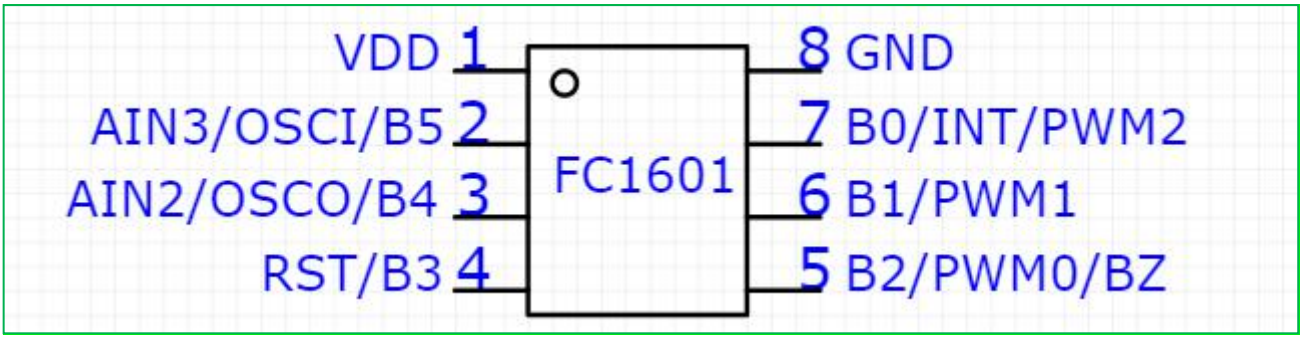
FC1601可访问1K×13的程序存储空间。

FC1601能直接或间接访问寄存器以及数据存储区，所有的特殊功能寄存器分布在数据存储区同时包含特定的程序指针。

结构图



管脚图(PDIP8， SOP8)



管脚功能描述

管脚名称	I/O	说明
IOB0/INT/PWM2	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部中断输入脚 PWM2输出引脚
IOB1/PWM1	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 PWM1输出引脚
IOB2/T0CKI/PWM0	I/O	双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部计数输入脚 Buzzer输出引脚/PWM0输出引脚
IOB3/RSTB	I/O	双向I/O同时具有系统唤醒功能 软件设置为上拉/硬件设置开漏 系统复位脚. 低电平复位.设置为复位脚时上拉自动开启
IOB4/OSCO	I/O	双向I/O口同时具有系统唤醒功能 (RCOUT 可选择IRC/ERIC，ERC模式) 软件设置为上拉/开漏 晶体振荡器输出脚 (HF， LF，IRC_RTC模式) 基于指令周期晶体振荡器输出 (RCOUT 可选择IRC/ERIC，ERC模式)
IOB5/OSCI	I/O	双向I/O口同时具有系统唤醒功能 (IRC 模式) 软件设置为上拉/开漏 晶体振荡器输入脚 (HF， LF，IRC_RTC模式) 外部实时时钟输入脚(ERIC， ERC模式)
AIN2/AIN3	A	电压比较器的输入端，可检测32级端口的电压值
Vdd	-	电源
Vss	-	地

Legend: I=输入， O=输出， I/O=输入/输出，A=模拟输入

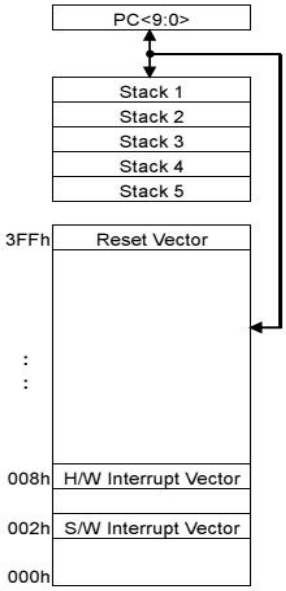
1.0 存储器结构

FC1601 存储器包含程序存储器和数据存储器。

1.1 程序存储器

FC1601有一个10位PC指针能访问1K×13的存储空间。
FC1601的复位地址为3FFh。
H/W中断向量地址008h.， S/W中断向量地址002h。
FC1601的CALL/GOTO能指向在同一个程序页面（一个程序页面为1K）的所有存储空间

程序存储器分布图和堆栈结构



FC1601

1.2数据存储器

数据存储器包含特殊功能器组和通用寄存器组，所有通用寄存器可以直接寻址或者通过FSR寄存器间接寻址。特殊功能寄存器用来控制CPU或外围功能模块的工作。

表 1.1: FC1601寄存器列表

地址	Bank0	Bank1	Bank2
00h	INDF		
01h	TMR0		
02h	PCL		
03h	STATUS		
04h	FSR		
05h	PORTA		
06h	PORTB		
07h	通用寄存器	T1M	
08h	PCON	T1C	
09h	WUCON	T1R	
0Ah	PCHBUF	PWMCON	
0Bh	PDCON	PWM0R	
0Ch	ODCON	PWM1R	
0Dh	PHCON	PWM2R	E2PCON
0Eh	INTEN	SYSM	E2PADR
0Fh	INTFLAG	ADCSEL	E2PDATA
10 - 3Fh	通用寄存器		10 - 1Fh

表1.2: 通过OPTION 或IOST指令控制的寄存器

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	T0TB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
06h (w)	IOSTB	Port B I/O控制寄存器							

表1.3: 寄存器列表

地址	说明	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							
01h (r/w)	TMR0	8位定时/计数器							
02h (r/w)	PCL	低8位PC指针							
03h (r/w)	STATUS	RST	GP1	LVDF	/TO	/PD	Z	DC	C
04h (r/w)	FSR	RP1	RP0	间接地址访问指针（RAM选择寄存器）					
06h (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

地址Bank0	说明	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	SRAM	通用寄存器							
08h (r/w)	PCON	WDTE	EIS	LVDTE					
09h (r/w)	WUCON	WUB7	WUB6	WUB5	WUB4	WUB3	WUB2	WUB1	WUB0
0Ah (r/w)	PCHBUF	-	-	-	-	-	-	2 MSBs Buffer of PC	
0Bh (r/w)	PDCON	/PDB3	/PDB2	/PDB1	/PDB0				
0Ch (r/w)	ODCON			ODB5	ODB4	-	ODB2	ODB1	ODB0
0Dh (r/w)	PHCON			/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0
0Eh (r/w)	INTEN	GIE	*	E2PIE	-	T1IE	INTIE	PBIE	T0IE
0Fh (r/w)	INTFLAG	-	-	E2PIF	-	T1IF	INTIF	PBIF	T0IF

地址Bank1	说明	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	T1M	T1EN	T1PS2	T1PS1	T1PS0	T1CKS-	ALOAD	BUZE	PWM0E
08h (r/w)	T1C	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0
09h (r/w)	T1R	T1R7	T1R6	T1R5	T1R4	T1R3	T1R2	T1R1	T1R0
0Ah (r/w)	PWMCON	PWM0OE	PWM1OE	PWM2OE	-	PWMMD	PWMINV	PWM1E	PWM2E
0Bh (r/w)	PWM0R	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0
0Ch (r/w)	PWM1R	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0
0Dh (r/w)	PWM2R	TC2EN	TC2PS2	TC2PS1	TC2PS0	TC2CKS	ALOAD2	TC2OUT	PWM2EN
0Eh (r/w)	SYSM	-	-	-	-	RTCWP	RTCCON	CLKMD	STOPHX
0Fh (r/w)	ADCSEL	-	-	-	ADCSEL4	ADCSEL3	ADCSEL2	ADCSEL1	ADCSEL0

Legend: - = unimplemented， read as '0'， * = unimplemented， read as '1'， NG= no used bit

地址Bank2	说明	B7	B6	B5	B4	B3	B2	B1	B0
0DH(r/w)	E2PCON	-	-	-	BER	CER	SER	WR	RD
0EH(r/w)	E2PADR	-	-	E2PADR5	E2PADR4	E2PADR3	E2PADR2	E2PADR1	E2PADR0
0FH(r/w)	E2PDATA	E2P读写寄存器							

2.0 功能介绍

2.1 寄存器操作

2.1.1 INDF (间接寻址寄存器)

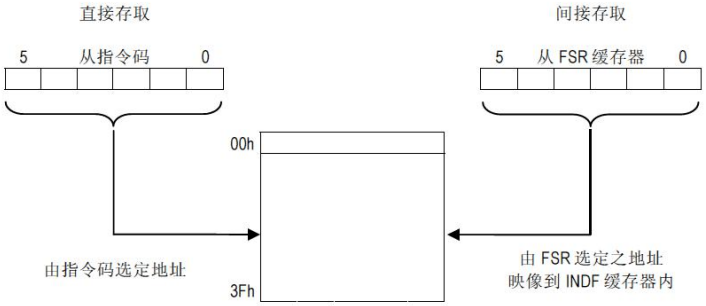
地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	通过FSR 访问数据区(不是一个实际的物理地址)							

INDF不是一个实际的物理地址，间接寻址时INDF通过RAM选择寄存器（FSR）来访问其所指向的地址。间接寻址读操作直接读地址00h(FSR="0")， 间接寻址不能对INDF直接进行写操作（尽管有些状态会发生改变）。
FSR的5-0位可以用来选择64个寄存器（地址:00h ~ 3Fh）。

例 2.1:间接寻址

- 地址38内容为10h
- 地址39内容为0Ah
- 将38写入FSR 中
- 通过A读INDF返回10h
- FSR加1 (@FSR=39h)
- 通过A读INDF返回0A h

图2.1:直接/间接存取



2.1.2 TMR0 (定时/计数器 Time lock/Counter register)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
01h (r/w)	TMR0	8位定时/计数器							

TMR0是一个8位定时/计数器寄存器，Timer0的时钟源可以取值于指令周期或外部实时钟（T0CKI pin），使用外部时钟需要设置OPTION的T0CS(T0CS=5)位为1。
使用TMR0的预置器需要设置OPTION的PSA (PSA =3)位为0 ，这种模式下TMR0值的改变，预置器被清零。

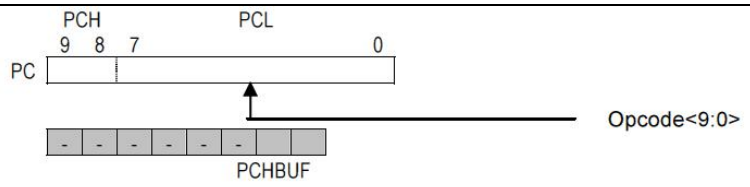
2.1.3 PCL (Low Bytes of Program Counter) & Stack

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
02h (r/w)	PCL	PC低8位							

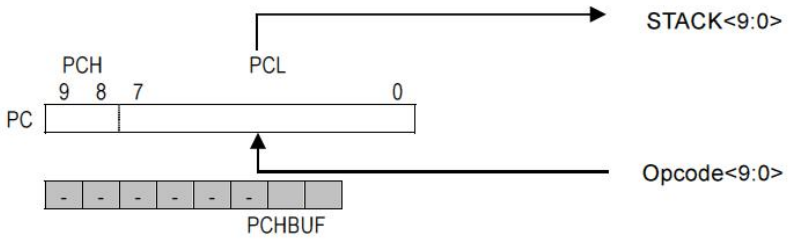
FC1601的PC指针和堆栈的位数为10位， 堆栈有5级，低位的PC指针为PCL寄存器，该寄存器时可读写的，高位的PC指针为PCH寄存器，该寄存器包含PC<9:8> 位，该寄存器不能直接读写。. PCH寄存器的改变是通过PCHBUF寄存器来实现的。.每一条指令执行的时候他的PC指针包含下一条指令的操作地址。指令没有改变PC内容时候、在每一个指令周期PC指针自动加1。
对于GOTO指令有PC<9:0>，PCL 映射成PC<7:0>，PCHBUF不变。
对于CALL指令有PC<9:0>，下一条指令地址被推进堆栈，PCL 映射成PC<7:0>，PCHBUF不变。
对于RETIA， RETFIE， RETURN指令有PC<9:0>，PC的内容更改为出栈信息，PCL 映射成PC<7:0>，PCHBUF不变。
对于其他指令，PCLj就是目标信息， PC<7:0>的内容就是指令地址或。不管怎样， PC<9:8> 来源于 PCHBUF<1:0> 位 (PCHBUF→PCH)。 PCHBUF不会改变，从而PCH不会改变。

图2.2:不同的指令调用PC指针跳转方式

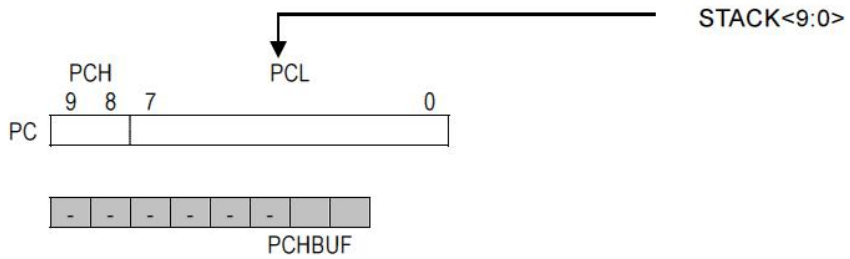
1、GOTO指令



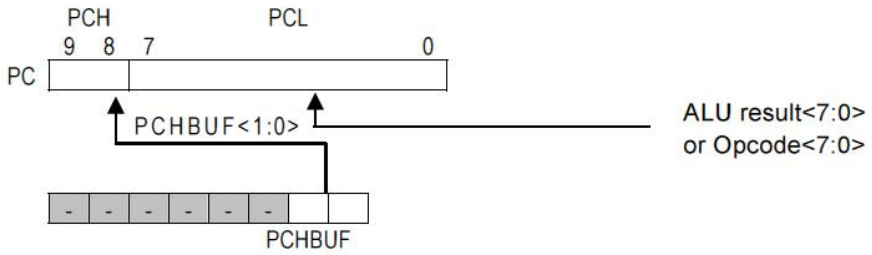
2、CALL**指令**



3、RETIA， RETFIE， RETURN**指令**



4、以PCL为目的的指令



注释1. PCHBUF只有在PCL内容是目标地址才有效，当PCL是运算结果时候，PCHBUF不起作用。

2.1.4 STATUS (状态字寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
03h (r/w)	STATUS	RST	GP1	LVDF	/TO	/PD	Z	DC	C

状态字寄存器包含运算标志，结果标志。

指令执行以后可能会影响STATUS寄存器的Z、DC 、C标志位，则不能直接对这三个标志位进行写操作，这些标志位的设置由MCU的逻辑自动完成。同时，TO 和 PD 位也是不能通过指令直接改变写操作。因此，与STATUS作为目标寄存器的指令后，结果可能会与预期的不同。例如:运行CLRR STATUS将把STATUS 的高三位置零和Z标志位置1 同时该寄存器的内容如下

0	0	0	u	u	1	u	u

u表示为指令执行前后该位没有发生改变

C :进位标志

ADDAR， ADDIA

= 1，有进位

= 0，无进位

SUBAR， SUBIA

= 1，无借位

= 0，有借位

注释 ：减法是通过将2的补第二个操作数的执行。旋转（RRR，RLR）指令，该位装载高或低位源寄存器位。

DC :辅助进位/借位标志.(低四位向高四位进位/借位标志)

ADDAR， ADDIA

= 1， 底4位有进位

= 0， 底4位无进位

SUBAR， SUBIA

= 1， 底4位无借位

= 0， 底4位有借位

Z : 零标志位

= 1， 算术或逻辑运算结果为“0”时

= 0， 算术或逻辑运算结果不为“0”时

/PD :系统休眠标志位

= 1， 当系统上电时或执行“CLRWDT”指令后

= 0， 当执行“SLEEP”指令后

/TO :看门狗溢出标志位

= 1， 当系统上电时或执行“CLRWDT”或SLEEP指令后

= 0，看门狗定时器溢出

LVDF:高精度LVD标志位

= 1，VDD低于高精度LVD电压检测点

= 0，VDD高于高精度LVD电压检测点

GP1 :通用寄存器读/写位

RST :定义系统复位类型位.

= 1， 唤醒SLEEP或Port B脚位变化唤醒SLEEP

= 0， 其他类型唤醒SLEEP.

2.1.5 FSR (间接寻址指针)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
04h (r/w)	FSR	RP1	RP0	间接寻址指针					

Bit5:Bit0 : 用来选择访问间接寻址时目标寄存器地址. 具体描述见2.1.1。

Bit7:Bit6 : 用来选择控制寄存器的页，控制寄存器页选择只对地址07H~0FH有效，地址00H~06H和控制寄存器页选择无关。

= 00， 选择bank0
= 01， 选择bank1
= 10， 选择bank2
= 11， 没有使用

注：写入RP1，RP0寄存器时，只能BSR,BCR指令，其他指令无效。

2.1.6 PORTA， PORTB (Port 寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
05h (r/w)	PORTA	-	-	-	-	-	-	-	-
06h (r/w)	PORTB	-	-	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

读端口(PORTB 寄存器)的状态依赖于该端口是输入/输出模式，写端口是向锁存器写数据。
PORTB是一个8位端口数据寄存器，只有低6位有效。

2.1.7 PCON (电源控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	PCON	WDTE	EIS	LVDTE	HCMPEN	-	-	HCMPIN1	HCMPIN0

HCMPIN1：HCMPIN0（高精度电压检测输入端）

0x VDD 检测电源VDD的值

10 PB4 检测PB4脚位的电压

11 PB5 检测PB5脚位的电压

B3~B2：未用，可做通用寄存器

HCMPEN : 高精度电压检测使能位
= 0， 禁止高精度电压检测
= 1， 使能高精度电压检测

LVDTE : LVDT (低电压检测) 使能位
= 0， 关闭 LVDT
= 1， 使能 LVDT

EIS : 定义管脚B0/INT功能位
= 0， IOB0 (双向I/O 口) is selected. 屏蔽了INT功能.
= 1， INT (外部中断输入脚) ，在这种模式下，PORTB 的IOB0必须置“1”. IOB0作为I/O口输入功能通过硬件屏蔽了，读取INT管脚信息的与读PORTB.方式相同

WDTE : WDT (watch-dog timer) 使能看门狗定时器
= 0，关闭WDT
= 1，使能WDT

2.1.8 WUCON (Port B输入改变/唤醒控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	WUCON			WUB5	WUB4	WUB3	WUB2	WUB1	WUB0

WUB0 : = 0，禁止IIOB0 输入改变/唤醒功能
=1，使能IOB0 输入改变/唤醒功能

WUB1 : = 0，禁止IIOB1 输入改变/唤醒功能
=1，使能IOB1 输入改变/唤醒功能

WUB2 : = 0，禁止IIOB2 输入改变/唤醒功能
=1，使能IOB2 输入改变/唤醒功能

WUB3 : = 0，禁止IIOB3 输入改变/唤醒功能
=1，使能IOB3 输入改变/唤醒功能

WUB4 : = 0，禁止IIOB4 输入改变/唤醒功能
=1，使能IOB4 输入改变/唤醒功能

WUB5 : = 0，禁止IIOB5 输入改变/唤醒功能
=1，使能IOB5 输入改变/唤醒功能

2.1.9 PCHBUF (PC指针高位缓冲区)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	PCHBUF	-	-	-	-	-	-	2MSBs 缓冲	

Bit1:Bit0 : 见2.1.3

Bit7:Bit2 : 没有使用，置 0

2.1.10 PDCON (I/O下拉控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PDCON	/PDB3	/PDB2	/PDB1	/PDB0	-	-	-	-

/PDB0: = 0，使能IOB 0内部下拉
=1，禁止IOB0内部下拉

/PDB1: = 0，使能IOB1内部下拉
=1，禁止IOB1内部下拉

/PDB2: = 0，使能IOB2内部下拉
=1，禁止IOB2内部下拉

/PDB3: = 0，使能IOB3内部下拉
=1，禁止IOB3内部下拉

2.1.11 ODCON (I/O开漏控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	ODCON			ODB5	ODB4	-	ODB2	ODB1	ODB0

ODB0 : = 0，禁止IOB0内部开漏
=1，使能 IOB0内部开漏

ODB1 : = 0，禁止IOB1内部开漏
=1，使能 IOB1内部开漏

ODB2 : = 0，禁止IOB2内部开漏
=1，使能 IOB2内部开漏

Bit3 : 没有使用，置0

ODB4 : = 0，禁止IOB4内部开漏
=1，使能 IOB4内部开漏

ODB5 : = 0，禁止IOB5内部开漏
=1，使能 IOB5内部开漏

2.1.12 PHCON (I/O上拉控制寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	PHCON			/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0

/PHB0 : = 0，使能IOB0内部上拉.
= 1，禁止IOB0内部上拉

/PHB1 : = 0，使能IOB1内部上拉.
= 1，禁止IOB1内部上拉

/PHB2 : = 0，使能IOB2内部上拉
= 1，禁止IOB2内部上拉

/PHB3: = 0，使能IOB3内部上拉
= 1，禁止IOB3内部上拉

/PHB4 : = 0，使能IOB4内部上拉.
= 1，禁止IOB4内部上拉

/PHB5 : = 0，使能IOB5内部上拉
= 1，禁止IOB5内部上拉

2.1.13 INTEN (中断屏蔽寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	INTEN	GIE	*	*	*	T1IE	INTIE	PBIE	T0IE

T0IE : Timer0溢出中断屏蔽位。
= 0，禁止Timer0溢出中断
= 1，使能Timer0溢出中断

PBIE : Port B输入改变中断屏蔽位
= 0，禁止Port B输入改变中
= 1，使能Port B输入改变中

INTIE : 外部中断屏蔽位
= 0，禁止外部中断.
= 1，使能外部中断

T1IE : T1溢出中断屏蔽位。
= 0，禁止T1溢出中断
= 1，使能T1溢出中断

Bit6:Bit5:没有使用，置1

GIE : 中断允许总控位
= 0，禁止所有中断. 对于睡眠唤醒模式的中断事件，MCU将执行SLEEP后的指令。
= 1，使能所有没有屏蔽的中断. 对于睡眠唤醒模式的中断事件，MCU将跳转到中断地址（008h）。
注释: 在中断事件发生时， GIEB被硬件清零并禁止一切中断，所以GIE以及与该中断相关的中断屏蔽位需要重开启。RETFIE 为退出中断程序并重新设置GIE =1允许中断。

2.1.14 INTFLAG (中断标志寄存器)

地址Bank0	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Fh (r/w)	INTFLAG	-	-	-	-	T1IF	INTIF	PBIF	T0IF

T0IF : 溢出中断标志，发生Timer0溢出中断置1， 软件设置清零

PBIF : Port B输入改变中断标志 interrupt flag. Port B输入改变时置1， 软件设置清零

INTIF : 外部中断标志. 当管脚INT上升沿/下降沿（是上升沿/下降沿由 INTEDG 位 (OPTION<6>)决定）时置1， 软件设置清零

T1IF : 溢出中断标志，发生T1溢出中断置1， 软件设置清零

Bit7:BIT5 : 没有使用，置0

2.1.15 T1M(T1控制寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
07h (r/w)	T1M	T1EN	T1PS2	T1PS1	T1PS0	T1CKS	ALOAD	BUZE	PWM0E

PWM0E: PWM0输出控制
= 0，禁止PWM0
= 1，使能PWM0

BUZE: T1溢出输出信号控制，仅当PWM0E=0时有效
= 0，禁止，IOB2为GPIO引脚
= 1，使能，IOB2输出TC0OUT信号

ALOAD: 自动装载控制
= 0，禁止T1自动装载
= 1，使能T1自动装载

T1CKS: T1时钟选择
= 0，T1 选择Fcpu作为时钟输入
= 1，T1 选择Fhosc作为时钟输入

T1PS[2:0]: TC0分频选择位
= 000，Ft1/128
= 001，Ft1/64
= 010，Ft1/32
= 011，Ft1/16
= 100，Ft1/8
= 101，Ft1/4
= 110，Ft1/2
= 111，Ft1/1

T1EN: T1启动控制位
= 0，禁止T1定时器
= 1，开启T1定时器

2.1.16 T1C(T1计数寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	T1C	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0

8位计数器T1C溢出时，T1IF置1并由程序清零，用来控制T1的中断间隔时间。首先须写入正确的值到T1C和T1R寄存器，并使能T1定时器以保证第一个周期正确。T1溢出后，T1R的值自动装入T1C。

T1C初始值的计算公式如下:

T1C初始值 = N - （T1中断间隔时间 * T1时钟rate ）

2.1.17 T1R(T1自动装载寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	T1R	T1R7	T1R6	T1R5	T1R4	T1R3	T1R2	T1R1	T1R0

T1内置自动重装功能，T1R寄存器存储重装值。T1C溢出时，T1R的值自动装入T1C中。T1定时器工作在计时模式时，要通过修改T1R寄存器来修改T1的间隔时间，而不是通过修改T1C寄存器。在T1定时器溢出后，新的T1C值会被更新，T1R会将新的值装载到T1C寄存器中。但在初次设置T1M时，必须要在开启T1定时器前把T1C以及T1R设置成相同的值。

T1为双重缓存器结构。若程序对T1R进行了修改，那么修改后的T1R值首先被暂存在T1R的第一个缓存器中，T1溢出后，T1R的新值就会被存入T1R缓存器中，从而避免T1中断时间出错。

2.1.18 PWMCON(PWM控制寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	PWMCON	PWM0OE	PWM1OE	PWM2OE	-	PWMMD	PWMINV	PWM1E	PWM2E

PWM2E: PWM2输出控制
= 0，禁止PWM2
= 1，使能PWM2

PWM1E: PWM1输出控制
= 0，禁止PWM1
= 1，使能PWM1

PWMINV: PWM逻辑操作选择（PWMMD=1有效）
= 0，PWM1同或PWM2
= 1，PWM1异或PWM2

PWMMD: PWM输出选择
= 0，IOB1输出PWM1同时IOB0输出PWM2
= 1，IOB1输出PWM0同时IOB0输出（PWM1异或PWM2）或（PWM1同或PWM2）

Bit4: 未使用

PWM2OE: PWM2输出选择
= 0，禁止PWM2输出，IOB0作为I/O
= 1，使能PWM2输出，IOB0输出PWM2信号

PWM1OE: PWM1输出选择
= 0，禁止PWM1输出，IOB1作为I/O
= 1，使能PWM1输出，IOB1输出PWM1信号

PWM0OE: PWM0输出选择
= 0，禁止PWM0输出，IOB2作为I/O
= 1，使能PWM0输出，IOB2输出PWM0/BUZ信号

2.1.19 PWM0R(PWM0占空比寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PWM0R	PWM0R7	PWM0R6	PWM0R5	PWM0R4	PWM0R3	PWM0R2	PWM0R1	PWM0R0

用于设置PWM0高电平时间，PWM0高电平时间= PWM0R - T1R*ALOAD

2.1.20 PWM1R(PWM1占空比寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	PWM1R	PWM1R7	PWM1R6	PWM1R5	PWM1R4	PWM1R3	PWM1R2	PWM1R1	PWM1R0

用于设置PWM1高电平时间，PWM1高电平时间= PWM1R - T1R*ALOAD

2.1.21 PWM2R(PWM2占空比寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	PWM2R	PWM2R7	PWM2R6	PWM2R5	PWM2R4	PWM2R3	PWM2R2	PWM2R1	PWM2R0

用于设置PWM2高电平时间，PWM2高电平时间= PWM2R - T1R*ALOAD

2.1.22 ACC (Accumulator)累加器

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (r/w)	ACC	累加器							

累加器是一个内部数据转化、指令操作和存放操作结果的存储单元，不能被访问。

2.1.23 OPTION Register（选项寄存器）

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	T0TB	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

通过OPTION 指令访问，在执行OPTION 指令时候，该数据单元由ACC（累加器） 转化为选项寄存器（OPTION Register）。选项寄存器是一个7位只写寄存器，它的一些控制位主要用来配置与Timer0/WDT 分频器， Timer0， 外部中断选项相关信息。除INTEDG位以外其他位是只写并可以置1。

PS2:PS0 : 分频率选择控制位

PS2:PS0	Timer0 Rate	WDT Rate
0 0 0	1:2	1:1
0 0 1	1:4	1:2
0 1 0	1:8	1:4
0 1 1	1:16	1:8
1 0 0	1:32	1:16
1 0 1	1:64	1:32
1 1 0	1:128	1:64
1 1 1	1:256	1:128

PSA : 分频器选择位.
= 1， WDT (看门狗定时器)
= 0， TMR0 (Timer0)

T0SE : TMR0触发方式控制位
= 1， T0CKI脚下降沿触发计数
= 0， T0CKI脚上升沿触发计数

T0CS : TMR0 时钟源选择控制位
= 1， 外部T0CKI脚. 当IOST IOB2 = “0”.时，IOB2/T0CKI脚设置为输入
= 0， 内部指令时钟周期或者RTC时钟

INTEDG :中断触发方式控制位.
= 1， 中断触发方式为INT脚上升沿出发
= 0， 中断触发方式为INT脚下降沿出发

T0TB :T0时钟RTC选择位.
= 1， T0时钟选择RTC时钟（外部32768晶振）
= 0， T0时钟选择内部指令时钟

2.1.24 IOSTB (I/O口控制寄存器)

地址	名称	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	IOSTB	-	-	IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

通过IOST指令访问，通过指令IOST R (06h)把累加器A的内容加载到I/O控制寄存器，按位将IOSTB设为1表示该脚为输入（高阻抗）、设为0时表示该脚为输出。
IOST寄存器只写，系统复位以后设置为输入（高阻抗）。

2.1.25 SYSM (系统工作模式控制寄存器)

地址Bank1	名称	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	SYSM	-	-	-	--	RTCWP	RTCCON	CLKMD	STOPHX

STOPHX: 高速振荡器控制
= 0，振荡器振荡
= 1，有效（从睡眠模式唤醒时自动清0）,振荡器停振

CLKMD: 系统工作模式选择
= 0，高速模式。
= 1，低速模式。系统时钟是内部ILRC的2分频，16K左右。

RTCCON: RTC模式时的睡眠模式
= 0，睡眠时 RTC的外置振荡器也关闭。
= 1，一直打开外置振荡器

RTCWP: RTC可唤醒控制
= 0，睡眠时，T0处于RTC模式无法唤醒
= 1，可唤醒

2.1.16 E2PCON(E2P控制寄存器)

地址bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0D (R/W)	E2PCON				BER	CER	SER	WR	RD

Bit6~bit5：读时数据为0。

RD: E2P的read使能。

0：未读。

1：read启动，自动清0。此标志位不用查询，置1后，可以直接读E2PDATA寄存器的值。

CER:E2P的CHIP擦除使能。擦除整个E2P的数据。

0：擦除完成。

1：擦除启动，擦完后自动清0

SER: E2P的SECTOR擦除使能。

0：擦除完成。

1：擦除启动，擦完后自动清0

注：16个byte为一页，共4页，每页地址由,E2PADR的高2位地址确定。

BER: E2P的BYTE擦除使能。

0：擦除完成。

1：擦除启动，擦完后自动清0

注：此位是byte擦除使能，可以擦除E2PADR指定地址的数据。

WR: E2P的WRITE使能。

0：编程完成。

1：编程启动，擦完后自动清0。普通模式编程需要1ms。

2.1.16.1 E2PADR(E2P地址寄存器)

地址bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0E (R/W)	E2PADR	E2PADR7	E2PADR6	E2PADR5	E2PADR4	E2PADR3	E2PADR2	E2PADR1	E2PADR0

Bit7:BIT0: 内置E2P地址，地址000H~3FH共64字节。高2位无效，可做通用寄存器

2.1.16.2 E2PDATA(E2P数据寄存器)

地址bank2	名称	B7	B6	B5	B4	B3	B2	B1	B0
0F (R/W)	E2PDIN	E2PDIN7	E2PDIN6	E2PDIN5	E2PDIN4	E2PDIN3	E2PDIN2	E2PDIN1	E2PDIN0

E2P数据输入寄存器，在烧写内置E2P前将要写数据写入E2PDIN寄存器中。
E2P数据读出寄存器。读E2P时，直接读入ACC中.如，MOVR E2PDAT,0

E2P例程：此案例为用到了RAM中10-1F地址的寄存器，需要注意切BANK

```
#define    BER    E2PCON,4
#define    CER    E2PCON,3
#define    SER    E2PCON,2
#define    WR     E2PCON,1
#define    RD     E2PCON,0
```

```
E2PCON          EQU     0DH
E2PADR          EQU     0EH
E2PDIN          EQU     0FH
E2PDATA         EQU     0FH
;;=============================================================
;; USE RAM EQUISTER ( 10H - 3FH )
;;=============================================================
E2_ADDR         REG     010H
E2_DATA         REG     011H

;*****************************************************
;片擦除(fc):
;*****************************************************
CLR_CHIP_E2:
                BSR     RP1
                BSR     E2PCER
CHECK_CHIP:
                CLRWDT
                BTRSC   E2PCER
                GOTO    CHECK_CHIP
                BCR     RP1
                RETURN
;;=============================================================
;;      向24c02写一字节数据，WR_DATA_8BIT为写入数据
;;=============================================================
WRITE_1_BYTE:
                BCR     GIE
                MOVR    E2_ADDR,A
                BSR     RP1
                MOVAR   E2PADR
LOOP_BYTE1:
                BSR     E2PBER
CHECK_BYTE1:
                CLRWDT
                BTRSC   E2PBER
                GOTO    CHECK_BYTE1
                BCR     RP1
                CALL    DELAY_2US
                MOVR    E2_DATA,A
                BSR     RP1
                MOVAR   E2PDIN
LOOP_WR:
                BSR     E2PWR
CHECK_WR:
                CLRWDT
                BTRSC   E2PWR
                GOTO    CHECK_WR
                BCR     RP1     ;
                BSR     GIE
                RETURN

;;=============================================================
;;
;;=============================================================
READ_1_BYTE:
                BCR     GIE
                MOVR    E2_ADDR,A
                BSR     RP1
                MOVAR   E2PADR
```

```
                BSR         E2PRD
                NOP
                NOP
                NOP
                MOVR        E2PDIN,A
                BCR         RP1
                MOVAR       E2_DATA
                BSR         GIE
                RETURN

DELAY_2US :
                CLRWDT
                GOTO        $+1
                RETURN
```

E2P例程：此案例为未用到了RAM中10-1F地址的寄存器

```
;--------------------------寄存器定义-----------------------------------
//FSR     04h
#define         BANK02          FSR,7
#define         BANK01          FSR,6
;BANK2
E2PCON          EQU         0Dh
E2PADR          EQU         0Eh
E2PDATA         EQU         0Fh

#define         BER             E2PCON,4
#define         CER             E2PCON,3
#define         SER             E2PCON,2
#define         WR              E2PCON,1
#define         RD              E2PCON,0
 ;-------------------------------------------------------------
                BSR         BANK02
                MOVIA       0X02
                MOVAR       E2PADR          ;对E2P的第二个地址编程。
;==========================================================================
;页擦除，当前值下为第一页。擦除后，可以全部写完后再下次擦除。更新数据时也可擦除。
  LOOP_SECTCE:
                BSR         SER
  CHECK_SER:
                BTRSC       SER
                GOTO        CHECK_SER   ;需要100ms擦除，小心wdt复位
;byte和chip擦除与sector类似。
 ;==========================================================================
;编程
                MOVIA       0X02
                MOVAR       E2PDATA         ;对E2P的地址编程,编程的数据为02h。
LOOP_WR:
                BSR         WR
CHECK_WR:
                BTRSC       WR
                GOTO        CHECK_WR    ;时间约1ms
;==========================================================================
;READ
                BSR         RD              ;启动读
                ;可插入NOP指令，NOP越多，表示READ的时间越长
                MOVR        E2PDATA,0       ;E2P的数据送入A;
                MOVAR       TEMP1           ;数据存入TEMP1
```

2.2 I/O Ports

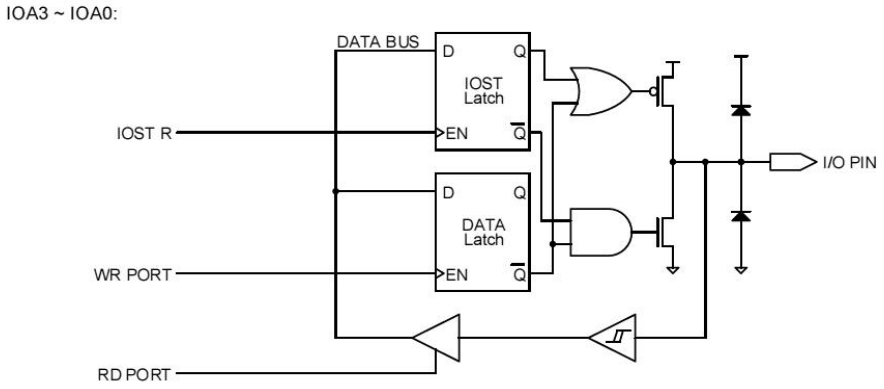
PortB为双向三态I/O 口。 Port B 为5脚I/O口。IOB2需要通过选项寄存器（Option） 的T0CS（(OPTION<5>)）位控制，所有的I/O的输入/输出方式由I/O控制寄存器(IOSTB)设置。 IOB<5:0>有相应的上拉控制位(PHCON 寄存器)来设置使能内部上拉 ，如果设置为输出模式，内部上拉功能会自动关闭。IOB<2:0>有相应的下拉控制位(PDCON寄存器)来设置使能内部下拉，如果设置为输出模式，内部下拉功能会自动关闭。IOB<5:4>和IOB<2:0>有相应的开漏控制位(ODCON寄存器)来设置使能开漏来设置输出为开漏输出。

IOB<5:0> 有输入改变中断/唤醒功能.它的每个管脚是否具有该功能通过取决于WUCON寄存器的相应位。

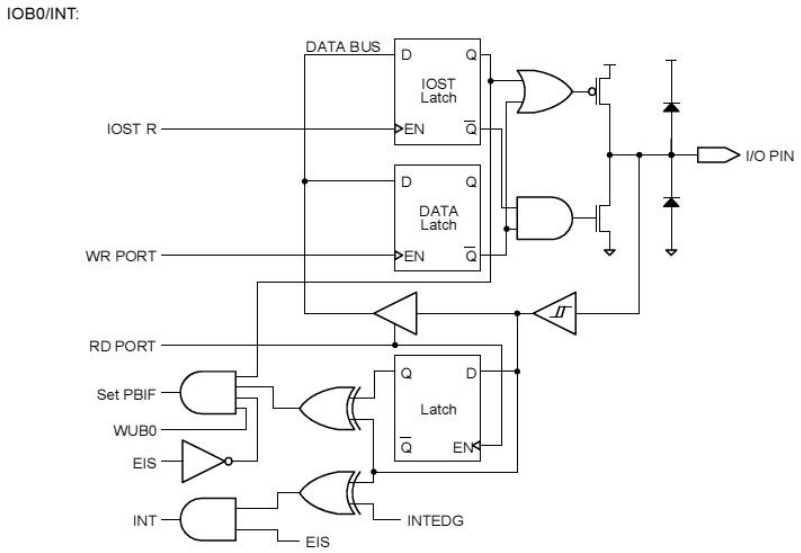
当EIS(PCON<6>)=1 时， IOB0作为外部中断输入脚，在该模式下IOB0 输入改变中断/唤醒功能被硬件屏蔽，即使软件已经设置为中断/唤醒功能可用也不可启用该功能。

配置字能交替设置I/O口的不同功能，功能交替设置完以后，读的I/O的值为0。

图2.3: I/O 脚的结构图

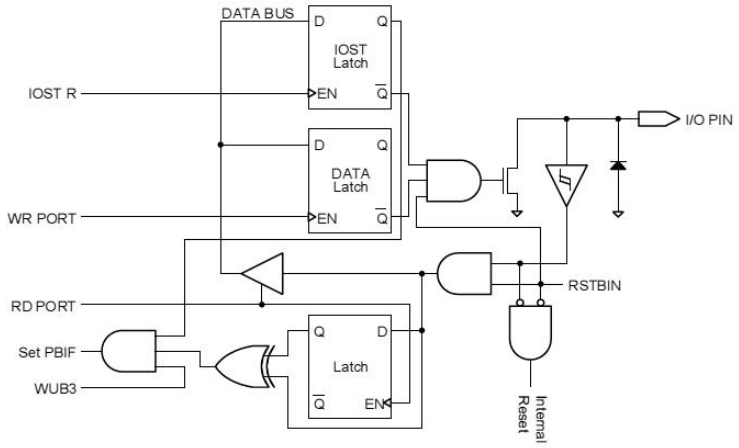


下拉在图中未显示



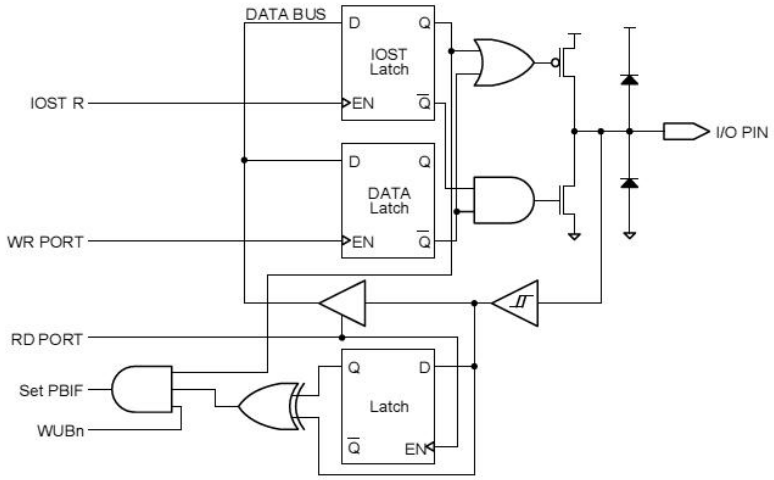
上拉/下拉和漏极开漏在图中未显示

IOB3:



电压在这个引脚禁止超过**VDD**

IOB7 ~ IOB4, IOB2 ~ IOB1:



上拉/下拉和漏极开漏在图中未显示

2.3 Timer0/WDT & Prescler/T1

2.3.1 Timer0

Timer0为8位定时/计数器， Timer0 的时钟源可以是内部或外部时钟源(T0CKI pin)

2.3.1.1 使用内部时钟: 定时模式

T0CS(OPTION<5>)=0为定时模式，定时模式在没有预置器的情况下，定时寄存器每个指令周期自动加1，设置TMR0以后，定时器将在两个时钟周期以后开始自增。

2.3.1.2 使用外部时钟: 计数模式

T0CS(OPTION<5>)=1为计数模式，是选择通过T0CK管脚的上升或下降沿触发Timer0寄存器的增加由T0SE 位(OPTION<4>)决定，外在时钟要求与内部时钟(Tosc)同步。同步以后，Timer0 实际增加有一个延迟。
在没有预置器的情况下，外部时钟输入同样也可以作为预置器输出；T0CKI与内部时钟同步时能方便处理在T2 和 T4周期上的预分频. 因此T0CKI为高或低电平必须要保持两个以上时钟周期才有效。
有预置分频时器，外部时钟输入被异步分频器平分，这种常用来计算波形。因此:因此T0CKI的一个波形周期至少4Tosc才能被 预置器平分。

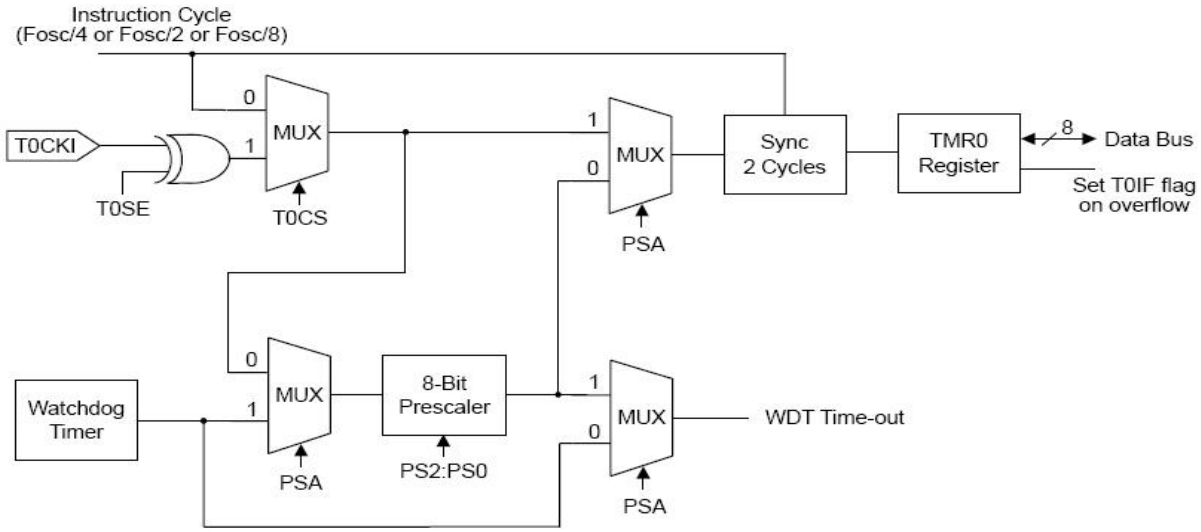
2.3.2 看门狗定时器 (WDT)

看门狗定时器（WDT）的运行依赖于芯片里的RC振荡器，无需任何额外电路即能工作。不管时钟OSCI和OSCO管脚是否关闭，它都能运行，如在睡眠模式。在一般操作或睡眠模式情况下，看门狗定时器的溢出都会导致MCU复位同时TO (STATUS<4>)位被清零。.
如WDTE 位(PCON<7>)清零. 看门狗定时器不能工作。
在没有预置器时看门狗的溢出为18 ms， 4.5ms， 288ms ， 72ms这个时间可以通过SUT<1:0> 设置。
需要看门狗的t溢出周期变长可以通过设置OPTION寄存器的看门狗定时器分频大于1:128.，因此最长的看门狗溢出周期为 36.8 秒。.
CLRWDT指令能使WDT和预置器清零，启用看门狗可以防止超时，如果超时MCU能复位。
SLEEP 指令重置WDT和预置器， 启用看门狗就给机器分派了一个最大睡眠时间。

2.3.3 Prescaler（预置器）

有一个8位的向下计数器作为Timer0和看门狗定时器(WDT)的预置器。注意该预置器只能分配给Timer0 或 WDT使用，不能两者同时使用。PSA 位(OPTION<3>) 决定预置器是指派给Timer0还是WDT. PS<2:0> 位(OPTION<2:0>) 配置分频。当作为Timer0的预置器的时候， TMR0会被预置器清零。. 当作为WDT的预置器的时候， CLRWD 指令会清除预置器内容。预置器不能读写，机器复位，预置器各位全为1。
为了避免机器非正常复位，当Timer0 或 WDT的预置器发生改变的时候，需要执行CLRWDT 或 CLRR TMR0 指令，反之亦然。

图2.4: Timer0/WDT Prescaler结构图



2.3.4 T1

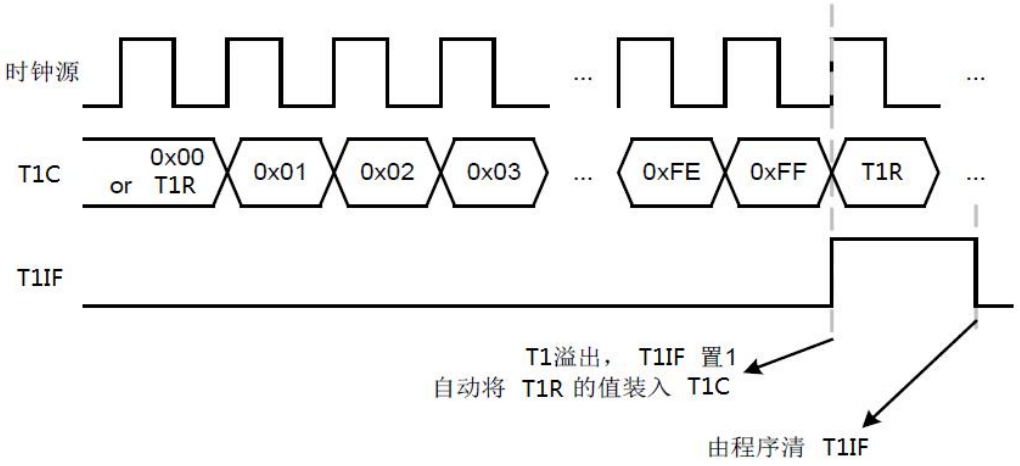
8位二进制定时/计数器具有基本定时器、Buzzer和PWM功能。基本定时器功能可以支持标志显示（T1IF）和中断操作（中断向量）。由T1M、T1C、T1R寄存器控制T1的中断间隔时间。T1还内置Buzzer和PWM功能，Buzzer和PWM的周期和分辨率由T1PS[2:0]、T1R寄存器控制，故具有良好性能的Buzzer和PWM可以处理IR载波信号，马达控制和光度调节等。
T1的主要用途如下：

1. 8位可编程定时器：根据选择的时钟信号，产生周期中断；
2. 中断功能：T1定时器支持中断，当T1溢出时，T1IF置1，当T1IE=1和GIE=1时系统执行中断；
3. PWM输出：由T1PS[2:0]、T1R 和PWM0R、PWM1R、PWM2R寄存器控制占空比/周期；
4. Buzzer输出：Buzzer的输出信号是T1间隔时间的1/2个周期；

2.3.4.1 T1定时模式

T1定时器由T1EN控制。当T1EN=0时，T1停止工作；当T1EN=1时，T1开始计数。使能T1之前，先要设定好T1的功能模式，如基本定时器、T1中断等。T1C溢出（从0FFH到00H）时，T1IF置1以显示溢出状态并由程序清零。在不同的功能模式下，T1C不同的值对应不同的操作，若改变T1C的值影响到操作，会导致功能出错。T1内置双重缓存器以避免此种状况的发生。在T1C计数的过程中不断的刷新T1C，保证将最新的值存入T1R（重装缓存器）中，当T1溢出后，T1R的值由自动存入T1C。进入下一个周期后，T1进入新的工作状态。定时器模式下，由ALOAD控制自动重装功能；PWM模式下，使能T1时，可选择打开或关闭T1的自动重装功能。如果使能T1中断功能（T1IE=1），在T1溢出时系统执行中断服务程序，在中断时必须由程序清T1IF。

图2.5:T1定时溢出示意图

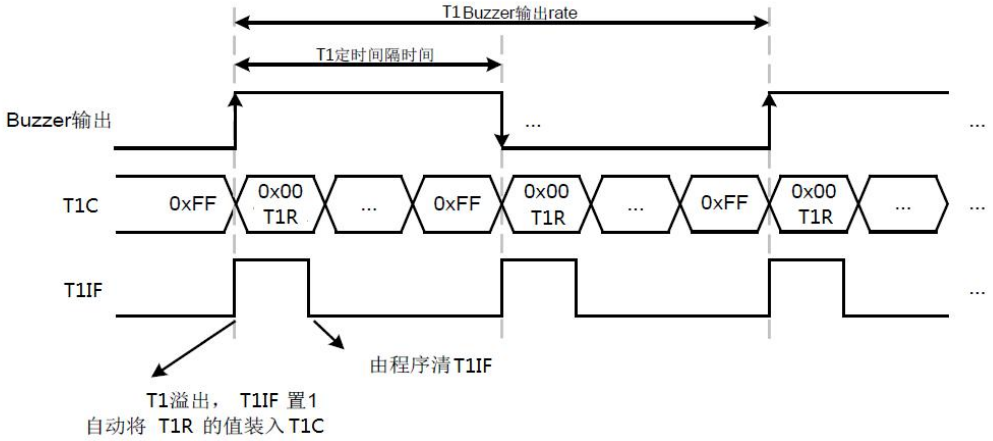


T1PS[2:0]	T1时钟	T1间隔时间			
		Fhosc=16MHz Fcpu=Fhosc/4 (4T)		Fhosc=4Mhz Fcpu=Fhosc/4 (4T)	
		max.(ms)	Unit(us)	max.(ms)	Unit(us)
000b	Ft1/128	8.192	32	32.768	128
001b	Ft1/64	4.096	16	16.384	64
010b	Ft1/32	2.048	8	8.192	32
011b	Ft1/16	1.024	4	4.096	16
100b	Ft1/8	0.512	2	2.048	8
101b	Ft1/4	0.256	1	1.024	4
110b	Ft1/2	0.128	0.5	0.512	2
111b	Ft1/1	0.064	0.25	0.256	1

2.3.4.2 T1 BUZZER输出

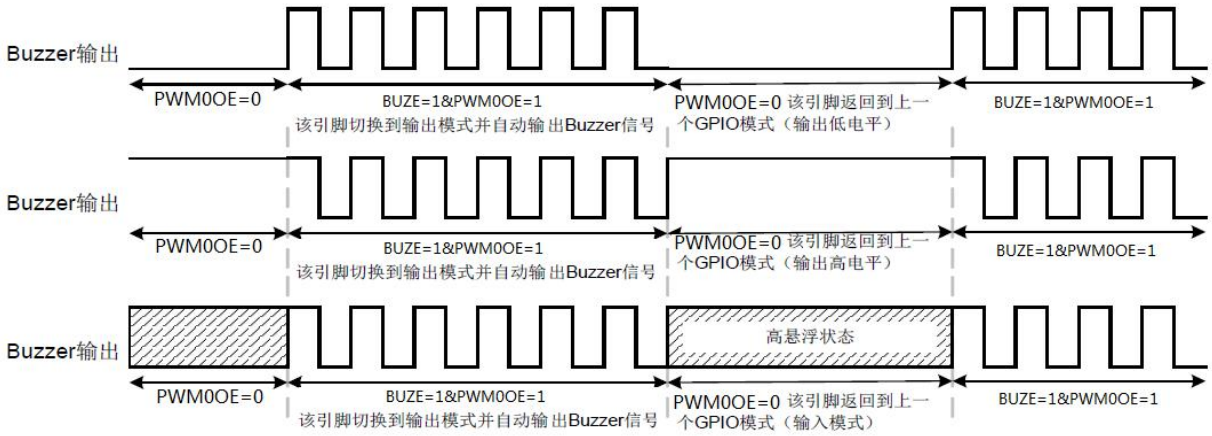
Buzzer输出是一个简单的1/2占空比信号输出，由T1产生。当T1溢出时，Buzzer开始输出一个方波，中断间隔时间频率2分频后作为Buzzer输出的频率。Buzzer输出的波形图如下所示：

图2.6:Buzzer输出示意图



T1溢出后，Buzzer输出时，T1IF有效，且当T1IE=1时，使能T1中断功能。但强烈建议小心同时使用Buzzer和T1定时器，以确保两种功能都能正常工作。
Buzzer输出引脚与GPIO引脚共用，BUZE=1&PWM0OE=1时，该引脚自动设为Buzzer输出引脚。如清BUZE位以禁止Buzzer（BUZE=0&PWM0OE=0）输出后，该引脚自动返回到最后一个GPIO模式。

图2.7:Buzzer输出/IO切换状态示意图



2.3.4.2 T1 PWM输出（脉宽调制）

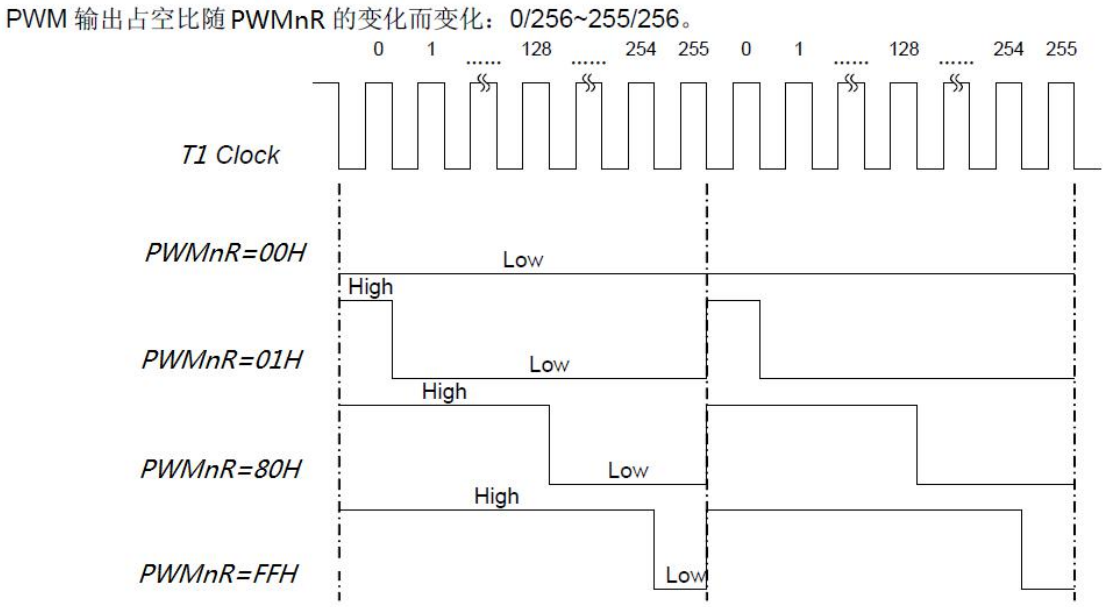
可编程控制占空比/周期的PWM可以提供不同的PWM信号。使能T1定时器且PWM0E=1&PWM0OE=1时，由PWM0输出引脚（IOB2）输出PWM0信号；PWM1E=1&PWM1OE=1时，由PWM1输出引脚（IOB1）输出PWM1信号；PWM2E=1&PWM2OE=1时，由PWM2输出引脚（IOB0）输出PWM2信号。PWM首先输出高电平，然后输出低电平。T1PS[2:0]、ALOAD和T1R控制PWM的周期，ALOAD和T1R决定PWM的分辨率，PWM0R、PWM1R、PWM2R和T1R寄存器决定PWM的占空比（脉冲高电平的长度）。开启T1定时器且定时器溢出后，T1C的初始值为0。当T1C=PWMnR时，PWM输出低电平；T1溢出时（T1C的值从0FFH到00H或T1R），整个PWM周期完成，并进入下一个周期。T1溢出时，PWM的一个周期完成。在PWM输出的过程由程序更改PWM的占空比，则在下一个周期开始输出新的占空比的PWM信号。

PWM周期时间= (256 - T1R*ALOAD) x T1计数周期

PWM 高电平时间= (PWMnR - T1R*ALOAD) x T1计数周期

PWM占空比= (PWMnR - T1R*ALOAD) / (256 - T1R*ALOAD)

图2.8:PWM输出示意图



PWM的分辨率由ALOAD和T1R决定，以实现高速PWM信号。当ALOAD = 0时，PWM的分辨率为1/256；ALOAD = 1时，PWM的分辨率为1 /（256 - T1R）。若需调制PWM的分辨率，T1R PWM的占空比控制范围必须调制到一个合适的分辨率。PWM输出过程中，T1溢出时，T1IF有效，T1IE=1时，则使能T1中断。但强烈建议小心同时使用PWM和T1定时器功能，保证两种功能都能正常工作。

PWM输出引脚和GPIO引脚共用，PWMnE=1&PWMnOE=1时，该引脚自动输出PWM信号。如果清PWMnOE位以禁止PWM时，该引脚返回到最后一个GPIO模式。

2.4 中断方式

FC1601系统具备有四种中断方式:
1. INT 管脚的外部中断
2. TMR0 溢出中断
3. Port B 输入改变中断 (IOB7:IOB0脚)
4. T1 溢出中断

INTFLAG为中断标志寄存器，决定该寄存器机器所发生的中断状态。
中断允许总控位GIE（INTEN<7>）， 能使所有中断被开放（GIE=1）或屏蔽所有中断(GIE=0)，每中断能否启用决定INTEN 寄存器同时保证GIE=1。
中断发生时GIE 位（在中断发生前GIE 位和该中断相关的中断屏蔽位置1）被硬件清零从而禁止进一步中断（FC1601不区分中断优先级别），同时下条指令跳到008h后开始执行。中断标志位在中断允许总控位GIE重新置1的时候需要被软件清零以防止重复中断。一个中断标志位（PBIF除外的）会被它的中断事件置1，而不管与它相关的中断屏蔽位是否启用。通过INTFLAG 和INTEN 的相应中位来判断是否发生中断以及中断类型。当通过INT指令发生软中断时，下条指令跳到002后开始执行。

2.4.1 外部中断

外部中断INT管脚上升沿还是下降沿触发由 INTEDG 位 (OPTION<6>)决定，当一个有效的跳变发生时标志位INTIF置1，如INTIE位(INTEN<2>)清零，该中断被屏蔽。
在睡眠之前INTIE 位已被置1，INT管脚可以作为系统睡眠条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

2.4.2 Timer0 中断

TMR0发生溢出 (FFh → 00h)时T0IF标志位置1 (INTFLAG<0>). T0IE 位(INTEN<0>)清零，该中断被屏蔽。

2.4.3 Port B 输入改变中断

输入改变中断触发时IOB<7:0> PBIF标志位置1 (INTFLAG<1>). PBIE位(INTEN<1>)清零，该中断被屏蔽。
在输入改变中断发生之前，必须读取port B信息
与PortB的管脚相对应的WUBn位 (WUCON<7:0>) i清零或设置为输出或IOB0 脚设置为外部中断输入脚INT 都拥有该功能。PBIE在睡眠之前置1， Port B 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前GIE位已被置1机器唤醒以后会执行中断服务程序，否则会运行睡眠以后的下一条指令。

2.4.4 T1 中断

T1发生溢出 (FFh → 00h)时T1IF标志位置1 (INTFLAG<3>). T1IE 位(INTEN<3>)清零，该中断被屏蔽。

2.5 省电模式 (SLEEP)

执行SLEEP 指令以后机器进入省电模式。
执行SLEEP 指令， /PD 位清零 (STATUS<3>) ，/TO位置1，看门狗清零同时保持运行状态，晶体停振。
I/O维持原状

2.5.1睡眠唤醒

在睡眠状态下，单片机能通过以下方式唤醒:
1. RSTB 管脚复位
2. 看门狗复位 (机器设置了看门狗)。
3. PB0/INT管脚中断，或PORTB 输入改变中断。
外部的RSTB管脚和看门狗溢出都能使机器复位. 通过查看 /PD 和/TO 位可以检测机器是哪种复位，/PD位置1为上电复位，置0为执行SLEEP， /TO 位置0为看门狗溢出复位。.
机器通过中断唤醒，该中断屏蔽位置1，中断唤醒不管GIE是否置1.。 当GIE位被清零， 机器唤醒以后执行SLEEP指令以后的指令；当GIE位被置1， 机器唤醒以后跳转到中断复位地址 (008h)。在高频或低频模式机器复位延迟时间为18/4.5/288/72ms (该延迟时间由SUT<1:0>设置) 加上64个振荡周期。
在IRC/ERIC or ERC 模式， 机器复位延迟时间为640us。

2.6 复位

FC1601单片机能通过以下方式复位:
1. 上电复位(POR)
2. 掉电复位(Brown-out Reset BOR)
3. RSTB 管脚复位
4. 看门狗WDT溢出复位
一些寄存器在一些复位条件下没有影响，在上电和其他一些复位情况下它们的状态是未知的。. 大多数寄存器会回到复位状态在上电复位，RSTB 管脚复位，看门狗WDT溢出复位。
对Vdd上升信号检测告之芯片是否加上上电复位脉冲信号。 要使用这个特点，用户需要把RSTB管脚连接到Vdd。
掉电复位作为一种典型应用主要用在 AC 或重载交换的应用上。
芯片上的低电压检测模块到电压低于一个固定的电压也会对使芯片复位，这样能保证芯片只能在正常电压范围内工作。
RSTB或WDT睡眠唤醒也导致芯片复位，其复位操作的不会在睡眠之前。
根据不同的复原状态设置对/TO和/PD位(STATUS<4 :3>)置1或清零。

2.6.1上电复位计数器(Power-up Reset Timer PWRT)

上电复位计数器提供一个 18/4.5/288/72ms 延迟时间 (该延迟时间由 SUT<1:0>设置) (或 640us，基于不同的振荡源和复位条件) 在 Power-on Reset (POR)， Brown-out Reset (BOR)， RSTB Reset 或 看门狗溢出复位。只要 PWRT 在运行，设备就一直保持的复位状态。Vdd、温度和其他变化而会影响 PWDT 控制的设备延迟时间。

表2.1: PWRT Period

Oscillator Mode	Power-on Reset Brown-out Reset	RSTB Reset WDT time-out Reset
ERC & IRC/ERIC	18/4.5/288/72 ms	640 us
HF & LF & IRC_RTC	18/4.5/288/72 ms	18/4.5/288/72ms

2.6.2_振荡启动计数器(Oscillator Start-up Timer OST)

在HF或 LF或IRC_RTC振荡模式下在PWRT 延迟 (18/4.5/288/72ms) 之后振荡启动计数器会再提供一个64个clock的延迟。这种延迟晶体谐振器能提供稳定的振荡源，这段时间内只要OST在工作，设备就一直保持的复位状态。
在 OSCI 信号的振幅到达振荡器输入最大振幅之后，该计数器只开始增加。

2.6.3复位顺序

FC1601复位时序如下:
1. 复位锁存器置1 ，PWRT & OST 清零。
2. 当内部的POR， BOR， RSTB 复位或 WDT溢出复位脉冲加载完成后， PWRT开始计数。
3. PWRT溢出以后， OST开始计数延迟。
4. OST延迟完成以后， 复位锁存器清零最后芯片得到一个复位信号。
在高频或低频振荡模式机器复位延迟时间为18/4.5/288/72ms加上64个振荡周期，在IRC/ERIC ，ERC振荡模式单片机会在Power-on Reset (POR)， Brown-out Reset (BOR)， 或RSTB复位以后在延迟640us ，看门狗溢出复位后再延迟18/4.5/288/72ms的时间。

图2.9: 复位电路结构图

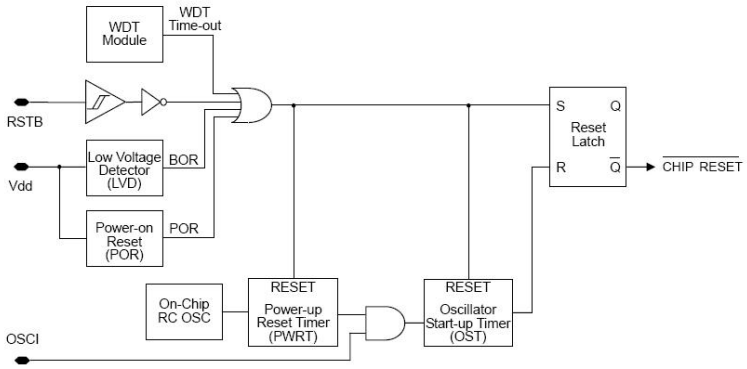


表2.2: 复位以后各个寄存器状态列表

寄存器	地址	上电复位 掉电复位	RSTB复位 WDT 复位
ACC	N/A	xxxx xxxx	uuuu uuuu
OPTION	N/A	0011 1111	0011 1111
IOSTA	N/A	---- 1111	---- 1111
IOSTB	N/A	1111 1111	1111 1111
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000# #uuu
FSR	04h	00xx xxxx	00uu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
General Purpose Registers	10 ~ 3Fh	xxxx xxxx	uuuu uuuu

Bank0	地址	上电复位 掉电复位	RSTB复位 WDT 复位
General Purpose Register	07h	xxxx xxxx	xxxx xxxx
PCON	08h	1010 0000	1010 0000
WUCON	09h	0000 0000	0000 0000
PCHBUF	0Ah	---- --00	---- --00
PDCON	0Bh	1111 1111	1111 1111
ODCON	0Ch	0000 0000	0000 0000
PHCON	0Dh	1111 1111	1111 1111
INTEN	0Eh	0--- 0000	0--- 0000
INTFLAG	0Fh	---- 0000	---- 0000

Bank1	地址	上电复位 掉电复位	RSTB复位 WDT 复位
T1M	07h	0000 0000	0000 0000
T1C	08h	0000 0000	0000 0000
T1R	09h	0000 0000	0000 0000
PWMCON	0Ah	0000 0000	0000 0000
PWM0R	0Bh	0000 0000	0000 0000
PWM1R	0Ch	0000 0000	0000 0000
PWM2R	0Dh	0000 0000	0000 0000
SYSM	0Eh	---- 0000	---- 0000

Legend: u = 不变， x = 未知， - = 不起作用， # =参见下表的值

表2.3: RST/ TO / PD 复位和唤醒后的状态

RST	/TO	/PD	复位方式
0	1	1	Power-on Reset
0	1	1	Brown-out reset
0	u	u	RSTB Reset during normal operation

0	1	0	RSTB Reset during SLEEP
0	0	1	WDT Reset during normal operation
0	0	0	WDT Wake-up during SLEEP
1	1	0	Wake-up on pin change during SLEEP

Legend: u =不变

表:2.4: /TO /PD状态位影响事件

事件	/TO	/PD
Power-on	1	1
WDT Time-Out	0	u
SLEEP instruction	1	0
CLRWDT instruction	1	1

Legend: u =不变

2.7 十六进制转化为十进制 (Hexadecimal Convert to Decimal HCD)

FC1601具有十进制格式化功能. 当一个寄存器里面的内容需要十进制转化的时候,在执行操作ALU以后必须把结果进行相应的进制转化。一个数据在处理过程中进行了转化成了十进制，那么所有对这个数进行的操作（包含存放该数据的RAM单元，accumulator (ACC)，立即数， 以及所要查表信息）都的进行十进制转化，这样的运算结果才正确。
DAA指令能在加法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC
转换操作在例子 2.2 中被说明。

例2.2: DAA 转化

```
Address  Code
NA       #include   <FC160.ASH>
n                   ...
n+1                 MOVIA    0x90     ;Set immediate data = decimal format number "90" (ACC ← 90h)
n+2                 MOVAR    0x30     ;Load immediate data "90" to data memory address 30H
n+3                 MOVIA    0x10     ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+4                 ADDAR    0x30,A   ;Contents of the data memory address 30H and ACC are binary-added
                                      ;the result loads to the ACC (ACC ← A0h, C ← 0)
n+5                 DAA               ;Convert the content of ACC to decimal format, and restored to ACC
                                      ;The result in the ACC is "00" and the carry bit C is "1". This represents the
                                      ;decimal number "100"
n+6                 ...
```

DAS指令能在减法运算完成以后将ACC 里的数据从十六进制转化为十进制重存给ACC
转换操作在例子2.3中被说明

例2.3: DAS 转化

```
Address  Code
NA       #include   <FC160.ASH>
n                   ...
n+1                 MOVIA    0x10     ;Set immediate data = decimal format number "10" (ACC ← 10h)
n+2                 MOVAR    0x30     ;Load immediate data "90" to data memory address 30H
n+3                 MOVIA    0x20     ;Set immediate data = decimal format number "20" (ACC ← 20h)
n+4                 SUBAR    0x30,A   ;Contents of the data memory address 30H and ACC are binary-subtracted
                                      ;the result loads to the ACC (ACC ← F0h, C ← 0)
n+5                 DAS               ;Convert the content of ACC to decimal format, and restored to ACC
                                      ;The result in the ACC is "90" and the carry bit C is "0". This represents the
                                      ;decimal number " -10"
n+6                 ...
```

2.8振荡器配置（Oscillator Configurations）

FC1601有七种不同的振荡模式，用户可通过编程Fosc配置位来选择相应的振荡方式:

- LF: 低频晶体谐振器
- HF: 高频晶体谐振器
- IRC:内部电阻内部电容振荡器
- ERIC:外部电阻内部电容振荡器

- ERC: 外部RC振荡器
- XT: 晶体/陶瓷振荡器
- IRC_RTC: Fcpu使用内部电阻电容振荡器/T0使用低频晶体谐振器

在 LF，XT ，HF 或 IRC_RTC 模式下， 一个晶体或陶瓷谐振器连接到 OSCI 和 OSCO 管脚建立振荡源。当在 LF， XT 或 HF 模式下，单片机通过 OSCI 脚接入外部时钟源。使用 ERC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，RC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

使用 IRC（IRC_RTC）/ERIC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，单片机具有 4 种不同的振荡频率， 8MHz，4MHz， 1MHz， 和 455KHz， 通过 (RCM<1:0>)来选择一种， 或则用户改变外部电阻来实现。 ERIC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

图2.10: HF，XT 和 LF 振荡器模式(晶体振荡器或陶瓷振荡器)

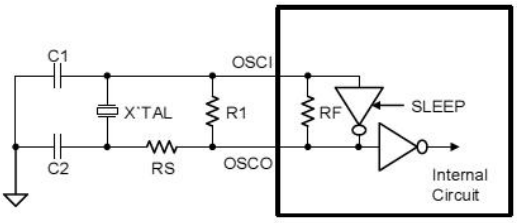


图2.11: HF，XT 和 LF 振荡器模式 (外部时钟输入操作)

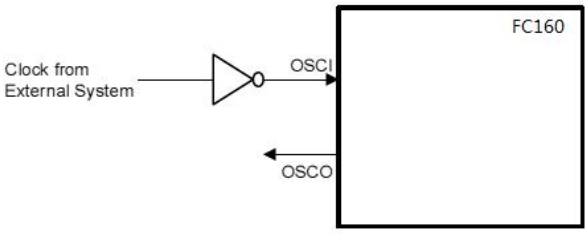


图2.12: ERC 振荡器模式 (外部电阻电容振荡器)

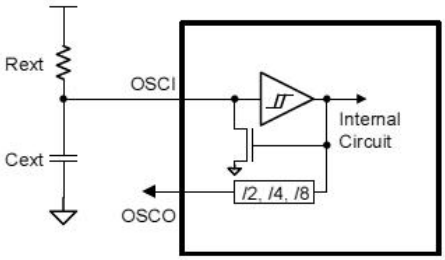


图2.13: ERIC 振荡器模式 (外部电阻 内部电容 振荡器)

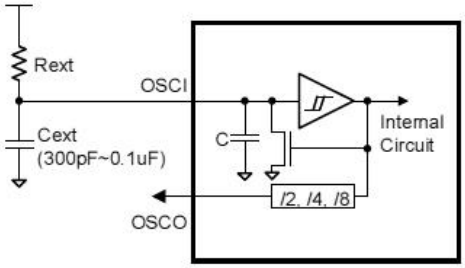
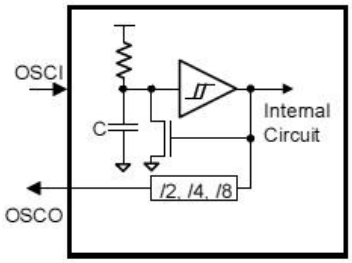


图2.14: IRC 振荡器模式 (内部电阻电容 振荡器)



3.0 工作模式

FC1601 可以在 3 种工作模式下以不同的时钟频率工作，这些模式可以控制振荡器的工作、程序的执行以及模拟电路的功能损耗。

- 普通模式：系统高速工作模式；
- 低速模式：系统低速工作模式；
- 省电模式：系统省电模式（睡眠模式）。

3.1 普通模式

普通模式是系统高速时钟正常工作模式，系统时钟源由高速振荡器提供。程序被执行。上电复位或任意一种复位触发后，系统进入普通模式执行程序。当系统从睡眠模式被唤醒后进入普通模式。普通模式下，高速振荡器正常工作，功耗最大。

- 程序被执行，所有的功能都可控制。
- 系统速率为高速。
- 高速振荡器和内部低速 RC 振荡器都正常工作。
- 通过 SYSM 寄存器，系统可以从普通模式切换到其它任何一种工作模式。
- 系统从睡眠模式唤醒后进入普通模式。
- 低速模式可以切换到普通模式。

3.2 低速模式

低速模式为系统低速时钟正常工作模式。系统时钟源由内部低速RC 振荡器提供。低速模式由SYSM 寄存器的CLKMD位控制。当 CLKMD=0 时，系统为普通模式；当 CLKMD=1 时，系统进入低速模式。切换进入低速模式后，不能自动禁止高速振荡器，必须通过 STOPHX 位来禁止以减少功耗。低速模式下，系统速率被固定为 Flosc/2（Flosc 为内部低速 RC 振荡器频率）。

- 程序被执行，所有的功能都可控制。
- 系统速率位低速（Flosc/2）。
- 内部低速 RC 振荡器正常工作，高速振荡器由 STOPHX=1 控制。低速模式下，强烈建议停止高速振荡器。
- 通过 SYSM 寄存器，低速模式可以切换进入其它的工作模式。
- 从低速模式切换到睡眠模式，唤醒后返回到低速模式，但 STOPHX 会自动置 0。
- 普通模式可以切换进入低速模式。

3.3 睡眠模式

睡眠模式是系统的理想状态，不执行程序，振荡器也停止工作。整个芯片的功耗低于 1uA。。

- 执行 sleep 指令，程序停止执行，所有的功能被禁止。
- 所有的振荡器，包括外部高速振荡器、内部高速振荡器和内部低速振荡器都停止工作。
- 功耗低于 1uA。

注： RTC模式时，外置振荡可选择不停。

例程：

进入低速模式：

```
BSR    BANK1     ;进入bank 1寄存器地址
BSR    CLKMD     ;进入低速模式
BSR    STOPHX    ;停止高速振荡器
BCR    BANK1     ;退出bank1
```

返回普通模式：

```
BSR    BANK1     ;进入bank 1寄存器地址
BCR    STOPHX    ;打开高速振荡器
BCR    CLKMD     ;返回普通模式
BCR    BANK1     ;退出bank1
```

4.0 配置选项

表4.0: 配置选项0

位	名称	说明
2~0	Fosc<2:0>	振荡源选择位 = 0 0 0→ERC mode (external R & C) (默认) IOB4/OSCO管脚为取OSCOUT功能 = 1 1 0→ HF mode = 1 0 1→XT mode = 1 0 0→LF mode = 0 1 1→IRC mode (internal R & C) IOB4/OSCO管脚为取OSCOUT功能 = 0 0 1→ERIC mode (external R & internal C) IOB4/OSCO管脚为取OSCOUT功能 = 1 1 1→IRC_RTC mode (external R & internal C for Fcpu/LF for T0)
5~3	LVDT<2:0>	低电压检测选择位 = 0 0 0→禁止低电压检测(默认) = 0 0 1→enable, LVDT voltage = 2.0V, 睡眠模式关闭 = 0 1 0→enable, LVDT voltage = 2.0V = 0 1 1→enable, LVDT voltage = 3.6V = 1 0 0→enable, LVDT voltage = 2.9V = 1 0 1→enable, LVDT voltage = 2.2V = 1 1 0→enable, LVDT voltage = 2.4V = 1 1 1→enable, LVDT voltage = 2.6V
7~6	RCM<1:0>	IRC选择位 = 0 0→4MHz (默认) = 0 1→8MHz = 1 0→1MHz = 1 1→455KHz
10~8	SUT<2:0>	PWRT & WDT计数周期选择位 = 0 0 0→PWRT = WDT prescaler rate = 18ms (默认) = 0 0 1→PWRT = WDT prescaler rate = 4.5ms = 0 1 0→PWRT = WDT prescaler rate = 288ms = 0 1 1→PWRT = WDT prescaler rate = 72ms = 1 0 0→PWRT = 640us, WDT prescaler rate = 18ms = 1 0 1→PWRT = 640us, WDT prescaler rate = 4.5ms = 1 1 0→PWRT = 640us, WDT prescaler rate = 288ms = 1 1 1→PWRT = 640us, WDT prescaler rate = 72ms
11	OSCOUT	IRC/ERIC/ERC 模式下IOB4/OSCO功能选择位置 = 0, IOB4(默认) = 1, OSCO
12	RSTBIN	IOB3/RSTB选择位置 = 0, IOB3 (默认) = 1, RSTB

表4.1: 配置选项1

位	名称	说明
0	WDTEN	看门狗使能位 = 0，禁止WDT(默认) = 1，使能WDT
1	PROTECT	代码保护选择位 = 0→代码不加密EPROM code protection off (默认) = 1→代码加密EPROM code protection on
3~2	OSCD<1:0>	指令运行周期选择位 = 0 0→ 4个振荡周期 (默认) = 1 0→ 2个振荡周期 = 1 1→ 8个振荡周期
4	PBWKEN	IOB口中断唤醒使能 = 0， IOB口中断唤醒使能通过WUCON寄存器控制，LVDTE软件控制（默认） = 1， IOB口默认打开中断唤醒使能，LVDTE使能打开（无法通过软件设置）
5	RDPORT	IO作为输出时，读端口方式控制位 = 0， 从寄存器读 (默认) = 1， 从管脚读
6	SCHMITT	I/O输入缓冲控制位 = 0， 通过Schmitt触发器(默认) = 1， 不通过Schmitt触发器
7	IOB3OD-	IOB3开漏输出控制位 = 0， IOB3口为I/O口通过IOSTB3控制输入输出(默认) = 1， IOB3口作为输出时为开漏输出
12~8	OTPBANK<4:0>	OTP bank选择（每个bank 256条指令，bank只能按bank0到bank3的顺序选择; 每个bank512条指令,bank只能按bank0到bank1的顺序选择） = 0 0 0 0 0，OTP 1K容量(默认) = 1 1 0 0 0，OTP 选择bank_256_0 (PC 000~0FF) = 1 1 0 0 1，OTP 选择bank_256_1 (PC 100~1FF) = 1 1 0 1 1，OTP 选择bank_256_2 (PC 200~2FF) = 1 1 1 1 1，OTP 选择bank_256_3 (PC 300~3FF) = 1 0 0 0 0，OTP 选择bank_512_0 (PC 000~1FF) = 1 0 1 0 0，OTP 选择bnak_512_1 (PC 200~3FF)

表4.2: 配置选项2

位	名称	说明
6	IHRC2X	IHRC的频率是否加倍 = 0，不加倍 = 1，加倍。最快达到16M/2分频
8：7	上拉电阻选择	00=100k 01=70k 10=50k 11=20k
9	OTP_RDS	0：1/4*Fcpu 1：1/2*Fcpu（读取宽度越宽，低压特性越好，但工作电路会大）
10	Drive_power_low	B0,B1,B2大驱动时是否需要省电容 = 0，省 = 1，电源地需要加电容。

表4.3: 配置选项3

位	名称	说明
7~0	IHRC修调	
8	POWER_SAVE	节能选择位 = 0，IRC选择455K时降低功耗（默认） = 1，关闭455K节能模式
9	IO_PH_EN	IO口上拉选择使能位（PB3此位无效） = 1，IO口上拉只在输入模式下有效（默认） = 0，IO口上拉在输入，输出模式下都有效。

表4.4: 配置选项4

位	名称	说明
1~0	PB0 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
3~2	PB0 NMOS drive select	00=5ma 01=10m 10=50m 11=80m
5~4	PB1 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
7~6	PB1 NMOS drive select	00=5ma 01=10m 10=50m 11=80m
9~8	PB2 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
11~10	PB2 NMOS drive select	00=5ma 01=10m 10=30m 11=80m

表4.5: 配置选项5

位	名称	说明
1~0	PB3 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
3~2	PB3 NMOS drive select	00=5ma 01=10m 10=30m 11=50m
5~4	PB4 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
7~6	PB4 NMOS drive select	00=5ma 01=10m 10=30m 11=50m
9~8	PB5 PMOS drive select	00=5ma 01=10m 10=20m 11=30m
11~10	PB5 NMOS drive select	00=5ma 01=10m 10=30m 11=50m

表4.6: Selection of IOB5/OSCI and IOB4/OSCO Pins

振荡方式	IOB5/OSCI	IOB4/OSCO
IRC/ERIC	IOB5 (OSCIN=0)	IOB4/OSCO selected by OSCOUT bit
	OSCI (OSCIN=1)	IOB4/OSCO selected by OSCOUT bit
ERC	OSCI	IOB4/OSCO selected by OSCOUT bit
HF	OSCI	OSCO
LF	OSCI	OSCO

5.0 指令集合

操作语法	说明	操作内容	指令周期	影响标志位
BCR R, bit	Clear bit in R	0→R<b>	1	-
BSR R, bit	Set bit in R	1→R<b>	1	-
BTRSC R, bit	Test bit in R， Skip if Clear	Skip if R<b> = 0	1/2 (1)	-
BTRSS R, bit	Test bit in R， Skip if Set	Skip if R<b> = 1	1/2 (1)	-
NOP	No Operation	No operation	1	-
CLRWDT	Clear Watchdog Timer	00h→WDT， 00h →WDT prescaler	1	/TO，/PD
OPTION	Load OPTION register	ACC→ OPTION	1	-
SLEEP	Go into power-down mode	00h→ WDT， 00h→ WDT prescaler	1	/TO，/PD
DAA	Adjust ACC's data format from HEX to DEC after any addition operation	ACC(hex) → ACC(dec)	1	C
DAS	Adjust ACC's data format from HEX to DEC after any subtraction operation	ACC(hex) → ACC(dec)	1	-
RETURN	Return from subroutine	Top of Stack→ PC	2	-
RETFIE	Return from interrupt， set GIE bit	Top of Stack→ PC， 1→ GIE	2	-
INT	S/W interrupt	PC + 1→Top of Stack， 002h→ PC	2	-
IOST R	Load IOST register	ACC→ IOST register	1	-
CLRA	Clear ACC	00h→ ACC	1	Z
CLRR R	Clear R	00h→ R	1	Z
MOVAR R	Move ACC to R	ACC→ R	1	-
MOVR R, d	Move R	R→ dest	1	Z
DECR R, d	Decrement R	R - 1 →dest	1	Z
DECRSZ R, d	Decrement R， Skip if 0	R - 1→ dest， Skip if result = 0	1/2 (1)	-
INCR R, d	Increment R	R + 1→ dest	1	Z
INCRSZ R, d	Increment R， Skip if 0	R + 1→ dest， Skip if result = 0	1/2 (1)	-
ADDAR R, d	Add ACC and R	R + ACC→ dest	1	C， DC， Z
SUBAR R, d	Subtract ACC from R	R - ACC→ dest	1	C， DC， Z
ADCAR R, d	Add ACC and R with Carry	R + ACC + C → dest	1	C， DC， Z
SBCAR R, d	Subtract ACC from R with Carry	R + ACC + C→dest	1	C， DC， Z
ANDAR R, d	AND ACC with R	ACC and R→dest	1	Z
IORAR R, d	Inclusive OR ACC with R	ACC or R→ dest	1	Z
XORAR R, d	Exclusive OR ACC with R	R xor ACC→ dest	1	Z
COMR R, d	Complement R	R→dest	1	Z
RLR R, d	Rotate left f through Carry	R<7>→C， R<6:0>→ dest<7:1>， C→ dest<0>	1	C
RRR R, d	Rotate right f through Carry	C ←dest<7>， R<7:1> ← dest<6:0>， R<0>← C	1	C
SWAPR R, d	Swap R	R<3:0> →dest<7:4>， R<7:4> →dest<3:0>	1	-
MOVIA I	Move Immediate to ACC	I →ACC	1	-
ADDIA I	Add ACC and Immediate	I + ACC →ACC	1	C， DC， Z
SUBIA I	Subtract ACC from Immediate	I - ACC →ACC	1	C， DC， Z
ANDIA I	AND Immediate with ACC	ACC and I →ACC	1	Z
IORIA I	OR Immediate with ACC	ACC or I →ACC	1	Z

操作语法	说明	操作内容	指令周期	影响标志位
XORIA I	Exclusive OR Immediate to ACC	ACC xor I →ACC	1	Z
RETIA I	Return， place Immediate in ACC	I ACC， Top of Stack → PC	2	-
CALL I	Call subroutine	PC + 1 →Top of Stack， I → PC	2	-
GOTO I	Unconditional branch	I →PC	2	-

注释: 1. 两周期指令为分支跳转指令

2. bit : Bit 地址为8位寄存器R中的某一位

R : 寄存器地址 (00h to 3Fh)

I :立即数

ACC : 累加器

d : 目的选择:

=0 (结果存放在ACC)

=1 (结果存放在R)

dest : 目的地

PC : 程序指针

PCHBUF : 高位缓冲程序指针

WDT : 看们狗计数器

GIE :中断允许总控制位

TO : 计数溢出位

PD : 省电模式选择位

C : 进位/借位标志

DC : 辅助进位/借位标志.(低四位向高四位进位/借位标志)

Z : 零标志

ADCAR(带进位加法)	Add ACC and R with Carry
语　法	ADCAR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	R + ACC + C→ dest
受影响的标志	C, DC, Z
说　明	將A寄存器的内含值加上R寄存器的内含值（带进位），如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

ADDAR (加法指令)	ACC and R with Carry
语　法	ADDAR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	R + ACC→ dest
受影响的标志	C, DC, Z
说　明	將A寄存器的内含值加上R寄存器的内含值（不带进位），如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

ADDIA	Add ACC and Immediate
语　法	ADDIA I
操作数	0≤ I ≤ 255
操作内容	ACC + I → ACC
受影响的标志	C, DC, Z
说　明	將A寄存器的内含值加上立即数I，结果在ACC中存放。
指令执行周期	1

ANDAR	AND ACC and R
语　法	ANDAR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	ACC and R → dest
受影响的标志	Z
说　明	將A寄存器内含值和R寄存器做相与操作，如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1

ANDIA	AND Immediate with ACC
语　法	ANDIA I
操作数	0≤ I ≤ 255
操作内容	ACC and I → dest
受影响的标志	Z
说　明	將A寄存器的内含值与立即数I做相与操作，结果在ACC中存放
指令执行周期	1

BCR	Clear Bit in R
语　法	BCR R, b
操作数	0≤ R≤ 63 0 ≤ b ≤ 7
操作内容	0 → R<b>
受影响的标志	无
说　明	R寄存器的位“b” 被设成0
指令执行周期	1

BSR	Set Bit in R

语　法	BSR R， b
操作数	0≤ R≤ 63 0 ≤ b ≤ 7
操作内容	1 → R<b>
受影响的标志	无
说　明	R寄存器的位“b” 被设成1
指令执行周期	1

BTRSC　　Test Bit in R， Skip if Clear

语　法	BTRSC R， b
操作数	0≤ R≤ 63 0 ≤ b ≤ 7
操作内容	当R<b> = 0跳过下条指令
受影响的标志	无
说　明	R<b> = 0跳过下条指令 R<b> = 0时，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)

BTRSS　　Test Bit in R， Skip if Set

语　法	BTRSS R， b
操作数	0≤ R≤ 63 0 ≤ b ≤ 7
操作内容	当R<b> = 1跳过下条指令
受影响的标志	无
说　明	R<b> =1跳过下条指令 R<b> = 1时，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)

CALL　　Subroutine Call

语　法	CALL I
操作数	0≤ I ≤ 1023
操作内容	PC +1 → Top of Stack: I → PC
受影响的标志	无
说　明	子程序调用。 首先下一条指令地址(PC+1)进栈。 10 位立即地址被装载入PC指针的位<9 :0>. CALL是二周期指令。
指令执行周期	2

CLRA　　Clear ACC

语　法	CLRA
操作数	无
操作内容	00h → ACC I → Z
受影响的标志	Z
说　明	ACC被清零，Z标志为置1
指令执行周期	1

CLRR　　Clear R

语　法	CLRR R
操作数	0≤ R≤ 63
操作内容	00h → R I → Z
受影响的标志	Z
说　明	R被清零，Z标志为置1
指令执行周期	1

CLRWDT　　Clear Watchdog Timer

语　法	CLRWDT
操作数	无
操作内容	00h → WDT: 00h → WDT prescaler (已经设置了WDT预置器): 1 → TO: 1 → PD
受影响的标志	TO，PD
说　明	CLRWDT指令重置WDT，如已经设置了WDT预置器，也重置WDT预置器；并把TO，PD位置1
指令执行周期	1

COMR　　Complement R

语　法	COMR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R → dest
受影响的标志	Z
说　明	将R内含内容取反，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

DAA　　Adjust ACC's data format from HEX to DEC

语　法	DAA
操作数	无
操作内容	ACC(hex) → ACC(dec)
受影响的标志	C
说　明	在有些加法操作以后把ACC内值的十六进制转化十进制，
指令执行周期	1

DAS　　Adjust ACC's data format from HEX to DEC

语　法	DAS
操作数	无
操作内容	ACC(hex) → ACC(dec)
受影响的标志	C
说　明	在有些减法操作以后把ACC内值的十六进制转化十进制，
指令执行周期	1

DECR　　Decrement R

语　法	DECR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R - 1 → dest
受影响的标志	Z
说　明	递减R寄存器的值，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

DECRSZ　　Decrement R， Skip if 0

语　法	DECRSZ R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R - 1 → dest 如果结果等于0，跳过下条指令
受影响的标志	无
说　明	递减R寄存器的值，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。 如果结果等于0，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)

GOTO　　Unconditional Branch

语　法	GOTO I
操作数	0≤ I ≤ 1023
操作内容	I → PC
受影响的标志	无
说　明	无条件跳转。10位立即地址被装载入PC指针的位<9 :0>. GOTO是二周期指令。
指令执行周期	2

INCR　　Increment R

语　法	INCR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	R + 1 → dest
受影响的标志	Z
说　明	将被指定R寄存器的内含值加1，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

INCRSZ　　Increment R, Skip if 0

语　法	INCRSZ R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	R + 1 → dest 如果结果等于0，跳过下条指令
受影响的标志	无
说　明	将被指定R寄存器的内含值加1，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。 如果结果等于0，该指令周期中提取的下条指令被丢弃，并以执行NOP操作来替换这条2周期指令。
指令执行周期	1(2)

INT　　S/W Interrupt

语　法	INT
操作数	无
操作内容	PC +1 → Top of Stack: 002h → PC
受影响的标志	无
说　明	子程序调用。 首先下一条指令地址(PC+1)进栈。 10 位地址002h被装载入PC指针的位<9 :0>. CALL是二周期指令。
指令执行周期	2

IORAR　　OR ACC with R

语　法	**IORAR**
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	ACC or R → dest
受影响的标志	Z
说　明	將A寄存器内含值和R寄存器做或操作，如果d是0结果在ACC中存放。如果d是1结果在R中存放。
指令执行周期	1

IORIA　　OR Immediate with ACC

语　法	IORIA I
操作数	0≤ I ≤ 255
操作内容	ACC or I → dest
受影响的标志	Z
说　明	將A寄存器的内含值与立即数I做相与操作，结果在ACC中存放
指令执行周期	1

IOST　　Load IOST Register

语　法	IOST R
操作数	R = 5 or 6
操作内容	ACC → IOST register R
受影响的标志	无
说　明	將A寄存器的内含值加载到IOST register R中
指令执行周期	1

MOVAR　　Move ACC to R

语　法	MOVAR R
操作数	0≤ R≤ 63
操作内容	ACC → R
受影响的标志	无
说　明	将数据从ACC传送到R
指令执行周期	1

MOVIA　　Move Immediate to ACC

语　法	MOVIA I
操作数	0≤ I ≤ 255
操作内容	I → ACC
受影响的标志	无
说　明	将立即值载入A寄存器中
指令执行周期	1

MOVR　　Move Immediate to ACC

语　法	MOVR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R → dest
受影响的标志	无
说　明	将R寄存器内容载入R中，，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。d 为1用来测试该寄存器对标志Z 是否有影响
指令执行周期	1

NOP　　No Operation

语　法	NOP
操作数	无
操作内容	无操作
受影响的标志	无
说　明	不做任何操作
指令执行周期	1

OPTION　　Load OPTION Register

语　法	OPTION
操作数	无
操作内容	ACC → OPTION
受影响的标志	无
说　明	将A寄存器内容载入OPTION中
指令执行周期	1

RETFIE　　Return from Interrupt， Set ‘GIE’ Bit

语　法	RETFIE
操作数	无
操作内容	Top of Stack→PC
受影响的标志	无
说　明	程序计数器载入堆栈返回地址。 GIE位被设置到1。 这是二周期指令。
指令执行周期	2

RETIA　　Return with Immediate in ACC

语　法	RETIA I
操作数	0≤ I ≤ 255
操作内容	I → ACC: Top of Stack→PC
受影响的标志	无
说　明	程序计数器载入堆栈返回地址，并把立即数送入A中。这是二周期指令。
指令执行周期	2

RETURN **Return from Subroutine**

语　法	RETURN
操作数	无
操作内容	Top of Stack→PC
受影响的标志	无
说　明	程序计数器载入堆栈返回地址。这是二周期指令。
指令执行周期	2

RLR **Rotate Left f through Carry**

语　法	RLR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	R<7> → C: R<6:0> → dest<7:1>: C → dest<0>
受影响的标志	C
说　明	R寄存器的内含值又移1bit，右移时包含C(进位标志)，如下图，结果存放由d决定，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

RRR **Rotate Right f through Carry**

语　法	RRR R, d
操作数	0≤ R≤ 63 d∈[0, 1]
操作内容	C → dest<7>: R<7:1> → dest<6:0>: R<0> → C
受影响的标志	C
说　明	R寄存器的内含值又移1bit，右移时包含C(进位标志)，如下图，结果存放由d决定，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

SLEEP **SLEEP**

语　法	SLEEP
操作数	无
操作内容	00h→WDT: 00h→ WDT prescaler: 1→TO: 0→ PD
受影响的标志	TO，PD
说　明	TO位置1。PD位清零，WDT和WDT预置器清零 单片机进入睡眠模式
指令执行周期	1

SBCAR (带借位加法)　　Subtract ACC from R with Carry

语　法	SBCAR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	（R -ACC-C） → dest
受影响的标志	C， DC， Z
说　明	将R寄存器的内含值减去A寄存器的内含值（带借位），如果d是0 结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

SUBAR　　Subtract ACC from R

语　法	SUBAR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R - ACC→ dest
受影响的标志	C， DC， Z
说　明	将R寄存器的内含值减去A寄存器的内含值（不带借位），如果d是0 结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

SUBIA　　Subtract ACC from Immediate

语　法	SUBIA I
操作数	0≤ I ≤ 255
操作内容	ACC - I → ACC
受影响的标志	C， DC， Z
说　明	将A寄存器的内含值减去立即数I，结果在ACC中存放。
指令执行周期	1

SWAPR　　Swap nibbles in R

语　法	SWAPR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	R<3:0> → dest<7:4>: R<7:4> → dest<3:0>
受影响的标志	无
说　明	将所选定的寄存器，高4位以及低4位，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

XORAR　　Exclusive OR ACC with R

语　法	XORAR R， d
操作数	0≤ R≤ 63 d∈[0，1]
操作内容	ACC xor R → dest R
受影响的标志	Z
说　明	将A寄存器的值和R寄存器的值异或在一起，如果d是0结果在ACC中存放。 如果d是1结果在R中存放。
指令执行周期	1

XORIA	**Exclusive OR Immediate with ACC**
语　法	XORIA I
操作数	0≤ I ≤ 255
操作内容	ACC xor I → ACC
受影响的标志	Z
说　明	将A寄存器的值和立即数I 异或在一起，结果在ACC中存放。
指令执行周期	1

6.0 绝对最大额定值

操作环境温度	0℃到+70℃
存储器额定温度	-65℃到+150℃
DC 电源电压(Vdd)	0V 到+6.0V
输入电压(对地电压 (Vss))	-0.3V 到(Vdd + 0.3)V

7.0 操作条件

DC 供电电压	+2.0V到+5.5V
操作温度	0℃到+70℃

*细节详见 6.1

8.0 电气特性

8.1 FC1601 电气特性

电气特性是在四时钟指令周期和 WDT & LVDT 禁用情况下 Ta=25℃

Sym	Description	Conditions	Min.	Typ.	Max.	Unit
V_{DD}	Supply voltage	0Hz~1MHz		2.0	5.5	V
		1MHz~4MHz		2.0	5.5	
		4MHz~8MHz		2.4	5.5	
		8MHz~10MHz		2.6	5.5	
		10MHz~16MHz		3.0	5.5	
		16MHz~20MHz		3.5	5.5	
T_{PWR}	Power rising time	Vdd=0V to Vdd	0.8		2.6	ms/V
F_{HF}	X'tal oscillation range	HF mode, Vdd=5V	4		20	MHz
		HF mode, Vdd=3V	4		16	
F_{XT}	X'tal oscillation range	XT mode, Vdd=5V	0.455		20	MHz
		XT mode, Vdd=3V	0.455		8	
F_{LF}	X'tal oscillation range	LF mode, Vdd=5V	32		455	MHz
		LF mode, Vdd=3V	32		455	
F_{ERC}	RC oscillation range	ERC mode, Vdd=5V	DC		16	MHz
		ERC mode, Vdd=3V	DC		13	
$F_{IRC/ERIC}$	RC oscillation range	ERIC mode, external R, Vdd=5V	DC		16	MHz
		ERIC mode, external R, Vdd=3V	DC		16	
		IRC mode, internal R, Vdd=5V	0.455		8	
		IRC mode, internal R, Vdd=3V	0.455		8	
V_{IH}	Input high voltage	With Schmitt-trigger				V
		I/O ports, Vdd=5V	2.2		VDD	
		RSTB, T0CKI pins, Vdd=5V	2.2		VDD	
		I/O ports, Vdd=3V	1.7		VDD	
		RSTB, T0CKI pins, Vdd=3V	1.7		VDD	
		Without Schmitt-trigger				
		I/O ports, Vdd=5V	2.0		VDD	
		RSTB, T0CKI pins, Vdd=5V	2.0		VDD	
		I/O ports, Vdd=3V	1.5		VDD	
		RSTB, T0CKI pins, Vdd=3V	1.5		VDD	
V_{IL}	Input low voltage	With Schmitt-trigger				V
		I/O ports, Vdd=5V	VSS		0.8	
		RSTB, T0CKI pins, Vdd=5	VSS		0.8	
		I/O ports, Vdd=3V	VSS		0.5	
		RSTB, T0CKI pins, Vdd=3V	VSS		0.5	
		Without Schmitt-trigger				
		I/O ports, Vdd=5V	VSS		0.8	
		RSTB, T0CKI pins, Vdd=5V	VSS		0.8	
		I/O ports, Vdd=3V	VSS		0.6	
		RSTB, T0CKI pins, Vdd=3V	VSS		0.6	
Sym	Description	Conditions	Min.	Typ.	Max.	Unit
V_{OH}	Output high voltage	I_{OH}=-5.4mA, Vdd=5V	3.6			V
V_{OL}	Output low voltage	I_{OL}=8.7mA, Vdd=5V			0.6	V
I_{PH}	Pull-high current	Input pin at Vss, Vdd=5V	-50	-66	-80	uA
I_{PL}	Pull-down current	Input pin at Vdd, Vdd=5V	30	44	60	uA
I_{WDT}	WDT current (18mS)	Vdd=5V		6		uA
		Vdd=3V		1.5		
T_{WDT}	WDT period (18mS)	Vdd=3V		20.2		mS
		Vdd=4V		17.5		
		Vdd=5V		15.9		
I_{LVDT}	LVDT current	Vdd=5V, LVDT=3.6V		1.89		uA
		Vdd=5V, LVDT=2.4V		2.4		
		Vdd=3V, LVDT=2.4V		1		
I_{SB}	Power down current	Sleep mode, Vdd=5V, WDT LVDT disable		1	2	uA
		Sleep mode, Vdd=3V, WDT LVDT disable		0.5	1	
I_{DD}	Operating current	HF mode, Vdd=5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF				mA
		20MHz		3.03		
		16MHz		2.61		
		HF mode, Vdd=3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF				
		20MHz		1.27		
		16MHz		1.07		

I_{DD}	Operating current	HF mode, Vdd=5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF				mA
		20MHz		4.36		
		16MHz		3.41		
		HF mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF				
		20MHz		1.94		
		16MHz		1.46		
I_{DD}	Operating current	XT mode, Vdd=5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF				mA
		20MHz		2.13		
		16MHz		1.77		
		XT mode, Vdd=5V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF				
		455KHz		0.42		
		32KHz		-		
I_{DD}	Operating current	XT mode, Vdd=3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF				mA
		16MHz		0.76		
		8MHz		0.43		
		XT mode, Vdd=3V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF				
		455KHz		0.16		
		32KHz		0.10		
I_{DD}	Operating current	XT mode, Vdd=5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF				mA
		20MHz		3.12		
		16MHz		2.57		
		XT mode, Vdd=5V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF				
		455KHz		0.44		

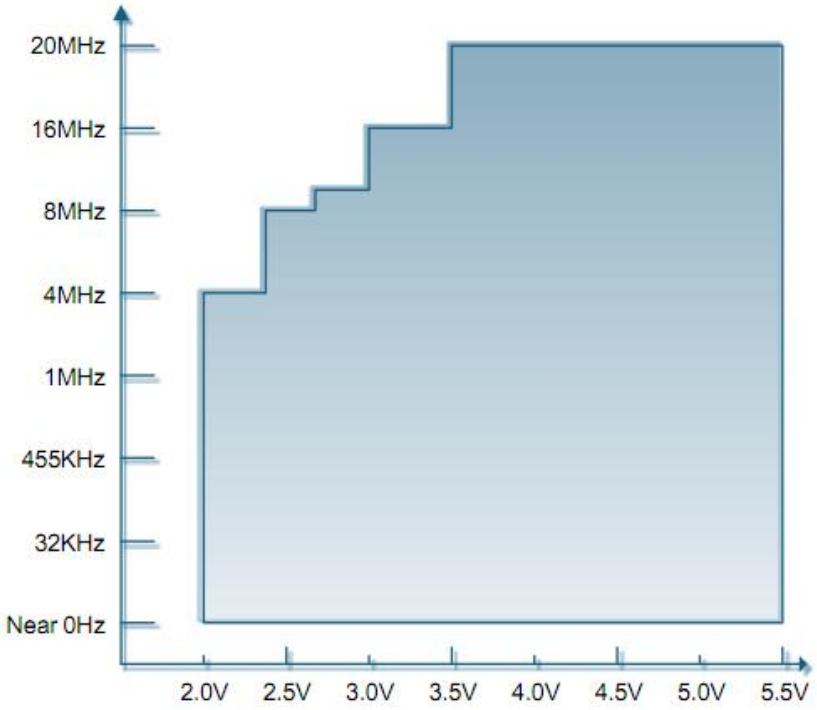
Sym	Description	Conditions			Min.	Typ.	Max.	Unit
		32KHz				-		
I_{DD}	Operating current	XT mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF						mA
		16MHz				-		
		8MHz				0.62		
		XT mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF						
		455KHz				0.44		
		32KHz				-		
I_{DD}	Operating current	LF mode, Vdd=5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF						uA
		455KHz				106.3		
		32KHz				42.5		
		LF mode, Vdd=3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF						
		455KHz				38.7		
		32KHz				12.1		
I_{DD}	Operating current	LF mode, Vdd=5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF						uA
		455KHz				130.9		
		32KHz				44.8		
		LF mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF						
		455KHz				50.4		
		32KHz				13		
I_{DD}	Operating current	ERC mode, Vdd=5V, 4 clock instruction						mA
		C=3P	R=10Kohm	F=7.4MHz		1.2		
			R=3.3Kohm	F=16.4MHz		2.87		
		ERC mode, Vdd=3V, 4 clock instruction						
		C=3P	R=10Kohm	F=7MHz		0.61		
			R=3.3Kohm	F=13.6MHz		1.41		
I_{DD}	Operating current	ERC mode, Vdd=5V, 2 clock instruction						mA
		C=3P	R=10Kohm	F=7.6MHz		1.87		
			R=3.3Kohm	F=15.1MHz		3.94		
		ERC mode, Vdd=3V, 2 clock instruction						
		C=3P	R=10Kohm	F=6.7MHz		0.87		
			R=3.3Kohm	F=13.7MHz		2		
I_{DD}	Operating current	ERIC mode, external R, Vdd=5V, 4 clock instruction						mA
		F=8MHz	R=103.2Kohm			1.11		
		F=4MHz	R=206.2Kohm			0.58		
		ERIC mode, external R, Vdd=3V, 4 clock instruction						
		F=8MHz	R=99.4Kohm			0.56		
		F=4MHz	R=187.6Kohm			0.29		
I_{DD}	Operating current	ERIC mode, external R, Vdd=5V, 2 clock instruction						mA
		F=8MHz	R=103.2Kohm			1.78		
		F=4MHz	R=206.2Kohm			0.9		
		ERIC mode, external R, Vdd=3V, 2 clock instruction						
		F=8MHz	R=99.4Kohm			0.89		

		F=4MHz	R=187.6Kohm		0.46		

Sym	Description	Conditions	Min.	Typ.	Max.	Unit
I_{DD}	Operating current	IRC mode, internal R, Vdd=5V, 4 clock instruction				mA
		F=8MHz		1.77		
		F=4MHz		0.93		
		F=1MHz		0.32		
		F=455KHz		0.21		
		IRC mode, internal R, Vdd=3V, 4 clock instruction				
		F=8MHz		0.97		
		F=4MHz		0.55		
		F=1MHz		0.2		
		F=455KHz		0.14		
I_{DD}	Operating current	IRC mode, internal R, Vdd=5V, 2 clock instruction				mA
		F=8MHz		1.75		
		F=4MHz		0.95		
		F=1MHz		0.33		
		F=455KHz		0.22		
		IRC mode, internal R, Vdd=3V, 2 clock instruction				
		F=8MHz		0.99		
		F=4MHz		0.55		
		F=1MHz		0.21		
		F=455KHz		0.15		

8.2 FC1601电气特性表

8.2.1 操作频率 vs 操作电压（Ta=25℃）



This datasheet contains new product information.SZFC reserves the rights to modify the product specification without notice.No liability is assumed as a result of the use of this product.No rights under any patent accompany the sales of the product.
本中文版内容若与英文版内容有争议时，则以英文版的内容为准。