

G80F980

集成MDU+CORDIC协处理器、ADC和PWM的增强型8051微控制器

Ver1.10 2016.06

选型指南

订购型号	FLASH ROM	RAM	类 EEPROM	I/O MAX	TIMER	ADC	PWM	EUART	其它	温度范围	封装
G80F980	31K	16K	1K	46	5*16	11*12	6*16 3*8	1	WDT /LVR/SPI 温度传感器等	-40℃ ~ +105℃	TQFP48

技术咨询

南京立超电子科技有限公司
 中国南京市和燕路251号金港大厦A幢2406室
 ZIP:210028
 Tel: 0086-25-83306839/83310926
 Fax: 0086-25-83737785
 Website: <http://www.dycmcu.com/>

目录	
1.	特性4
2.	概述5
3.	方框图.....6
4.	引脚配置.....7
5.	引脚描述.....9
6.	SFR映像.....11
7.	标准功能.....22
7.1	CPU 22
7.2	RAM 24
7.3	FLASH/RAM程序存储器 25
7.4	SSP 29
7.5	系统时钟和振荡器 34
7.6	I/O端口 39
7.7	定时器 47
7.8	8BIT脉冲宽度调制模块 (PWM30/PWM31/PWM32) 62
7.9	中断 65
8.	增强功能74
8.1	电机控制模块MCM (MOTOR CONTROL MOD) PWM0/1/2 74
8.2	协处理器单元 (CORDIC+MDSU) 114
8.3	运算放大器和比较器模块 122
8.4	增强型通用异步收发器 (EUART) 132
8.5	模/数转换模块 (ADC) 142
8.6	温度传感器 (TEMPRETURE SENSOR) 163
8.7	串行外部设备接口 (SPI) 165
8.8	输入捕捉模块 172
8.9	实时监控模块 (RM) 179
8.10	电压复位 (LVR) 180
8.11	看门狗定时器 (WDT) , 程式超范围溢出 (OVL) 复位及其它复位状态 181
8.12	电源管理 183
8.13	预热计数器 185
8.14	高速程序RAM (PRAM) 186
8.15	CRC校验模块 188
8.16	代码选项 190
9.	指令集.....191
10.	电气特性.....196
11.	封装信息.....200
12.	格更改记录.....201
13.	业务联络.....202
14.	免责声明.....203

1. 特性

- 基于8051指令流水线结构的8位单片机
 - 最高84MHz系统工作频率
- 集成算术协处理器 (MDU+CORDIC)
 - 单周期32bit硬件移位单元
 - 单周期16×16bit硬件乘法
 - 8周期32 / 16 bit硬件除法
 - 硬件CORDIC协处理器 (圆函数)
 - 两套操作数结果寄存器组可切换
- Program ROM(Flash): 31K字节
- Program RAM: 16K 字节
- Data RAM: 1.5K 字节
- 类EEPROM: 1K 字节
- 工作电压:
 - $f_{osc} = 400k - 84Hz, V_{DD} = 2.4V - 5.5V$
- 高频振荡器 (代码选项二选一):
 - 晶体谐振器: 400k - 16MHz
 - 内部RC高频振荡器: 12MHz (常温0.5% 精度)
 - 内部集成无需外接电容的PLL, 最高输出84MHz
 - · · (84MHz @-40~125°C, 100MHz @-40~105°C)
- 低频振荡器: 内部RC低频振荡器: 128KHz
- 46个CMOS双向I/O管脚
- I/O内建上拉电阻
- P3口可增强灌电流能力(代码选项可选增强或标准)
- P0/P1口可增强拉电流能力(代码选项可选增强或标准)
- 5个16位定时器/计数器T0/T1/T2/T3/T4
- 三路捕捉功能, 保存捕捉触发条件下的捕捉定时器值
- 脉冲宽度调制模块 (PWM)
 - 6路 (3对) 16位PWM输出, 每对PWM支持互补或独立模式, 输出极性可单独控制
 - 3种计数模式, 单次计数模式, 边沿对齐模式, 中心对齐模式(中心对齐互补模式下支持对称和非对称波形)
 - 带死区产生逻辑及故障检测功能
 - PWM周期内可设置多个时刻硬件触发ADC功能
 - 可设置后分频系数
 - PWM管脚顺序可定义 (代码选项)
- 中断源:
 - 定时器0/1/2/3/4
 - 外部中断0 - 1
 - 外部中断4: 7输入
 - 捕捉中断0/1/2
 - I2C, SPI
 - PWM周期, 归零和占空比中断
 - 模拟比较器1/2
 - PWM30/31/32周期中断
 - CRC中断
- 11+5 通道12位1M sps高速模数转换器 (ADC), 自动触发ADC功能, 16通道转换结果缓冲功能. (增加起始转换通道指针, 目前都是从0通道开始转换)
- 内建温度传感器, 利用AD读取结果。
- 内建两个多输入模拟施密特比较器CMP1, CMP2(可使用内部参考源分压值作为比较基准)
- 内建4个高速模拟放大器OP1, OP2, OP3, OP4, 带内置可变增益系数 (也可外置, 二选一)
- 增强型UART
- 三路8bit无死区PWM输出
- SPI接口 (主/从模式)
- 内建的低电压复位功能 (代码选项)
 - LVR电压1: 4.1V
 - LVR电压2: 3.7V
 - LVR电压3: 2.5V
- 内建CRC校验模块, 可对高速PRAM区做校验
- CPU机器周期:
 - 1振荡周期
- 看门狗定时器 (WDT)
- 内建振荡器预热计数器
- 低功耗工作模式:
 - 空闲模式
 - 掉电模式
- Flash型
- 40位可读MCU识别码
- 工作环境温度-40~+105°C
- 单线仿真接口 (同时支持四线仿真接口)
- 封装: - TQFP48

2. 概述

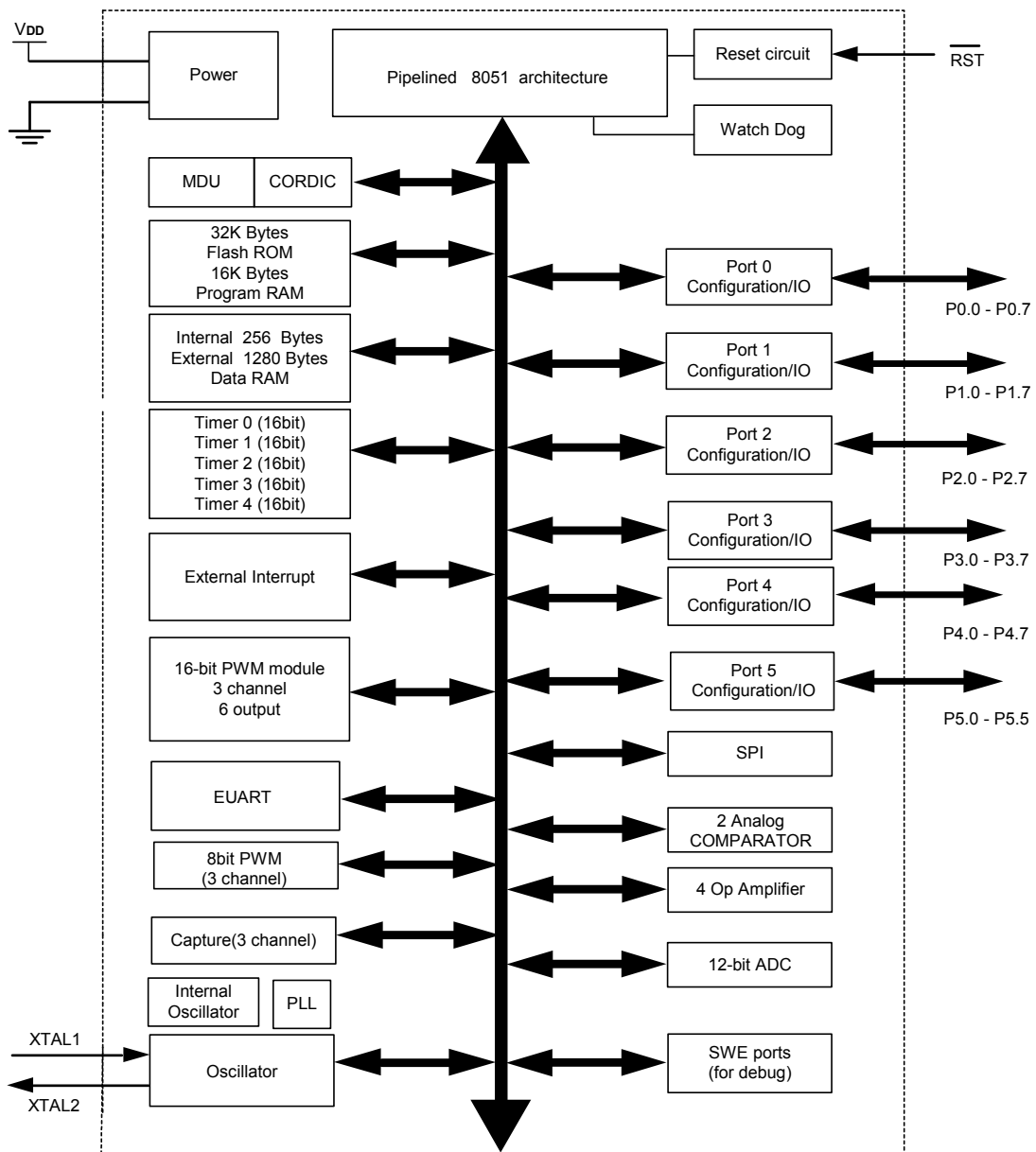
G80F980是一种高速高效率8051兼容单片机。在同样振荡频率下，较之传统的8051芯片具有运行更快速，性能更优越的特性。

G80F980保留了标准8051芯片的大部分特性，包括内置256字节RAM和2个16位定时器/计数器。此外，G80F980还集成了1280字节外部扩展RAM，以及存储程序的31K字节flash以及16K字节Program RAM，此外还集成1K类EEPROM，用于在系统掉电后保存数据。

G80F980还集成了两个多输入模拟比较器，四路放大器，12位集成数字比较功能的高速ADC，以及6输出带死区时间控制的电机控制PWM模块，非常适合于直流无刷电机/永磁同步电机控制。

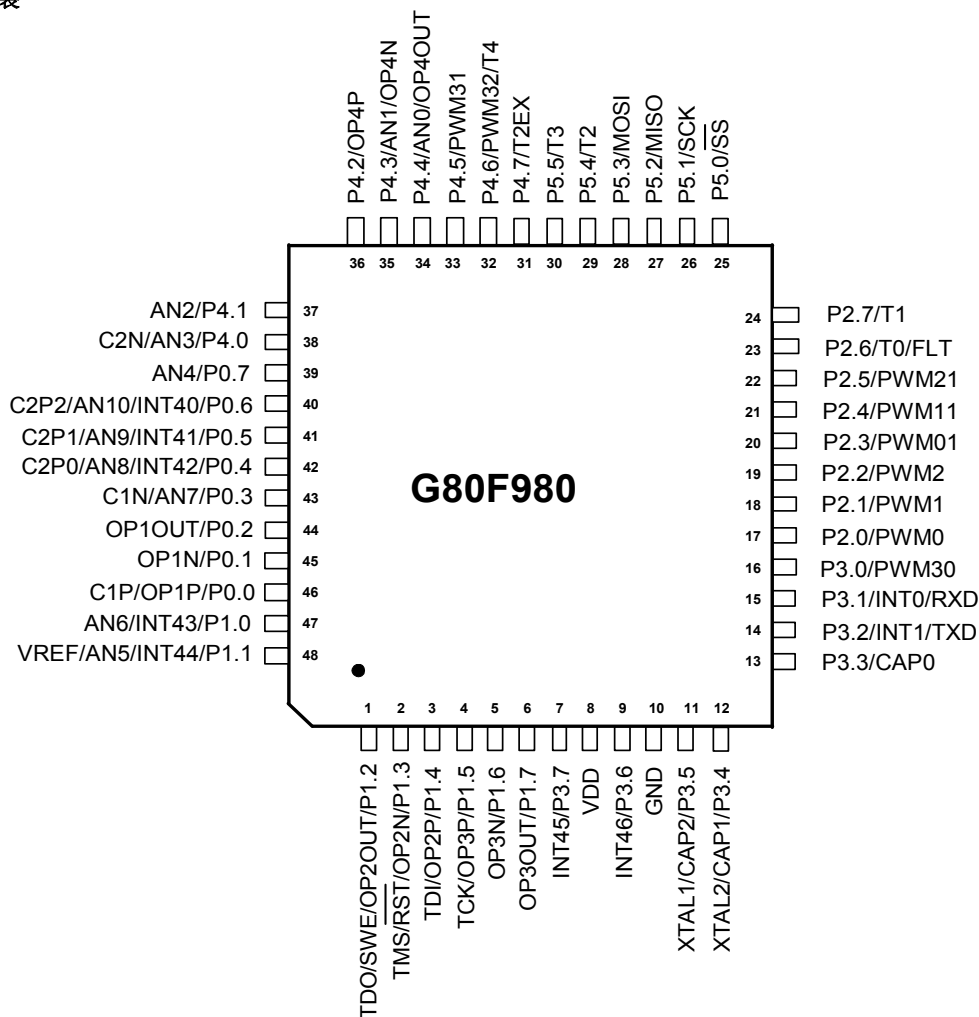
此外，G80F980集成了看门狗定时器，上电复位和低电压复位等功能，提供了2种低功耗省电模式。

3. 方框图



4. 引脚配置

48脚TQFP封装

**注意:**

引脚命名中，写在最外侧的引脚功能具有最高优先级，最内侧的引脚功能具有最低优先级（参见引脚配置图）。当一个引脚被高优先级的功能占用时，即使低优先级功能被允许，也不能作为低优先级功能的引脚。只有当软件禁止引脚的高优先级功能，相应引脚才能被释放作为低优先级端口使用。

但也有同优先级情况，比如P0.4 - P0.6，它们可以同时用作AD输入通道和比较器同相输入端，具体端口共用请参考7.6.4“端口共用”章节。

引脚功能

引脚编号	引脚命名	默认功能	引脚编号	引脚命名	默认功能
1	TDO/SWE/OP2OUT/P1.2	P1.2	25	SS/P5.0	P5.0
2	TMS/ $\overline{\text{RST}}$ /OP2N/P1.3	$\overline{\text{RST}}$	26	SCK/P5.1	P5.1
3	TDI/OP2P/ P1.4	P1.4	27	MISO/P5.2	P5.2
4	TCK/OP3P/P1.5	P1.5	28	MOSI/P5.3	P5.3
5	OP3N/P1.6	P1.6	29	T2/P5.4	P5.4
6	OP3OUT/P1.7	P1.7	30	T3/P5.5	P5.5
7	INT45/P3.7	P3.7	31	T2EX/P4.7	P4.7
8	V _{DD}	----	32	T4/PWM32/P4.6	P4.6
9	INT46/P3.6	P3.6	33	PWM31/P4.5	P4.5
10	GND	----	34	OP4OUT/ AN0/P4.4	OP4OUT
11	XTAL1/CAP2 /P3.5	P3.5	35	OP4N/ AN1/P4.3	OP4N
12	XTAL2/CAP1/P3.4	P3.4	36	OP4P/P4.2	OP4P
13	CAP0/P3.3	P3.3	37	AN2/P4.1	P4.1
14	TXD/INT1/P3.2	P3.2	38	C2N/AN3/P4.0	P4.0
15	RXD/INT0/P3.1	P3.1	39	AN4/P0.7	P0.7
16	PWM30/P3.0	P3.0	40	C2P2/AN10/INT40/P0.6	P0.6
17	PWM0/P2.0	P2.0	41	C2P1/AN9/INT41/P0.5	P0.5
18	PWM1/P2.1	P2.1	42	C2P0/AN8/INT42/P0.4	P0.4
19	PWM2/P2.2	P2.2	43	C1N/ AN7/P0.3	P0.3
20	PWM01/P2.3	P2.3	44	OP1OUT/P0.2	P0.2
21	PWM11/P2.4	P2.4	45	OP1N/P0.1	P0.1
22	PWM21/P2.5	P2.5	46	C1P/OP1P/P0.0	P0.0
23	P2.6/T0/FLT	P2.6	47	AN6/INT43/P1.0	P1.0
24	T1/P2.7	P2.7	48	VREF/AN5/INT44/P1.1	P1.1

5. 引脚描述

引脚编号	类型	说明
I/O端口		
P0.0 - P0.7	I/O	8位双向I/O端口
P1.0 - P1.7	I/O	8位双向I/O端口
P2.0 - P2.7	I/O	8位双向I/O端口
P3.0 - P3.7	I/O	8位双向I/O端口
P4.0 - P4.7	I/O	8位双向I/O端口
P5.0 - P5.5	I/O	6位双向I/O端口
定时器		
T0	I/O	定时器0外部输入/比较功能输出
T1	I/O	定时器1外部输入/比较功能输出
T2	I/O	定时器2外部输入/比较功能输出
T2EX	I	定时器2重载/捕捉/方向控制
T3	I/O	定时器3外部输入/比较功能输出
T4	I/O	定时器4外部输入/比较功能输出
PWM		
PWM01, 11, 21	O	16BIT PWM互补输出
PWM0, 1, 2	O	16BIT PWM输出
PWM30 - PWM32	O	8BIT PWM独立输出
FLT	I	16BIT PWM模块故障输入信号
模数转换器		
AN0 - AN10	I	ADC输入通道
V _{REF}	I	ADC参考电压输入
中断, 复位, 时钟, 电源		
INT0, 1	I	外部中断输入
INT40 - INT46	I/O	外部中断7输入源
$\overline{\text{RST}}$	I	该引脚上保持10 μs 以上的低电平, CPU将复位。由于有内建30k Ω 上拉电阻连接到V _{DD} , 所以仅接一个外部电容即可实现上电复位。 注意: 在使用JTAG功能(仿真或工具烧写)时, 外接电容不能大于0.01 μF , 否则就应将外接电容断开。
XTAL1	I	谐振器输入
XTAL2	O	谐振器输出
GND	P	接地
V _{DD}	P	电源(2.4 - 5.5V)

续上表

EUART		
RXD	I/O	串口数据输入
TXD	O	串口数据输出
SPI		
MOSI	I/O	SPI主输出从输入引脚
MISO	I/O	SPI主输入从输出引脚
SCK	I/O	SPI串行时钟引脚
\overline{SS}	I	SPI从设备选择引脚
内部模拟比较器		
C1P	I	比较器1反相输入端（比较器1同相输入端和放大器1同相输入端OP1P共用）
C1N	I	比较器1反相输入端
C2N	I	比较器2反相输入端
C2P0 - 2	I	比较器2同相多路输入端
内部模拟放大器		
OP1P	I	放大器1同相输入端
OP1N	I	放大器1反相输入端
OP1OUT	O	放大器1输出端
OP2P	I	放大器2同相输入端
OP2N	I	放大器2反相输入端
OP2OUT	O	放大器2输出端
OP3P	I	放大器3同相输入端
OP3N	I	放大器3反相输入端
OP2OUT	O	放大器3输出端
OP4P	I	放大器4同相输入端
OP4N	I	放大器4反相输入端
OP4OUT	O	放大器4输出端
编程器接口		
TDO (P1.2)	O	四线仿真接口：测试数据输出
TMS (P1.3)	I	四线仿真接口：测试模式选择
TDI (P1.4)	I	四线仿真接口：测试数据输入
TCK (P1.5)	I	四线仿真接口：测试时钟输入
SWE(P1.2)	I/O	单线仿真接口
注意： 当P1.2~P1.5作为仿真接口时，原有功能被禁止		

6. SFR 映像

G80F980置256字节的直接寻址寄存器，包括通用数据存储器和特殊功能寄存器（SFR），G80F980的SFR有以下几种：

CPU内核寄存器：	ACC, B, PSW, SP, DPL, DPH
CPU内核增强寄存器：	AUXC, DPL1, DPH1, INSCON
电源时钟控制寄存器：	PCON, SUSLO
Flash寄存器：	IB_OFFSET, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5, FLASHCON
数据页面控制寄存器：	XPAGE
看门狗定时器寄存器：	RSTSTAT
系统时钟控制寄存器：	CLKCON, CLKCON1
中断寄存器：	IEN0, IEN1, IENC, IPH0, IPL0, IPH1, IPL1, EXF1, EXCON
I/O端口寄存器：	P0, P1, P2, P3, P4, P5, P0CR, P1CR, P2CR, P3CR, P4CR, P5CR, P0PCR, P1PCR, P2PCR, P3PCR, P4PCR, P5PCR
定时器寄存器：	TCON, TMOD, TH0, TH1, TL0, TL1, T2CON, T2MOD, TH2, TL2, RCAP2L, RCAP2H, TCON1, T3CON, TL3, TH3, T4CON, TL4, TH4
EUART寄存器：	SCON, SBUF, SADEN, SADDR, PCON, SBRTL, SBRTH, BFINE
ADC寄存器：	ADCON1, ADCON2, ADT, SEQADDR, SEQCON, ADCH1, ADCH2, SEQCHX, ADDXL, ADDXH, ADDGTL, ADDGTH, ADDLTL, ADDLTH, ADCMPCON
SPI寄存器：	SPCON, SPCON, SPDAT
PWM3寄存器：	PWM3EN, PWM30CON, PWM30P, PWM30D, PWM31CON, PWM31P, PWM31D, PWM32CON, PWM32P, PWM32D
OP&CMP寄存器：	CMP1CON, CMP2CON1, CMP2CON2, CMPINT1, CMPINT2, OPCON1, OPCON2, CMPVRS
MDSU寄存器：	CMDSCON1, OPRDYL, OPRDYH, OPRDZL, OPRDZH, OPRDXL, OPRDXH
CORDIC寄存器：	CMDSCON1, CDCXL, CDCXH, CDCYL, CDCYH, CDCZL, CDCZH
MCM寄存器：	PTCON, PWMCON1, PWMCON2, PWMCON3, PWMOE, PMANUALCON1, PMANUALCON2, FLTCON, PWMINTEN, PWMINTF, PWMRLDEN, PWMPL, PWMPH, PWMCL, PWMCH, PWM0DL, PWM0DH, PWM1DL, PWM1DH, PWM2DL, PWM2DH, PWM01DL, PWM01DH, PWM11DL, PWM11DH, PWM21DL, PWM21DH, PWMDT0L, PWMDT0H, PWMDT1L, PWMDT1H, PWMCMP1L, PWMCMP1H, PWMCMP2L, PWMCMP2H
CAPTURE寄存器	CAPCON1, CAPCON2, CAPINT1, CAPINT2, CAPTINT, CAPTOPL, CAPTOPH, CAPBOTL, CAPBOTH, CAPTL, CAPTH, CAPSTAT
PRAM寄存器	PRAMLOCK, PRAMCON
实时监控寄存器	RMDATA1, RMDATA2, RMDATA3, RMDATA4, RMDATA5, RMDATA6
CRC校验寄存器	CRCCON, CRCDL, CRCDH

Table 6.1 C51核SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ACC	E0H	累加器	00000000	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
B	F0H	B寄存器	00000000	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
AUXC	F1H	C寄存器	00000000	C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
PSW	D0H	程序状态字	00000000	CY	AC	F0	RS1	RS0	OV	F1	P
SP	81H	堆栈指针	00001111	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
DPL	82H	数据指针低位字节	00000000	DPL0.7	DPL0.6	DPL0.5	DPL0.4	DPL0.3	DPL0.2	DPL0.1	DPL0.0
DPH	83H	数据指针高位字节	00000000	DPH0.7	DPH0.6	DPH0.5	DPH0.4	DPH0.3	DPH0.2	DPH0.1	DPH0.0
DPL1	84H	数据指针1低位字节	00000000	DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0
DPH1	85H	数据指针1高位字节	00000000	DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0
INSCON	86H	数据指针选择	----00-0	-	-	-	-	DIV	MUL	-	DPS

Table 6.2 电源时钟控制SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PCON	87H	电源控制	00000000	SMOD	SSTAT	IT41	IT40	GF1	GF0	PD	IDL
SUSLO	8EH	电源控制保护字	00000000	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0

Table 6.3 Flash控制SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_OFF SET	FBH	可编程flash低位字节偏移	00000000	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
IB_DATA	FCH	可编程flash数据寄存器	00000000	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
IB_CON1	F2H	flash控制寄存器1	00000000	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
IB_CON2	F3H	flash控制寄存器2	----0000	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
IB_CON3	F4H	flash控制寄存器3	----0000	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
IB_CON4	F5H	flash控制寄存器4	----0000	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
IB_CON5	F6H	flash控制寄存器5	----0000	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
XPAGE	F7H	flash页寄存器	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
FLASHCON	A7H	访问控制寄存器	-----0	-	-	-	-	-	-	-	FAC

Table 6.4 WDT SFR

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
RSTSTAT	B1H	看门狗定时器控制寄存器	*-***000	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0

注意: *表示不同情况的复位决定RSTSTAT寄存器中的复位值, 详见WDT章节

Table 6.5 时钟控制SFR

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CLKCON	B2H	系统时钟选择	-110-01-	-	CLKS1	CLKS0	SCMIF	-	CLKSEL1	CLKSEL0	-
CLKCON1	CFH	时钟控制寄存器1	00-00000	OSC2ON	OSC2F	-	PLLON	PLLLOCK	PLLMUL2	PLLMUL1	PLLMUL0

Table 6.6 中断 SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IEN0	A8H	中断允许控制0	00000000	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
IEN1	A9H	中断允许控制1	00000000	ESCM	EPWM3	EMCM	ET3	EX4	ET4	ECMP1	ESPI
IEN2	AAH	中断允许控制2	-----00	-	-	-	-	-	-	ECMP2	ECAP
IENC	BAH	外部中断4通道允许控制	-0000000	-	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
IPH0	B4H	中断优先级控制高位0	00000000	PINTH	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
IPL0	B8H	中断优先级控制低位0	00000000	PINTL	PADCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
IPH1	B5H	中断优先级控制高位1	00000000	PSCMH	PPWMH	PMCMH	PT3H	PX4H	PT4H	PCMPH	PSPIH
IPL1	B9H	中断优先级控制低位1	00000000	PSCML	PPWML	PMCML	PT3L	PX4L	PT4L	PCMPL	PSPIL
EXF1	D8H	外部中断寄存器1	-0000000	-	IF46	IF45	IF44	IF43	IF42	IF41	IF40
EXCON	95H	外部中断采样次数控制寄存器	00000000	I1PS1	I1PS0	I1SN1	I1SN0	I0PS1	I0PS0	I0SN1	I0SN0

Table 6.7 端口 SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
P0	80H	8位端口0	00000000	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1	90H	8位端口1	00000000	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2	A0H	8位端口2	00000000	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3	B0H	8位端口3	00000000	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4	C0H	8位端口4	00000000	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
P5	F8H	8位端口5	----0000	-	-	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
P0CR	E1H	端口0输入/输出方向控制	00000000	P0CR.7	P0CR.6	P0CR.5	P0CR.4	P0CR.3	P0CR.2	P0CR.1	P0CR.0
P1CR	E2H	端口1输入/输出方向控制	00000000	P1CR.7	P1CR.6	P1CR.5	P1CR.4	P1CR.3	P1CR.2	P1CR.1	P1CR.0
P2CR	E3H	端口2输入/输出方向控制	00000000	P2CR.7	P2CR.6	P2CR.5	P2CR.4	P2CR.3	P2CR.2	P2CR.1	P2CR.0
P3CR	E4H	端口3输入/输出方向控制	00000000	P3CR.7	P3CR.6	P3CR.5	P3CR.4	P3CR.3	P3CR.2	P3CR.1	P3CR.0
P4CR	E5H	端口2输入/输出方向控制	00000000	P4CR.7	P4CR.6	P4CR.5	P4CR.4	P4CR.3	P4CR.2	P4CR.1	P4CR.0
P5CR	E6H	端口2输入/输出方向控制	----0000	-	-	P5CR.5	P5CR.4	P5CR.3	P5CR.2	P5CR.1	P5CR.0
P0PCR	E9H	端口0内部上拉允许	00000000	P0PCR.7	P0PCR.6	P0PCR.5	P0PCR.4	P0PCR.3	P0PCR.2	P0PCR.1	P0PCR.0
P1PCR	EAH	端口1内部上拉允许	00000000	P1PCR.7	P1PCR.6	P1PCR.5	P1PCR.4	P1PCR.3	P1PCR.2	P1PCR.1	P1PCR.0
P2PCR	EBH	端口2内部上拉允许	00000000	P2PCR.7	P2PCR.6	P2PCR.5	P2PCR.4	P2PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
P3PCR	ECH	端口3内部上拉允许	00000000	P3PCR.7	P3PCR.6	P3PCR.5	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	P3PCR.0
P4PCR	EDH	端口4内部上拉允许	00000000	P4PCR.7	P4PCR.6	P4PCR.5	P4PCR.4	P4PCR.3	P4PCR.2	P4PCR.1	P4PCR.0

P5PCR	EEH	端口5内部上拉允许	----0000	-	-	P5PCR.5	P2PCR.4	P5PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
-------	-----	-----------	----------	---	---	---------	---------	---------	---------	---------	---------

Table 6.8 定时器 SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TCON	88H	定时器/计数器0和1控制寄存器	00000000	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89H	定时器/计数器0和1模式寄存器	00000000	GATE1	$\overline{C/T1}$	M11	M10	GATE0	$\overline{C/T0}$	M01	M00
TL0	8AH	定时器/计数器0低位字节	00000000	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	8CH	定时器/计数器0高位字节	00000000	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1	8BH	定时器/计数器1低位字节	00000000	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1	8DH	定时器/计数器1高位字节	00000000	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
T2CON	C8H	定时器/计数器2控制寄存器	000-0000	TF2	EXF2	TRG	-	EXEN2	TR2	$\overline{C/T2}$	CP/RL2
T2MOD	C9H	定时器/计数器2模式寄存器	11---00	TCLKP21	TCLKP20	-	-	-	-	T2OE	DCEN
RCAP2L	CAH	定时器/计数器2重载/截获低位字节	00000000	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	CBH	定时器/计数器2重载/截获高位字节	00000000	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	CCH	定时器/计数器2低位字节	00000000	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	CDH	定时器/计数器2高位字节	00000000	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
TCON1	CEH	定时器0/1比较功能允许	00111100	TCLK_S1	TCLK_S0	TCLKP11	TCLKP10	TCLKP01	TCLKP00	TC1	TC0
T3CON	B3H	定时器/计数器3控制寄存器	0-00-000	TF3	-	T3PS.1	T3PS.0	-	TR3	T3CLKS.1	T3CLKS.0
TL3	ACH	定时器/计数器3低位字节	00000000	TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
TH3	ADH	定时器/计数器3高位字节	00000000	TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
T4CON	ABH	定时器/计数器4控制寄存器	0000-00	TF4	TC4	T4PS.1	T4PS.0	-	-	TR4	T4CLKS
TL4	AEH	定时器/计数器4低位字节	00000000	TL4.7	TL4.6	TL4.5	TL4.4	TL4.3	TL4.2	TL4.1	TL4.0
TH4	AFH	定时器/计数器4高位字节	00000000	TH4.7	TH4.6	TH4.5	TH4.4	TH4.3	TH4.2	TH4.1	TH4.0

Table 6.9 EUART SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SCON	98H	串行控制	00000000	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
SBUF	99H	串行数据缓冲器	00000000	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SADEN	9BH	从属地址掩码	00000000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
SADDR	9AH	从属地址	00000000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
PCON	87H	电源和串行控制	00—0000	SMOD	SSTAT	IT41	IT40	GF1	GF0	PD	IDL
SBRTL	FDH	波特率发生器寄存器	00000000	SBRT.7	SBRT.6	SBRT.5	SBRT.4	SBRT.3	SBRT.2	SBRT.1	SBRT.0

SBRTH	FEH	波特率发生器寄存器	00000000	SBRTEN	SBRT.14	SBRT.13	SBRT.12	SBRT.11	SBRT.10	SBRT.9	SBRT.8
BFINE	A4H	波特率发生器微调寄存器	----0000	-	-	-	-	BFINE.3	BFINE.2	BFINE.1	BFINE.0

Table 6.10 SPI SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SPCON	A2H	SPI控制寄存器	00000000	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
SPSTA	A1H	SPI状态寄存器	00000---	SPEN	SPIF	MODF	WCOL	RXOV	-	-	-
SPDAT	A3H	SPI数据寄存器	00000000	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0

Table 6.11 ADC SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCON1	93H	ADC控制寄存器1	00000000	ADON	ADCIF	SC	REFC	ADCIE	PWMTRGEN	TIMTRGEN	GO/DONE
ADCON2	92H	ADC控制寄存器2	000000-0	GRP3	GRP2	GRP1	GRP0	MODE	ADDRST	-	ALR
ADT	FF80H	ADC时钟控制寄存器	00000000	TADC3	TADC2	TADC1	TADC0	TS3	TS2	TS1	TS0
SEQADDR	94H	ADC采样通道指针寄存器	----0000	-	-	-	-	ADDR3	ADDR2	ADDR1	ADDR0
SEQCON	A6H	映射控制寄存器	00000000	TGAP2	TGAP1	TGAP0	REGSEL	REG3	REG2	REG1	REG0
ADCH1	FF81H	ADC通道配置寄存器1	00000000	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADCH2	FF82H	ADC通道配置寄存器2	00010000	CH10	CH9	CH8	VBIAS	TPSEN	TPSCHOP	BGEN	BGCHOP
SEQCHx	96H	通道寄存器	----0000	-	-	-	-	SEQx3	SEQx2	SEQx1	SEQx0
ADDxL	96H	ADC结果寄存器低位	00000000	A7	A6	A5	A4	A3	A2	A1	A0
ADDxH	97H	ADC结果寄存器高位	00000000	A15	A14	A13	A12	A11	A10	A9	A8
ADDGTH	FF85H	ADC上限比较寄存器高位	00000000	GT15	GT14	GT13	GT12	GT11	GT10	GT9	GT8
ADDGTL	FF86H	ADC上限比较寄存器低位	00000000	GT7	GT6	GT5	GT4	GT3	GT2	GT1	GT0
ADDLTH	FF83H	ADC下限比较寄存器高位	00000000	LT15	LT14	LT13	LT12	LT11	LT10	LT9	LT8
ADDLTL	FF84H	ADC下限比较寄存器低位	00000000	LT7	LT6	LT5	LT4	LT3	LT2	LT1	LT0
ADCMPCON	91H	比较寄存器	00000000	ADLIE	ADGIE	ADLIF	ADGIF	CSEL3	CSEL2	CSEL1	CSEL0

Table 6.12 CMDSU SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMDSCON1	C1H	操作控制寄存器1	00000000	MD2	MD1	MD0	SIGNEN	OPRDY	OPRDZ	OPRDZ	RUN
CMDSCON2	BBH	操作控制寄存器2	0000-00	BSYMDS	BSYCDC	DIVERR	OVF	-	-	Bank	ALUSEL
OPRDYL	C2H	操作数结果Y寄存器	00000000	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

OPRDYH	C3H	操作数结果Y寄存器	00000000	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
OPRDZL	C4H	操作数结果Z寄存器	00000000	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
OPRDZH	C5H	操作数结果Z寄存器	00000000	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24
OPRDXL	C6H	操作数结果X寄存器	00000000	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
OPRDXH	C7H	操作数结果X寄存器	00000000	DB15	DB14	DB13	DB12	DB11	DB10	DB9	DB8

Table 6.14 PWM3 SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM3EN	FFA0H	PWM3使能寄存器	----000	-	-	-	-	-	PWM32EN	PWM31EN	PWM30EN
PWM30CON	FFA1H	PWM30控制寄存器	00000000	PWM30S	PWM30CK2	PWM30CK1	PWM30CK0	FLT30EN	PWM30IE	PWM30IF	PWM30OE
PWM30P	FFA2H	PWM30周期寄存器	00000000	PWM30P.7	PWM30P.6	PWM30P.5	PWM30P.4	PWM30P.3	PWM30P.2	PWM30P.1	PWM30P.0
PWM30D	FFA3H	PWM30占空比寄存器	00000000	PWM30D.7	PWM30D.6	PWM30D.5	PWM30D.4	PWM30D.3	PWM30D.2	PWM30D.1	PWM30D.0
PWM31CON	FFA9H	PWM31控制寄存器	00000000	PWM31S	PWM31CK2	PWM31CK1	PWM31CK0	FLT31EN	PWM31IE	PWM31IF	PWM31OE
PWM31P	FFAAH	PWM31周期寄存器	00000000	PWM31P.7	PWM31P.6	PWM31P.5	PWM31P.4	PWM31P.3	PWM31P.2	PWM31P.1	PWM31P.0
PWM31D	FFABH	PWM31占空比寄存器	00000000	PWM31D.7	PWM31D.6	PWM31D.5	PWM31D.4	PWM31D.3	PWM31D.2	PWM31D.1	PWM31D.0
PWM32CON	FFB1H	PWM32控制寄存器	00000000	PWM32S	PWM32CK2	PWM32CK1	PWM32CK0	FLT32EN	PWM32IE	PWM32IF	PWM32OE
PWM32P	FFB2H	PWM32周期寄存器	00000000	PWM32P.7	PWM32P.6	PWM32P.5	PWM32P.4	PWM32P.3	PWM32P.2	PWM32P.1	PWM32P.0
PWM32D	FFB3H	PWM32占空比寄存器	00000000	PWM32D.7	PWM32D.6	PWM32D.5	PWM32D.4	PWM32D.3	PWM32D.2	PWM32D.1	PWM32D.0

Table 6.15 OP & CMP SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMP1CON	FF8EH	比较器1控制寄存器	00000000	CMP1EN	C1NCHS	C1PCHS1	C1PCHS0	OP1PCHS	OP1CHOP	C1DEB1	C1DEB0
CMP2CON1	FF8BH	比较器2控制寄存器1	00000000	CMP2EN	C2NCHS	C2PCHS2	C2PCHS1	C2PCHS0	PWMTRGS1	PWMTRGS0	TRGPOL
CMP2CON2	FF8CH	比较器2控制寄存器2	00000000	C2FT1EN	C2FT1S2	C2FT1S1	C2FT1S0	C2FT2EN	C2FT2S2	C2FT2S1	C2FT2S0
CMPINT1	FF8AH	比较器1中断寄存器	-----00	-	-	-	-	-	-	C1IF	C1OUT
CMPINT2	FF87H	比较器2中断寄存器	--00--00	-	-	C2IFS1	C2IFS0	-	-	C2IF	C2OUT
OPCON1	FF88H	放大器控制寄存器1	00000000	OP2EN	OP2G2	OP2G1	OP2G0	OP1EN	OP1G2	OP1G1	OP1G0
OPCON2	FF89H	放大器控制寄存器2	10000000	OP4EN	OP4G2	OP4G1	OP4G0	OP3EN	OP3G2	OP3G1	OP3G0
CMPVRS	FF8DH	比较器参考电压选择寄存器	-000-000	-	CMP2VRS2	CMP2VRS1	CMP2VRS0	-	CMP1VRS2	CMP1VRS1	CMP1VRS0
CMPST	FF8FH	比较器施密特控制寄存器	----0000	-	-	-	-	C2SMT1	C2SMT0	C1SMT1	C1SMT0

Table 6.16 MCM SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PTCON	FF90H	PWM时基控制寄存器	00000000	PTMOD1	PTMOD0	PTCLK2	PTCLK1	PTCLK0	POSTPS2	POSTPS1	POSTPS0
PWMCON1	D9H	PWM模块控制寄存器1	0-000000	POUTMOD	PWMSYM	PWM21S	PWM11S	PWM01S	PWM2S	PWM1S	PWM0S
PWMCON2	DAH	PWM模块控制寄存器2	00000000	ZETIM1	ZETIM0	PEAD	ZEAD	CMP21	CMP20	CMP11	CMP10
PWMCON3	DBH	PWM模块控制寄存器3	000-0000	PDLDEN	DT1	DT0	-	OSYNC	PDCON2	PDCON1	PDCON0
PWMOE	EFH	PWM输出使能寄存器	0-000000	PWMEN	-	PWM21OE	PWM11OE	PWM01OE	PWM2OE	PWM1OE	PWM0OE
PMANUAL CON1	B6H	PWM手动输出寄存器1	-000000	-	-	PMANUAL21	PMANUAL11	PMANUAL01	PMANUAL2	PMANUAL1	PMANUAL0
PMANUAL CON2	B7H	PWM手动输出寄存器2	-000000	-	-	POUT21	POUT11	POUT01	POUT2	POUT1	POUT0
FLTCON	D1H	PWM故障检测保护寄存器	00000000	FLT1EN	FLT1SEL	FLT2EN	FLT2S	FLT2DEB1	FLT2DEB0	FLTM	FLTSTAT
PWMINTEN	FF91H	PWM中断使能寄存器	00000000	PWMPIE	PWMZIE	PTDD2IE	PTUD2IE	PTDD1IE	PTUD1IE	PTDD0IE	PTUD0IE
PWMINTF	E8H	PWM中断标志寄存器	00000000	PWMPIF	PWMZIF	PTDD2IF	PTUD2IF	PTDD1IF	PTUD1IF	PTDD0IF	PTUD0IF
PWMRLDEN	E7H	PWM定时器锁定寄存器	00000000	RLDEN.7	RLDEN.6	RLDEN.5	RLDEN.4	RLDEN.3	RLDEN.2	RLDEN.1	RLDEN.0
PWMPH	FF98H	PWM周期寄存器高位	00000000	PP.15	PP.14	PP.13	PP.12	PP.11	PP.10	PP.9	PP.8
PWMPH	FF99H	PWM周期寄存器低位	00000000	PP.7	PP.6	PP.5	PP.4	PP.3	PP.2	PP.1	PP.0
PWMCH	FF9AH	PWM计数器寄存器高位	00000000	PC.15	PC.14	PC.13	PC.12	PC.11	PC.10	PC.9	PC.8
PWMCL	FF9BH	PWM计数器寄存器低位	00000000	PC.7	PC.6	PC.5	PC.4	PC.3	PC.2	PC.1	PC.0
PWM0DL	DCH	PWM0占空比控制寄存器低位	00000000	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0
PWM0DH	DDH	PWM0占空比控制寄存器高位	00000000	PD0.15	PD0.14	PD0.13	PD0.12	PD0.11	PD0.10	PD0.9	PD0.8
PWM1DL	DEH	PWM1占空比控制寄存器低位	00000000	PD1.7	PD1.6	PD1.5	PD1.4	PD1.3	PD1.2	PD1.1	PD1.0
PWM1DH	DFH	PWM1占空比控制寄存器高位	00000000	PD1.15	PD1.14	PD1.13	PD1.12	PD1.11	PD1.10	PD1.9	PD1.8
PWM2DL	D2H	PWM2占空比控制寄存器低位	00000000	PD2.7	PD2.6	PD2.5	PD2.4	PD2.3	PD2.2	PD2.1	PD2.0
PWM2DH	D3H	PWM2占空比控制寄存器高位	00000000	PD2.15	PD2.14	PD2.13	PD2.12	PD2.11	PD2.10	PD2.9	PD2.8
PWM01DL	D4H	PWM01占空比控制寄存器低位	00000000	PD01.7	PD01.6	PD01.5	PD01.4	PD01.3	PD01.2	PD01.1	PD01.0
PWM01DH	D5H	PWM01占空比控制寄存器高位	----0000	PD01.15	PD01.14	PD01.13	PD01.12	PD01.11	PD01.10	PD01.9	PD01.8
PWM11DL	D6H	PWM11占空比控制寄存器低位	00000000	PD11.7	PD11.6	PD11.5	PD11.4	PD11.3	PD11.2	PD11.1	PD11.0
PWM11DH	D7H	PWM11占空比控制寄存器高位	----0000	PD11.15	PD11.14	PD11.13	PD11.12	PD11.11	PD11.10	PD11.9	PD11.8
PWM21DL	F9H	PWM21占空比控制寄存器低位	00000000	PD21.7	PD21.6	PD21.5	PD21.4	PD21.3	PD21.2	PD21.1	PD21.0
PWM21DH	FAH	PWM21占空比控制寄存器高位	00000000	PD21.15	PD21.14	PD21.13	PD21.12	PD21.11	PD21.10	PD21.9	PD21.8
PWMCMP1L	BCH	事件触发比较寄存器1低位	00000000	PCMP1.7	PCMP1.6	PCMP1.5	PCMP1.4	PCMP1.3	PCMP1.2	PCMP1.1	PCMP1.0
PWMCMP1H	BDH	事件触发比较寄存器1高位	00000000	PCMP1.7	PCMP1.6	PCMP1.5	PCMP1.4	PCMP1.3	PCMP1.2	PCMP1.1	PCMP1.0

PWMCMP2L	BEH	事件触发比较寄存器2低位	00000000	PCMP2.7	PCMP2.6	PCMP2.5	PCMP2.4	PCMP2.3	PCMP2.2	PCMP2.1	PCMP2.0
PWMCMP2H	BFH	事件触发比较寄存器2高位	00000000	PCMP2.7	PCMP2.6	PCMP2.5	PCMP2.4	PCMP2.3	PCMP2.2	PCMP2.1	PCMP2.0
PWMDT0H	FF92H	死区控制寄存器0高位	0000					DT0.11	DT0.10	DT0.9	DT0.8
PWMDT0L	FF93H	死区控制寄存器0低位	00000000	DT0.7	DT0.6	DT0.5	DT0.4	DT0.3	DT0.2	DT0.1	DT0.0
PWMDT1H	FF94H	死区控制寄存器1高位	0000					DT1.11	DT1.10	DT1.9	DT1.8
PWMDT1L	FF95H	死区控制寄存器1低位	00000000	DT1.7	DT1.6	DT1.5	DT1.4	DT1.3	DT1.2	DT1.1	DT1.0

Table 6.17 Capture SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPCON1	FFB8H	捕捉控制寄存器1	00000000	CAP2EN	CAP1EN	CAP0EN	CAPTG1	CAPTG0	CAPDEB2	CAPDEB1	CAPDEB0
CAPCON2	FFB9H	捕捉控制寄存器2	000-0000	CAPPSC	CAPPSC1	CAPPSC0	-	CAPTPS2	CAPTPS1	CAPTPS0	CAPTR
CAPSTAT	A5H	捕捉状态寄存器	0-000000	CAP0IE	-	CAPSTAT	CAPSTAT	CAP0IF	CAP2	CAP1	CAP0
CAPINT1	8FH	输入捕捉中断寄存器1	0---0---	CAP1IE	-	-	-	CAP1IF	-	-	-
CAPINT2	FFBCH	输入捕捉中断寄存器2	0---0---	CAP2IE	-	-	-	CAP2IF	-	-	-
CAPTINT	FFBDH	输入捕捉定时器中断寄存器	0---0---	CAPTIE	-	-	-	CAPTIF	-	-	-
CAPTOPL	9CH	捕捉存储栈顶寄存器低位	00000000	CAPTOP	CAPTOPL.6	CAPTOPL.5	CAPTOPL.4	CAPTOPL.3	CAPTOPL.2	CAPTOPL.1	CAPTOPL.0
CAPTOPH	9DH	捕捉存储栈顶寄存器高位	00000000	CAPTOP.7	CAPTOPH.6	CAPTOPH.5	CAPTOPH.4	CAPTOPH.3	CAPTOPH.2	CAPTOPH.1	CAPTOPH.0
CAPBOTL	9EH	捕捉存储栈底寄存器低位	00000000	CAPBOT	CAPBOTL.6	CAPBOTL.5	CAPBOTL.4	CAPBOTL.3	CAPBOTL.2	CAPBOTL.1	CAPBOTL.0
CAPBOTH	9FH	捕捉存储栈底寄存器高位	00000000	CAPBOT.7	CAPBOTH.6	CAPBOTH.5	CAPBOTH.4	CAPBOTH.3	CAPBOTH.2	CAPBOTH.1	CAPBOTH.0
CAPTH	FFBAH	捕捉定时器计数寄存器高位	00000000	CAPTH.7	CAPTH.6	CAPTH.5	CAPTH.4	CAPTH.3	CAPTH.2	CAPTH.1	CAPTH.0
CAPTL	FFBBH	捕捉定时器计数寄存器低位	00000000	CAPTL.7	CAPTL.6	CAPTL.5	CAPTL.4	CAPTL.3	CAPTL.2	CAPTL.1	CAPTL.0

Table 6.18 PRAM SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PRAMLOCK	FFC8H	PRAM控制锁定寄存器	00000000	LOCK.7	LOCK.6	LOCK.5	LOCK.4	LOCK.3	LOCK.2	LOCK.1	LOCK.0
PRAMCON	FFC9H	PRAM控制寄存器	0-----00	PRAMLDF	-	-	-	-	-	EXEC	LOAD

Table 6.19 CRC SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CRCCON	FFF8H	CRC控制寄存器	00-----	CRC_GO	CRCIF	-	-	-	-	-	-
CRCDH	FFF9H	CRC校验结果寄存器高位	00000000	CRCD15	CRCD14	CRCD13	CRCD12	CRCD11	CRCD10	CRCD9	CRCD8
CRCDL	FFFAH	CRC校验结果寄存器低位	00000000	CRCD7	CRCD6	CRCD5	CRCD4	CRCD3	CRCD2	CRCD1	CRCD0

Table 6.19 RMDAT SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
RMDAT1	FFC0H	实时监控寄存器1	00000000	RMDAT1.7	RMDAT1.6	RMDAT1.5	RMDAT1.4	RMDAT1.3	RMDAT1.2	RMDAT1.1	RMDAT1.0
RMDAT2	FFC1H	实时监控寄存器2	00000000	RMDAT2.7	RMDAT2.6	RMDAT2.5	RMDAT2.4	RMDAT2.3	RMDAT2.2	RMDAT2.1	RMDAT2.0
RMDAT3	FFC2H	实时监控寄存器3	00000000	RMDAT3.7	RMDAT3.6	RMDAT3.5	RMDAT3.4	RMDAT3.3	RMDAT3.2	RMDAT3.1	RMDAT3.0
RMDAT4	FFC3H	实时监控寄存器4	00000000	RMDAT4.7	RMDAT4.6	RMDAT4.5	RMDAT4.4	RMDAT4.3	RMDAT4.2	RMDAT4.1	RMDAT4.0
RMDAT5	FFC4H	实时监控寄存器5	00000000	RMDAT5.7	RMDAT5.6	RMDAT5.5	RMDAT5.4	RMDAT5.3	RMDAT5.2	RMDAT5.1	RMDAT5.0
RMDAT6	FFC5H	实时监控寄存器6	00000000	RMDAT6.7	RMDAT6.6	RMDAT6.5	RMDAT6.4	RMDAT6.3	RMDAT6.2	RMDAT6.1	RMDAT6.0

注意：-：保留位。

SFR映像图

直接寻址SFR

可接位寻址		不可接位寻址							
0/8		1/9	2/A	3/B	4/C 5/D		6/E	7/F	
F8H	P5	PWM21DL	PWM21DH	IB_OFFSET	IB_DATA	SBRTL	SBRTH		FFH
F0H	B	AUXC	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7H
E8H	PWMINTF	P0PCR	P1PCR	P2PCR	P3PCR	P4PCR	P5PCR	PWMOE	EFH
E0H	ACC	P0CR	P1CR	P2CR	P3CR	P4CR	P5CR	PWMRLDEN	E7H
D8H	EXF1	PWMCON1	PWMCON2	PWMCON3	PWM0DL	PWM0DH	PWM1DL	PWM1DH	DFH
D0H	PSW	FLTCON	PWM2DL	PWM2DH	PWM01DL	PWM01DH	PWM11DL	PWM11DH	D7H
C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TCON1	CLKCON1	CFH
C0H	P4	CMDSCON1	OPRDYL	OPRDYH	OPRDZL	OPRDZH	OPRDXL	OPRDXH	C7H
B8H	IPL0	IPL1	IENC	CMDSCON2	PWMCMP1L	PWMCMP1H	PWMCMP2L	PWMCMP2H	BFH
B0H	P3	RSTSTAT	CLKCON	T3CON	IPH0	IPH1	PMANUAL CON1	PMANUAL CON2	B7H
A8H	IEN0	IEN1	IEN2	T4CON	TL3	TH3	TL4	TH4	AFH
A0H	P2	SPSTA	SPCON	SPDAT	BFINE	CAPSTAT	SEQCON	FLASHCON	A7H
98H	SCON	SBUF	SADDR	SADEN	CAPTOPL	CAPTOPH	CAPBOTL	CAPBOTH	9FH
90H	P1	ADCMPCON	ADCON2	ADCON1	SEQADDR	EXCON	ADDxL/ SEQCHx	ADDxH	97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	SUSLO	CAPINT1	8FH
80H	P0	SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87H
	0/8 1/9		2/A	3/B	4/C 5/D		6/E	7/F	

间接寻址SFR

		不可接位寻址							
0/8		1/9	2/A	3/B	4/C 5/D		6/E	7/F	
FFF8H	CRCCON	CRCDH	CRCDL						FFF8H
FFF0H									FFF0H
FFE8H									FFE8H
FFE0H									FFE0H
FFD8H									FFD8H
FFD0H									FFD0H
FFC8H	PRAMLOCK	PRAMCON							FFC8H
FFC0H	RMDAT1	RMDAT2	RMDAT3	RMDAT4	RMDAT5	RMDAT6			FFC0H
FFB8H	CAPCON1	CAPCON2	CAPTH	CAPTL	CAPINT2	CAPTINT			FFB8H
FFB0H		PWM32CON	PWM32P	PWM32D					FFB0H
FFA8H		PWM31CON	PWM31P	PWM31D					FFA8H
FFA0H	PWM3EN	PWM30CON	PWM30P	PWM30D					FFA0H
FF98H	PWMPH	PWMPL	PWMCH	PWMCL					FF98H
FF90H	PTCON	PWMINTEN	PWMMDT0H	PWMMDT0L	PWMMDT1H	PWMMDT1L			FF90H
FF88H	OPCON1	OPCON2	CMPINT1	CMP2CON1	CMP2CON2	CMPVRS	CMP1CON	CMPSMT	FF88H
FF80H	ADT	ADCH1	ADCH2	ADDLTH	ADDLTL	ADDGTH	ADDGTL	CMPINT2	FF80H
	0/8 1/9		2/A	3/B	4/C 5/D		6/E	7/F	

注意：未使用的SFR地址禁止读写。

7. 标准功能

7.1 CPU

7.1.1 内核特殊功能寄存器

特性

- CPU内核寄存器：ACC, B, PSW, SP, DPL, DPH

累加器

累加器ACC是一个常用的专用寄存器，指令系统中采用A作为累加器的助记符。

B寄存器

在乘法指令中，会用到B寄存器。在其它指令中，B寄存器可作为暂存器来使用。

栈指针（SP）

栈指针SP是一个8位专用寄存器，在执行PUSH、各种子程序调用、中断响应等指令时，SP先加1，再将数据压栈；执行POP、RET、RETI等指令时，数据退出堆栈后SP再减1。堆栈栈顶可以是片上内部RAM（00H-FFH）的任意地址，系统复位后，SP初始化为07H，使得堆栈事实上由08H地址开始。

程序状态字（PSW）寄存器

程序状态字（PSW）寄存器包含了程序状态信息。

Table 7.1 PSW寄存器

D0H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PSW	C	AC	F0	RS1	RS0	OV	F1	P
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	C	进位标志位 0: 算术或逻辑运算中，没有进位或借位发生 1: 算术或逻辑运算中，有进位或借位发生
6	AC	辅助进位标志位 0: 算数逻辑运算中，没有辅助进位或借位发生 1: 算数逻辑运算中，有辅助进位或借位发生
5	F0	F0标志位 用户自定义标志位
4-3	RS[1:0]	R0-R7寄存器页选择位 00: 页0（映射到00H-07H） 01: 页1（映射到08H-0FH） 10: 页2（映射到10H-17H） 11: 页3（映射到18H-1FH）
2	OV	溢出标志位 0: 没有溢出发生 1: 有溢出发生
1	F1	F1标志位 用户自定义标志位
0	P	奇偶校验位 0: 累加器A中值为1的位数为偶数 1: 累加器A中值为1的位数为奇数

数据指针（DPTR）

数据指针DPTR是一个16位专用寄存器，其高位字节寄存器用DPH表示，低位字节寄存器用DPL表示。它们既可以作为一个16位寄存器DPTR来处理，也可以作为2个独立的8位寄存器DPH和DPL来处理。

7.1.2 CPU增强内核特殊功能寄存器

- 扩展的'MUL'和'DIV'指令：16位*8位，16位/8位
- 双数据指针
- CPU增强内核寄存器：AUXC，DPL1，DPH1，INSCON

G80F980cpu核扩展了'MUL'和'DIV'的指令，使用一个新寄存器-AUXC寄存器保存运算数据的高8位，以实现16位运算。在16位乘法指令中，会用到AUXC寄存器。在其它指令中，AUXC寄存器可作为暂存器来使用。

CPU在复位后进入标准模式，'MUL'和'DIV'的指令操作和标准8051指令操作一致。当INSCON寄存器的相应位置1后，'MUL'和'DIV'指令的16位操作功能被打开。

	操作		结果		
			AB		AUXC
MUL	INSCON.2 = 0; 8位模式	(A)*(B)	低位字节	高位字节	---
	INSCON.2 = 1; 16位模式	(AUXC A)*(B)	低位字节	中位字节	高位字节
DIV	INSCON.3 = 0; 8位模式	(A)/(B)	高低位字节	余数	---
	INSCON.3 = 1; 16位模式	(AUXC A)/(B)	商低位字节	余数	商高位字节

注意：此功能为CPU核提供，执行时间较长；G80F980另外集成了一个硬件乘除法单元，执行时间短，参见8-2（乘除法/移位单元）

双数据指针

使用双数据指针能加速数据存储移动。标准数据指针被命名为DPTR而新型数据指针命名为DPTR1。

数据指针DPTR1与DPTR类似，是一个16位专用寄存器，其高位字节寄存器用DPH1表示，低位字节寄存器用DPL1表示。它们既可以作为一个16位寄存器DPTR1来处理，也可以作为2个独立的8位寄存器DPH1和DPL1来处理。

通过对INSCON寄存器中的DPS位置1或清0选择两个数据指针中的一个。所有读取或操作DPTR的相关指令将会选择最近一次选择的数据指针。

寄存器

Table 7.2 数据指针选择寄存器

86H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
INSCON	-	-	-	-	DIV	MUL	-	DPS
读/写	-	-	-	-	读/写	读/写	-	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	-	0

位编号	位符号	说明
3	DIV	16位/8位除法选择位 0: 8位除法 1: 16位除法
2	MUL	16位/8位乘法选择位 0: 8位乘法 1: 16位乘法
0	DPS	数据指针选择位 0: 数据指针 1: 数据指针1

7.2 RAM

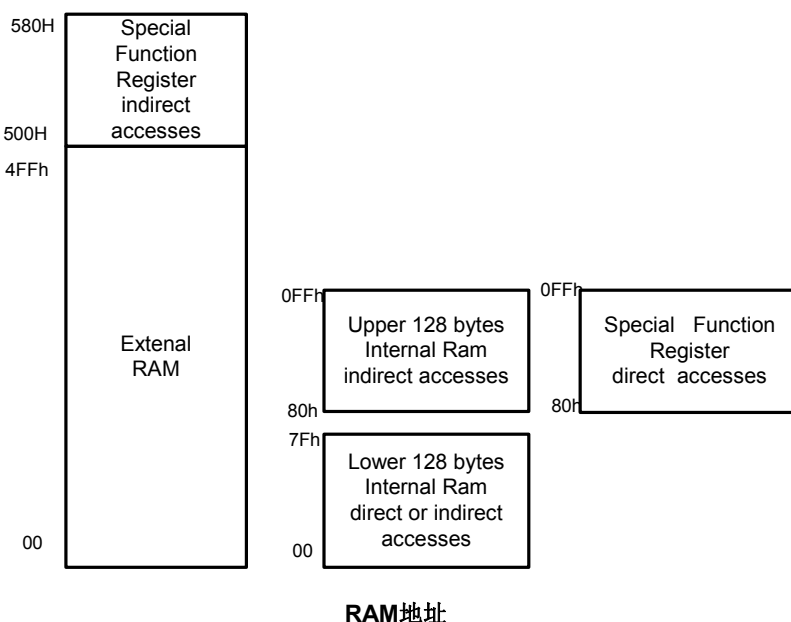
为了提高系统数据变量的处理能力，支持高级语言，G80F980提供了额外的RAM空间。

G80F980除内部RAM 256字节外，还扩展了外部1280字节的RAM，分为以下四个独立的空间：

- 低位128字节的RAM（地址从00H到7FH）可直接或间接寻址。
- 高位128字节的RAM（地址从80H到FFH）只能间接寻址。
- 特殊功能寄存器（SFR，地址从80H到FFH）只能直接寻址。
- 外部RAM可通过MOVX指令间接访问。

高位128字节的RAM占用的地址空间和SFR相同，但在物理上与SFR的空间是分离的。当一个指令访问高于地址7FH的内部位置时，CPU可以根据访问的指令类型来区分是访问高位128字节数据RAM还是访问SFR。

注意：未使用的SFR地址禁止读写。



G80F980支持传统的访问外部RAM方法。可以使用MOVXA, @Ri或MOVX@Ri, A；来访问外部低256字节RAM；使用MOVX A, @DPTR或MOVX@DPTR, A来访问外部1280字节RAM。

用户也能用XPAGE寄存器来访问外部RAM，仅用MOVX A, @Ri或MOVX@Ri, A指令即可。用户能用XPAGE来表示高于256字节的RAM地址。

在Flash SSP模式下，XPAGE也能用作分段选择器（详见SSP章节）。

Table 7.3 数据存储页寄存器（XPAGE）

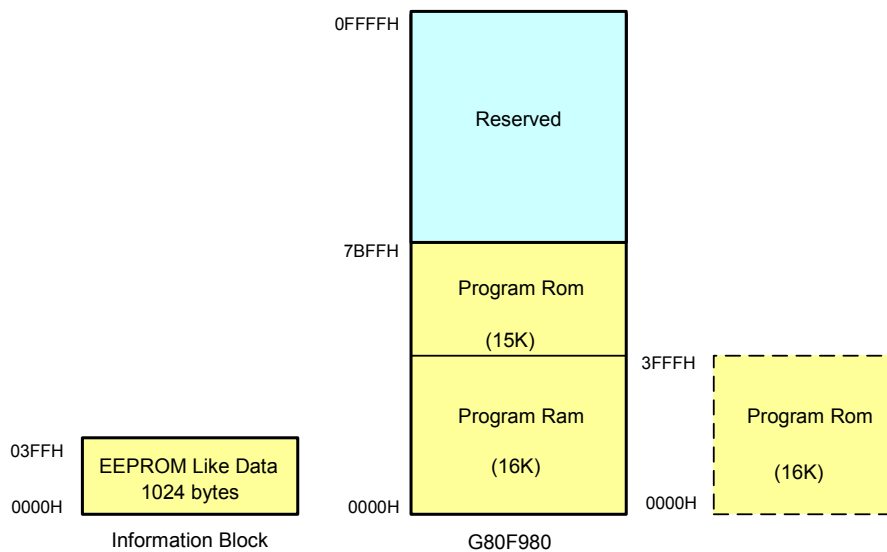
F7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	XPAGE[7:0]	RAM页选择控制位

7.3 Flash/RAM程序存储器

7.3.1 特性

- Flash 存储器包括 62 X 512Byte，总共 31KB
- 集成类 EEPROM 存储器 2 X 512B，总共 1KB
- 16KByte 高速程序 RAM，支持最大 11 MIPS 程序执行(详见上电 Autoload 功能章节)
- 在工作电压范围内都能进行编程和擦除操作
- 支持 3 种代码保护模式
- 在线编程（ICP）操作支持写入、读取和擦除操作
- 支持整体/扇区擦除和编程
- 编程/擦除次数：程序区：至少10,000次
类 EEPROM 区：至少 100,000 次
- 数据保存年限：至少 10 年
- 低功耗



G80F980为存储程序代码内置31K可编程Flash程序存储区（Program Memory Block），支持在线编程（ICP）模式和扇区自编程（SSP）模式对Flash存储器操作。每个扇区512字节。

G80F980还内置1024字节的类EEPROM存储区用于存放用户数据。每个扇区512字节，总共2个扇区。

G80F980内置16Kbyte高速程序RAM，支持代码的高速运行(最大11MHz)，详细请参考[高速程序RAM](#)章节

Flash操作定义：

在线编程（ICP）模式：通过Flash编程器对Flash存储器进行擦、读、写操作。

扇区自编程（SSP）模式：用户程序代码在Flash代码区中运行，对Flash存储器（包括Flash代码区和类EEPROM区）进行擦、读、写操作，但无法擦除代码自身所在的扇区，SSP只对Flash区有效，无法对Program Ram区域执行操作。

Flash存储器支持以下操作：

(1) 代码保护控制模式编程

G80F980的代码保护功能为用户代码提供了高性能的安全措施。共提供4种保护模式。

代码保护模式0：对烧写器加密，允许/禁止任何编程器的写入/读取操作（不包括整体擦除），以4K（8个扇区）为单位，可以分开保护。

代码保护模式1：对MOVC指令加密，允许/禁止在其他扇区中通过MOVC指令进行读取操作，以4K（8个扇区）为单位，可以分开保护。

代码保护模式2：SSP功能允许/禁止控制，选中后，芯片对code区域的SSP操作（擦除或者写入，不包括读取）是禁止的，包括本扇区对自身的SSP，但是不会禁止芯片对类EEPROM的操作。

代码保护模式3：客户密码保护，可由客户自设密码，密码由6字节组成。如果将此功能开启，表示在烧写器或仿真器工具对芯片做任何操作（读出，写入，擦除或者仿真）之前先输入这个密码，如果这个密码正确，则芯片允许烧写器或仿真器工具进行相应的操作，反之则报错，无法执行相应操作。

用户必须使用下列方式才能完成代码保护控制模式的设定：

Flash编程器在ICP模式设置相应的保护位，以进入所需的保护模式，SSP模式不支持代码保护控制模式编程。

(2) 整体擦除

无论代码保护控制模式的状态如何，整体擦除操作都将会擦除所有程序，代码选项，代码保护位，但是不会擦除类EEPROM存储区。

用户必须使用下列方式才能完成整体擦除：

Flash编程器在ICP模式发出整体擦除指令，进行整体擦除。

SSP模式不支持整体擦除。

(3) 扇区擦除

扇区擦除操作将会擦除所选扇区中内容。用户程式（SSP）和Flash编程器都能执行该操作。

若需用户程式执行该操作，必须禁止所选扇区的代码保护控制模式1和保护模式2。

若需编程器执行该操作，必须禁止所选扇区的代码保护控制模式0，如果保护模式3使能，必须输入正确密码。

用户必须使用下列2种方式之一才能完成扇区擦除：

1. Flash编程器在ICP模式发出扇区擦除指令，进行扇区擦除。

2. 通过SSP功能发出扇区擦除指令，进行扇区擦除（详见在扇区自编程章节）。

(4) 类EEPROM存储区擦除

类EEPROM存储区擦除操作将会擦除类EEPROM存储区中的内容。用户程式（SSP）和Flash编程器都能执行该操作。

用户必须使用下列2种方式之一才能完成类EEPROM存储区擦除：

1. Flash编程器在ICP模式发出类EEPROM存储区擦除指令，进行类EEPROM存储区擦除。

2. 通过SSP功能发出类EEPROM存储区擦除指令，进行类EEPROM存储区擦除（详见在扇区自编程章节）。

(5) 写/读代码

读/写代码操作可以将代码从Flash存储器中读出或写入。用户程式（SSP）和编程器都能执行该操作。

若需用户程式执行读代码操作，必须禁止所选扇区的代码保护控制模式1。但不管保护位如何设置，用户程序都能读程式自身所在扇区（512Byte为单位）。

若需用户程式执行写代码操作，必须禁止所选扇区的代码保护模式1。

注：若只使能扇区的代码保护控制模式1，用户程序不能写其它扇区，但能写程式自身所在扇区（512Byte为单位），

若需编程器执行该操作，必须禁止所选扇区的代码保护控制模式0。

用户必须使用下列2种方式之一才能完成写/读代码：

1. Flash编程器在ICP模式发出写/读代码指令，进行写/读代码。

2. 通过SSP功能发出写代码指令，进行写代码操作；通过MOVC指令执行读代码操作。

(6) 写/读类EEPROM存储区

读/写类EEPROM存储区操作可以将数据从类EEPROM存储区中读出或写入。用户程式（SSP）和Flash编程器都能执行该操作。

用户必须使用下列2方式之一才能完成写/读类EEPROM存储区：

1. Flash编程器在ICP模式发出写/读类EEPROM存储区指令，进行写/读类EEPROM存储区。

2. 过SSP功能发出写类EEPROM存储区指令，进行写类EEPROM操作；通过MOVC指令执行读类EEPROM操作。

Flash存储器操作汇总

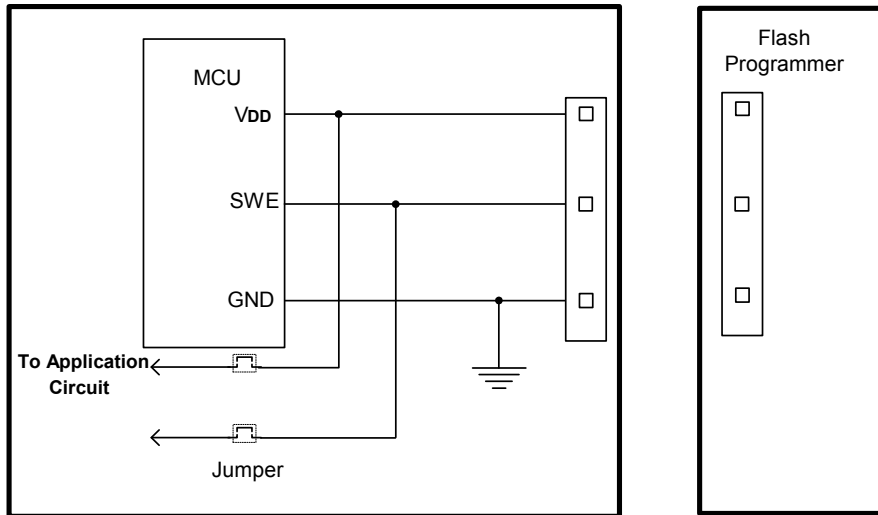
操作	ICP SSP	
代码保护	支持	支持
扇区擦除	支持（无安全位）	支持（无安全位）
整体擦除	支持	不支持
类EEPROM擦除	支持	支持
写/读代码	支持（无安全位）	支持（无安全位）
读/写类EEPROM	支持	支持
代码保护	支持	不支持

7.3.2 ICP模式下的Flash操作（单线模式）

ICP模式为通过Flash编程器对MCU进行编程，可以在MCU焊在用户板上以后编程。ICP模式下，用户系统必须关机后Flash编程器才能通过ICP编程接口刷新Flash存储器。ICP编程接口包括3个引脚（V_{DD}，GND，SWE）。

编程器使用SWE引脚进入编程模式。只有将特定波形输入SWE引脚后，CPU才能进入编程模式。如需详细说明请参考Flash编程器用户指南。

在ICP模式中，通过3线接口编程器能完成所有Flash操作。因为编程信号非常敏感，所以使用编程器编程时用户需要先用2个跳线将芯片的编程引脚（V_{DD}，SWE）从应用电路中分离出来，如下图所示。



当采用ICP模式进行操作时，建议按照如下步骤进行操作：

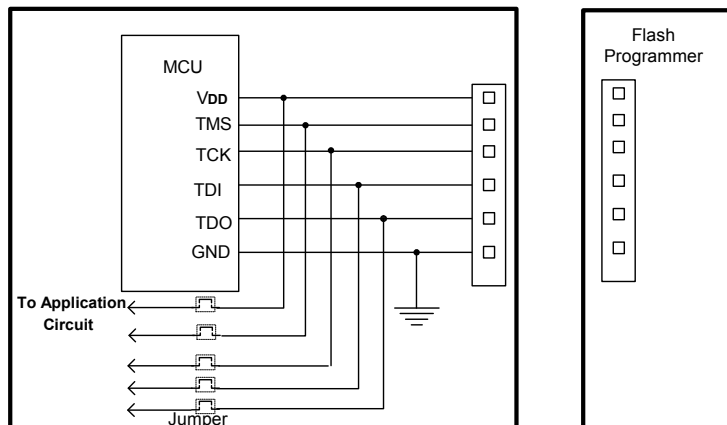
- (1) 在开始编程前断开跳线（jumper），从应用电路中分离编程引脚；
- (2) 将芯片编程引脚连接至 Flash 编程器编程接口，开始编程；
- (3) 编程结束后断开 Flash 编程器接口，连接跳线恢复应用电路。

7.3.3 ICP模式下的Flash操作（四线模式）

ICP模式为通过Flash编程器对MCU进行编程，可以在MCU焊在用户板上以后编程。ICP模式下，用户系统必须关机后Flash编程器才能通过ICP编程接口刷新Flash存储器。ICP编程接口包括6个引脚（V_{DD}，GND，TCK，TDI，TMS，TDO）。

编程器使用4个JTAG引脚（TDO，TDI，TCK，TMS）进入编程模式。只有将特定波形输入4个引脚后，CPU才能进入编程模式。如需详细说明请参考Flash编程器用户指南。

在ICP模式中，通过6线接口编程器能完成所有Flash操作。因为编程信号非常敏感，所以使用编程器编程时建议用户需要先6个跳线将芯片的编程引脚（V_{DD}，GND，TCK，TDI，TMS，TDO）从应用电路中分离出来，如下图所示。



当采用ICP模式进行操作时，建议按照如下步骤进行操作：

1. 在开始编程前断开跳线（jumper），从应用电路中分离编程引脚
2. 将芯片编程引脚连接至Flash编程器编程接口，开始编程
3. 编程结束后断开Flash编程器接口，连接跳线恢复应用电路

如果不加跳线，需保证电源线上的电容负载不超过100uF，4根信号线上的电容负载不超过0.01uf，电阻负载不小于1K阻值。

7.4 SSP

7.4.1 寄存器

擦除/编程用扇区选择和编程用地址偏移量寄存器

此寄存器用来选择待擦除或者待编程扇区的区号，配合IB_OFFSET寄存器来表示待编程字节在扇区内的地址偏移量。

对于程序存储区，一个扇区为512字节，寄存器定义如下：

Table 7.4 编程用地址选择寄存器

F7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-1	XPAGE[7:1]	被编程的存储单元扇区号，0000000代表扇区0，以此类推
0	XPAGE[0]	被编程的存储单元最高位地址

Table 7.5 编程用地址偏移寄存器

FBH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_OFFSET	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	IB_OFFSET[7:0]	被编程的存储单元低8位地址

XPAGE[0]和IB_OFFSET[7:0]共9位，可以表示1个程序存储扇区内全部512个字节的偏移量。

类EEPROM扇区，一个扇区为512字节，共2个扇区，寄存器定义如下：

Table 7.6 擦除/编程用扇区选择寄存器

F7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-2	XPAGE[7:2]	在擦除/编程扇区时无意义
1	XPAGE[1]	被擦除/编程的扇区选择位 00: 扇区0 01: 扇区1
0	XPAGE[0]	被编程的存储单元最高位地址

类EEPROM块区的访问可通过指令“MOVC A, @A+DPTR”或“MOVC A, @A+PC”实现。

注意：需要将FLASHCON寄存器中的FAC位置1。

Table 7.7 编程用地址偏移寄存器

FBH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_OFFSET	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	IB_OFFSET[7:0]	被编程的存储单元低8位地址

XPAGE[0]和IB_OFFSET[7:0]共9位，可以表示1个块区内全部512个字节的偏移量。

Table 7.8 编程用数据寄存器

FCH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_DATA	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	IB_DATA[7:0]	待编程数据

Table 7.9 SSP型选择寄存器

F2H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_CON1	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
读/写	-	-	-	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	0	0	0	0	0

位编号	位符号	说明
7-0	IB_CON1[7:0]	SSP操作选择 0xE6: 扇区擦除 0x6E: 存储单元编程

Table 7.10 SSP流程控制寄存器1

F3H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_CON2	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	IB_CON2[3:0]	必须为05H，否则Flash编程将会终止

Table 7.11 SSP流程控制寄存器2

F4H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_CON3	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	IB_CON3[3:0]	必须为0AH, 否则Flash编程将会终止

Table 7.12 SSP流程控制寄存器3

F5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_CON4	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

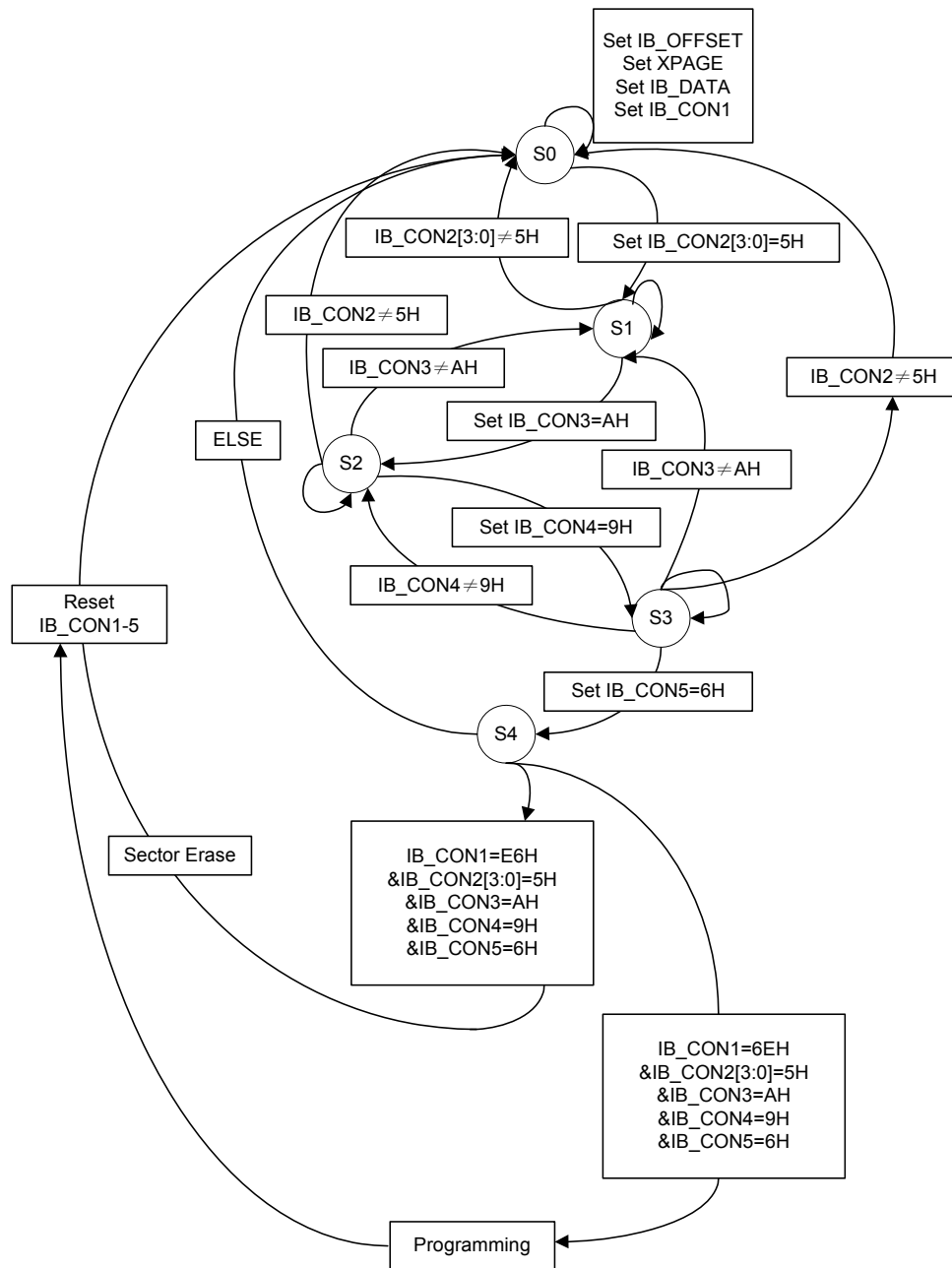
位编号	位符号	说明
3-0	IB_CON4[3:0]	必须为09H, 否则Flash编程将会终止

Table 7.13 SSP流程控制寄存器4

F6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IB_CON5	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	IB_CON5[3:0]	必须为06H, 否则Flash编程将会终止

7.4.2 Flash控制流程图



7.4.3 SSP编程注意事项

为确保顺利完成SSP编程，用户软件必须按以下步骤设置：

(1) 用于代码/数据编程：

1. 关闭中断；
2. 根据地址设置XPAGE, IB_OFFSET；
3. 按编程需要，设置IB_DATA；
4. 按照顺序设置IB_CON1 - 5；
5. 添加4个NOP指令；
6. 开始编程，CPU将进入IDLE模式；编程完成后自动退出IDLE模式；
7. 如需继续写入数据，跳转至第2步；
8. XPAGE寄存器清0；恢复中断设置。

(2) 用于扇区擦除：

1. 关闭中断；
2. 按相应的扇区设置XPAGE；
3. 按照顺序设置IB_CON1 - 5；
4. 添加4个NOP指令；
5. 开始擦除，CPU将进入IDLE模式；擦除完成后自动退出IDLE模式；
6. 如需继续擦除数据，跳转至第2步；
7. XPAGE寄存器清0；恢复中断设置。

(3) 读取：

使用“MOVC A, @A+DPTR”或者“MOVC A, @A+PC”。

7.4.4 可读识别码

G80F980每颗芯片出厂后都固化有一个40位的可读识别码，它的值为0 - 0xffffffff的随机值，它是无法擦除的（存放在地址信息存储区0x127b - 127f），可以由程序或编程工具读出。

程序读出示例：

```
Unsigned char Temp1, Temp2, Temp3, Temp4, Temp5;
FLASHCON = 0x01;
Temp1 = CBYTE[0x127b];
Temp2 = CBYTE[0x127c];
Temp3 = CBYTE[0x127d];
Temp4 = CBYTE[0x127e];
Temp5 = CBYTE[0x127f];
FLASHCON = 0x00;
```

FLASHCON寄存器的描述如下：

Table 7.14 Flash访问控制寄存器

A7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
FLASHCON	-	-	-	-	-	-	-	FAC
读/写	-	-	-	-	-	-	-	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	-	-	0

位编号	位符号	说明
0	FAC	访问控制 0: MOVC指令或者SSP功能访问主程序区域 1: MOVC指令或者SSP功能访问类EEPROM区域

7.5 系统时钟和振荡器

7.5.1 特性

- 高低频双振荡器系统
- 支持4种高频振荡器类型：晶体谐振器、陶瓷谐振器和内部12M RC振荡器
- 内建倍频系数可选锁相环（PLL）振荡器
- 内建128k低频RC振荡器
- 内部12M 高频RC振荡器
- 内建系统时钟分频器

7.5.2 时钟定义

S80F980几个内部时钟定义如下：

CRYCLK: 从XTAL输入的400k~16MHz晶体/陶瓷谐振器的时钟或外部。 f_{CRY} 定义为CRYCLK的频率。 t_{CRY} 定义为CRYCLK的周期。

LRCCLK: 内部128kHz RC振荡器时钟。 f_{LRC} 定义为LRCCLK的频率。 t_{LRC} 定义为LRCCLK的周期。

HRCCLK: 内部12MHz RC振荡器时钟。 f_{HRC} 定义为HRCCLK的频率。 t_{HRC} 定义为HRCCLK的周期。

PLLCLK: 锁相环（PLL）振荡器时钟。 f_{PLL} 定义为PLLCLK的频率， t_{PLL} 定义为PLLCLK的周期

SCMCLK: 内部2MHz时钟监控RC振荡器时钟。 f_{SCM} 定义为SCMCLK的频率。 t_{SCM} 定义为SCMCLK的周期。

WDTCLK: 内部的2kHz看门狗RC振荡器时钟。 f_{WDT} 定义为WDTCLK的频率。 t_{WDT} 定义为WDTCLK的周期。

OSC1CLK: 通过代码选项OP_OSC（详见代码选项章节）选中的振荡器1的时钟。 f_{OSC1} 定义为OSC1CLK的频率。 t_{OSC1} 定义为OSC1CLK的周期。

OSC2CLK: 通过代码选项OP_OSC（详见代码选项章节）选中的振荡器2的时钟。 f_{OSC2} 定义为OSC2CLK的频率。 t_{OSC2} 定义为OSC2CLK的周期。

OSCSCLK: 系统时钟频率分频器的输入时钟。这个时钟可能为LRCCLK、HRCCLK、CRYCLK 或PLLCLK。 f_{OSCS} 定义为OSCSCLK的频率。 t_{OSCS} 定义为OSCSCLK的周期。

SYSCLK: 系统时钟，系统频率分频器的输出时钟。这个时钟为CPU指令周期的时钟。 f_{SYS} 定义为SYSCLK的频率。 t_{SYS} 定义为SYSCLK的周期。

概述

G80F980采用双振荡器系统架构，高频振荡器支持3种振荡器类型：晶体谐振器（400kHz-16MHz），陶瓷谐振器（400kHz-16MHz）和内部RC振荡器（12MHz），振荡器类型的选择由代码选项OP_OSC决定（详见代码选项章节）。低频振荡器采用内置的128KRC振荡器。由振荡器产生的基本时钟脉冲提供系统时钟支持CPU及片上外围设备。

G80F980内建一个锁相环振荡器，当选择高频时钟源时，PLL振荡器能提供最高11MHz振荡频率。CLKCON1控制寄存器能禁止或使能PLL振荡器及控制PLL倍频系数（PLL_MUL）；当选择低频时钟源时，PLL功能不起作用。

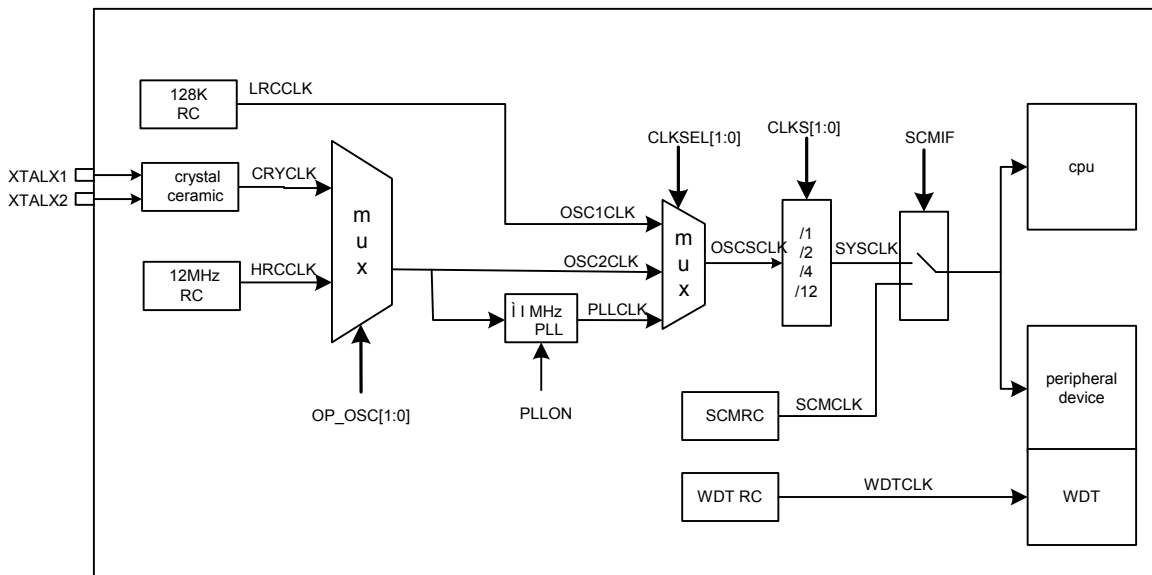


Table 7.15 系统时钟控制寄存器

B2H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CLKCON	-	CLKS1	CLKS0	SCMIF	-	CLKSEL1	CLKSEL0	-
读/写	-	读/写	读/写	只读	-	读/写	读/写	-
复位值 (POR/WDT/LVR/PIN)	-	1	1	0	-	0	1	-

位编号	位符号	说明
6-5	CLKS[1:0]	<p>系统时钟预分频器</p> <p>00: $f_{\text{SYS}} = f_{\text{OSC}}$</p> <p>01: $f_{\text{SYS}} = f_{\text{OSC}}/2$</p> <p>10: $f_{\text{SYS}} = f_{\text{OSC}}/4$</p> <p>11: $f_{\text{SYS}} = f_{\text{OSC}}/12$</p> <p>注意: 系统复位后此寄存器默认为11, 即初始化默认系统时钟12分频。当选择OSC1CLK (内部128K RC) 为OSCSCLK时, 系统时钟分频无效</p>
4	SCMIF	<p>系统时钟监控标志位</p> <p>0: 表示系统时钟正常运行</p> <p>1: 表示系统时钟故障, 硬件自动切换到内部2M RC振荡器</p>
2-1	CLKSEL[1:0]	<p>系统时钟切换控制位</p> <p>00: 选择OSC2CLK为OSCSCLK</p> <p>01: 选择OSC1CLK为OSCSCLK</p> <p>10: 选择PLLCLK为OSCSCLK</p> <p>其他值无效 (无法写入)</p>

Table 7.16 时钟控制寄存器1

CFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CLKCON1	OSC2ON	OSC2F	-	PLLON	PLLLOCK	PLLMUL2	PLLMUL1	- PLLMUL0
读/写	读/写	读	-	读/写	读	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	-	0	0	0	0	0

位编号	位符号	说明
7	OSC2ON	<p>高频振荡器控制位</p> <p>0: 高频振荡器关闭</p> <p>1: 高频振荡器打开</p> <p>注: 当PLL打开时 (PLLON=1) 及CLKSEL=00或10时, OSC2ON不能清零; 在开启高频振荡器前, 需提前 25us 开启辅助电源 VBIAS (设置ADCH2[4] = 1)</p>
6	OSC2F	<p>高频振荡器准备就绪标志位</p> <p>0: 硬件清0表示高频振荡器正在预热或者没有使能</p> <p>1: 硬件置1表示高频振荡器准备就绪, 只能硬件置1</p> <p>注: OSC2ON 开启后务必等待OSC2F置1振荡器稳定后, 再将系统时钟切换至OSC2CLK; OSC2F在OSC2ON=0时同时清0</p>

4	PLLON	锁相环（PLL）振荡器使能位 0: PLL振荡器关闭 1: PLL振荡器打开 <i>注：只有在高频振荡器打开且准备就绪后（OSC2ON=1且OSC2F=1），PLLON设置为1才有效</i>
3	PLLLOCK	锁相环（PLL）振荡器准备就绪标志位 0: 硬件清0表示PLL振荡器输出未稳定或者没有使能 1: 硬件置1表示PLL振荡器准备就绪，只能硬件置1 <i>注：PLLON开启后务必等待PLLLOCK置1PLL稳定后，再将系统时钟切换至PLLCLK</i>
2-0	PLLMUL[2:0]	PLL倍频系数选择位 000: X 2 001: X 3 010: X 4 011: X 6 100: X 8 101: X 7 110: X 10 111: X 10

注意：

- 当OSC2CLK或PLLCLK作为OSCSCLK时（也就是说CLKSEL[1:0]=00或10），OSC2ON不能软件清0
- 当PLLCLK作为OSCSCLK时（也就是说CLKSEL[1:0]=10），PLLON不能软件清0
- 当OSCSCLK从LRCCLK（128KHz低频RC）切换到OSC2CLK或PLLCLK时，假如当时高频时钟OSC2CLK为关闭状态，则必须按以下步骤依次设置：
 - 设置VBIAS=1，开启辅助电源（默认为已开启）
 - 设置OSC2ON=1，打开OSC2CLK（高频时钟类型由代码选项OP_OSC确定）
 - 至少等待振荡器预热时间（详见**振荡器预热**章节）及功耗切换时间（共约2ms）
 - 设置CLKSEL[1:0]=00，选择OSC2CLK作为OSCSCLK
 - 如无需开启PLL，则设置完成。
 - 如需开启PLL，则设置PLLON=1，设置PLLMUL倍频系数
 - 查询PLLLOCK标志位，等待PLL稳定后，设置CLKSEL[1:0]=10，选择PLLCLK作为OSCSCLK，设置完成
 - 如运行中改变PLLMUL倍频系数则需要先设置CLKSEL[1:0]=00，选择OSC2CLK作为OSCSCLK
 - 关闭PLL（PLLON=0），等待至少100ns；
 - 开启PLL（设置PLLON=1），同时设置新的PLLMUL倍频系数
 - 查询PLLLOCK标志位，等待稳定后设置CLKSEL[1:0]=10完成
- 当OSCSCLK从OSC2CLK或PLLCLK切回到128K时，则必须按以下步骤依次设置：
 - 如PLL已开启，则选择CLKSEL=00，选择OSC2CLK作为OSCSCLK
 - 设置PLLON=0，关闭PLL，可降低系统功耗
 - 设置CLKSEL=01，选择LRCCLK（128KHz低频RC）作为OSCSCLK
 - 添加1个NOP指令
 - OSC2ON位清0，关闭高频时钟源，可进一步降低系统功耗

7.5.3 系统时钟监控（SCM）

为了增强系统的可靠性，G80F980含有一个系统时钟监控（SCM）模块。在系统选择高频时钟时（不管是单时钟模式还是双时钟模式），如果系统时钟出现故障（例如：高频振荡器停振等），内建SCM模块会将OSCCLK自动切换到内部2M振荡器，同时系统时钟监控标志位（SCMIF）被置1。如果EA和ESCM位均为1，SCM模块将会产生中断。如果高频振荡器恢复工作，SCM会将OSCCLK自动切换回高频振荡器，然后SCMIF位自动清0。

注意：

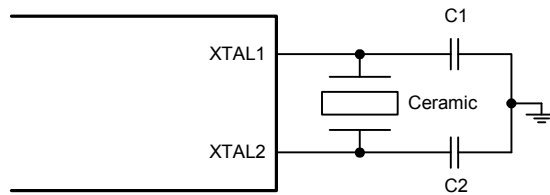
SCMIF为只读位，只能由硬件清0或者置1。

如果SCMIF清0，SCM将系统时钟自动切换到系统时钟出故障前的状态。

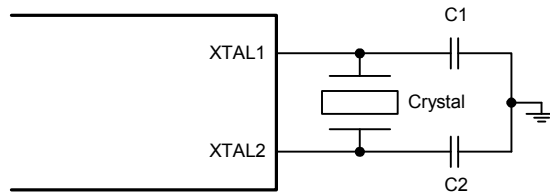
如果系统工作在128K低频振荡器，则系统时钟监控功能不可用。

7.5.4 振荡器类型

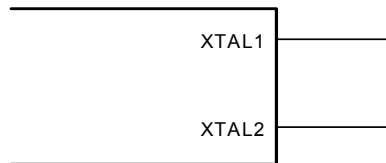
(1) 陶瓷谐振器: 400kHz - 16MHz



(2) 晶体谐振器: 400k - 16MHz



(3) 内部RC振荡器: 12MHz (不需要外部电路)



7.5.5 谐振器负载电容选择

陶瓷谐振器		
频率	C1	C2
455kHz	47 - 100pF	47 - 100pF
11.0592MHz	8 - 15pF	8 - 15pF

* 表示已经内建负载电容。

注意:

1. **表中负载电容仅供参考!**
2. 以上电容值可通过谐振器基本的起振和运行测试，**并非最优值。**
3. 请注意印制板上的杂散电容，用户应在超过应用电压和温度的条件下测试谐振器的性能。
在应用陶瓷谐振器/晶体谐振器之前，用户需向谐振器生产厂要求相关应用参数以获得最佳性能。

7.6 I/O端口

7.6.1 特性

- 46个双向I/O端口
- I/O端口可与其他功能共用

G80F980提供46个可编程双向I/O端口。端口数据在寄存器Px中。端口控制寄存器（PxCRy）控制端口是作为输入或者输出。当端口作为输入时，每个I/O端口带有由PxPCRY控制的内部上拉电阻（x = 0-5, y = 0-7）。

G80F980的有些I/O引脚能与选择功能共享。当所有功能都允许时，在CPU中存在优先权以避免功能冲突。（详见端口共享章节）。

7.6.2 寄存器

Table 7.16 端口控制寄存器

E1H - E6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
P0CR (E1H)	P0CR.7	P0CR.6	P0CR.5	P0CR.4	P0CR.3	P0CR.2	P0CR.1	P0CR.0
P1CR (E2H)	P1CR.7	P1CR.6	P1CR.5	P1CR.4	P1CR.3	P1CR.2	P1CR.1	P1CR.0
P2CR (E3H)	P2CR.7	P2CR.6	P2CR.5	P2CR.4	P2CR.3	P2CR.2	P2CR.1	P2CR.0
P3CR (E4H)	P3CR.7	P3CR.6	P3CR.5	P3CR.4	P3CR.3	P3CR.2	P3CR.1	P3CR.0
P4CR (E5H)	P4CR.7	P4CR.6	P4CR.5	P4CR.4	P4CR.3	P4CR.2	P4CR.1	P4CR.0
P5CR (E6H)	-	-	P5CR.5	P5CR.4	P5CR.3	P5CR.2	P5CR.1	P5CR.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	PxCR.y x = 0-5, y = 0-7	端口输入/输出控制 0: 输入模式 1: 输出模式

Table 7.17 端口上拉电阻控制寄存器

E9H- EEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
P0PCR (E9H)	P0PCR.7	P0PCR.6	P0PCR.5	P0PCR.4	P0PCR.3	P0PCR.2	P0PCR.1	P0PCR.0
P1PCR (EAH)	P1PCR.7	P1PCR.6	P1PCR.5	P1PCR.4	P1PCR.3	P1PCR.2	P1PCR.1	P1PCR.0
P2PCR (EBH)	P2PCR.7	P2PCR.6	P2PCR.5	P2PCR.4	P2PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
P3PCR (ECH)	P3PCR.7	P3PCR.6	P3PCR.5	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	P3PCR.0
P4PCR (EDH)	P4PCR.7	P4PCR.6	P4PCR.5	P4PCR.4	P4PCR.3	P4PCR.2	P4PCR.1	P4PCR.0
P5PCR (EEH)	-	-	P5PCR.5	P5PCR.4	P5PCR.3	P5PCR.2	P5PCR.1	P5PCR.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

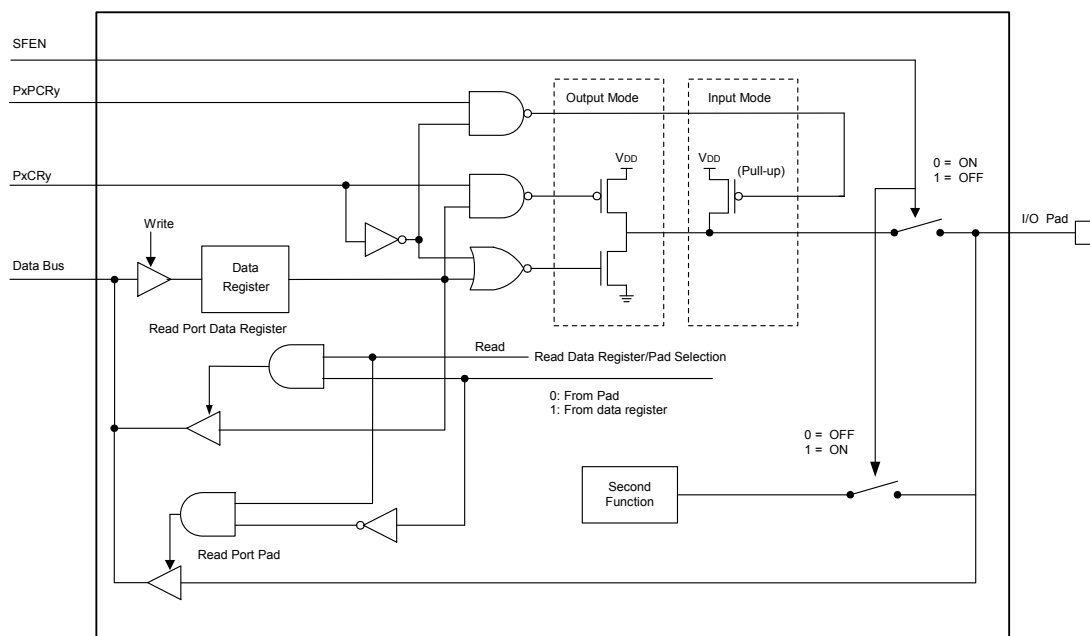
位编号	位符号	说明
7-0	PxPCR.y x = 0-5, y = 0-7	输入端口内部上拉电阻控制 0: 内部上拉电阻关闭 1: 内部上拉电阻开启

Table 7.18 端口数据寄存器

80H - F8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
P0 (80H)	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1 (90H)	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2 (A0H)	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3 (B0H)	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4 (C0H)	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
P5 (F8H)	-	-	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	Px.y x = 0-5, y = 0-7	端口数据寄存器

7.6.3 端口模块图



注意:

1. 输入端口读操作直接读引脚电平。
2. 输出端口读操作的输入源有两种，一种是从端口数据寄存器读取，另一种是直接读引脚电平。用读取指令来区分：读-改写指令是读寄存器，而其它指令读引脚电平。
3. 不管端口是否共用为其他功能，对端口写操作都是针对端口数据寄存器。

7.6.4 端口共用

46个双向I/O端口也能共享作为第二或第三种特殊功能。共享优先级按照外部最高内部最低的规则：

在**引脚配置图**中引脚最外边标注功能享有最高优先级，最里边标注功能享有最低优先级。这意味着一个引脚已经使用较高优先级功能（如果被允许的话），就不能用作较低优先级功能，即使较低优先级功能被允许。只有较高优先级功能由硬件或软件关闭后，相应的引脚才能用作较低优先级功能。上拉电阻也由相同规则控制。

当允许端口复用为其它功能时，用户可以修改PxCR、PxPCR（x = 0 - 5），但在复用的其它功能被禁止前，这些操作不会影响端口状态。

当允许端口复用为其它功能时，任何写入端口操作只会影响到数据寄存器；而对于读操作，如果I/O复用为中断功能，则返回引脚电平，而复用为其它功能时，返回端口寄存器的值。

PORT0:

- OP1P (P0.0) : OP1输入正端
- C1P (P0.0) : 比较器1输入正端
- OP1N (P0.1) : OP1输入负端
- OP1OUT (P0.2) : OP1输出端
- C1N (P0.3) : 比较器1输入负端
- AN4, AN7-AN10 (P0.7, P0.3 - P0.6) : ADC输入通道
- C2P0-C2P2 (P0.4 - P0.6) : 比较器2的三个输入正端
- INT40-INT42 (P0.6 - P0.4) : 外部中断4的三个输入端

Table 7.19 PORT0共用列表

引脚编号	优先级	功能	允许位
46	1	C1P/OP1P	CMP1CON寄存器的CMP1EN位置1且C1PCHS清0或OPCON1寄存器的OP1EN位置1。 <i>注：比较器功能与放大器功能可同时实现（CMP1EN=1&C1PCHS=0且OP1EN=1）</i>
	2	P0.0	CMP1CON寄存器的CMP1EN位清0，OPCON1寄存器的OP1EN位清0
45	1	OP1N	OPCON1寄存器的OP1EN位置1，同时OP1G位为0
	2	P0.1	OPCON1寄存器的OP1EN位清0，或OP1EN位置1但OP1G位不为0
44	1	OP1OUT	OPCON1寄存器的OP1EN位置1，同时OP1G位为0
	2	P0.2	OPCON1寄存器的OP1EN位清0，或OP1EN位置1但OP1G位不为0
43	1	C1N	CMP1CON寄存器的CMP1EN位置1，且C1NCHS清零
	2	AN7	ADCH1寄存器的CH7位置1 <i>注：比较器功能与ADC功能可以同时实现</i>
	3	P0.3	CMP1CON寄存器的CMP1EN位清0，或CM1EN置1但C1NCHS清零
42-40	1	C2P0-C2P2/ AN8-AN10	ADCH2寄存器的CH8, CH9, CH10位置1作C2P0-C2P2或AN8-AN10的模拟端口。 CMP2CON1寄存器的CMP2EN位置1，C2P0-C2P2由C2PCHS[1:0]选择作为比较器正端输入 <i>注：比较器功能与ADC功能可以同时实现</i>
	2	INT40-INT42	不满足优先级1的情况下，IEN1寄存器的EX4置1，IENC寄存器EXS40 - EXS42置1，且IO为输入。此时I/O口的输入功能仍然有效。
	3	P0.4-P0.6	不满足优先级1的情况下都作I/O口
39	1	AN4	ADCH1寄存器的CH4位置1
	2	P0.7	ADCON1寄存器的CH4清0

PORT1:

- AN5 – AN6 (P1.1 – P1.0) : ADC 输入通道
- INT43-INT44 (P1.0 – P1.1) : 外部中断4的两个输入端
- TDO, TMS, TDI, TCK (P1.2 - P1.5) : 四线仿真 JTAG 接口
- OP2OUT (P1.2) : OP2输出端
- SWE (P1.2) : 单线仿真接口
- OP2N (P1.3) : OP2输入负端
- RST (P1.3) : PIN 复位输入
- OP2P (P1.4) : OP2输入正端
- OP3P (P1.5) : OP3输入正端
- OP3N (P1.6) : OP3输入负端
- OP3OUT (P1.7) : OP3输出端

Table 7.20 PORT1共用列表

引脚编号	优先级	功能	允许位
47	1	AN6	ADCH1寄存器的CH6位置1
	2	INT43	不满足优先级1的情况下, IEN1寄存器的EX4置1, IENC寄存器EXS43置1, 且IO为输入。此时I/O口的输入功能仍然有效。
	3	P1.0	不满足优先级1的情况下都作I/O口
48	1	VREF	ADCON1寄存器的REFC位置1
	2	AN5	不满足优先级1的情况下, ADCH1寄存器的CH5位置1
	3	INT44	不满足优先级1、2的情况下, IEN1寄存器的EX4置1, IENC寄存器EXS44置1, 且IO为输入。此时I/O口的输入功能仍然有效。
	4	P1.1	不满足优先级1、2的情况下都作I/O口
1	1	TDO	进入仿真模式
	2	SWE	进入仿真模式
	3	OP2OUT	非仿真模式下, OPCON1寄存器的OP2EN位置1并且OP2G位清0
	4	P1.2	不满足优先级1、2的情况下始终作为I/O。
2	1	TMS	进入仿真模式
	2	RST	代码选项选择OP_RST作为reset pin
	3	OP2N	OPCON1寄存器的OP2EN位置1并且OP3G位清0
	4	P1.3	不满足优先级1、2的情况下始终作为I/O。
3	1	TDI	进入仿真模式
	2	OP2P	OPCON1寄存器的OP2EN位置1
	3	P1.4	不满足优先级1的情况下始终作为I/O。
4	1	TCK	进入仿真模式
	2	OP3P	OPCON2寄存器的OP3EN位置1
	3	P1.5	不满足优先级1的情况下始终作为I/O
5	1	OP3N	OPCON2寄存器的OP3EN位置1并且OP3G位清0
	2	P1.6	不满足优先级1的情况下始终作为I/O
6	1	OP3OUT	OPCON2寄存器的OP3EN位置1并且OP3G位清0
	2	P1.7	不满足优先级1的情况下始终作为I/O

PORT2:

- PWMn (n = 0.1.2), PWMn1 (n = 0.1.2) (P2.0-P2.5): PWM输出
- T0, T1 (P2.6, P2.7): 定时器0, 1输入或比较功能输出
- FLT (P2.6): 故障检测输入

Table 7.21 PORT2共用列表

引脚编号	优先级	功能	允许位
17-22	1	PWMn PWMn1 (n = 0.1.2)	PWMOE寄存器的PWMnOE (n = 0.1.2), PWMn1OE (n = 0.1.2) 位置1
	2	P2.0-P2.5	PWMOE寄存器的PWMnOE (n = 0.1.2), PWMn1OE (n = 0.1.2) 位清0
23	1	FLT	PWM0/1/2模块的FLTCON寄存器的FLT2EN位置1
	2	T0	FLTCON寄存器的FLT2EN位清0, TCON寄存器的TR0位置1, 且TMOD寄存器的C/T0位或TCON1寄存器的TC0位 其中一个置1
	3	P2.6	FLTCON寄存器的FLT2EN位清0, TCON寄存器的TR0位清0, 或TMOD寄存器的C/T0位和TCON1寄存器的TC0位 都为0
24	1	T1	TCON寄存器的TR1位置1, 且TMOD寄存器的C/T1位或TCON1寄存器的TC1位 其中一个置1
	2	P2.7	TCON寄存器的TR1位清0, 或TMOD寄存器的C/T1位和TCON1寄存器的TC1位 都为0

PORT3:

- PWM30 (P3.0) : PWM30的输出
- RXD (P3.1) : EUART的输入
- TXD (P3.2) : EUART的输出
- INT0, INT0 (P3.1, P3.2) : 外部中断0/1输入
- XTAL1, XTAL2 (P3.4, P3.5) : 振荡器输入输出
- CAP0 – CAP2 (P3.3 – P3.5): 输入捕捉口
- INT45 - INT46 (P3.7 - P3.6) : 外部中断4的两个输入端

Table 7.22 PORT3共用列表

引脚编号	优先级	功能	允许位
16	1	PWM30	PWM3EN寄存器的PWM30EN位置1, 同时PWM30OE置1
	2	P3.0	不满足优先级1情况下, 始终为IO
15	1	RXD	SCON寄存器的REN位置1 (自动上拉)
	2	INT0	IEN0寄存器的EA位和EX0位置1, 且IO为输入作中断口功能INT0或T0的GATE=1时, 作T0定时计数器控制输入端INT0。此时外中断功能仍然有效。
	3	P3.1	SCON寄存器的REN位清0, 始终作为I/O
14	1	TXD	对SBUF寄存器写操作
	2	INT1	UART非传输过程中, IEN0寄存器的EA位和EX1位置1, 且IO为输入, 作中断口功能INT1或T1的GATE=1时, 作T1定时计数器控制输入端INT1。此时外中断功能仍然有效。
	3	P3.2	对SBUF寄存器无操作, 始终作为I/O
13	1	CAP0	CAPCON1寄存器的CAP0EN置1, 且IO为输入。此时I/O口的输入功能仍然有效。
	2	P3.3	CAPCON1寄存器的CAP0EN位清0, 始终作为I/O
12, 11	1	XTAL1, XTAL2	代码选项OP_OSC选为晶振或陶振
	2	CAP1, CAP2	代码选项OP_OSC选为内部RC。CAPCON1寄存器的CAP1EN, CAP2EN置1, 且IO为输入。此时I/O口的输入功能仍然有效。
	3	P3.4, P3.5	代码选项OP_OSC选为内部RC。CAPCON1寄存器的CAP1EN, CAP2EN清0, 始终作为I/O
7, 9	1	INT45 – INT46	IEN1寄存器的EX4置1, IENC寄存器EXS45, EXS46置1, 且IO为输入。此时I/O口的输入功能仍然有效。
	2	P3.7, P3.6	不满足上述情况

PORT4:

- AN0 – AN3 (P4.4-P4.3, P4.1-P4.0) : ADC的4路模拟通道
- C2N (P4.0) : 比较器2输入负端
- OP4P (P4.2) : OP4输入正端
- OP4N (P4.3) : OP4输入负端
- OP4OUT (P4.4) : OP4输出端
- PWM31, PWM32 (P4.5, P4.6) : PWM31, PWM32输出
- T4 (P4.6) : 定时器4输入
- T2EX (P4.7) : 定时器4输入

Table 7.23 PORT4共用列表

引脚编号	优先级	功能	允许位
38	1	C2N AN3	CMP2CON1寄存器的CMP2EN开启, 且 C2N清0。ADCH1寄存器的CH3位置1 <i>注: 比较器功能与ADC功能可以同时实现</i>
	2	P4.0	不满足优先级1的情况下, 始终作为IO口
37	1	AN2	ADCH1寄存器的CH2位置1
	2	P4.1	不满足优先级1的情况下, 始终作为IO口
36	1	OP2P	OPCON2寄存器的OP4EN位置1
	2	P4.2	不满足上述情况, 始终做IO口
35	1	OP4N	OPCON2寄存器的OP4EN位置1, 同时OP4G位为0
	2	AN1	ADCH1寄存器的CH1位置1
	3	P4.3	不满足上述情况, 始终做IO口
34	1	OP4OUT	OPCON2寄存器的OP4EN位置1, 同时OP4G位为0
	2	AN0	ADCH0寄存器的CH1位置
	3	P4.4	不满足上述情况, 始终做IO口
33	1	PWM31	PWM3EN寄存器的PWM31EN位置1, 同时PWM31OE置1
	2	P4.5	不满足优先级1情况下, 始终为IO
32	1	T4	T4CLK[1:0]=01的情况下, 作TIMER4的时钟源输入口T4
	2	PWM32	PWM3EN寄存器的PWM32EN位置1, 同时PWM32OE置1
	3	P4.6	不满足优先级1情况下, 始终为IO
31	1	T2EX	在0, 2, 3模式下T2CON寄存器的EXEN2位置1 或在1模式下T2CON寄存器的DCEN位置1
	2	P4.7	T2CON寄存器的TR2位清0

PORT5:

- MOSI, MISO, SCK, SS (P5.0-P5.3) : SPI的输入输出接口以及时钟口和主从设备选择口
- T2, T3 (P5.4, P5.5) : 定时器2, 3输入或比较功能输出

Table 7.24 PORT5共用列表

引脚编号	优先级	功能	允许位
25	1	SS	SPI的SPSTA寄存器的SPEN置1, 当主模式SSDIS=1时或从模式CPHA=1且SSDIS=1时也能作I/O口用
	2	P5.0	不满足上述情况
28-26	1	MOSI, MISO, SCK	SPI的SPSTA寄存器的SPEN置1

	2	P5.1 - P5.3	SPI的SPSTA寄存器的SPEN清0
29	1	T2	T2CON寄存器的TR2位置1, C/T2位置1
	2	P5.4	不满足上述情况
30	1	T3	T3CLK[1:0]=01的情况下, 作TIMER3的时钟源输入口T3
	2	P5.5	不满足上述情况

7.7 定时器

7.7.1 特性

- G80F980有5个定时器（定时器0, 1, 2, 3, 4）
- 定时器0兼容标准的8051
- 定时器1兼容标准的8051
- 定时器2有递增递减计数和可编程输出功能，兼有硬件自动单次触发功能
- 定时器3是16位自动重载定时器，且可以工作在掉电模式，可作为捕捉模块时基
- 定时器4是16位自动重载定时器，可作为捕捉模块时基

7.7.2 定时器0和定时器1

每个定时器的两个数据寄存器（THx & TLx (x = 0, 1)）可作为一个16位寄存器来访问。它们由寄存器TCON和TMOD控制。IEN0寄存器的ET0和ET1位置1能允许定时器0和定时器1中断。（详见中断章节）

定时器x的方式 (x = 0, 1)

通过计数器/定时器方式寄存器（TMOD）的方式选择位Mx1-Mx0，选择定时器工作方式。

方式0：13位计数器/定时器

在方式0中，定时器x为13位计数器/定时器。THx寄存器存放13位计数器/定时器的高8位，TLx存放低5位（TLx.4-TLx.0）。TLx的高三位（TLx.7-TLx.5）是不确定的，在读取时应该被忽略。当13位定时器寄存器递增，溢出时，系统置起定时器溢出标志TFx。如果定时器x中断被允许，将会产生一个中断。C/Tx位选择计数器/定时器的时钟源。

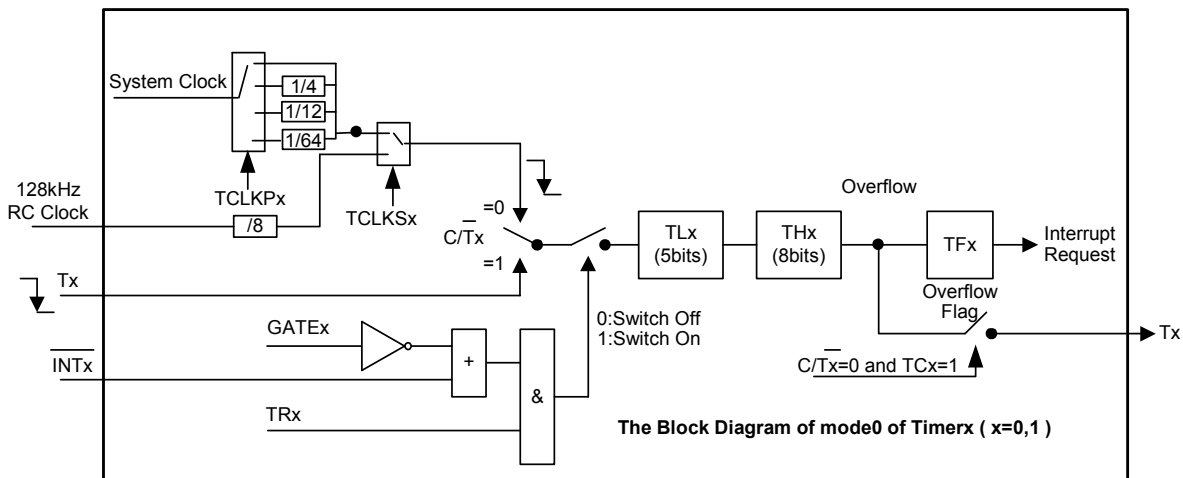
如果 $C/\overline{Tx} = 1$ ，定时器x输入引脚（Tx）的电平从高到低跳变，使定时器x数据寄存器加1。如果 $C/\overline{Tx} = 0$ ，选择系统时钟为定时器x的时钟源。

当 $GATEx = 0$ 或 $GATEx = 1$ 且输入信号 \overline{INTx} 有效时，TRx置1打开定时器。GATEx置1允许定时器由外部输入信号 \overline{INTx} 控制，便于测量 \overline{INTx} 的正脉冲宽度。TRx位置1不强行复位定时器，这意味着如果TRx置1，定时器寄存器将从上次TRx清0时的值开始计数。所以在允许定时器之前，应该设定定时器寄存器的初始值。

当作为定时器应用时，可配置寄存器TCON1中的TCLKSx (x = 0, 1) 位选择系统时钟或低频128kHz的8分频作为定时器x (x = 0, 1) 的时钟源。

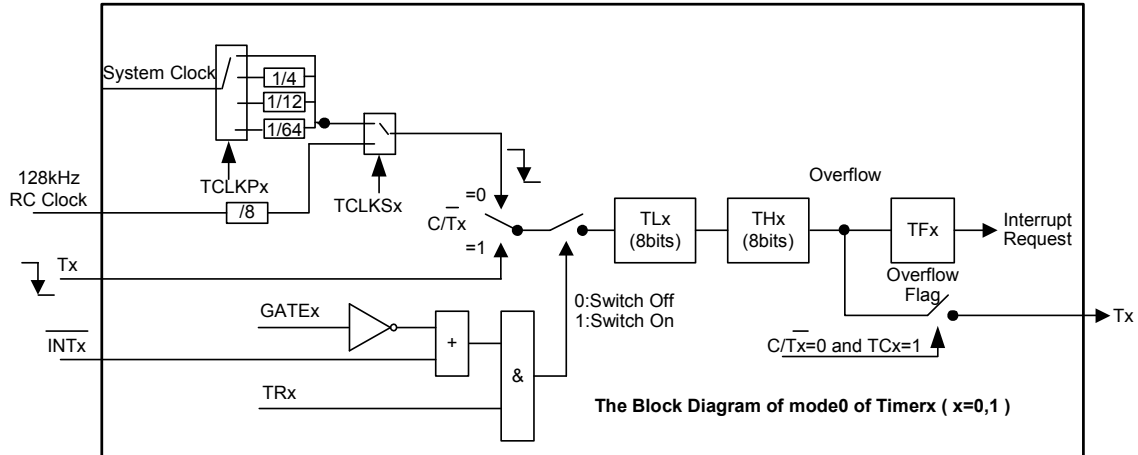
可配置寄存器TCON1中的TCLKPx (x = 0, 1) 位选择系统时钟或系统时钟的1/12作为定时器x (x = 0, 1) 的时钟源。

当作为定时器应用时，可配置寄存器TCON1中的TC0/1位使定时器0/1溢出时T0/T1脚自动翻转。如果TC0/1被置1，T0/T1引脚自动设置为输出。



方式1：16位计数器/定时器

除了使用16位定时器/计数器之外，方式1的运行与方式0一致。打开和配置计数器/定时器也如同方式0。

**方式2：8位自动重载计数器/定时器**

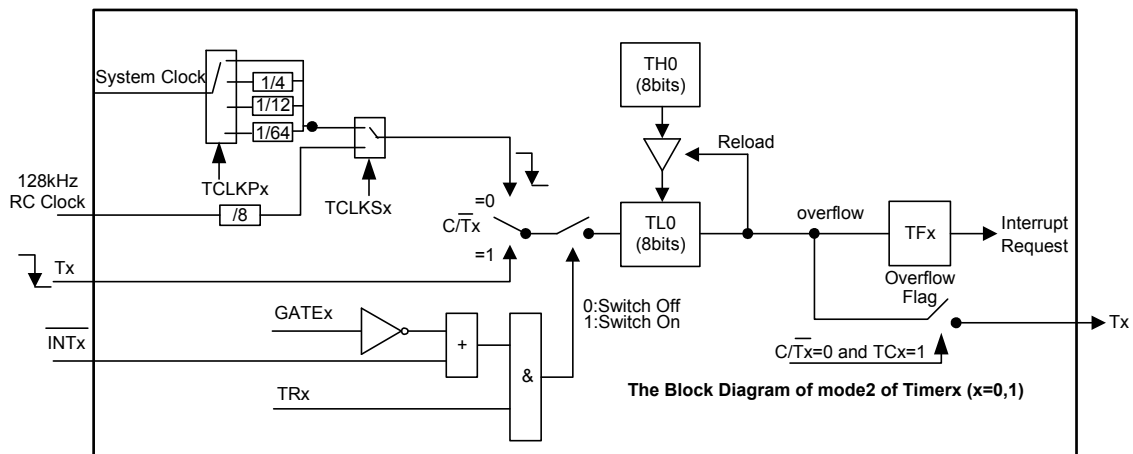
方式2中，定时器x是8位自动重载计数器/定时器。TLx存放计数值，THx存放重载值。当在TLx中的计数器溢出至0x00时，置起定时器溢出标志TFx，寄存器THx的值被重载入寄存器TLx中。如果定时器中断使能，当TFx置1时将产生一个中断。而在THx中的重载值不会改变。在允许定时器正确计数开始之前，TLx必须初始化为所需的值。

除了自动重载功能外，方式2中的计数器/定时器的使能和配置与方式1和0是一致的。

当作为定时器应用时，可配置寄存器TCON1中的TCLKSx (x=0, 1) 位选择系统时钟或128kHz的8分频作为定时器x (x=0, 1) 的时钟源。

可配置寄存器TCON1中的TCLKPx (x=0, 1) 位选择系统时钟或系统时钟的1/12作为定时器x (x=0, 1) 的时钟源。

当作为定时器应用时，可配置寄存器TCON1中的TC0/1位使定时器0/1溢出时T0/T1脚自动翻转。如果TC0/1被置1，T0/T1引脚自动设置为输出。



方式3：两个8位计数器/定时器（只限于定时器0）

在方式3中，定时器0用作两个独立的8位计数器/定时器，分别由TL0和TH0控制。TL0使用定时器0的控制（在TCON中）和状态（在TMOD中）位：TR0，C/T0，GATE0和TF0。TL0能用系统时钟或128kHz的8分频或外部输入信号作为时钟源。

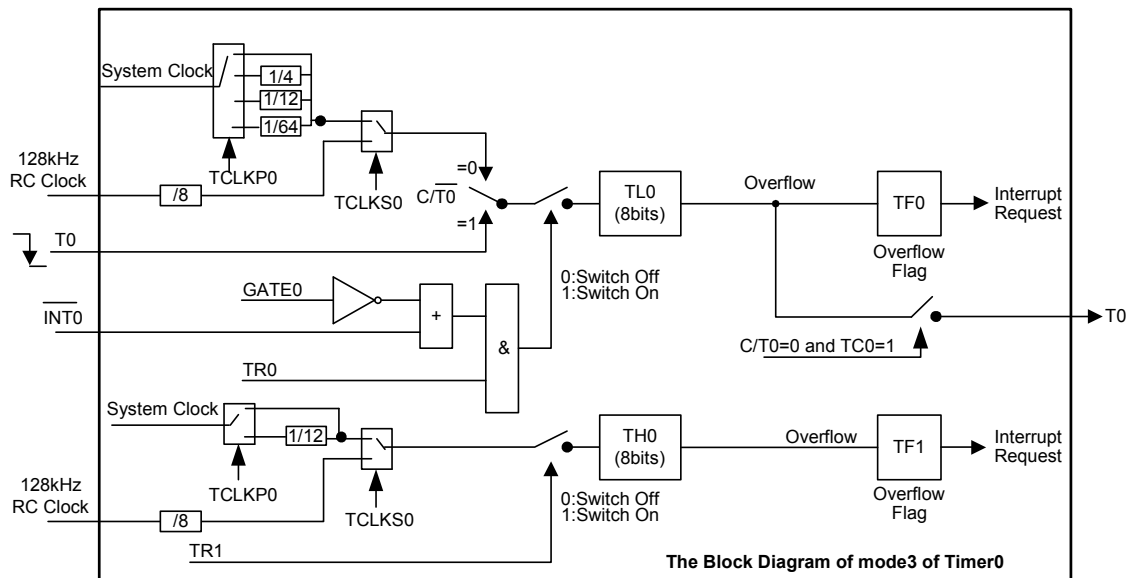
TH0只能用作定时器功能，时钟源来自系统时钟。TH0由定时器1的控制位TR1控制使能，溢出时定时器1溢出标志TF1置1，控制定时器1中断。

定时器0工作在方式3时，定时器1可以工作在方式0、1或2，但是不能置1 TF1标志和产生中断。TH1和TL1只能用作定时器功能，时钟源来自系统时钟，GATE1位无效。T1输入脚的上拉电阻也无效。定时器1由方式控制使能与否，因为TR1被定时器0占用。定时器1在方式0、1或2时使能，在方式3时被关闭。

当作为定时器应用时，可配置寄存器TCON1中的TCLKSx（x = 0, 1）位选择系统时钟或128kHz的8分频作为定时器x（x = 0, 1）的时钟源。

可配置寄存器TCON1中的TCLKPx（x = 0, 1）位选择系统时钟或系统时钟的1/12作为定时器x（x = 0, 1）的时钟源。

当作为定时器应用时，可配置寄存器TCON1中的TC0/1位使定时器0/1溢出时T0/T1脚自动翻转。如果TC0/1被置1，T0/T1引脚自动设置为输出。



注意：此时定时器1溢出率不作为波特率发生器，此功能仅作屏蔽，G80F980芯片的可变波特率仅由寄存器SBRTL/H设置。

寄存器

Table 7.25 定时器/计数器x控制寄存器 (x = 0, 1)

88H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7. 5	TFx x = 0, 1	定时器x溢出标志位 0: 定时器x无溢出, 可由软件清0 1: 定时器x溢出, 由硬件置1; 若由软件置1将会引起定时器中断
6. 4	TRx x = 0, 1	定时器x启动, 停止控制位 0: 停止定时器x 1: 启动定时器x
3. 1	IEx x = 0, 1	外部中断x请求标志位
2. 0	ITx x = 0, 1	外部中断x触发方式选择位

Table 7.26 定时器/计数器x方式寄存器 (x = 0, 1)

89H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TMOD	GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7. 3	GATEx x = 0, 1	定时器x门控位 0: TRx置1, 定时器x即被允许 1: 只有INTx在高电平期间TRx置1, 定时器x才被允许
6. 2	C/Tx x = 0, 1	定时器/计数器方式选择位 0: 定时器方式 1: 计数器方式
5-4 1-0	Mx[1:0] x = 0, 1	定时器x定时器方式选择位 00: 方式0, 13位向上计数计数器/定时器, 忽略TLx的第7-5位 01: 方式1, 16位向上计数计数器/定时器 10: 方式2, 8位自动重载向上计数计数器/定时器 11: 方式3 (只用于定时器0), 两个8位向上计数定时器

Table 7.27 定时器/计数器x数据寄存器1 (x = 0, 1)

8AH-8DH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TL0 (8AH)	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0 (8CH)	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1 (8BH)	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1 (8DH)	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	TLx.y, THx.y x = 0-1, y = 0-7	定时器x低及高字节计数器

Table 7.28 定时器/计数器x控制寄存器1 (x = 0, 1)

CEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TCON1	TCLK_S1	TCLK_S0	TCLKP11	TCLKP10	TCLKP01	TCLKP00	TC1	TC0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	1	1	1	1	0	0

位编号	位符号	说明
7-6	TCLK_Sx x=0, 1	定时器x时钟源控制位 0: 系统时钟作为定时器x的时钟源 1: 选择128kHz RC时钟的8分频作为定时器x的时钟源
5-2	TCLKPx[1:0] x=0, 1	定时器x时钟源预分频控制位 00: 选择系统时钟的1/64作为定时器x的时钟源 01: 选择系统时钟的1/12作为定时器x的时钟源 10: 选择系统时钟的1/4作为定时器x的时钟源 11: 选择系统时钟作为定时器x的时钟源
1-0	TCx x=0, 1	比较输出功能允许位 0: 禁止定时器x比较输出功能 1: 允许定时器x比较输出功能

7.7.3 定时器2

两个数据寄存器（TH2和TL2）串联后可作为一个16位寄存器来访问，由寄存器T2CON和T2MOD控制。设置IEN0寄存器中的ET2位能允许定时器2中断。（详见中断章节）

C/T2选择系统时钟（定时器）或外部引脚T2（计数器）作为定时器时钟输入。通过所选的引脚设置TR2允许定时器2/计数器2数据寄存器计数。

定时器2方式

定时器2有4种工作方式：捕获/重载、带递增或递减计数器的自动重载方式、单次触发和可编程时钟输出。通过设置寄存器T2CON和T2MOD选择这些方式。

定时器2方式选择

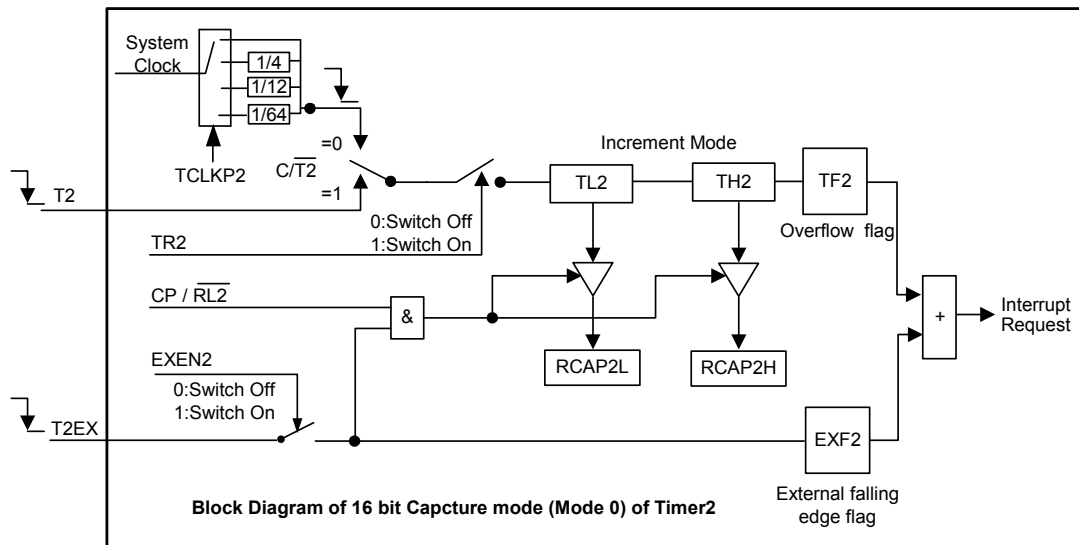
C/T2	T2OE DCEN	TR2	CP/RL2	TRG	方式		
X	0	X	1	1	0	0	16位捕获
X	0	0或1	1	0	0	1	16位自动重载定时器
X	0	0	1	X	1	2	单次触发模式
0	1	X	1	X	X	3	可编程时钟输出
X	X	X	0	X	0	X	定时器2停止，T2EX通路仍旧允许

方式0：16位捕获

在捕获方式中，T2CON的EXEN2位有两个选项。

如果EXEN2 = 0，定时器2作为16位定时器或计数器，如果ET2被允许的话，定时器2能设置TF2溢出产生一个中断。

如果EXEN2 = 1，定时器2执行相同操作，但是在外部输入T2EX上的下降沿也能引起在TH2和TL2中的当前值分别被捕获到RCAP2H和RCAP2L中，此外，在T2EX上的下降沿也能引起在T2CON中的EXF2被设置。如果ET2被允许，EXF2位也像TF2一样也产生一个中断。



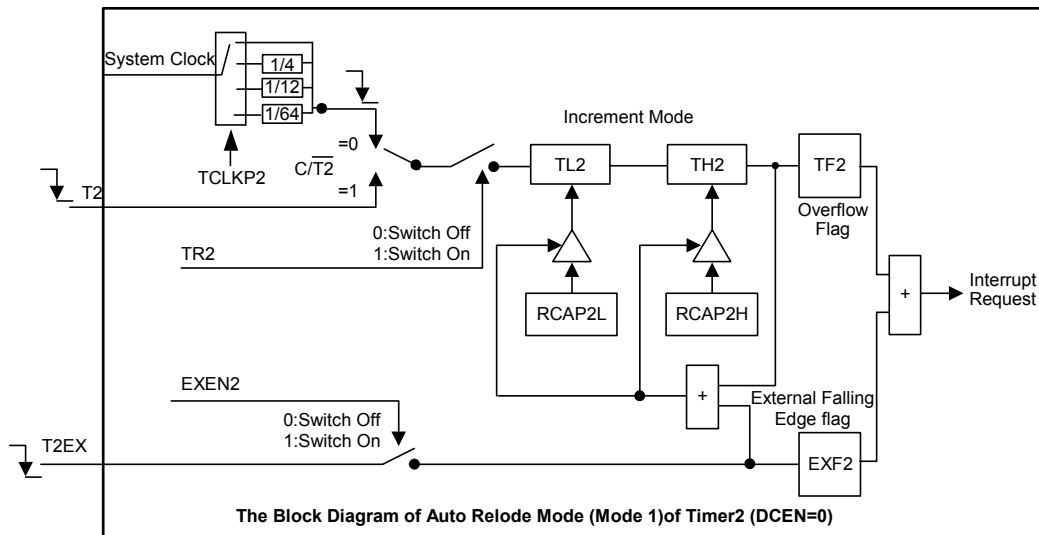
方式1：16位自动重载定时器

在16位自动重载方式下，定时器2可以被选为递增计数或递减计数。这个功能通过T2MOD中的DCEN位（递减计数允许）选择。系统复位后，DCEN位复位值为0，定时器2默认递增计数。当DCEN置1时，定时器2递增计数或递减计数取决于T2EX引脚上的电平。

当DCEN = 0，通过在T2CON中的EXEN2位选择两个选项。

如果EXEN2 = 0，定时器2递增到0FFFFH，在溢出后置起TF2位，同时定时器自动将用户软件写好的寄存器RCAP2H和RCAP2L的16位值装入TH2和TL2寄存器。

如果EXEN2 = 1，溢出或在外部输入T2EX上的下降沿都能触发一个16位重载，置起EXF2位。如果ET2被使能，TF2和EXF2位都能产生一个中断。

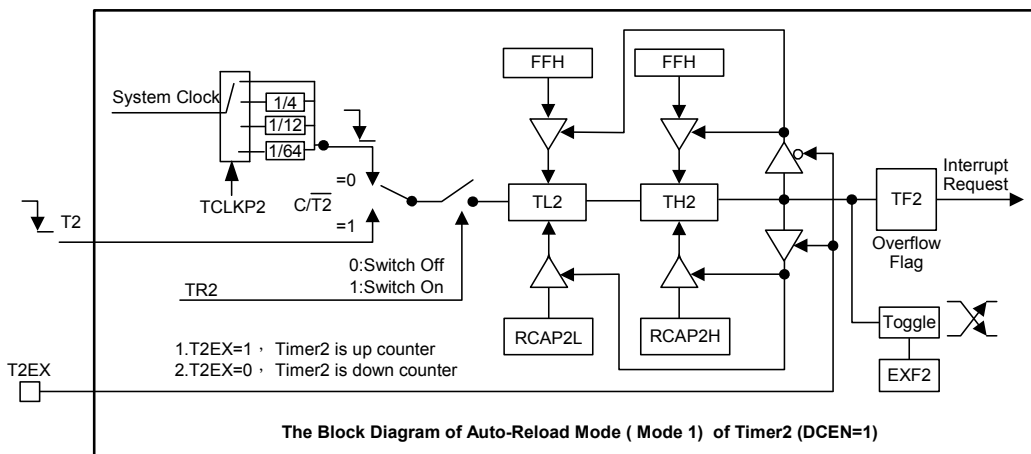


设置DCEN位允许定时器2递增计数或递减计数。当DCEN = 1时，T2EX引脚控制计数的方向，而EXEN2控制无效。

T2EX置1可使定时器2递增计数。定时器向0FFFFH溢出，然后设置TF2位。溢出也能分别引起RCAP2H和RCAP2L上的16位值重载入定时器寄存器。

T2EX清0可使定时器2递减计数。当TH2和TL2的值等于RCAP2H和RCAP2L的值时，定时器溢出。置起TF2位，同时0FFFFH重载入定时器寄存器。

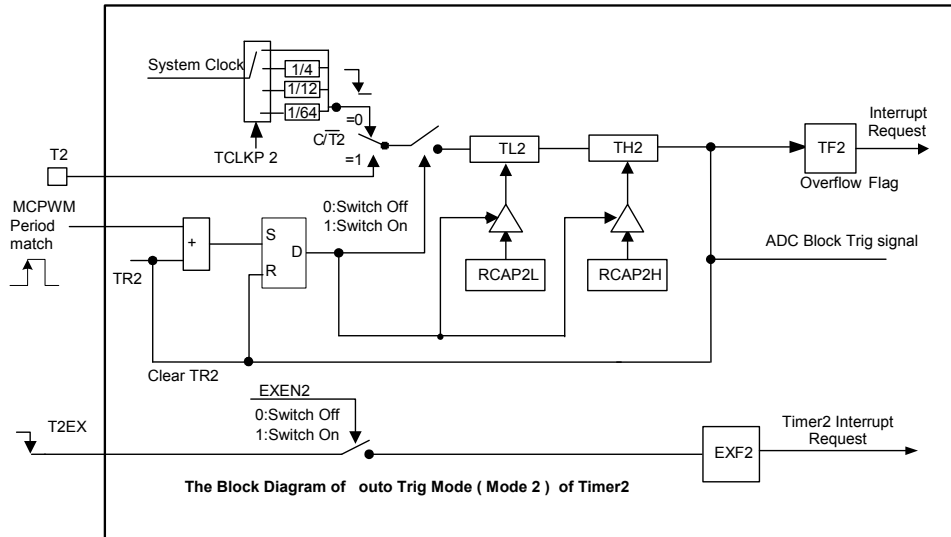
无论定时器2溢出与否，EXF2位都被用作结果的第17位。在此工作模式下，EXF2不作为中断标志。



方式2：单次触发模式

TRG = 1，进入此模式。在此模式下，TR2 = 1会引起Timer2自动重载并启动Timer2，将寄存器RCAP2H和RCAP2L的16位值装入TH2和TL2寄存器，然后开始计数，当计数溢出时，置位TF2，同时触发ADC模块并自动清0 TR2停止Timer2计数。

在此模式下，TR2可以由软件置1也可以硬件置1。硬件置1通过MCPWM模块实现，将PWMCON2寄存器的ZETIM置1，则MCPWM模块会送出硬件触发信号给Timer2（详见MCPWM模块），硬件置起TR2。

**注意：**

当在软件启动Timer2时，又有MCPWM硬件触发信号进来，定时器会停止先前的计数并重载数据，进行新的计数；相反，由MCPWM硬件信号触发启动Timer2时，软件再置位TR2会当成是无效操作，不会再重载定时器。

Timer2工作在方式2时，EXEN2 = 1时，T2EX的一个下降沿会置起EXF2，如果ET2被允许，EXF2和TF2的置起都将引发一个Timer2的中断。

方式3：可编程时钟输出

在这种方式中，T2输出占空比为50%的时钟：

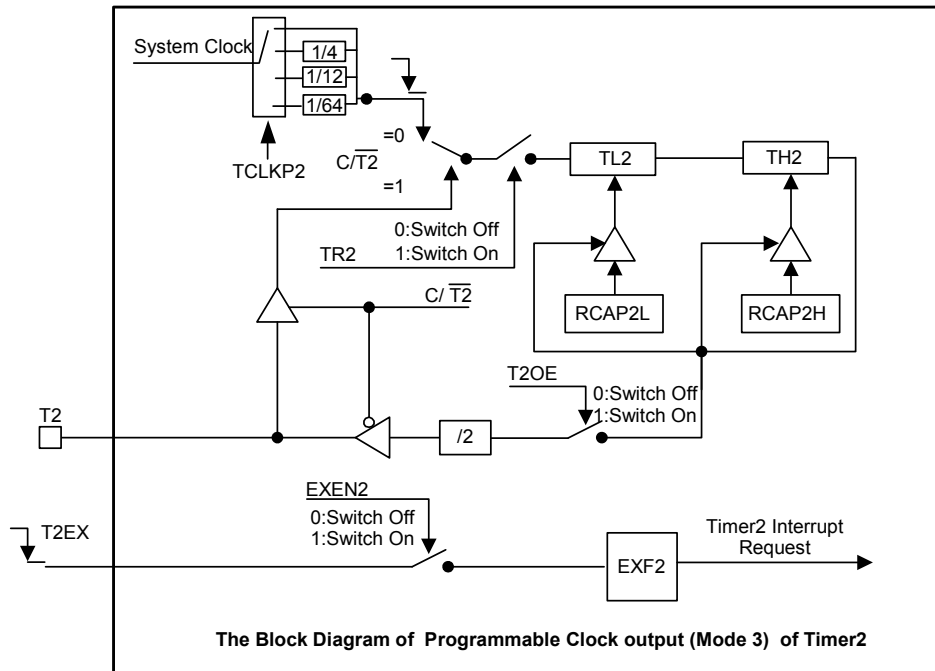
$$\text{Clock Out Frequency} = \frac{1}{2 \times 2 \times 64} \times \frac{f_{\text{SYS}}}{65536 - [\text{RCAP2H}, \text{RCAP2L}]} \quad ; \text{TCLKP2} = 00$$

$$\text{Clock Out Frequency} = \frac{1}{2 \times 2 \times 12} \times \frac{f_{\text{SYS}}}{65536 - [\text{RCAP2H}, \text{RCAP2L}]} \quad ; \text{TCLKP2} = 01$$

$$\text{Clock Out Frequency} = \frac{1}{2 \times 2 \times 4} \times \frac{f_{\text{SYS}}}{65536 - [\text{RCAP2H}, \text{RCAP2L}]} \quad ; \text{TCLKP2} = 10$$

$$\text{Clock Out Frequency} = \frac{1}{2 \times 2} \times \frac{f_{\text{SYS}}}{65536 - [\text{RCAP2H}, \text{RCAP2L}]} \quad ; \text{TCLKP2} = 11$$

定时器2溢出不产生中断，T2端口作时钟输出。

**注意：**

1. TF2和EXF2都能引起定时器2的中断请求，两者有相同的向量地址。
2. 当事件发生时或其它任何时间都能由软件设置TF2和EXF2为1，只有软件以及硬件复位才能使之清0。
3. 当EA = 1 且 ET2 = 1 时，设置 TF2 或 EXF2 为 1 能引起定时器 2 中断。

寄存器

Table 7.29 定时器2重载/捕获和数据寄存器

CAH-CDH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
RCAP2L	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	RCAP2L.x	定时器2重载/捕获数据低位高位, x = 0 - 7
	RCAP2H.x	
7-0	TL2.x	定时器2低位/高位计数器, x = 0 - 7
	TH2.x	

Table 7.30 定时器2控制寄存器

C8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
T2CON	TF2	EXF2	TRG	-	EXEN2	TR2	C/T2	CP/RL2
读/写	读/写	读/写	读/写	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

位编号	位符号	说明
7	TF2	定时器2溢出标志位 0: 无溢出 (必须由软件清0) 1: 溢出, 由硬件设1
6	EXF2	T2EX引脚外部事件输入 (下降沿) 被检测到的标志位 0: 无外部事件输入 (必须由软件清0) 1: 检测到外部输入 (如果EXEN2 = 1, 由硬件设1)
5	TRG	单次触发控制位 0: 关闭单次触发功能 1: 开启单次触发功能
3	EXEN2	T2EX引脚上的外部事件输入 (下降沿) 用作重载/捕获触发器允许/禁止控制位 0: 忽略T2EX引脚上的事件 1: 检测到T2EX引脚上一个下降沿, 产生一个捕获或重载
2	TR2	定时器2开始/停止控制位 0: 停止定时器2 1: 开始定时器2
1	C/T2	定时器2定时器/计数器方式选定位 0: 定时器方式, T2引脚用作I/O端口 1: 计数器方式, 内部上拉电阻被打开
0	CP/RL2	捕获/重载方式选定位 0: 16位带重载功能的定时器/计数器 1: 16位带捕获功能的定时器/计数器

Table 7.31 定时器2方式控制寄存器

C9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
T2MOD	TCLKP21	TCLKP20	-	-	-	-	T2OE	DCEN
读/写	读/写	读/写	-	-	-	-	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	1	1	-	-	-	-	0	0

位编号	位符号	说明
7-6	TCLKP2[1: 0]	分频选择控制位 00: 选择系统时钟的1/64作为定时器x的时钟源 01: 选择系统时钟的1/12作为定时器x的时钟源 10: 选择系统时钟的1/4作为定时器x的时钟源 11: 选择系统时钟作为定时器x的时钟源
1	T2OE	定时器2输出允许位 0: 设置P5.4/T2作为时钟输入或I/O端口 1: 设置P5.4/T2作为时钟输出
0	DCEN	递减计数允许位 0: 禁止定时器2作为递增/递减计数器, 定时器2仅作为递增计数器 1: 允许定时器2作为递增/递减计数器

7.7.4 定时器3

定时器3是16位自动重载定时器，通过两个数据寄存器TH3和TL3访问，由T3CON寄存器控制。IEN1寄存器的ET3位置1允许定时器3中断（详见中断章节）。

定时器3只有一个工作方式：16位自动重载计数器/定时器，可以设置预分频比，并可以工作在CPU掉电模式。

定时器3有一个16位计数器/定时器寄存器（TH3，TL3）。当TH3和TL3被写时，用作定时器重载寄存器，当被读时，被用作计数寄存器。TR3位置1使定时器3开始递增计数。定时器在0xFFFF到0x0000溢出并置TF3位为1。溢出同时，定时器重载寄存器的16位数据被重新载入计数寄存器中，TH3写操作也导致重载寄存器的数据重新载入计数寄存器。

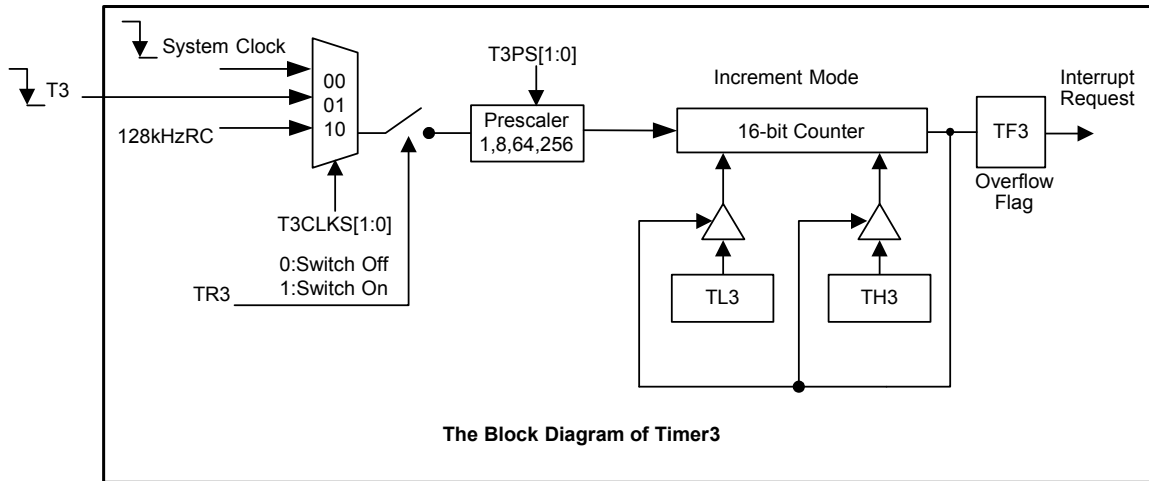
由于对TL3和TH3的读写操作需要2条指令才能完成。为保证精确计数，对TL3和TH3寄存器组的读写操作都以对TH3寄存器的读写操作作为基准。也就是说，写TL3寄存器不会立即生效，而是暂存在一个缓冲寄存器中，只有对TH3寄存器的写操作才会使TL3和TH3寄存器同时生效。

因此，TH3和TL3读写操作遵循以下顺序：

写操作：先低位后高位

读操作：先高位后低位

注意：虽然TH3和TL3寄存器地址相连，而且TL3地址在前，但由于上述限制，因此在C语言编程时，不能将此2个寄存器定义为SFR16。



定时器3可以工作在掉电模式。

但如果T3CLKS[1:0]为00，定时器3不能工作在掉电模式下。如果T3CLKS[1:0]为01，T3端口输入外部时钟，定时器3可以工作在普通模式或掉电模式。当T3CLKS[1:0]为10时，定时器3可以工作在普通模式或掉电模式。

注意：

1. 在读或写TH3和TL3时，要确保TR3 = 0。

寄存器

Table 7.32 定时器3控制寄存器

B3H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
T3CON	TF3	-	T3PS.1	T3PS.0	-	TR3	T3CLKS.1	T3CLKS.0
读/写	读/写	-	读/写	读/写	-	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	-	0	0	-	0	0	0

位编号	位符号	说明
7	TF3	定时器3溢出标志位 0: 无溢出 (硬件清0) 1: 溢出 (硬件置1)
5-4	T3PS[1:0]	定时器3预分频比选择位 00: 1/1 01: 1/8 10: 1/64 11: 1/256
2	TR3	定时器3允许控制位 0: 关闭定时器3 1: 打开定时器3
1-0	T3CLKS[1:0]	定时器3时钟源选择位 00: 系统时钟的1/4作为时钟源, T3端口作为I/O口 01: T3端口输入外部时钟, 自动上拉 10: 内部128KHz RC振荡器 11: 保留

Table 7.33 定时器3重载/计数数据寄存器

ACH - ADH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TL3 (ACH)	TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
TH3 (ADH)	TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	TL3.x	定时器3低位高位计数器, x = 0 - 7
	TH3.x	

7.7.4 定时器4

定时器4是16位自动重载定时器。两个数据寄存器TH4和TL4可作为一个16位寄存器来访问。由T4CON寄存器控制。IEN1寄存器的ET4位置1允许定时器4中断（详见中断章节）。

当TH4和TL4被写时，用作定时器重载寄存器，当被读时，被用作计数寄存器。TR4位置1使定时器4开始递增计数。定时器在0xFFFF到0x0000溢出并置TF4位为1。溢出同时，定时器重载寄存器的16位数据重新载入计数寄存器中，对TH4的写操作也导致重载寄存器的数据重新载入计数寄存器。

TH4和TL4读写操作遵循以下顺序：

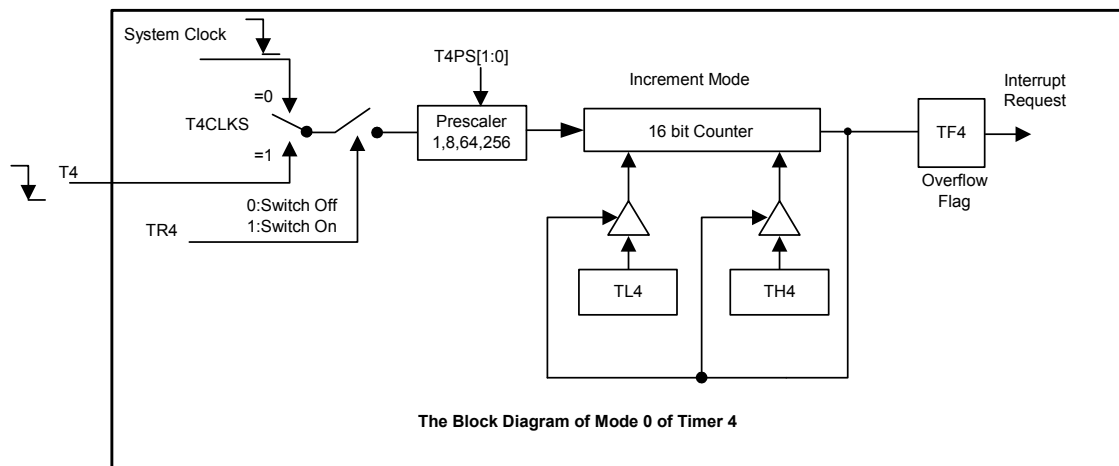
写操作：先低位后高位

读操作：先高位后低位

T4CON.0寄存器的T4CLKS位选择时钟源。当T4CLKS = 1时，定时器4的时钟源为外部时钟，预分频后，计数器数据寄存器增加。当T4CLKS = 0，定时器4的时钟源为系统时钟。

T4CON.1寄存器的TR4位置1允许定时器4，且不清定时器4的计数器。在允许定时器4之前，将希望的初始值写入定时器重载寄存器中。

在比较方式中，T4端口自动被硬件设为输出。定时器4从TH4和TL4预设值开始向0xFFFF计数，当计数器溢出时，T4端口输出反转，同时定时器4中断标志位被置起。



寄存器

Table 7.32 定时器4控制寄存器

ABH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
T4CON	TF4	TC4	T4PS.1	T4PS.0	-	-	TR4	T4CLKS
读/写	读/写	读/写	读/写	读/写	-	-	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	-	-	0	0

位编号	位符号	说明
7	TF4	定时器4溢出标志位 0: 无溢出（硬件清0） 1: 溢出（硬件置1）
6	TC4	比较功能允许位 0: 禁止定时器4比较功能 1: 允许定时器4比较功能

5-4	T4PS[1:0]	定时器4预分频比选择位 00: 1/1 01: 1/8 10: 1/64 11: 1/256
1	TR4	定时器4允许控制位 0: 关闭定时器4 1: 打开定时器4
0	T4CLKS	定时器4时钟源选择位 0: 系统时钟, T4端口作为I/O口 1: T4端口输入外部时钟, 自动上拉

Table 7.33 定时器4重载/计数数据寄存器

AEH - AFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TL4 (AEH)	TL4.7	TL4.6	TL4.5	TL4.4	TL4.3	TL4.2	TL4.1	TL4.0
TH4 (AFH)	TH4.7	TH4.6	TH4.5	TH4.4	TH4.3	TH4.2	TH4.1	TH4.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	TL4.x	定时器4低位高位计数器, x = 0 - 7
	TH4.x	

7.8 8bit脉冲宽度调制模块 (PWM30/ PWM31/ PWM32)

7.8.1 特性

- 三路独立8位分辨率PWM模块，提供预分频功能
- 提供每个PWM周期溢出中断
- 输出极性可选择，带输出独立允许位，允许位为0时可做通用定时器使用

G80F980内建3个8位PWM模块。PWM模块可以产生周期和占空比分别可以调整的脉宽调制波形。寄存器PWM3CON用于控制PWM模块的时钟源，寄存器PWMP用于设置PWM模块的周期，寄存器PWMD用于设置PWM模块的占空比。

7.8.2 寄存器

Table 7.34 PWM3使能寄存器

FFA0H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM3EN	-	-	-	-	-	PWM32EN	PWM31EN	PWM30EN
读/写	-	-	-	-	-	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	0	0	0

2	PWM32EN	PWM3x模块控制位 0: 关闭PWM32模块 1: 打开PWM32模块
1	PWM31EN	PWM3x模块控制位 0: 关闭PWM31模块 1: 打开PWM31模块
0	PWM30EN	PWM3x模块控制位 0: 关闭PWM30模块 1: 打开PWM30模块

Table 7.34 PWM3x控制寄存器

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM30CON(FFA1H)	PWM30S	PWM30CK2	PWM30CK1	PWM30CK0	FLT30EN	PWM30IE	PWM30IF	PWM30OE
PWM31CON(FFA9H)	PWM31S	PWM31CK2	PWM31CK1	PWM31CK0	FLT31EN	PWM31IE	PWM31IF	PWM31OE
PWM32CON(FFB1H)	PWM32S	PWM32CK2	PWM32CK1	PWM32CK0	FLT32EN	PWM32IE	PWM32IF	PWM32OE
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	PWM3xS (x = 0 - 2)	PWM3输出模式 0: 高电平有效, PWM3x占空比有效期间输出高电平, 其余时间输出低电平 1: 低电平有效, PWM3x占空比有效期间输出低电平, 其余时间输出高电平
6-4	PWM3xCK2:0] (x = 0 - 2)	PWM3时钟选择位 000: 系统时钟/1 001: 系统时钟/8 010: 系统时钟/64 011: 系统时钟/256 100: 系统时钟/2048 其他值无效

3	FLT3xEN (x = 0 - 2)	PWM30故障检测使能位 0: 故障检测功能禁止 1: 故障检测功能允许, 比较器1的输出信号作为PWM3x的故障检测输入信号, 如果检测到比较器1的输出变高, 将立即关闭PWM输出, 当比较器输出变低后, 不会立即恢复输出, 需要与PWM3x周期溢出信号同步, 然后PWM3x才会恢复输出
2	PWM3xIE (x = 0 - 2)	PWM3x中断允许位 (只有当IEN1中的EPWM3位也为1时, 功能才有效) 0: 禁止PWM3周期溢出中断 1: 允许PWM3周期溢出中断
1	PWM3xIF (x = 0 - 2)	PWM3x中断标志位 0: PWM3周期计数器没有溢出 1: PWM3周期计数器溢出, 由硬件置1
0	PWM3xOE (x = 0 - 2)	PWM3x输出控制位 0: PWM3输出禁止, 用作I/O功能 <i>注: 如果此位为0而PWM3xEN = 1, 则整个PWM3x模块仍然正常运行, 只是波形输出被禁止, PWM3x模块可以做一个定时器来使用。</i> 1: PWM3x输出允许 <i>注: 如果此位设为1而PWM3xEN位为0, 则PWM3x输出非选电平 (选为高电平有效则出低电平, 低电平有效则出高电平)</i>

Table 7.35 PWM3周期控制寄存器

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM30P (FFA2H)	PWM30P.7	PWM30P.6	PWM30P.5	PWM30P.4	PWM30P.3	PWM30P.2	PWM30P.1	PWM30P.0
PWM31P (FFAAH)	PWM31P.7	PWM31P.6	PWM31P.5	PWM31P.4	PWM31P.3	PWM31P.2	PWM31P.1	PWM31P.0
PWM32P (FFB2H)	PWM32P.7	PWM32P.6	PWM32P.5	PWM32P.4	PWM32P.3	PWM32P.2	PWM32P.1	PWM32P.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	PWM3xP.7-0 (x = 0 - 2)	PWM3x输出周期 = PWM3xP * PWM时钟 当PWM3xP = 00H时, 如果PWM3xS = 0, PWM3x引脚输出低 当PWM3xP = 00H时, 如果 PWM3xS = 1, PWM3x引脚输出高

Table 7.36 PWM3占空比控制寄存器

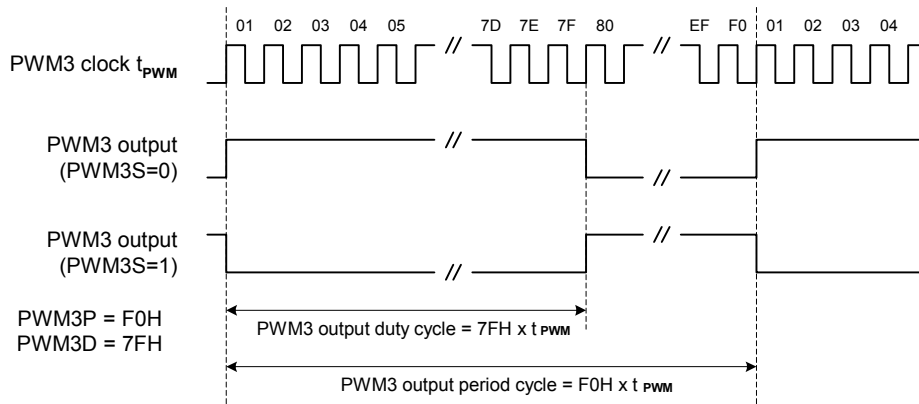
AFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM30D (FFA3H)	PWM30D.7	PWM30D.6	PWM30D.5	PWM30D.4	PWM30D.3	PWM30D.2	PWM30D.1	PWM30D.0
PWM31D (FFABH)	PWM31D.7	PWM31D.6	PWM31D.5	PWM31D.4	PWM31D.3	PWM31D.2	PWM31D.1	PWM31D.0
PWM32D (FFB3H)	PWM32D.7	PWM32D.6	PWM32D.5	PWM32D.4	PWM32D.3	PWM32D.2	PWM32D.1	PWM32D.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
-----	-----	----

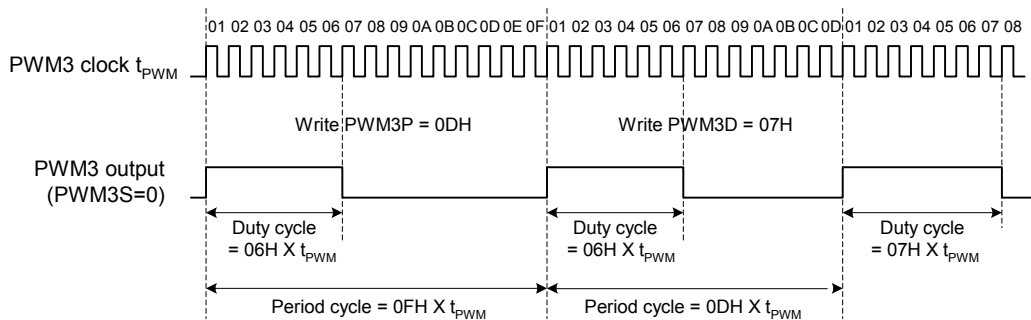
7-0	PWM3xD.7-0 (x = 0 - 2)	<p>PWM3x占空比控制，控制PWM波形占空比的输出时间</p> <p>特殊情况：</p> <ol style="list-style-type: none"> 当 $PWM3xP \leq PWM3xD$ 时 如果 $PWM3xS = 0$，则PWM3x引脚输出高电平 如果 $PWM3xS = 1$，则PWM3x引脚输出低电平 当 $PWM3xD = 00H$ 时 如果 $PWM3xS = 0$，则PWM3x引脚输出低电平 如果 $PWM3xS = 1$，则PWM3x引脚输出高电平
-----	-----------------------------------	---

注意事项：

1. PWM3xEN 位控制PWM3x 模块打开。
2. PWM3xOE 位能选择Pxx.0 端口是作为I/O 端口还是PWM3 输出端口。
3. 在IEN1 寄存器中的EPWM3 位能允许禁止PWM3 中断,三路PWM 共用PWM3 中断源。
4. 如果 PWM3xEN 置 1, PWMx3 模块打开, 但 PWM3xOE = 0, PWMx3 输出关闭, 此时 PWMx3 模块可以用作一个 8bit timer, 此时如果中断控制寄存器 IEN1 的 EPWM3 位置 1, PWM3 中断照样发生。



PWM3x(x = 0 - 2)输出范例



PWM3x(x = 0 - 2)输出周期或者占空比改变范例

7.9 中断

7.9.1 特性

- 15个中断源
- 4层中断优先级

G80F980有18个中断源：3个外部中断（INT0，INT1和INT4，INT4共7个中断源INT40-46共享一个向量地址），5个定时器中断（定时器0-4），1个EUART中断，系统时钟监控中断，1个SPI中断，ADC中断和PWM3中断，电机控制PWM模块中断，2个模拟比较器中断，输入捕捉中断和CRC中断。G80F980中断采用4级优先级结构，为处理18个中断源提供了极大的灵活性。

另外，G80F980为INT4提供了4种外部中断触发方式，可以通过寄存器选择。

7.9.2 中断允许控制

任何一个中断源均可通过对寄存器IEN0和IEN1中相应的位置位或清零，实现单独允许或禁止此中断功能。IEN0寄存器中还包含了一个全局使能位EA，它可以禁止所有的中断。如果要打开某个中断源，必须将EA控制位和IEN0/1中相应的控制位都设置为1。在复位后，所有中断允许位被清0，所有中断被禁止。

Table 7.37 初级中断允许寄存器

A8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IEN0	EA	EADC	ET2	ES	ET1	EX1	ET0	EX0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	EA	所有中断允许位 0: 禁止所有中断 1: 允许所有中断
6	EADC	ADC中断允许位 0: 禁止ADC中断 1: 允许ADC中断
5	ET2	定时器2溢出中断允许位 0: 禁止定时器2溢出中断 1: 允许定时器2溢出中断
4	ES	EUART中断允许位 0: 禁止EUART中断 1: 允许EUART中断
3	ET1	定时器1溢出中断允许位 0: 禁止定时器1溢出中断 1: 允许定时器1溢出中断
2	EX1	外部中断1允许位 0: 禁止外部中断1 1: 允许外部中断1
1	ET0	定时器0溢出中断允许位 0: 禁止定时器0溢出中断 1: 允许定时器0溢出中断
0	EX0	外部中断0允许位 0: 禁止外部中断0 1: 允许外部中断0

Table 7.38 中断允许寄存器1

A9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IEN1	ESCM	EPWM3	EMCM	ET3	EX4	ET4	ECMP	ESPI
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	ESCM	SCM中断允许位 0: 禁止SCM中断 1: 允许SCM中断
6	EPWM3	PWM3中断允许位 0: 禁止PWM3中断 1: 允许PWM3中断
5	EMCM	电机控制模块MCM (Motor Control Module) 中断允许位 0: 禁止电机控制模块所有中断 1: 允许电机控制模块中断 (和PWM0/1/2中断独立允许控制位配合)
4	ET3	定时器3溢出中断允许位 0: 禁止定时器3溢出中断 1: 允许定时器3溢出中断
3	EX4	外部中断4允许位 0: 禁止外部中断4 1: 允许外部中断4
2	ET4	定时器4溢出中断允许位 0: 禁止定时器4溢出中断 1: 允许定时器4溢出中断
1	ECMP1	模拟比较器1中断允许位 0: 禁止模拟比较器1中断 1: 允许模拟比较器1中断
0	ESPI	SPI中断允许位 0: 禁止SPI中断 1: 允许SPI中断

Table 7.38 中断允许寄存器1

AAH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IEN2	-	-	-	-	-	ECRC	ECMP2	ECAP
读/写	-	-	-	-	-	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	0	0	0

位编号	位符号	说明
2	ECRC	CRC中断允许位 0: 禁止CRC中断 1: 允许CRC中断
1	ECMP2	模拟比较器2中断允许位 0: 禁止模拟比较器2中断 1: 允许模拟比较器2中断
0	ECAP	输入捕捉中断允许位 0: 禁止输入捕捉中断 1: 允许输入捕捉中断

Table 7.39 中断通道允许寄存器

BAH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IENC	-	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	0	0	0	0	0	0	0

位编号	位符号	说明
6-0	EXS4x x = 0 - 6	外部中断4通道选择位 (x = 0 - 6) 0: 禁止外部中断4x 1: 允许外部中断4x

注意:

1. 打开外部中断0/1/4时，相应的端口必须设置为输入状态。
3. 若要打开外部中断4，EX4和IENC中的EXS4x (x = 0 - 6) 相应位必须同时置1。

7.9.3 中断标志

每个中断源都有自己的中断标志，当产生中断时，硬件会置起相应的标志位，在中断汇总表中会列出中断标志位。

外部中断INT0/1产生外部中断INT x ($x = 0, 1$) 时，如果中断为边沿触发，CPU在响应中断后，标志IE x ($x = 0, 1$) 被硬件清零；如果中断是电平触发，外部中断源直接控制中断标志，如果是高电平，此标志位保持为1，否则清0。

外部中断INT4产生中断时，EXF1寄存器中的IF4 x ($x = 0 - 6$) 标志位置位，INT4共7个中断源共享一个中断向量地址，标志位需要用户软件清除。但是如果INT4为电平触发时，标志位不能被用户软件清除，只受INT4中断源引脚所接信号电平直接控制。

注意尽管外部中断被禁止，但是中断标志位仍然受外部中断源输入状态变化而改变，除非该输入引脚配置为其它功能。

定时器0/1中断，当定时器中断产生，中断标志位产生，当进入中断服务程序后中断标志位由硬件清除。

定时器2中断，T2CON寄存器的TF2或EXF2标志位置1时，产生定时器2中断，CPU在响应中断后，标志不能被硬件自动清0。事实上，中断服务程序必须决定是由TF2或是EXF2产生中断，因此标志必须由软件清0。

定时器3中断，定时器3的计数器溢出时，T3CON寄存器的TF3中断标志位置1，产生**定时器3**中断请求，CPU在响应中断后，TF3标志被硬件自动清0。

定时器4中断，定时器4的计数器溢出时，T4CON寄存器的TF4中断标志位置1，产生**定时器4**中断请求，CPU在响应中断后，TF4标志被硬件自动清0。

串行通讯中断，SCON寄存器的标志RI或TI置1时，产生**EUART**中断，CPU在响应中断后，标志不会被硬件自动清0。事实上，中断服务程序必须判断是收中断还是发中断，标志必须由软件清零。

系统时钟监控中断，SCM寄存器的SCMIF标志位置1时，产生**SCM**中断，标志由硬件清0。

SPI通讯中断，SPSTA寄存器的SPIF标志位或MODF标志位置1时，产生**SPI**中断，标志必须由软件清0。

ADC中断，ADCON寄存器的ADCIF标志位置1时，产生**ADC**中断。在每次转换中，如果转换结果小于比较值时，ADCIF标志位为0；如果转换结果大于等于比较值时，上限比较标志位ADGIF标志位置1，同理，还有下限比较标志位。因此ADC模块有3个中断标志位，分别有各自独立的中断允许位，中断标志必须由软件清0。

电机控制模块MCM中断，当PWMxIF0-2置起，中断产生，中断标志位由软件清零（PWM中断标志请参考PWM电机模块部分）。

PWM3中断，PWM3 x 是三路独立的8bit PWM模块，三路PWM3 x 共用一个中断矢量入口地址，此标志位在中断响应时由软件清0。

模拟比较器1中断，模拟比较器1中断标志位C1IF置起时，产生CMP1中断，标志必须由软件清0。

捕捉中断，当CAPIF0-2置起或CAPTIF置起时，中断产生，中断标志位由软件清零（CAP中断标志请参考输入捕捉部分）。

模拟比较器2中断，模拟比较器2中断标志位C2IF置起时，产生CMP2中断，标志必须由软件清0。

CRC校验中断，CRCCON寄存器的CRCIF标志位置1时，产生CRC中断，标志必须由软件清0。

Table 7.40 外部中断标志寄存器

88H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
3, 1	IE x ($x = 0, 1$)	外部中断 x 请求标志 0: 无中断挂起 1: 中断挂起
2, 0	IT x ($x = 0, 1$)	外部中断 x 触发方式 0: 低电平触发 1: 下降沿触发

Table 7.41 外部中断4设置寄存器

87H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PCON	SMOD	SSTAT	IT41	IT40	GF1	GF0	PD	IDL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	SMOD	UART波特率加倍器
6	SSTAT	UART SCON[7:5]功能选择
5-4	IT4 [1:0]	外部中断4触发模式位 00: 低电平触发 01: 下降沿触发 10: 上升沿触发 11: 双沿触发 IT4[1:0]控制外部中断4各中断源采用同一触发方式
3-2	GF[1:0]	用于软件的通用标志位
1	PD	掉电模式控制位
0	IDL	空闲模式控制位

Table 7.42 外部中断4标志寄存器

D8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
EXF1	-	IF46	IF45	IF44	IF43	IF42	IF41	IF40
读/写	-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	0	0	0	0	0	0	0

位编号	位符号	说明
6-0	IF4x (x = 0 - 6)	外部中断4请求标志, IF4x须被软件清0 0: 无中断请求 1: 有中断请求

7.9.4 中断向量

当一个中断产生时，程序计数器内容被压栈，相应的中断向量地址被载入程序计数器。中断向量的地址在中断摘要表中详细列出。

7.9.5 中断优先级

每个中断源都可被单独设置为4个中断优先级之一，分别通过清0或置1 IPL0, IPH0, IPL1, IPH1中相应位来实现。但OVL不可屏蔽中断无需IPH/IPL控制，在所有中断源中享有最高优先级（除复位外）。中断优先级服务程序描述如下：

响应一个中断服务程序时，可响应更高优先级的中断，但不能响应同优先级或低优先级的另一个中断。

响应最高级中断服务程序时，不响应其它任何中断。如果不同中断优先级的中断源同时申请中断时，响应较高优先级的中断申请。

如果同优先级的中断源在指令周期开始时同时申请中断，那么内部查询序列确定中断请求响应顺序。

捕捉中断及比较器2中断优先级共用同一优先级设置，二者同时申请中断时按内部轮询优先级响应。

中断优先级		
优先位		中断优先级
IPHx	IPLx	
0	0	等级0（最低优先级）
0	1	等级1
1	0	等级2
1	1	等级3（最高优先级）

Table 7.43 中断优先级控制寄存器

B8H, B4H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IPL0	PINTL	PADCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
IPH0	PINTH	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

B9H, B5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
IPL1	PSCML	PPWML	PMCML	PT3L	PX4L	PT4L	PCMP1L	PSPIL
IPH1	PSCMH	PPWMH	PMCMH	PT3H	PX4H	PT4H	PCMP1H	PSPIH
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
IPL0[7] IPH0[7]	PINTL/H	捕捉中断及比较器2中断优先级选择 注：捕捉中断及比较器2中断优先级共用优先级设置
IPL0[6:0] IPH0[6:0] IPL1[7:0] IPH1[7:0]	PxxxL/H	相应中断源xxx优先级选择

7.9.6 中断处理

中断标志在每个机器周期都会被采样获取。所有中断都在时钟的上升沿被采样。如果一个标志被置起，那么CPU捕获到后中断系统调用一个长转移指令（LCALL）调用其中断服务程序，但由硬件产生的LCALL会被下列任何条件阻止：

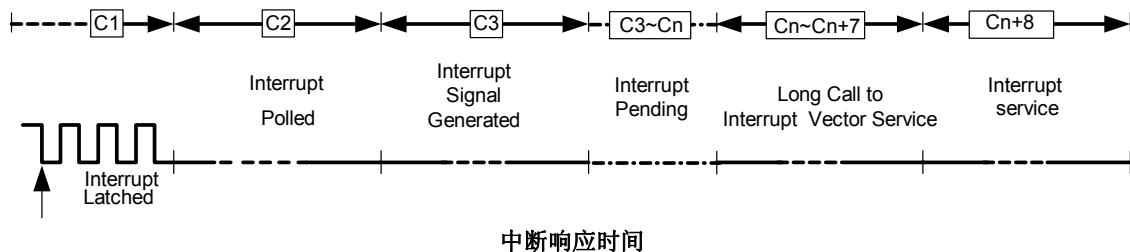
同级或更高级的优先级中断在运行中。

当前的周期不是执行中指令的最后一个周期。换言之，正在执行的指令完成前，任何中断请求都得不到响应。

正在执行的是一条RETI或者访问专用寄存器IEN0\1或是IPL\H的指令。换言之，在RETI或者读写IEN0\1或是IPL\H之后，不会马上响应中断请求，而至少在执行一条其它指令之后才会响应。

注意：因为更改优先级通常需要2条指令，在此期间，建议关闭相应的中断以避免在修改优先级过程中产生中断。如果当模块状态改变而中断标志不再有效时，将不会响应此中断。每一个轮询周期只查询有效的中断请求。

轮询周期/LCALL次序如下图所示：



由硬件产生的LCALL把程序计数器中的内容压入堆栈（但不保存PSW），然后将相应中断源的向量地址（参照中断向量表）存入程序计数器。

中断服务程序从指定地址开始，到RETI指令结束。RETI指令通知处理器中断服务程序结束，然后把堆栈顶部两字节弹出，重载入程序计数器中，执行完中断服务程序后程序回到原来停止的地方。RET指令也可以返回到原来地址继续执行，但是中断优先级控制系统仍然认为一个同一优先级的中断被响应，这种情况下，当同一优先级或低优先级中断将不会被响应。

7.9.7 中断响应时间

如果检测出一个中断，这个中断的请求标志位就会在被检测后的每个机器周期被置起。内部电路会保持这个值直到下一个机器周期，CPU会在第三个机器周期产生中断。如果响应有效，条件允许，在下一个指令执行的时候硬件LCALL指令将调用请求中断的服务程序，否则中断被挂起。LCALL指令调用程序需要7个机器周期。因而，从外部中断请求到开始执行中断程序中的第一条指令至少需要3+7个完整的机器周期。

当请求因前述的三个情况受阻时，中断响应时间会加长。如果同级或更高优先级的中断正在执行，额外的等待时间取决于正执行的中断服务程序的长度。

如果正在执行的指令还没有进行到最后一个周期，假如正在执行RETI指令，则完成正在执行的RETI指令，需要8个周期，加上为完成下一条指令所需的最长时间20个机器周期（如果该指令是16位操作数的DIV，MUL指令），若系统中只有一个中断源，再加上LCALL调用指令7个机器周期，则最长的响应时间是2+8+20+7个机器周期。

所以，中断响应时间一般大于10个机器周期小于37个机器周期。

7.9.8 外部中断输入

G80F980有3个外部中断输入。外部中断0-1分别有一个独立的中断源，外部中断4有8个中断源共用一个中断矢量地址。外部中断0/1可以通过设置TCON寄存器的IT1, IT0位来选择是电平触发或是边沿触发。当 $IT_x = 0$ ($x = 0, 1$)时，外部中断 INT_x ($x = 0, 1$)引脚为低电平触发；当 IT_x ($x = 0, 1$) = 1，外部中断 INT_x ($x = 0, 1$)为沿触发，在这个模式中，如果采样到一个系统时钟周期内 INT_x ($x = 0, 1$)引脚上采样为高电平而下个系统时钟周期采样为低电平，TCON寄存器的中断请求标志位置1，发出一个中断请求。由于外部中断引脚每个机器周期采样一次，输入高或低电平应当保持至少1个机器周期以确保能够被正确采样到。

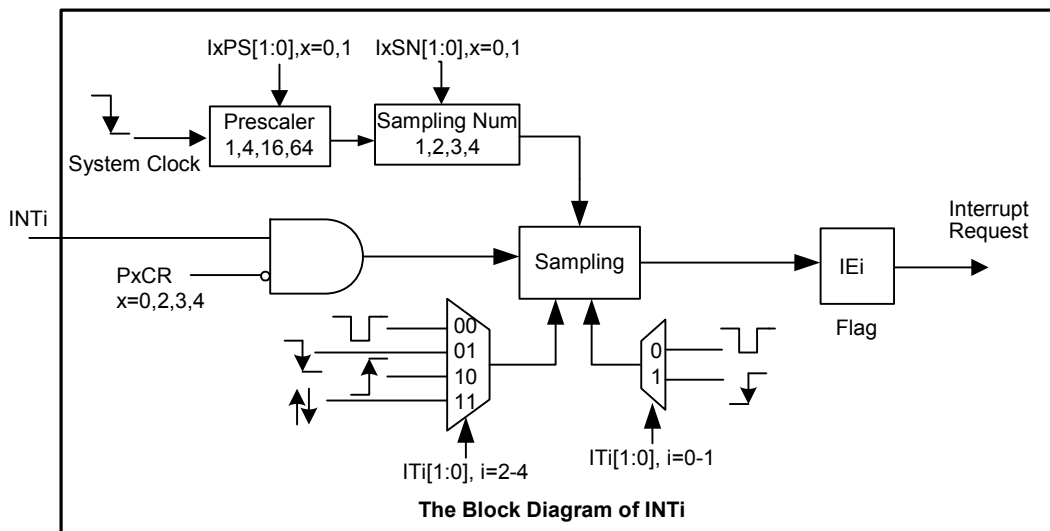
如果外部中断为下降沿触发，外部中断源应当将中断脚至少保持1个机器周期高电平，然后至少保持1个机器周期低电平。这样就确保了边沿能够被检测到以使 IE_x 置1。当调用中断服务程序后，CPU自动将 IE_x 清0。

如果外部中断为低电平触发，外部中断源必须一直保持请求有效，直到产生所请求的中断为止，此过程需要2个系统时钟周期。如果中断服务完成后而外部中断仍旧维持，则会产生下一次中断。当中断为电平触发时不必清除中断标志 IE_x ($x = 0, 1$)，因为中断只与输入口电平有关。

中断采样时钟分频比和连续采样次数均可以设置EXCON寄存器进行调节，满足不同中断响应要求。

外部中断4除了具有更多的中断触发方式外，与外部中断0, 1操作类似。

当G80F980进入空闲或是掉电模式，中断会唤醒处理器继续工作，详见电源管理章节。



注意：外部中断0-1的中断标志位在执行中断服务程序时被自动硬件清0，但外部中断4标志位IF40-46必须要软件清0。

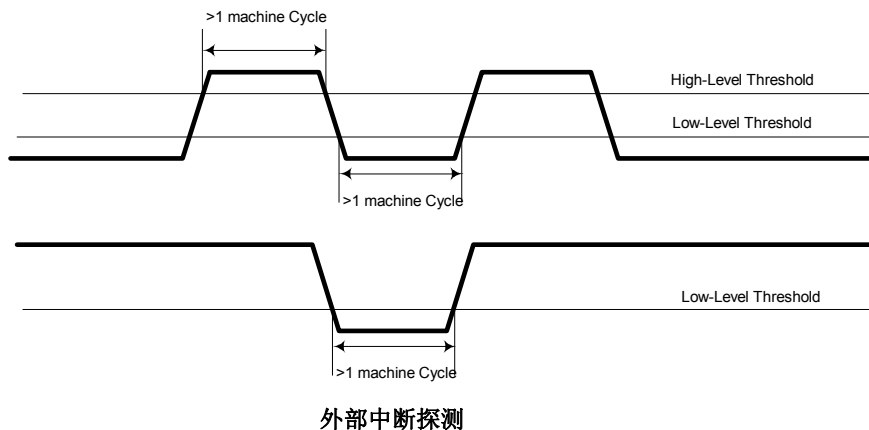


Table7.47 外部中断采样次数控制寄存器

95H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
EXCON	I1PS1	I1PS0	I1SN1	I1SN0	I0PS1	I0PS0	I0SN1	I0SN0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-6	I1PS[1:0]	外部中断INT4采样时钟预分频比选择位 00: 1/4 01: 1/16 10: 1/64 11: 1/256
5-4	I1SN[1:0]	外部中断INT4连续采样次数选择位 00: 1 01: 2 10: 3 11: 4
3-2	I0PS[1:0]	外部中断INT0,1采样时钟预分频比选择位 00: 1/4 01: 1/16 10: 1/64 11: 1/256
1-0	I0SN[1:0]	外部中断INT0,1连续采样次数选择位 00: 1 01: 2 10: 3 11: 4

注意: 1.若I0SN[1:0]=11, 则外部中断0, 1, 2, 3(下降沿触发), 连续采样4次低电平才会产生中断标志。

7.9.9 中断汇总

中断源	向量地址	允许位	标志位	轮询优先级	中断号(c语言)
Reset	0000H	-	-	0(最高级)	
INT0	0003H	EX0	IE0	1	0
Timer0	000BH	ET0	TF0	2	1
INT1	0013H	EX1	IE1	3	2
Timer1	001BH	ET1	TF1	4	3
EUART	0023H	ES	RI+TI	5	4
Timer2	002BH	ET2	TF2+EXF2	6	5
ADC	0033H	EADC	ADCIF/ADGIF/ADLIF	7	6
SPI	003BH	ESPI	SPIF	8	7
CMP1	0043H	ECMP1	C1IF	9	8
Timer4	004BH	ET4	TF4	10	9
INT4	0053H	EX4+IENC	IF4X	11	10
Timer3	005BH	ET3	TF3	12	11
MCM	0063H	EMCM+PWMP/Z/0/1/2IE	PWMP/Z/0/1/2IF	13	12
PWM3	006BH	EPWM3+PWM30/31/32IE	PWM30/31/32IF	14	13
SCM	0073H	ESCM	SCMIF	15	14
CAP	007BH	CAP0/1/2IE, CAPTIE	CAP0/1/2IF, CAPTIF	16	15
CMP2	0083H	ECMP2	C2IF	17	16
CRC	008BH	ECRC	CRCIF	18	17

8. 增强功能

8.1 电机控制模块MCM (Motor Control Module) PWM0/1/2

8.1.1 特性

- 重要寄存器受保护寄存器PWMRLDEN控制
- 16位时基计数器
- 三种时基计数模式：边沿对齐计数、中心对齐计数和单次计数模式
- 两种波形输出模式：互补模式与独立模式
- 互补模式下提供死区控制逻辑，并支持对称和非对称两种模式
- 提供PWM周期匹配、归零匹配和占空比比较匹配中断
- 3路互补PWM输出或6路独立PWM输出，输出极性可选择
- 提供故障检测功能可紧急关闭PWM输出
- 提供寄存器重载使能位以保证寄存器重载同步
- 4档预分频/后分频功能
- 可切换手动控制PWM输出
- 可选择自动触发Timer或ADC转换以及在一个PWM时基周期内任意两个时刻触发ADC
- PWM最终输出可配置成按不同顺序映射到管脚（参见代码选项章节）

G80F980内建一个16位电机控制PWM模块。PWM模块由时基模块（PWM Time Base Block）、波形发生模块（Wave Generator）、死区&极性控制模块（Dead&Polarity Logic）、故障检测（Fault Detect Logic）以及输出控制模块（Output Control Logic）组成，如图8-1-1所示。

时基模块中通过一个16位计数器（PWM Counter）对PWM时钟（PWM Clock）计数，用该计数器的值与周期寄存器和0比较以产生周期匹配信号与归零信号、与占空比寄存器比较以产生占空比匹配信号，结合波形发生模块产生原始的六路PWM信号Px_O/Px1_O，该原始PWM信号经过死区&极性控制模块后产生带死区和极性的PWM信号Px_D/Px1_D，最后的输出控制模块和故障检测模块决定是否将Px_D/Px1_D信号输出至PWM引脚PWMx/PWMx1上。

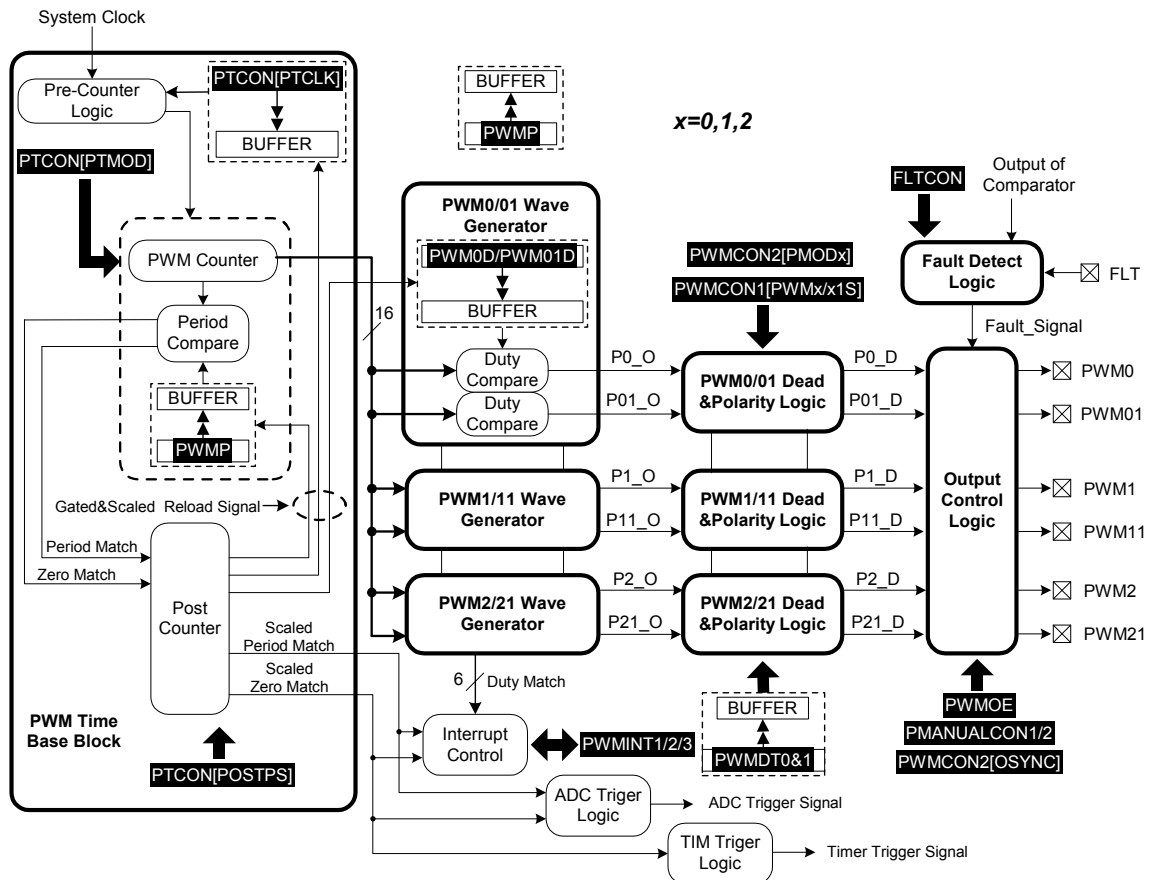


图8-1-1 PWM模块简图

8.1.2 寄存器

MCM模块所使用寄存器如下：

功能	名称	寄存器描述
PWM计数器时钟(时基)设置	PTCON	设置PWM时基的预分频、PWM模块中控制信号的后分频以及PWM计数器的运行模式
PWM控制寄存器	PWMCON1	设置PWM波形的输出极性 & 输出模式(互补/独立)
	PWMCON2	PWM触发事件的使能
	PWMCON3	重载, 死区方式及输出方式选择
PWM输出控制	PWMOE	设置PWM引脚输出PWM模块产生的信号或者作为普通IO口
	PMANUALCON1	设置PWM模块输出PWM波形或直接输出PMANUALCON2中定义的电平
	PMANUALCON2	定义PWM模块直接输出的电平
PWM死区控制	PWMDT0L	设置PWM原始波形上升沿触发死区时间
	PWMDT0H	
	PWMDT1L	设置PWM原始波形下降沿触发死区时间
	PWMDT1H	
故障检测保护	FLTCN	设置故障保护功能
PWM中断	PWMINTEN	PWM中断使能位控制
	PWMINTF	PWM中断标志位
寄存器重载使能控制	PWMRLDEN	禁止/使能占空比寄存器/周期寄存器/PWM时基预分频的重载
PWM周期寄存器	PWMPL	16位周期寄存器
	PWMPH	
PWM计数寄存器	PWMCL	16位计数寄存器
	PWMCH	
PWM占空比寄存器	PWM0DL	16位占空比寄存器 独立输出模式: PWMxDL/H (x = 0-2) 控制PWMx (x = 0-2) 原始波形 PWMx1DL/H (x = 0-2) 控制PWMx1 (x = 0-2) 原始波形 互补输出模式: 对称波形 PWM0DL/H同时控制PWM0/01原始波形 PWM1DL/H同时控制PWM1/11原始波形 PWM2DL/H同时控制PWM2/21原始波形 互补输出模式(非对称波形, 仅在中心对齐计数模式下存在): PWM0DL/H同时控制PWM0/01增计数过程的原始波形 PWM01DL/H同时控制PWM0/01减计数过程的原始波形 PWM1DL/H同时控制PWM1/11增计数过程的原始波形 PWM11DL/H同时控制PWM1/11减计数过程的原始波形 PWM2DL/H同时控制PWM2/21增计数过程的原始波形 PWM21DL/H同时控制PWM2/21减计数过程的原始波形
	PWM0DH	
	PWM01DL	
	PWM01DH	
	PWM1DL	
	PWM1DH	
	PWM11DL	
	PWM11DH	
	PWM2DL	
	PWM2DH	
	PWM21DL	
	PWM21DH	
PWM事件触发寄存器1	PWMCMP1L	16位事件触发比较寄存器1
	PWMCMP1H	
PWM事件触发寄存器2	PWMCMP2L	16位事件触发比较寄存器2
	PWMCMP2H	

Table 8.1 PWM时基控制寄存器

FF90H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PTCON	PTMOD1	PTMOD0	PTCLK2	PTCLK1	PTCLK0	POSTPS2	POSTPS1	POSTPS0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 6	PTMOD[1:0]	<p>PWM0/1/2时基模块中PWM计数器工作模式选择位段</p> <p>00: 边沿对齐计数模式, 在此模式下, 时基计数器只会发生归零匹配</p> <p>01: 中心对齐计数模式, 在此模式下, 时基计数器会发生归零匹配和周期匹配</p> <p>1x: 单次计数模式, 在此模式下, 时基计数器只会发生归零匹配</p>
5 - 3	PTCLK[2:0]	<p>PWM时基预分频系数选择位段</p> <p>000: PWM计数器时钟 = 系统时钟/1</p> <p>001: PWM计数器时钟 = 系统时钟/2</p> <p>010: PWM计数器时钟 = 系统时钟/4</p> <p>011: PWM计数器时钟 = 系统时钟/8</p> <p>100: PWM计数器时钟 = 系统时钟/16</p> <p>101: PWM计数器时钟 = 系统时钟/32</p> <p>110: PWM计数器时钟 = 系统时钟/64</p> <p>111: PWM计数器时钟 = 系统时钟/128</p>
2 - 0	POSTPS[2:0]	<p>后分频系数选择位段</p> <p>000: 无后分频功能</p> <p>001: 中断、重载信号与事件触发信号将2分频</p> <p>010: 中断、重载信号与事件触发信号将3分频</p> <p>011: 中断、重载信号与事件触发信号将4分频</p> <p>100: 中断、重载信号与事件触发信号将5分频</p> <p>101: 中断、重载信号与事件触发信号将6分频</p> <p>110: 中断、重载信号与事件触发信号将7分频</p> <p>111: 中断、重载信号与事件触发信号将8分频</p>

Table 8.4 PWM模块控制寄存器1

D9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCON1	POUTMOD	PWMSYM	PWM21S	PWM11S	PWM01S	PWM2S	PWM1S	PWM0S
读/写	读/写	读/写-	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	POUTMOD	PWM0/1/2输出模式（独立/互补）控制位 0: 整个模块配置为3通道互补输出 1: 整个模块配置为6路独立输出
6	PWMSYM	PWM波形对称性控制位（仅在中心对齐计数模式的互补输出时有效） 0: 互补输出对称PWM波形 1: 互补输出非对称PWM波形
5	PWM21S	PWM21输出有效电平（Active Polar）选择位 0: PWM21占空比期间输出低电平，其余时间输出高电平 1: PWM21占空比期间输出高电平，其余时间输出低电平
4	PWM11S	PWM11输出有效电平（Active Polar）选择位 0: PWM11占空比期间输出低电平，其余时间输出高电平 1: PWM11占空比期间输出高电平，其余时间输出低电平
3	PWM01S	PWM01输出有效电平（Active Polar）选择位 0: PWM01占空比期间输出低电平，其余时间输出高电平 1: PWM01占空比期间输出高电平，其余时间输出低电平
2	PWM2S	PWM2输出有效电平（Active Polar）选择位 0: PWM2占空比期间输出高电平，其余时间输出低电平 1: PWM2占空比期间输出低电平，其余时间输出高电平
1	PWM1S	PWM1输出有效电平（Active Polar）选择位 0: PWM1占空比期间输出高电平，其余时间输出低电平 1: PWM1占空比期间输出低电平，其余时间输出高电平
0	PWM0S	PWM0输出有效电平（Active Polar）选择位 0: PWM0占空比期间输出高电平，其余时间输出低电平 1: PWM0占空比期间输出低电平，其余时间输出高电平

Table 8.5 PWM模块控制寄存器2

DAH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCON2	ZETIM1	ZETIM0	PEAD	ZEAD	CMP21	CMP20	CMP11	CMP10
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-6	ZETIM[1:0]	<p>PWM时基计数器启动Timer2使能位</p> <p>00: PWM模块不会触发启动Timer2</p> <p>01: 中心对齐模式下, PWM时基计数器周期匹配时启动Timer2 边沿对齐或单次模式下, PWM时基计数器归零时刻启动Timer2</p> <p>10: 中心对齐模式下, PWM时基计数器归零匹配时启动Timer2 边沿对齐或单次模式下, PWM时基计数器归零时刻启动Timer2</p> <p>11: 中心对齐模式下, PWM时基计数器周期匹配和归零匹配时均启动Timer2 边沿对齐或单次模式下, PWM时基计数器归零时刻启动Timer2 (Timer2工作在单次触发模式下, 上述设置才有效)</p>
5	PEAD	<p>PWM时基计数器与周期寄存器匹配时刻启动AD转换使能位 (只在中心对齐模式下有效)</p> <p>0: PWM模块周期匹配时刻不会触发启动AD</p> <p>1: PWM时基计数器与周期寄存器匹配时刻启动AD (ADC的ADON和PWMTRGEN位置1的情况下)</p>
4	ZEAD	<p>PWM时基计数器归0时刻启动AD转换使能位</p> <p>0: PWM模块归0时刻不会触发启动AD</p> <p>1: PWM时基计数器归0时刻启动AD (ADC的ADON和PWMTRGEN位置1的情况下)</p>
3-2	CMP2[1:0]	<p>PWM时基计数器与比较寄存器2相等时刻启动AD转换使能位</p> <p>00: 不触发启动AD</p> <p>01: 只要PWM计数器与比较寄存器2相等时会触发启动AD</p> <p>10: 只有PWM计数器在增计数过程中与比较寄存器2相等时会触发启动AD (仅在PTMOD = 01中心对齐模式下有效)</p> <p>11: 只要PWM计数器在减计数过程中与比较寄存器2相等时会触发启动AD (仅在PTMOD = 01 (ADC的ADON和PWMTRGEN位置1的情况下)</p>
1-0	CMP1[1:0]	<p>PWM时基计数器与比较寄存器1相等时刻启动AD转换使能位</p> <p>00: 不触发启动AD</p> <p>01: 只要PWM计数器与比较寄存器1相等时会触发启动AD</p> <p>10: 只有PWM计数器在增计数过程中与比较寄存器1相等时会触发启动AD (仅在PTMOD = 01中心对齐模式下有效)</p> <p>11: 只要PWM计数器在减计数过程中与比较寄存器1相等时会触发启动AD (仅在PTMOD = 01中心对齐模式下有效) (ADC的ADON和PWMTRGEN位置1的情况下)</p>

Table 8.5 PWM模块控制寄存器3

DBH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCON3	PDLDEN	DT1	DT0	-	OSYNC	PDCON2	PDCON1	PDCON0
读/写	读/写	读/写	读/写	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

位编号	位符号	说明
7	PDLDEN	PWM计数器周期匹配时重载占空比死区寄存器使能位 （只在中心对齐模式下有效） 0：周期匹配时刻不允许重载占空比寄存器和死区寄存器以及死区方式选择位段DT[1:0]及DUTY区控制位 1：周期匹配时刻允许重载占空比寄存器和死区寄存器以及死区方式选择位段DT[1:0]及DUTY区控制位
6-5	DT[1:0]	死区寄存器方式选择位段 00：PWM0/1/2的上升沿死区由PWMDT0设置； PWM0/1/2的下降沿死区由PWMDT1设置 01：PWM1/2的上升沿死区和下降沿死区都由寄存器PWMDT0设置； PWM0的上升沿死区和下降沿死区都由寄存器PWMDT1设置 10：PWM0/2的上升沿死区和下降沿死区都由寄存器PWMDT0设置； PWM1的上升沿死区和下降沿死区都由寄存器PWMDT1设置 11：PWM0/1的上升沿死区和下降沿死区都由寄存器PWMDT0设置； PWM2的上升沿死区和下降沿死区都由寄存器PWMDT1设置
3	OSYNC	手动修改PWMx/PWMx1输出同步位： （手动修改通过设置PMANUALCON1/2寄存器实现，任一寄存器的修改，何时生效都由此寄存器控制） 0：手动修改后立即生效 1：手动修改与PWM周期同步后生效
2	PDCON2	PWM2/PWM21原始波形DUTY区控制位 0：计数器值小于占空比寄存器时定义为DUTY区 1：计数器值大于等于占空比寄存器时定义为DUTY区
1	PDCON1	PWM1/PWM11原始波形DUTY区控制位 0：计数器值小于占空比寄存器时定义为DUTY区 1：计数器值大于等于占空比寄存器时定义为DUTY区
0	PDCON0	PWM0/PWM01原始波形DUTY区控制位 0：计数器值小于占空比寄存器时定义为DUTY区 1：计数器值大于等于占空比寄存器时定义为DUTY区

Table 8.2 PWM周期寄存器低位

FF99H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMPL	PP.7	PP.6	PP.5	PP.4	PP.3	PP.2	PP.1	PP.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	PP.7 - PP.0	PWM周期控制寄存器低8位

Table 8.3 PWM周期寄存器高位

FF98H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMPH	PP.15	PP.14	PP.13	PP.12	PP.11	PP.10	PP.9	PP.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PP.15 - PP.8	PWM周期控制寄存器高8位						

Table 8.3 PWM计数寄存器低位

FF9BH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCL	PC.7	PC.6	PC.5	PC.4	PC.3	PC.2	PC.1	PC.0
读/写	读	读	读	读	读	读	读	读
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PC.7 - PC.0	PWM计数寄存器低8位						

Table 8.3 PWM计数寄存器高位

FF9AH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCH	PC.15	PC.14	PC.13	PC.12	PC.11	PC.10	PC.9	PC.8
读/写	读	读	读	读	读	读	读	读
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PC.15 - PC.8	PWM计数寄存器高8位						

由于对PWMCL和PWMCH的读操作需要2条指令才能完成。为保证精确计数，建议对PWMCL和PWMCH寄存器组的读操作都遵循先高位后低位顺序。

Table 8.6 PWM0占空比控制寄存器低位

DCH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM0DL	PD0.7	PD0.6	PD0.5	PD0.4	PD0.3	PD0.2	PD0.1	PD0.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD0.7-PD0.0	PWM0占空比寄存器低8位						

Table 8.7 PWM0占空比控制寄存器高位

DDH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM0DH	PD0.15	PD0.14	PD0.13	PD0.12	PD0.11	PD0.10	PD0.9	PD0.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD0.15-PD0.8	PWM0占空比寄存器高8位						

Table 8.8 PWM1占空比控制寄存器低位

DEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM1DL	PD1.7	PD1.6	PD1.5	PD1.4	PD1.3	PD1.2	PD1.1	PD1.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD1.7-PD1.0	PWM1占空比寄存器低8位						

Table 8.9 PWM1占空比控制寄存器高位

DFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM1DH	PD1.15	PD1.14	PD1.13	PD1.12	PD1.11	PD1.10	PD1.9	PD1.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD1.15-PD1.8	PWM1占空比寄存器高8位						

Table 8.10 PWM2占空比控制寄存器低位

D2H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM2DL	PD2.7	PD2.6	PD2.5	PD2.4	PD2.3	PD2.2	PD2.1	PD2.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD2.7-PD2.0	PWM2占空比寄存器低8位						

Table 8.11 PWM2占空比控制寄存器高位

D3H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM2DH	PD2.15	PD2.14	PD2.13	PD2.12	PD2.11	PD2.10	PD2.9	PD2.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD2.15-PD2.8	PWM2占空比寄存器高8位						

Table 8.12 PWM01占空比控制寄存器低位

D4H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM01DL	PD01.7	PD01.6	PD01.5	PD01.4	PD01.3	PD01.2	PD01.1	PD01.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD01.7-PD01.0	PWM01占空比寄存器低8位						

Table 8.13 PWM01占空比控制寄存器高位

D5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM01DH	PD01.15	PD01.14	PD01.13	PD01.12	PD01.11	PD01.10	PD01.9	PD01.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD01.15-PD01.8	PWM01占空比寄存器高8位						

Table 8.14 PWM11占空比控制寄存器低位

D6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM11DL	PD11.7	PD11.6	PD11.5	PD11.4	PD11.3	PD11.2	PD11.1	PD11.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD11.7-PD11.0	PWM11占空比寄存器低8位						

Table 8.15 PWM11占空比控制寄存器高位

D7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM11DH	PD11.15	PD11.14	PD11.13	PD11.12	PD11.11	PD11.10	PD11.9	PD11.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PD11.15-PD11.8	PWM11占空比寄存器高8位						

Table 8.16 PWM21占空比控制寄存器低位

F9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM21DL	PD21.7	PD21.6	PD21.5	PD21.4	PD21.3	PD21.2	PD21.1	PD21.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 0	PD21.7- PD21.0	PWM21占空比寄存器低8位

Table 8.17 PWM21占空比控制寄存器高位

FAH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM21DH	PD21.15	PD21.14	PD21.13	PD21.12	PD21.11	PD21.10	PD21.9	PD21.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 0	PD21.15- PD21.8	PWM21占空比寄存器高8位

PWM占空比寄存器	PWM0DL	16位占空比寄存器 独立输出模式: PWMxDL/H (x = 0- 2) 控制PWMx (x = 0-2) 原始波形 PWMx1DL/H (x = 0- 2) 控制PWMx1 (x = 0-2) 原始波形 互补输出模式: 对称波形 (PWM0DL/H同时控制PWM0/01原始波形 PWM1DL/H同时控制PWM1/11原始波形 PWM2DL/H同时控制PWM2/21原始波形 互补输出模式 (非对称波形, 仅在中心对齐计数模式下存在): PWM0DL/H同时控制PWM0/01增计数过程的原始波形 PWM01DL/H同时控制PWM0/01减计数过程的原始波形 PWM1DL/H同时控制PWM1/11增计数过程的原始波形 PWM11DL/H同时控制PWM1/11减计数过程的原始波形 PWM2DL/H同时控制PWM2/21增计数过程的原始波形 PWM21DL/H同时控制PWM2/21减计数过程的原始波形
	PWM0DH	
	PWM01DL	
	PWM01DH	
	PWM1DL	
	PWM1DH	
	PWM11DL	
	PWM11DH	
	PWM2DL	
	PWM2DH	
	PWM21DL	
	PWM21DH	

Table 8.16 PWMCMP1事件触发比较寄存器1低位

BCH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCMP1L	PCMP1.7	PCMP1.6	PCMP1.5	PCMP1.4	PCMP1.3	PCMP1.2	PCMP1.1	PCMP1.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 0	PCMP1.7- PCMP1.0	PWMCMP1事件触发寄存器1低8位

Table 8.16 PWMCMP1事件触发比较寄存器1高位

BDH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCMP1H	PCMP1.15	PCMP1.14	PCMP1.13	PCMP1.12	PCMP1.11	PCMP1.10	PCMP1.9	PCMP1.8

读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PCMP1.15- PCMP1.8	PWMCMP1事件触发寄存器1高8位						

Table 8.16 PWMCMP2事件触发寄存器2低位

BEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCMP2L	PCMP2.7	PCMP2.6	PCMP2.5	PCMP2.4	PCMP2.3	PCMP2.2	PCMP2.1	PCMP2.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PCMP2.7- PCMP2.0	PWMCMP1事件触发寄存器1低8位						

Table 8.16 PWMCMP2事件触发比较寄存器2高位

BFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMCMP2H	PCMP2.15	PCMP2.14	PCMP2.13	PCMP2.12	PCMP2.11	PCMP2.10	PCMP2.9	PCMP2.8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	PCMP2.15- PCMP2.8	PWMCMP2事件触发寄存器1高8位						

Table 8.18 PWM输出使能寄存器

EFH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMOE	PWMEN	-	PWM21OE	PWM11OE	PWM01OE	PWM2OE	PWM1OE	PWM0OE
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	-	0	0	0	0	0	0
位编号	位符号	说明						
7	PWMEN	MCM时基使能控制位 (注: PWM0/1/2 共用一个时基) 0: 关闭MCM (PWM0/1/2) 时基 1: 打开MCM (PWM0/1/2) 时基						
5	PWM21OE	PWM21输出使能位 0: PWM21引脚作为普通I/O口 1: PWM21引脚作为PWM输出口, 输出PWM波形或POUT21定义的电平						
4	PWM11OE	PWM11输出使能位 0: PWM11引脚作为普通I/O口 1: PWM11引脚作为PWM输出口, 输出PWM波形或POUT11定义的电平						
3	PWM01OE	PWM01输出使能位 0: PWM01引脚作为普通I/O口 1: PWM01引脚作为PWM输出口, 输出PWM波形或POUT01定义的电平						
2	PWM2OE	PWM2输出使能位 0: PWM2引脚作为普通I/O口 1: PWM2引脚作为PWM输出口, 输出PWM波形或POUT2定义的电平						

1	PWM1OE	PWM1输出使能位 0: PWM1引脚作为普通I/O口 1: PWM1引脚作为PWM输出口, 输出PWM波形或POUT1定义的电平
0	PWM0OE	PWM0输出使能位 0: PWM0引脚作为普通I/O口 1: PWM0引脚作为PWM输出口, 输出PWM波形或POUT0定义的电平

Table 8.19 PWM0/1/2手动输出设置寄存器1

B6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PMANUALCON1	-	-	PMANUAL 21	PMANUAL 11	PMANUAL 01	PMANUAL 2	PMANUAL 1	PMANUAL 0
读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	0	0	0	0	0	0

位编号	位符号	说明
2 - 0	PMANUALx x = 0-2	PWMx口输出控制位 (只有当PWMxOE = 1时有效, 修改立即生效) 0: PWMx口输出PWM波形 1: PWMx口输出由PMANUALCON2寄存器中的POUTx位控制
5 - 3	PMANUALx1 x = 0-2	PWMx1口输出控制位 (只有当PWMx1OE = 1时有效, 修改立即生效) 0: PWMx1口输出PWM波形 1: PWMx1口输出由PMANUALCON2寄存器中的POUTx1位控制

Table 8.20 PWM0/1/2手动输出设置寄存器2

B7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PMANUALCON2	-	-	POUT21	POUT11	POUT01	POUT2	POUT1	POUT0
读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	0	0	0	0	0	0

位编号	位符号	说明
5 - 3	POUTx1 x = 0-2	PMANUALx1 = 1时, 决定PWMx1口的输出电平 (修改立即生效) 0: 输出0 1: 输出1
2 - 0	POUTx x = 0-2	PMANUALx = 1时, 决定PWMx口的输出电平 (修改立即生效) 0: 输出0 1: 输出1

注意: 必须先写PMANUALCON2寄存器, 后写PMANUALCON1寄存器。

Table 8.21 PWM0/1/2故障检测保护寄存器

D1H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
FLTCON	FLT1EN	FLT1SEL	FLT2EN	FLT2S	FLT2DEB1	FLT2DEB0	FLTM	FLTSTAT
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	FLT1EN	故障检测1功能控制位（注1） 0：故障检测1功能禁止 1：故障检测1功能打开，输入源可以选择比较器1或比较器2
6	FLT1SEL	故障检测1输入源选择控制位 0：比较器1滤波后的输出作为PWM故障检测1输入源，高电平有效 1：比较器2滤波后的输出作为PWM故障检测1输入源，高电平有效
5	FLT2EN	故障检测2功能控制位（注1） 0：故障检测2功能禁止 1：故障检测2功能打开，输入源为FLT引脚
4	FLT2S	故障检测2有效电平选择位 0：故障检测2 FLT输入高电平有效 1：故障检测2 FLT输入低电平有效
3 - 2	FLT2DEB[1:0]	故障检测2输入滤波参数选择 00：无滤波 01：滤波时间常数为1us 10：滤波时间常数为4us 11：滤波时间常数为16us <i>注：上述滤波常数时间不是精确值，仅供参考</i>
1	FLTM	检测功能模式选择 0：锁存模式，当检测到故障输入有效，FLTSTAT 被硬件置1，PWM将立即停止输出，当故障输入变为无效时状态保持不变，只有当FLTSTAT被软件清零后，PWM波形才会在PWM时基计数器归0时刻或与周期寄存器PWMPH/LH的值匹配时恢复输出。（当故障输入一直有效时FLTSTAT无法被软件清除） 1：逐次模式，PWM输出直接由故障检测输入端来控制，如果故障输入有效，FLTSTAT被硬件置1，立即关闭PWM输出。如果故障输入变为无效，FLTSTAT自动清0，PWM0/1/2波形将在PWM时基计数器归0时刻自动恢复输出
0	FLTSTAT	故障检测标志位 0：PWM0/1/2模块处于正常输出状态 1：出现过流，PWM0/1/2模块处于停止输出状态，如果在锁存模式，可软件清0，恢复PWM输出

注1：故障检测输入1和2可同时打开，任一信号有效都会关闭PWM0/1/2模块输出。

Table 8.22 PWM中断使能控制寄存器

FF91H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMINTEN	PWMP1E	PWMZ1E	PTDD21E	PTUD21E	PTDD11E	PTUD11E	PTDD01E	PTUD01E
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	PWMP1E	PWM时基周期匹配中断允许位 （只在中心对齐模式下有效） 0：禁止PWM时基周期匹配中断 1：允许PWM时基周期匹配中断
6	PWMZ1E	PWM时基归零中断允许位 0：禁止PWM时基归零中断 1：允许PWM时基归零中断
5	PTDD21E	PWM时基计数器减计数时与占空比寄存器PWM2DL/H匹配时中断使能位 （只在中心对齐模式下有效） 0：禁止PWM计数器减计数时与PWM2DL/H匹配时触发中断 1：允许PWM计数器减计数时与PWM2DL/H匹配时触发中断
4	PTUD21E	PWM时基计数器增计数时与占空比寄存器PWM2DL/H匹配时中断使能位 0：禁止PWM计数器增计数时与PWM2DL/H匹配时触发中断 1：允许PWM计数器增计数时与PWM2DL/H匹配时触发中断
3	PTDD11E	PWM时基计数器减计数时与占空比寄存器PWM1DL/H匹配时中断使能位 （只在中心对齐模式下有效） 0：禁止PWM计数器减计数时与PWM1DL/H匹配时触发中断 1：允许PWM计数器减计数时与PWM1DL/H匹配时触发中断
2	PTUD11E	PWM时基计数器增计数时与占空比寄存器PWM1DL/H匹配时中断使能位 0：禁止PWM计数器增计数时与PWM1DL/H匹配时触发中断 1：允许PWM计数器增计数时与PWM1DL/H匹配时触发中断
1	PTDD01E	PWM时基计数器减计数时与占空比寄存器PWM0DL/H匹配时中断使能位 （只在中心对齐模式下有效） 0：禁止PWM计数器减计数时与PWM0DL/H匹配时触发中断 1：允许PWM计数器减计数时与PWM0DL/H匹配时触发中断
0	PTUD01E	PWM时基计数器增计数时与占空比寄存器PWM0DL/H匹配时中断使能位 0：禁止PWM计数器增计数时与PWM0DL/H匹配时触发中断 1：允许PWM计数器增计数时与PWM0DL/H匹配时触发中断

Table 8.23 PWM中断标志寄存器

E8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMINTF	PWMPIF	PWMZIF	PTDD2IF	PTUD2IF	PTDD1IF	PTUD1IF	PTDD0IF	PTUD0IF
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	PWMPIF	PWM时基周期匹配中断标志位 （只在中心对齐模式下有效） 0: 无PWM周期匹配中断 1: 发生PWM周期匹配中断
6	PWMZIF	PWM时基归零中断标志位 0: 无PWM时基归零中断 1: 发生PWM时基归零中断
5	PTDD2IF	PWM时基计数器减计数时与占空比寄存器PWM2DL/H匹配时中断标志位 （只在中心对齐模式下有效） 0: 未发生中断或发生过但被PTUD2IF信号清0。 1: 发生中断
4	PTUD2IF	PWM时基计数器增计数时与占空比寄存器PWM2DL/H匹配时中断标志位 0: 未发生中断或发生过但被PTDD2IF信号清0。 1: 发生中断
3	PTDD1IF	PWM时基计数器减计数时与占空比寄存器PWM1DL/H匹配时中断标志位 （只在中心对齐模式下有效） 0: 未发生中断或发生过但被PTUD1IF信号清0。 1: 发生中断
2	PTUD1IF	PWM时基计数器增计数时与占空比寄存器PWM1DL/H匹配时中断标志位 0: 未发生中断或发生过但被PTDD1IF信号清0。 1: 发生中断
1	PTDD0IF	PWM时基计数器减计数时与占空比寄存器PWM0DL/H匹配时中断标志位 （只在中心对齐模式下有效） 0: 未发生中断或发生过但被PTUD0IF信号清0。 1: 发生中断
0	PTUD0IF	PWM时基计数器增计数时与占空比寄存器PWM0DL/H匹配时中断标志位 0: 未发生中断或发生过但被PTDD0IF信号清0。 1: 发生中断

注意:

- 即使PWM中断允许位为0，中断标志位也会置1，但中断不会响应。
- PTDDxIF置1时，硬件会自动将PTUDxIF清零；同样，PTUDxIF置1时，硬件会自动将PTDDxIF清零；因此通过判断这两个标志位可以判断目前PWM波形处在有效状态还是无效状态。

Table 8.24 PWM死区控制寄存器0的低位

FF93H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMDT0L	DT0.7	DT0.6	DT0.5	DT0.4	DT0.3	DT0.2	DT0.1	DT0.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 0	DT0.7-DT0.0	PWM死区0控制寄存器低位

Table 8.25 PWM死区控制寄存器0的高位

FF92H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMDT0H	-	-	-	-	DT0.11	DT0.10	DT0.9	DT0.8
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0
位编号	位符号	说明						
3 - 0	DT0.11-DT0.8	PWM死区0控制寄存器高位						

Table 8.26 PWM死区控制寄存器1的低位

FF95H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMDT1L	DT1.7	DT1.6	DT1.5	DT1.4	DT1.3	DT1.2	DT1.1	DT1.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	DT1.7-DT1.0	PWM死区1控制寄存器低位						

Table 8.27 PWM死区控制寄存器1的高位

FF94H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMDT1H	-	-	-	-	DT1.11	DT1.10	DT1.9	DT1.8
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0
位编号	位符号	说明						
3 - 0	DT1.11-DT1.8	PWM死区1控制寄存器高位						

Table 8.28 PWM定时器锁定寄存器

E7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWMRLDEN	RLDEN.7	RLDEN.6	RLDEN.5	RLDEN.4	RLDEN.3	RLDEN.2	RLDEN.1	RLDEN.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7 - 0	RLDEN.7-0	<p>寄存器修改和重载控制寄存器</p> <p>0x55: 允许软件对模块寄存器的修改</p> <p>0xAA: 允许模块带缓冲的寄存器的重载</p> <p>注:</p> <p>1. PWM模块的寄存器, 除了中断标志寄存器PWMINTF和手动控制寄存器PMANUALCON2外, 其它寄存器的修改有限制条件, 只有当PWMRLDEN = 0x55时才允许软件修改, 否则修改无效</p> <p>2. PWM模块中有一些寄存器带有缓冲寄存器, 包括占空比寄存器、周期寄存器和PWM时钟预分频寄存器位段PTCLK[1:0]、死区寄存器、以及死区模式选择位段DT[1:0]。这些寄存器在软件修改后不会立即生效, 只有在归零或周期溢出时硬件重载, 才会真正生效。是否允许这些寄存器重载同样受控制, 只有当PWMRLDEN = 0xAA时才允许重载</p>						

8.1.3 PWM0/1/2时基模块（PWM Time Base Block）

PWM模块中的时基模块（Time Base Block）主要由一个16位的计数器结合一个预分频器和一个后分频器组成。如图8-1-1所示，波形发生器（Wave Generator）用时基模块中16位计数器的值与占空比寄存器比较并结合死区/极性逻辑（Dead&Polarity Logic）和输出控制逻辑（Output Control）最终产生PWM波形。图8-1-2给出了时基模块的逻辑简图。

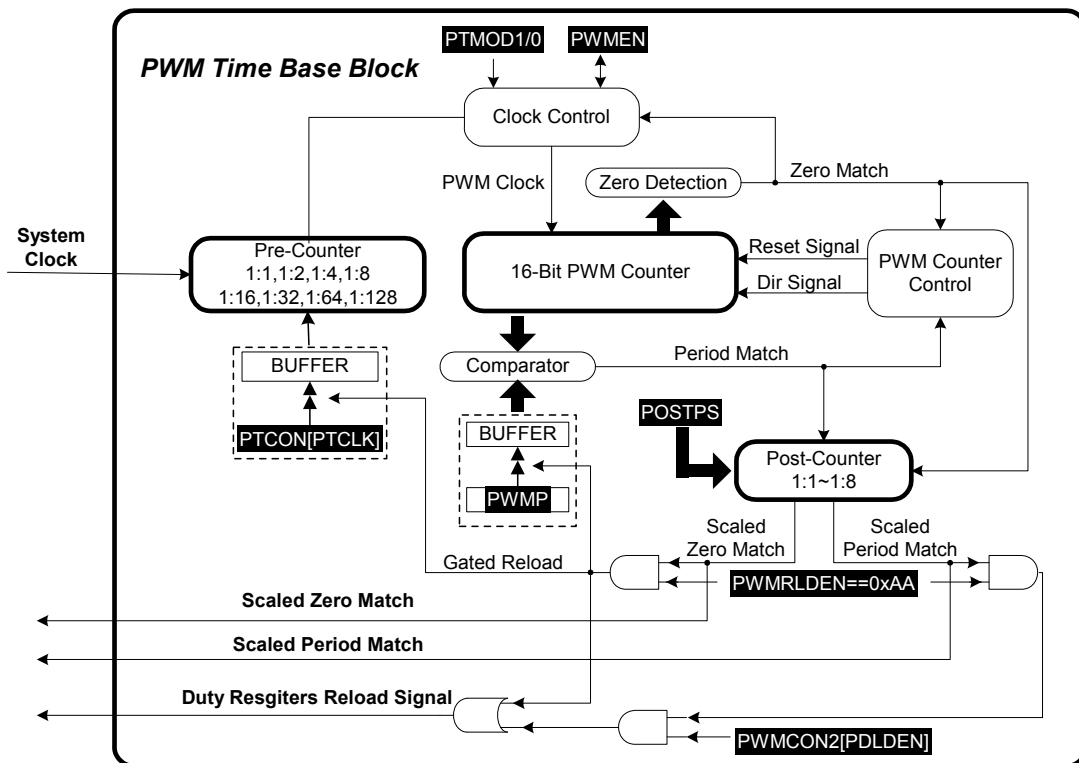


图8-1-2 PWM时基逻辑简图

PWM工作时钟

PWM时钟（PWM Clock）由系统时钟（System Clock）分频而来，通过预分频器（Pre-Counter）由寄存器PTCON中的位段[PTCLK1:PTCLK0]决定PWM模块时钟是系统时钟的1/1、1/2、1/4或1/8分频。

通过软件置位/清零PWOE寄存器中的PWMMEN位可以使能/禁止PWM时基模块。PWMMEN位从0置1时，PWM计数器（PWM Counter）开始对PWM时钟从0开始向上计数。PWMMEN位清0时，预分频计数器和PWM计数器都将清0。

注意：任何器件复位都将导致PWM时钟预分频计数器和PWM计数器清零。

PWM时基工作方式

PWM时基中16位的PWM计数器有3种工作方式，由PTCON寄存器中的[PMOD1:PMOD0]位段确定。计数模式的修改需要在PWM模块使能之前完成，如在PWM模块使能后进行计数模式修改可能会得到非预期结果。

边沿对齐计数模式

设置PTCON中PMOD位段为00，时基模块中的PWM计数器将工作在边沿对齐计数模式下。此模式下，软件令PWMEN位置1，PWM计数器将从0开始向上计数，直到与PWM周期寄存器PWMPH/H匹配，然后PWM计数器复位为0并继续向上计数，如此往复。PWM计数器复位为0的时刻即归零时刻（Zero Match）。软件令PWMEN位清0，PWM Counter将停止计数并在下一个系统时钟边沿复位为0并停止计数。如设置PWMPH/H = 5，则PWM计数器在此模式下的工作如图8-1-3所示。

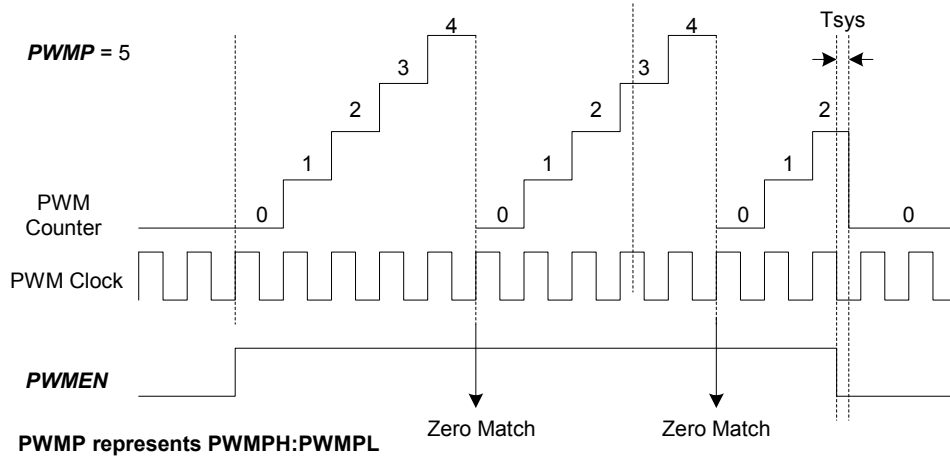


图8-1-3 边沿对齐计数模式

中心对齐计数模式

设置PTCON中PMOD位段为01，时基模块中的PWM计数器将工作在中心对齐计数模式下。此模式下，软件令PWMEN位置1，PWM计数器将从0开始向上计数，直到与PWM周期寄存器PWMPH/H匹配，然后PWM计数器开始向下计数直至归0，如此往复。PWM计数器向下计数至零的时刻即归零时刻（Zero Match），PWM计数器与周期寄存器匹配时刻为周期匹配时刻（Period Match）。软件令PWMEN位清0，PWM Counter将停止计数并在下一个系统时钟边沿复位为0并停止计数。如设置PWMPH/H = 5，则PWM计数器在此模式下的工作如图8-1-4所示。

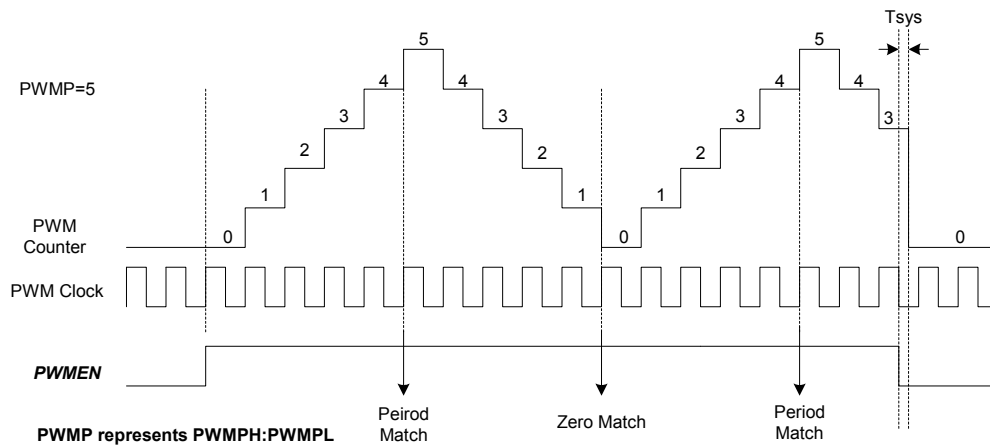


图8-1-4 中心对齐计数模式

单次计数模式

设置PTCON中PMD0位段为10或11，时基模块中的PWM计数器将工作在单次计数模式下。此模式下，软件令PWMEN位置1，PWM计数器将从0开始向上计数，直到与PWM周期寄存器PWMPL/H匹配时，PWM计数器复位为0，同时PWMEN位将由硬件清0，PWM输出高阻状态。PWM计数器复位为0的时刻即归零时刻（Zero Match）。

PWM计数器向上计数过程中若软件令PWMEN位清0，则PWM Counter将在下一个系统时钟边沿复位为0并停止计数。如设置PWMPL/H = 5，则PWM计数器在此模式下的工作如图8-1-5所示。

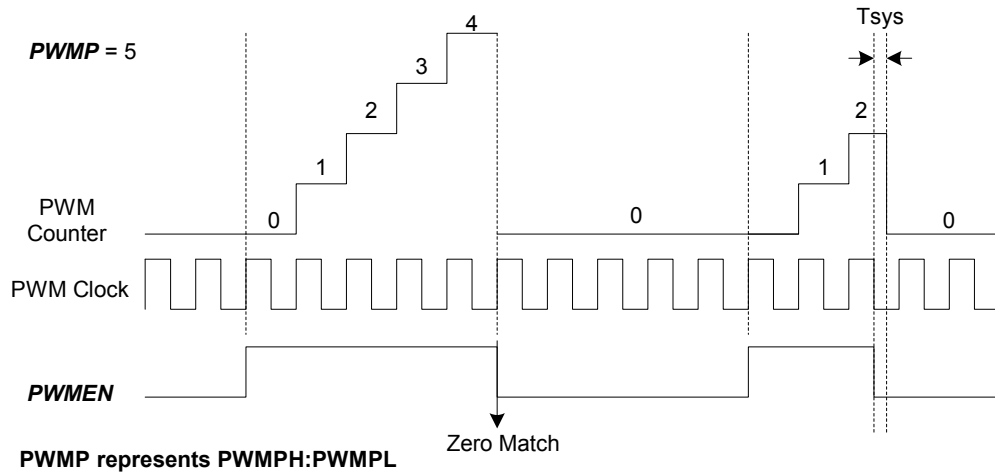


图8-1-5 单次计数模式

后分频

当不需要每个周期更新PWM的占空比时，后分频将非常有用。如图8-1-2所示，PWM计数器运行时产生的周期匹配信号（Period Match）和归零信号（Zero Match）经过后分频计数器（Post-Counter），可以实现1/1、1/2、1/3、1/4、1/5、1/6、1/7、1/8分频。通过寄存器PTCON中的POSTPS位段可以选择以上八档后分频。令POSTPS = 0则禁止后分频，周期匹配信号和归零信号不会被分频。

当PWMEN置1时，PWM开始计数，同时后分频计数器也开始以周期为单位开始增计数，也即是在每个计数周期计数值归零后分频计数器自动加1，当后分频计数器等于后分频系数设置值时，后分频有效，在该计数周期内产生的周期匹配信号和归零匹配信号有效，可有效触发周期/占空比/死区/比较值寄存器重载，周期中断，归零中断以及周期/归零/比较值匹配触发ADC，周期归零匹配触发Timer2。

首先，分频后的周期匹配信号（Scaled Period Match）和归零信号（Scaled Zero Match）结合PWMRLDEN寄存器值用于控制周期寄存器PWMPL/H、预分频系数PTCON[PTCLK1:PTCLK0]以及占空比寄存器的重载。

其次，分频后的周期匹配信号（Scaled Period Match）和归零信号（Scaled Zero Match）可用于产生归零中断和周期匹配中断，如图8-1-1所示。详见“**时基中断**”章节。

分频后的周期匹配信号（Scaled Period Match）和归零信号（Scaled Zero Match）可用于触发ADC，Timer2。在中心对齐模式下触发timer2，可用分频后的周期匹配信号触发也可用分频后的归零匹配信号触发；中心对齐模式触发ADC，可以选择在分频后的周期匹配信号触发，或分频后的归零信号触发，也可以同时选择两者。而在边沿对齐模式下，只能用分频后的归零信号触发。

最后，PWM比较寄存器触发AD信号也受后分频系数控制。在后分频有效周期内，PWM计数值与PWM比较值相等时可触发ADC。

但是，后分频只对周期匹配信号和归零信号有效，对占空比匹配时刻无效，比如，允许占空比中断和归零信号中断的情况下，设置后分频为8分频时，归零信号进中断会在8个PWM周期后才进一次中断，而占空比匹配信号则仍然会在每个PWM周期触发中断。

时基中断

时基中断包括分频后的周期匹配信号所触发的周期中断和分频后的归零信号所触发的归零中断。分频后的周期匹配信号发生时将令寄存器PWMINTF中的PWMPIF位置1，此时若PWMINTEN中的PWMPIE位为1则将触发周期中断；分频后的归零信号发生时将令寄存器PWMINTF中的PWMZIF位置1，此时若PWMINTEN中的PWMZIE位为1则将触发归零中断。PWMPIF位和PWMZIF位需由软件写0清除。

PTCON中POSTPS段等于0时，PWM计数器的周期匹配信号和归零信号不会被分频，故周期匹配信号每次发生时都将令PWMPIF位置1，归零信号每次发生时都将令PWMZIF位置1。

PWM周期寄存器与预分频系数，后分频系数的自动重载

周期寄存器PWMPL和PWMPH都有一个缓存寄存器（BUFFER），用户能够读写PWMPL/H但不能操作其缓存寄存器，如图8-1-2所示。PWM时基计数器运行时实际上是与周期寄存器的BUFFER进行比较以产生周期匹配信号。

PTCON寄存器中的PTCLK位段也有一个缓存寄存器（BUFFER），用户能够读写PTCLK位段但不能操作其缓存寄存器，如图8-1-2所示。系统时钟实际上是利用预分频系数位段PTCLK的BUFFER进行分频以产生PWM时钟（PWM Clock）。

PTCON寄存器中的POSTPS位段也有一个缓存寄存器（BUFFER），用户能够读写POSTPS位段但不能操作其缓存寄存器，如图8-1-2所示。系统时钟实际上是利用后分频系数位段POSTPS的BUFFER进行后分频操作。

由于存在缓冲寄存器，因此对周期寄存器，预分频系数和后分频系数寄存器位的读写有特殊规定：

当PWMRLDEN寄存器的值设为0x55时，才允许对这些寄存器进行修改，然后只有当PWMRLDEN寄存器中的值设为0xAA时，分频后的归零信号发生时才会将PWMPL和PWMPH寄存器和PTCLK位段锁存至对应的缓存寄存器（BUFFER）中，这样可以避免寄存器修改过程中发生重载，也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改，必须要有对PWMRLDEN寄存器的配合操作才能完成。

8.1.4 PWM0/1/2波形发生模块

PWM波形发生模块指图8-1-1中的Wave Generator模块。波形发生模块使用PWM时基模块中PWM计数器的值与6个16位占空比值进行比较，从而产生原始的6路PWM波形：P0_O、P01_O、P1_O、P11_O、P2_O和P21_O。该原始6路PWM波形信号经过死区&极性逻辑和输出控制逻辑最终反映在6个PWM输出引脚上，如图8-1-1所示。

波形发生器的逻辑简图如图8-1-6所示。

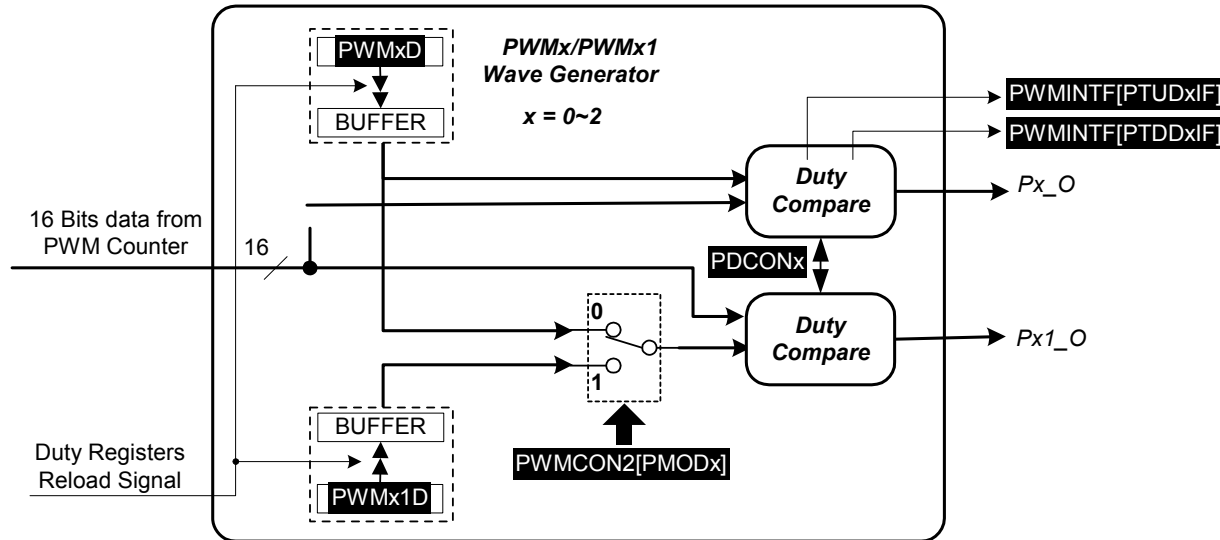


图8-1-6 波形发生器逻辑简图

PWM0/1/2输出模式

PWM模块中包含有3个波形发生模块，对应于3对PWM输出：PWM0/PWM01、PWM1/PWM11和PWM2/PWM21，如图8-1-1所示。对于每对PWM输出可以分别配置为互补模式或独立模式。

互补输出模式

令PWMCON1寄存器中POUTMOD位为0，PWMx/PWMx1工作在互补状态。

当工作在对称波形输出模式下（PWMSYM=0），16位PWM计数器与占空比寄存器PWMxDL/H比较以产生Px_O和Px1_O波形，因此PWMx和PWMx1引脚上最终输出的PWM波形使用的是同一占空比寄存器PWMxDL/H。在此输出模式下，可由寄存器PWMCON1设置PWMx/PWMx1引脚上输出波形的极性、并可由寄存器PWMDT0L/H和PWMDT1L/H设置死区（详见8.1.5节“PWM死区与极性控制模块”）。（x = 0 - 2）

当工作在非对称波形输出模式下（PWMSYM=1，仅在中心对齐技术模式下有效），16位PWM计数器在增计数过程中与占空比寄存器PWMxDL/H比较以产生Px_O和Px1_O波形，在减计数过程中与占空比寄存器PWMx1DL/H比较以产生Px_O和Px1_O波形。在此输出模式下，可由寄存器PWMCON1设置PWMx/PWMx1引脚上输出波形的极性、并可由寄存器PWMDT0L/H和PWMDT1L/H设置死区（详见8.1.5节“PWM死区与极性控制模块”）。（x = 0 - 2）

独立输出模式

令PWMCON1寄存器中POUTMOD位为1，PWMx/PWMx1工作在独立状态，16位PWM计数器与占空比寄存器PWMxDL/H比较以产生Px_O波形，16位PWM计数器与占空比寄存器PWMx1DL/H的值比较以产生Px1_O波形，因此PWMx和PWMx1引脚上最终输出的PWM波形使用的是不同的占空比寄存器PWMxDL/H与PWMx1DL/H。在独立输出模式下，可由寄存器PWMCON1设置PWMx/PWMx1引脚上输出波形的极性、但是寄存器PWMDT0L/H和PWMDT1L/H将无效，即独立输出模式下PWM的输出无死区（详见8.1.5节“PWM死区与极性控制模块”）。（x = 0 - 2）

占空比有效区域

在各种模式下，占空比有效区域都由PDxCON(x=0-2)定义。

当PDxCON=0时，边沿模式或独立模式时，计数值小于占空比寄存器定义为占空比有效区域（DutyZone），其余区域为非占空比区域（None-Duty Zone）；中心模式下，增计数时计数值小于占空比寄存器定义为占空比有效区域（DutyZone），其余

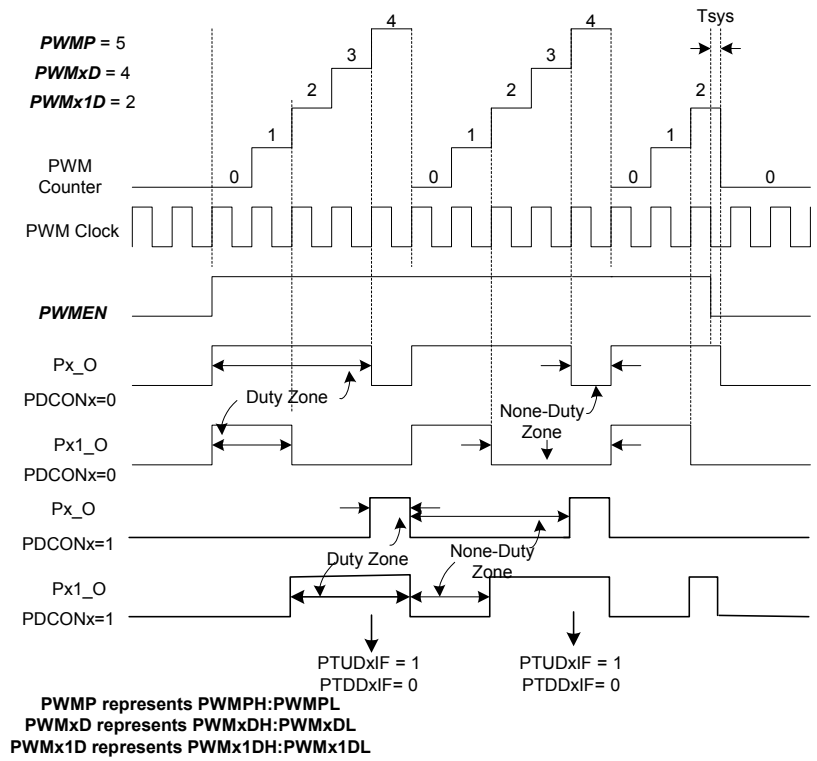


图8-1-8 边沿对齐计数、独立输出模式下的PWM原始波形

中心对齐计数模式下的PWM0/1/2原始波形

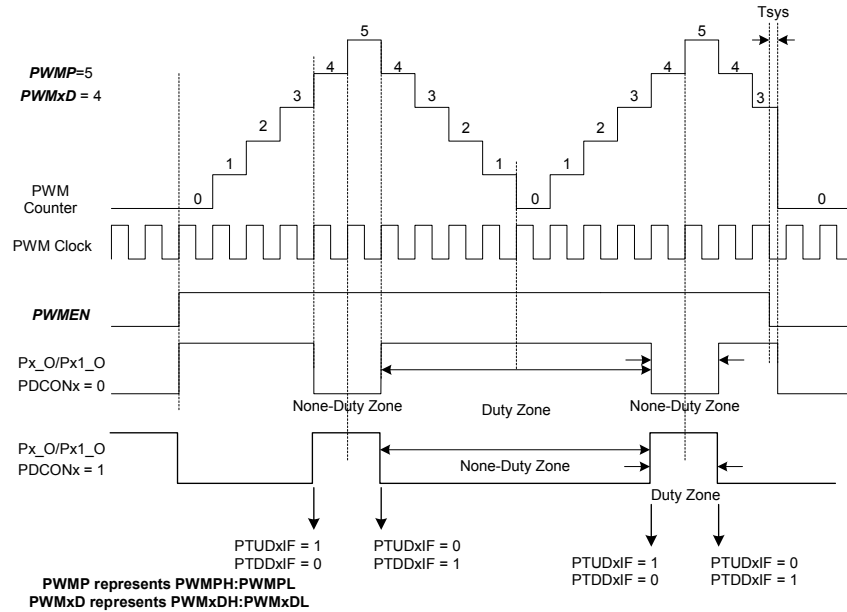


图8-1-9 中心对齐计数、互补输出模式下的PWM原始波形（对称波形）

令PTCON中PTMOD = 01，将PWM时基设置为中心对齐计数模式。

在互补输出模式下，令PWMCON1中PWMSYM = 0，PWM波形发生模块工作在对称波形输出模式，PWM原始输出波形Px_O和Px1_O如图8-1-9所示。PWMEN置1后，Px_O和Px1_O波形保持为高，PWM计数器增计数时与PWMxDL/H匹配后Px_O和Px1_O波形将变低直至PWM计数器减计数时再次与PWMxDL/H匹配，如此往复。

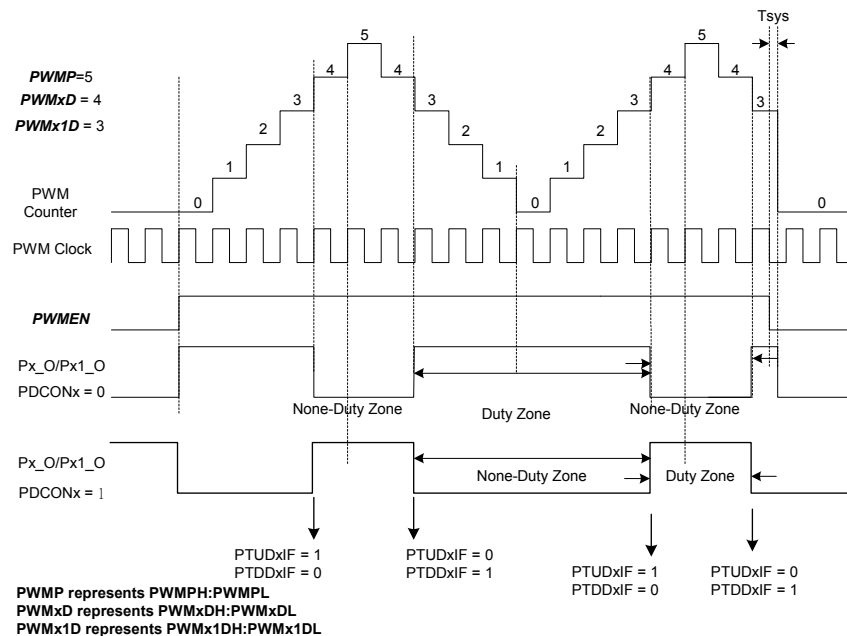


图8-1-9 中心对齐计数、互补输出模式下的PWM原始波形（非对称波形）

令PTCON中PTMOD = 01，将PWM时基设置为中心对齐计数模式。

在互补输出模式下，令PWMCON1中PWMSYM = 1，PWM波形发生模块工作在非对称波形输出模式，PWM原始输出波形Px_O和Px1_O如图8-1-9所示。PWMEN置1后，Px_O和Px1_O波形保持为高，PWM计数器增计数时与PWMxDL/H匹配后Px_O和Px1_O波形将变低直至PWM计数器减计数时与PWMx1DL/H匹配，如此往复。

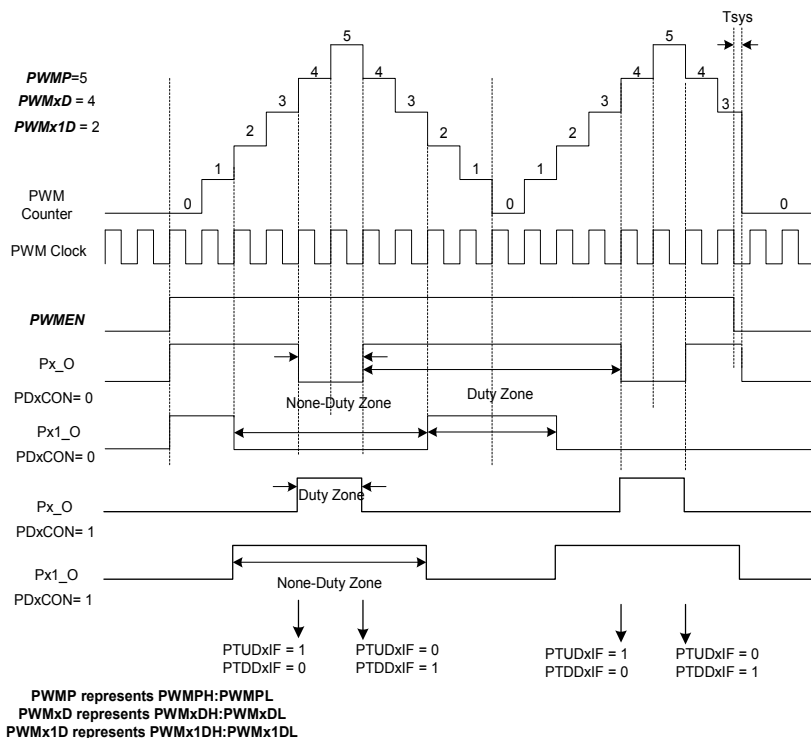


图8-1-10 中心对齐计数、独立输出模式下的PWM原始波形

独立输出模式下，PWM原始输出波形Px_O和Px1_O如图8-1-10所示。PWMEN置1后，Px_O和Px1_O波形变高，PWM计数器增计数时与PWMxDL/H匹配后Px_O波形将变低直至PWM计数器减计数时再次与PWMxDL/H匹配；PWM计数器增计数时与PWMx1DL/H匹配后Px1_O波形将变低直至PWM计数器减计数时再次与PWMx1DL/H匹配，如此往复。

单次计数模式下的PWM0/1/2原始波形

令PTCON中PTMOD = 10或11，将PWM时基设置为单次计数模式。在互补输出模式下，PWM原始输出波形Px_O和Px1_O如图8-1-11所示。PWMPEN置1后，Px_O和Px1_O波形变高，PWM计数器增计数时与PWMxDL/H匹配后Px_O和Px1_O波形将变低。

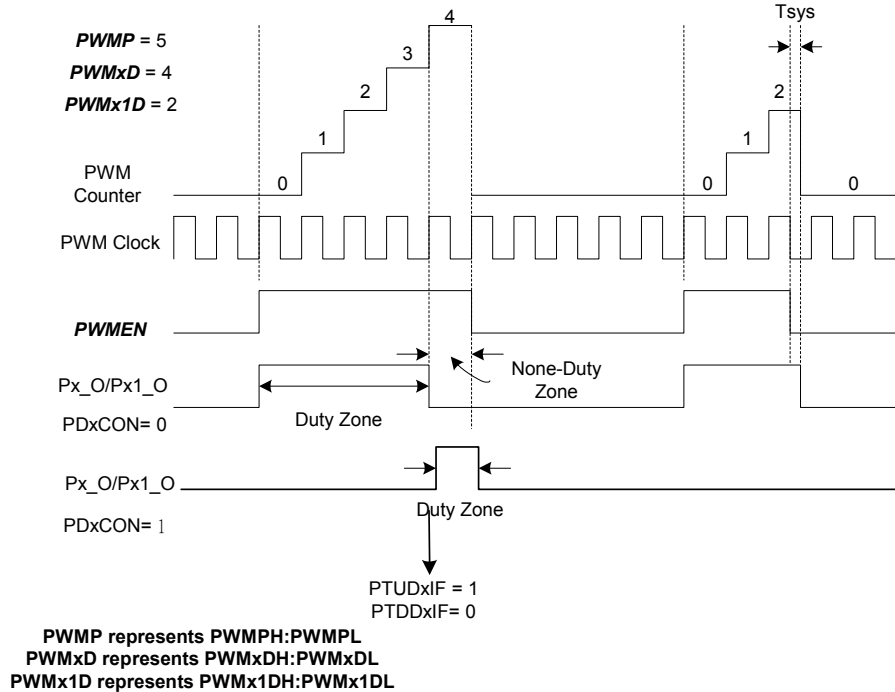


图8-1-11 单次计数、互补输出模式下的PWM原始波形

在独立输出模式下，PWMPEN置1后，Px_O和Px1_O波形变高(PDxCON=0)，PWM计数器增计数时与PWMxDL/H匹配后Px_O波形将变低；PWM计数器增计数时与PWMx1DL/H匹配后Px1_O波形将变低。

占空比定义及占空比寄存器的自动重载

如图8-1-6所示，6个占空比寄存器都带有对应的缓存寄存器（BUFFER），16位的PWM计数器实际上是与BUFFER比较从而控制PWM原始波形的变化。用户能够读写6个占空比寄存器，却不能操作其对应的缓存寄存器。结合图8-1-2和图8-1-6可以看出。

PWMCON2寄存器中的PDCONx（x=0-2）位段也有一个缓存寄存器（BUFFER），用户能够读写PDCONx位段但不能操作其缓存寄存器，如图8-1-6所示。

由于存在缓冲寄存器，和周期寄存器一样，对占空比定义和占空比寄存器的读写有特殊规定：

当PWMRLDEN寄存器的值设为0x55时，才允许对占空比定义和占空比寄存器进行修改，然后只有当PWMRLDEN寄存器中的值设为0xAA时，分频后的归零信号发生时才会将占空比定义和占空比寄存器锁存至对应的缓存寄存器（BUFFER）中，这样可以避免寄存器修改过程中发生重载，也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改，必须要有对PWMRLDEN寄存器的配合操作才能完成。

另外，在中心对齐模式下，和周期寄存器不同，占空比定义和占空比寄存器除了可以用归零信号重载外，还可以用周期匹配信号重载，如果将PWMCON3中PDLDEN位为1，则分频后的周期匹配信号也会将硬件自动将占空比定义和6个占空比寄存器锁存至对应的缓存寄存器中（称之为重载）。

因此在边沿计数模式和单次计数模式下，由于时基模块只有归零匹配信号，故PWM计数器每次复位为0时都将发生一次占空比定义和占空比寄存器的重载；中心对齐计数模式下，时基模块能产生归零匹配以及周期匹配信号，故PWM计数器每次复位为0以及与周期寄存器匹配时都可以重载占空比定义和占空比寄存器。

占空比中断

PWM0/1/2时基计数器向上计数时，不论在何种模式，当计数值与占空比寄存器PWMxDL/H匹配时能够令PWMINTF寄存器中PTUDxIF位置1，同时令PTDDxIF位清0，此时若PWMINTEN寄存器中的PTUDxIE位为1，则能够触发占空比中断，如图8-1-7/图8-1-9所示。在中心对齐计数模式中，PWM时基计数器向下计数时，当计数值与PWMxDL/H匹配时能够令PWMINTF寄存器中

PTDDxIF位置1，同时令PTUDxIF位清0，此时若PWMINTEN寄存器中的PTDDxIE位为1，则能够触发占空比中断，如图8-1-9所示。

注意：不论互补输出模式还是独立输出模式，占空比中断只是在PWM计数器与PWMxDL/H相关的比较时产生，与PWMx1DL/H相关比较匹配时不会改变占空比中断标志。

8.1.5 PWM死区与极性控制模块

如图8-1-1所示，波形发生器(Wave Generator)产生的PWM原始波形Px_O/Px1_O经过死区与极性控制模块(Dead&Polarity Logic)产生最终的PWM波形Px_D/Px1_D。带有死区与极性的PWM信号Px_D/Px1_D由输出控制模块(Output Control Logic)和故障检测模块(Fault Detect Logic)决定是否由引脚PWMx/PWMx1送出，(x = 0 - 2)。

当寄存器PWMDCON2中的DT[1:0] = 00时死区与极性控制模块的逻辑简图如图8-1-12所示。

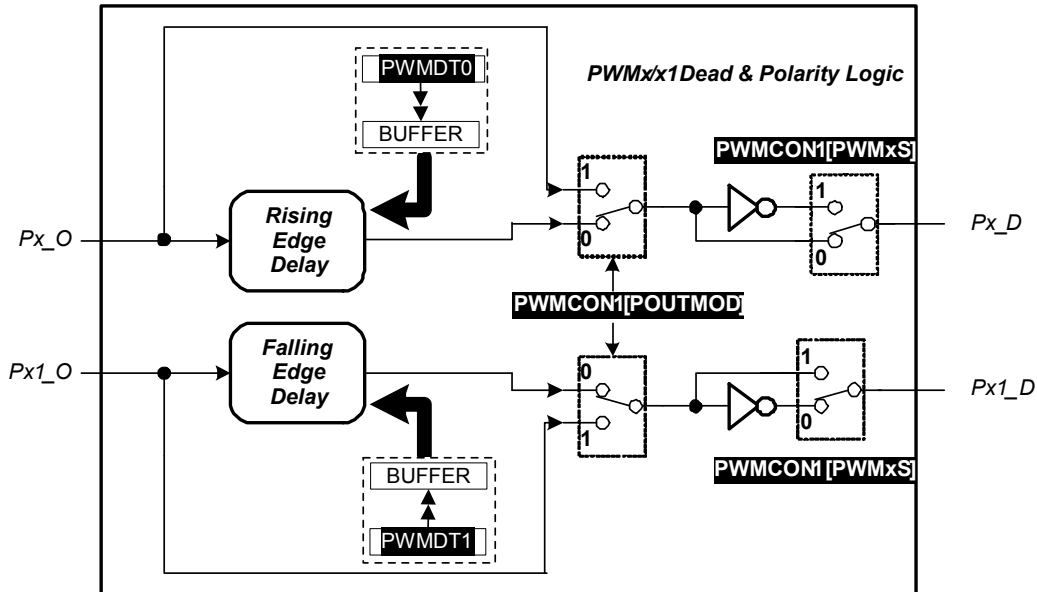


图8-1-12 死区与极性控制逻辑简图

PWM死区的实现

如图8-1-12所示，死区逻辑通过将PWM原始信号Px_O的上升和Px1_O的下降沿延迟一段时间而实现。PWMCON1寄存器中POUTMOD位为1时，PWM原始信号Px_O/Px1_O不会经过死区逻辑，即PWMx/PWMx1引脚输出为独立模式时，无死区；PWMCON1寄存器中POUTMOD位为0时，PWM原始信号Px_O/Px1_O将经过死区逻辑，即PWMx/PWMx1引脚为互补模式时，将引入死区。死区时间由寄存器PWMDT0/1决定：

上升沿延迟时间 (Rising Edge Delay) = PWMDT0 * T_{PWM CLOCK}

下降沿延迟时间 (Falling Edge Delay) = PWMDT1 * T_{PWM CLOCK}

注意：只在PWMCON2中的DT[1:0] = 00的情况如此，其他情况请参照PWMCON2寄存器部分的描述。

在不同的极性设置下，插入死区后的PWM波形也将不同，详见“不同极性设置下带死区的PWM波”章节。

注意：PWMDT0 = 0时，原始PWM波形Px_O的上升沿不会触发延时；PWMDT1 = 0时，原始PWM波形Px1_O的下降沿也不会触发延时。

PWM极性设置

为了有效说明PWM引脚上波形的极性，将一个PWM周期划分为“占空比时区 (Duty Zone)”与“非占空比时区 (None-Duty Zone)”，各种时基计数模式下，占空比时区和非占空比时区的划分如图8-1-7 - 图8-1-11所示。由寄存器PWMCON1中的位PWMxS/PWMx1S来决定占空比时区期间Px_D和Px1_D的电平。

PWMxS控制Px_D的极性。PWMxS = 0时，Px_D的占空比时区期间为高电平，非占空比时区期间为低电平；PWMxS = 1时，Px_D的占空比时区期间为低电平，非占空比时区期间为高电平。

PWMx1S控制Px1_D的极性。PWMx1S = 0时，Px1_D的占空比时区期间为低电平，非占空比时区期间为高电平；PWMx1S = 1时，Px1_D的占空比时区期间为高电平，非占空比时区期间为低电平。

结合图8-1-7 - 图8-1-11所示的PWM原始波形Px_O/Px1_O和图8-1-12可以清晰的反映死区与极性的实现。

PWM死区和死区方式选择位段的自动重载

PWM死区也和占空比、周期一样，有自己的缓冲寄存器（BUFFER），自动重载功能类似于占空比的自动重载，可以在周期匹配和归零信号发生时重载。对死区寄存器的修改也必须PWMRLDEN寄存器的值设为0x55时，然后只有当PWMRLDEN寄存器中的值设为0xAA时，重载生效。

死区的方式选择位段DT[1:0]同样也有缓冲寄存器（BUFFER），自动重载功能类似于占空比的自动重载，可以在周期匹配和归零信号发生时重载。对死区寄存器的修改也必须PWMRLDEN寄存器的值设为0x55时，然后只有当PWMRLDEN寄存器中的值设为0xAA时，重载生效。

不同极性设置下带死区的PWM波形

图8-1-13给出了一对在DT[1:0] = 00时的PWM输出Px_D/Px1_D设置为互补输出模式，所有的极性组合下带有死区的波形。

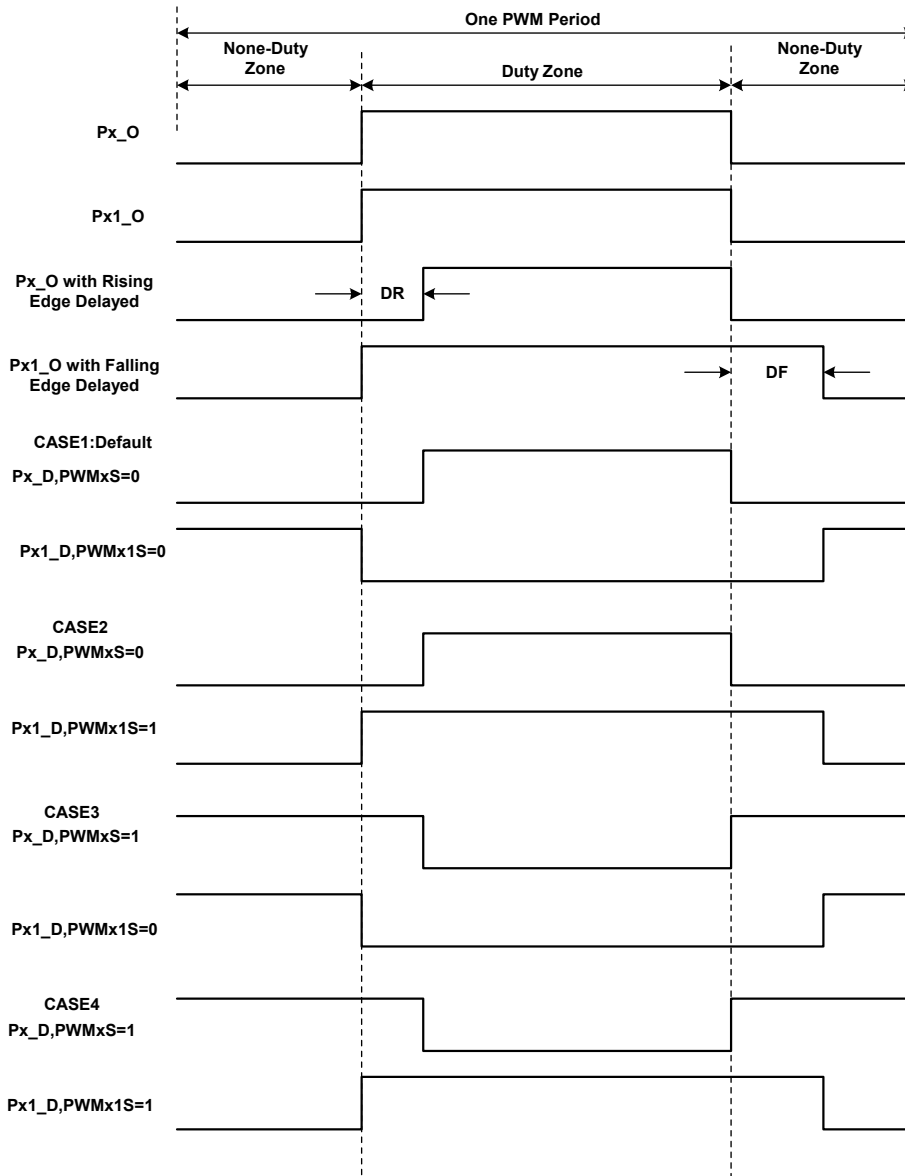


图8-1-13 不同极性设置下的死区波形

注：图中的DR表示上升沿死区，DF表示下降沿死区。上图是DT[1:0] = 00时的情况，DT[1:0]为其他值时的情况，可参考PWMCON2寄存器部分的说明，这里就不再多阐述。

PWM0/1/2输出控制与故障检测

PWM输出控制与故障检测模块（Output Control Logic & Fault Detect Logic）用于决定最终的PWM波形Px_D是否在PWMx引脚上输出、Px1_D是否在PWMx1引脚上输出。输出控制与故障检测模块的逻辑简图如图8-1-14所示。

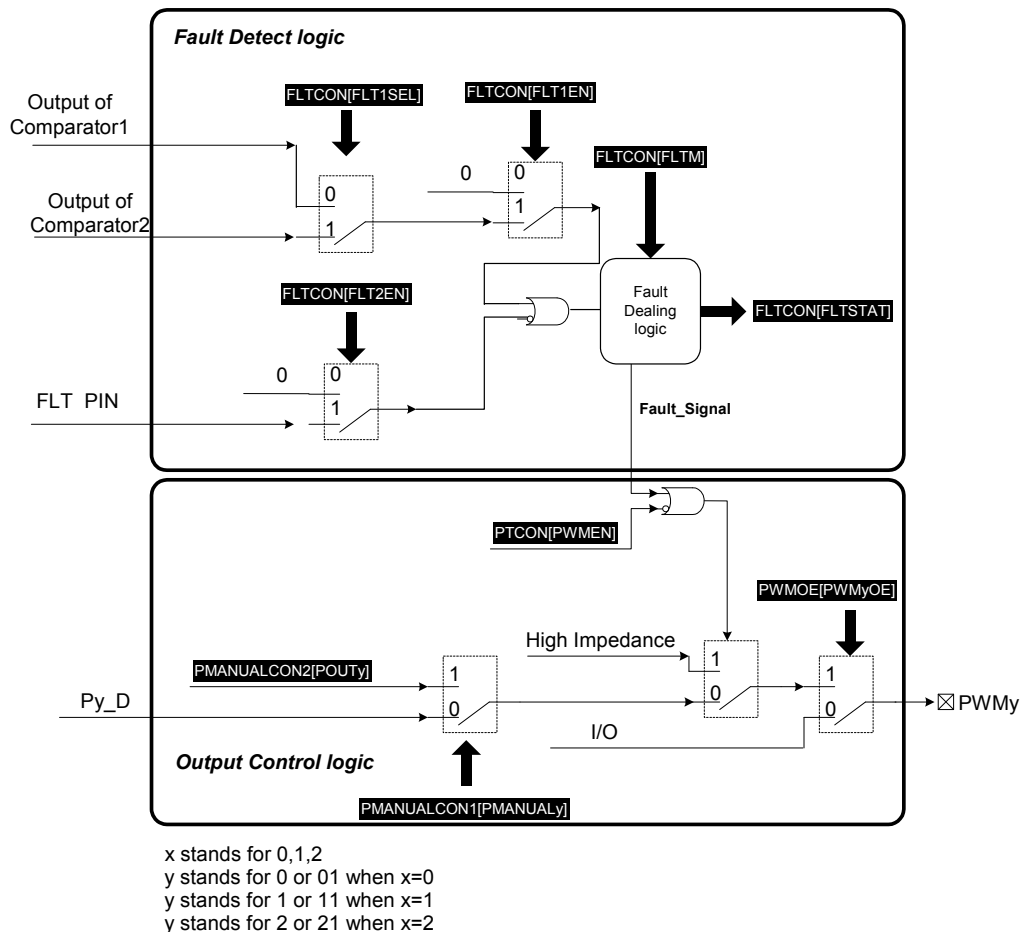


图8-1-14 输出控制与故障检测逻辑简图

8.1.6 故障检测模块

PWM模块中有一个故障检测模块（Fault Detection logic），当FLTCON寄存器中FLT1EN位或FLT2EN位为1时，使能该模块功能，若FLT1EN和FLT2EN都为0，则禁止该模块功能。PWM故障检测主要目的是：当故障发生时（比如过流）可以切断PWM的输出，进入无效驱动状态（输出高阻态），进而达到保护外部功率器件的目的，由于它是由硬件控制，因此响应速度非常快。故障检测模块的逻辑简图如图8-1-14所示，结合输出控制模块（Output Control Logic）能够实现前述的保护功能。

故障信号输入源的选择

可以选择比较器（1/2）的输出或FLT引脚输入作为故障检测输入信号，如图8-1-14所示。无论故障信号来自比较器1/2输出或FLT，高电平为有效故障信号，即比较器输出由低变高时将自动切断PWM的输出。

如图8-1-14所示，若FLT1EN = 1，则比较器1或2的输出会作为故障检测输入信号。一旦比较器的输出变高，并保持滤波时间（下节介绍），则六路PWM引脚PWMx/PWMx1立即输出高阻态（x = 0 - 2）。同理，若FLT2EN = 1，则FLT引脚输入作为故障检测输入信号，一旦FLT引脚输入变高（FLT2S = 0时），并保持滤波时间，则六路PWM引脚PWMx/PWMx1立即输出高阻态（x = 0 - 2）。如果FLT2EN = 0，则禁止FLT引脚的故障检测功能，FLT引脚作为普通IO使用（但比较器的故障检测功能不受影响）。

当使用PWM模块来控制图8-1-15所示的逆变桥，六个管子为高电平驱动功率管：设置PWMxS/PWMx1S = 0（x = 0 - 2），则PWM0/1/2的输出在占空比时区期间为高电平，PWM01/11/21的输出在占空比时区期间为低电平（见“PWM极性的设置”章节），设置PWMCON1寄存器中POUTMOD位为0，即PWM0与PWM01互补输出、PWM1与PWM11互补输出、PWM2与PWM21互补输出。使能故障检测后，若发生故障，则六路PWM立即输出高阻态。

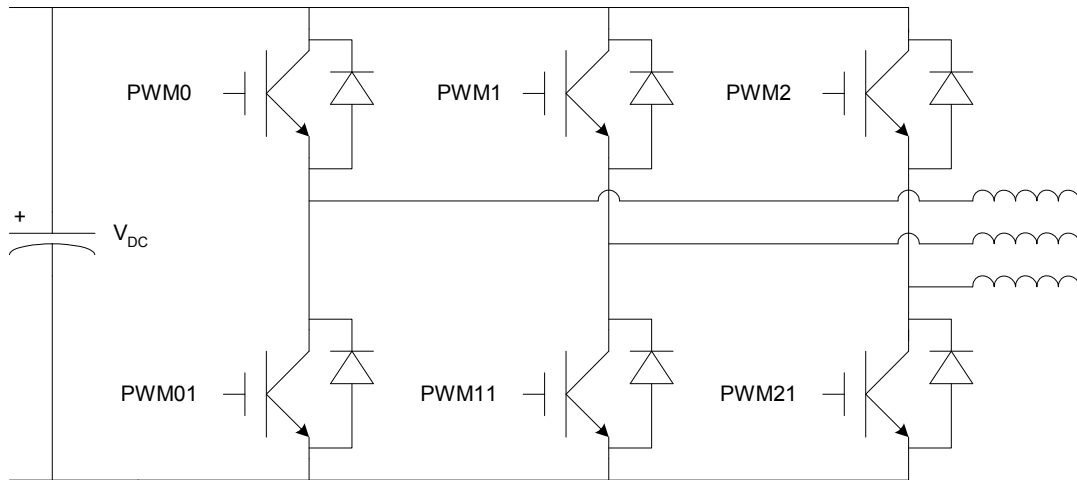


图8-1-15 PWM控制逆变桥示例图

FLT引脚上的信号与比较器输出信号的滤波

当FLT2EN = 1时，FLT引脚的输入信号会作为故障检测输入信号使用，此时可以通过设置寄存器FLTCON中的FLT2DEB[1:0]位来调整对此信号的滤波时间。FLT2DEB[1:0] = 0时，无滤波作用。当FLT2S = 1时，故障检测输入信号的电平由低变高时立即触发保护；FLT2DEB[1:0] = 1 - 3时，可以将滤波时间设置为1 μ s、4 μ s、16 μ s共3档，故障检测输入信号由低变高时，高电平需要至少保持FLT2DEB[1:0]定义的时间长度，故障检测模块才认为故障检测输入信号的电平变高了，从而触发保护；故障检测输入信号的电平由高变低时，低电平需要至少保持FLT2DEB[1:0]定义的时间长度，故障检测模块才认为FLT引脚上的有效故障电平消失了，六路PWM的输出根据保护模式（下节介绍）动作。当FLT2S = 0时，FLT故障检测信号变为低电平有效，情况与FLT2S = 1时相反。根据功率管的特性，恰当的设置滤波时间可以滤除故障检测输入信号上的噪声。

FLT1EN = 1时，可选比较器1、2的输出信号作为故障检测输入信号使用，当比较器输出由0变1时触发保护，此时比较器1、2的输出滤波时间和算法分别在寄存器CMP1CON和CMP2CON2中设置。（详见运算放大器和比较器模块章节）

保护模式

有两种故障检测模式：锁存模式与逐次模式。

(1) 锁存模式

设置FLTCON寄存器中FLTM位为0将使能锁存模式。当故障检测模块探测到有效故障信号，六路PWM引脚PWMx/PWMx1立即切换为高阻态。

若有效故障信号没有消失，六路PWM引脚PWMx/PWMx1将保持输出高阻态（x = 0 - 2），软件无法清除FLTSTAT标志位，六路PWM引脚无法恢复正常输出。

当有效故障信号消失后，六路PWM引脚PWMx/PWMx1（x = 0 - 2）亦不会恢复正常输出，只有软件将FLTSTAT位清0后，PWMx/PWMx1才会在FLTSTAT被清零之后的最近一次PWM计数器归0或与周期寄存器PWMPH/LH的值匹配时恢复正常输出（边沿对齐模式中，在PWM计数器归0时恢复输出；中心对齐模式中，在PWM计数器与周期寄存器PWMPH/LH匹配或归0时恢复输出）。

(2) 逐次模式

设置FLTCON寄存器中FLTM位为1将使能逐次模式。在逐次模式下，PWMx/PWMx1引脚是否输出PWM波形直接由比较器滤波后的输出信号控制。FLTSTAT位也都由滤波之后的故障信号直接控制，若滤波之后的故障信号为高，则FLTSTAT位为1；若滤波之后的故障信号为低，则FLTSTAT位为0。

当故障检测模块探测到有效故障信号，六路PWM引脚PWMx/PWMx1立即输出高阻态（x = 0 - 2），FLTSTAT位亦会被硬件置1。

若有效故障信号没有消失，六路PWM引脚PWMx/PWMx1将保持输出高阻态（x = 0 - 2），软件无法使得六路PWM引脚恢复正常输出。

当有效故障信号消失后，六路PWM引脚PWMx/PWMx1（x = 0 - 2）会在有效故障信号消失后的最近一次PWM计数器归0或与周期寄存器PWMPH/LH的值匹配时自动恢复正常输出，FLTSTAT位也将归0。

8.1.7 PWM输出控制模块

PWM输出引脚与IO功能复用

PWM引脚PWMx/PWMx1与IO功能复用($x=0-2$)，由寄存器PWMOE控制。令PWMOE中的PWMxOE位为1，可以将PWMx引脚设置为PWM输出功能，反之为IO功能；令PWMOE中的PWMx1OE位为1，可以将PWMx1引脚设置为PWM输出功能，反之为IO功能。具体如图8-1-14所示。

手动控制PWM输出

当PWMxOE/PWMx1OE位为1，且PWMEN位为1时，若PMANUALCON1寄存器中的PMANUALx/PMANUALx1位为1，则PWMx/PWMx1引脚将输出PMANUALCON2寄存器中POUTx/POUTx1位的值；若PMANUALCON1寄存器中的PMANUALx/PMANUALx1位为0，则PWMx/PWMx1引脚将输出PWM波形Px_D/Px1_D。此功能和PORT口的输出功能类似，唯一的区别是，在手动控制PWM输出模式下，其输出会受到PWM故障检测输入信号控制，而在I/O输出模式下，不受该信号控制。具体如图8-1-14所示。

更改PMANUALCON1和PMANUALCON2寄存器时，PWMx/PWMx1引脚输出的改变与系统时钟同步，输出状态改变立即生效。

注意：无论PMANUALx/PMANUALx1是否为1，当寄存器PWMOE中的PWMxOE/PWMx1OE为0时，PWMx/PWMx1引脚都将成为普通IO口。

时基停止时PWM引脚的输出

PWMOE中PWMEN位为0时，时基模块将停止工作，同时若PWMOE中PWMxOE/PWMx1OE位为1，则PWMx/PWMx1引脚将输出高阻态。具体如图8-1-14所示。

8.1.8事件触发功能

PWM时基模块产生的分频后的归零信号(Scaled Zero Match)和分频后的周期匹配信号(Scaled Period Match)可以用来触发启动ADC或者Timer2。

PWM计数器触发Timer2

若ZETIM = 0，则PWM模块不会自动启动Timer2。若寄存器PWMCON2中的ZETIM位为1且Timer2工作在自动触发模式(TRG = 1)，如果是中心对齐模式，则分频后的周期匹配信号可以自动启动Timer2，如果是边沿对齐模式或单次模式，则是分频后的归零信号启动Timer2；若寄存器PWMCON2中的ZETIM位为10且Timer2工作在自动触发模式(TRG = 1)，如果是中心对齐模式，则分频后的归零匹配信号可以自动启动Timer2，如果是边沿对齐模式或单次模式，则是分频后的归零信号启动Timer2；寄存器PWMCON2中的ZETIM位为11且Timer2工作在自动触发模式(TRG = 1)，如果是中心对齐模式，则分频后的周期匹配和归零匹配信号都可以自动启动Timer2，如果是边沿对齐模式或单次模式，则是分频后的归零信号启动Timer2。见图8-1-4。

PWM0/1/2计数器触发ADC

若寄存器PWMCON2中的ZEAD = 1，则分频后的归零信号可以自动触发一次ADC序列的转换(如果ADC模块中的ADCON1寄存器的ADON位和PWMTRGEN位被允许，将会启动一次序列转换)；若ZEAD = 0，则归零信号不会自动触发ADC序列。若寄存器PWMCON2中的PEAD = 1，则分频后的周期匹配信号可以自动触发一次ADC序列的转换(如果ADC模块中的ADCON1寄存器的ADON位和PWMTRGEN位被允许，将会启动一次序列转换)；若PEAD = 0，则周期匹配信号不会自动触发ADC序列。

若寄存器PWMCON2中的CMPx = 1 ($x=0-1$)，则在PWM计数期间当计数器值与PWMCMPx ($x=0-1$)设定值相等时可以自动触发一次ADC序列的转换(如果ADC模块中的ADCON1寄存器的ADON位和PWMTRGEN位被允许，将会启动一次序列转换)；CMPx = 2 ($x=0-1$)，则仅在PWM增计数期间当计数器值与PWMCMPx ($x=0-1$)设定值相等时可以自动触发一次ADC序列的转换(如果ADC模块中的ADCON1寄存器的ADON位和PWMTRGEN位被允许，将会启动一次序列转换)；CMPx = 3 ($x=0-1$)，则仅在PWM减计数期间当计数器值与PWMCMPx ($x=0-1$)设定值相等时可以自动触发一次ADC序列的转换(如果ADC模块中的ADCON1寄存器的ADON位和PWMTRGEN位被允许，将会启动一次序列转换)。PWM比较寄存器触发AD信号也受后分频系数控制

PWMCMPx寄存器自动重载

和周期及占空比寄存器一样，比较值寄存器也有对应的缓存寄存器(BUFFER)。

当PWMRLDEN寄存器的值设为0x55时，才允许对比较值寄存器进行修改，然后只有当PWMRLDEN寄存器中的值设为0xAA时，分频后的归零信号发生时才会将比较值寄存器锁存至对应的缓存寄存器(BUFFER)中，这样可以避免寄存器修改过程中发生重载，也避免寄存器修改立即生效可能会出现毛刺。因此对上述寄存器的修改，必须要有对PWMRLDEN寄存器的配合操作才能完成。

另外，在中心对齐模式下，比较值寄存器除了可以用归零信号重载外，还可以用周期匹配信号重载，如果将PWMCON3中PDLLEN位为1，则分频后的周期匹配信号也会将硬件自动将比较值寄存器锁存至对应的缓存寄存器中(称之为重载)。

因此在边沿计数模式和单次计数模式下，由于时基模块只有归零匹配信号，故PWM计数器每次复位为0时都将发生一次比较值寄存器的重载；中心对齐计数模式下，时基模块能产生归零匹配以及周期匹配信号，故PWM计数器每次复位为0以及与周期寄存器匹配时都可以重载比较值寄存器。

比较器事件触发设置位PWMCON2的CMP1/CMP2的重载也受归零信号或周期匹配信号控制。

PWM模块的中断总汇

PWM模块的中断包括时基模块产生的PWM计数器归零中断、PWM计数器值与周期寄存器匹配产生的周期中断（详见“时基中断”章节）以及PWM计数器值与占空比寄存器PWMxDL/H（ $x = 0 - 2$ ）匹配时产生的占空比中断（详见“占空比中断”章节）。这些中断共享一个位于0063H地址的中断向量。

8.1.9 注意事项

边沿对齐计数模式中占空比寄存器值为0或周期值

边沿对齐计数模式中，若占空比寄存器为0或周期值，则根据不同的极性设置，PWMx/PWMx1应输出占空比为0%或100%的波形。令PWMOE = 0xBF、PWMDT0L/H = 0、PWMDT1L/H = 0在边沿对齐时基计数、互补输出模式下，不同占空比值的PWM引脚输出的波形、中断标志触发位置、事件触发位置、寄存器重载位置如图8-1-16 - 图8-1-18所示。

注意：PWM时基Enable时刻，虽然时基计数器为0，但不会产生归零（zero match）信号，因此不会置PWMZIF标志，也不会产生事件触发信号，而寄存器重载信号在PWMMEN = 0时是立即发生的。

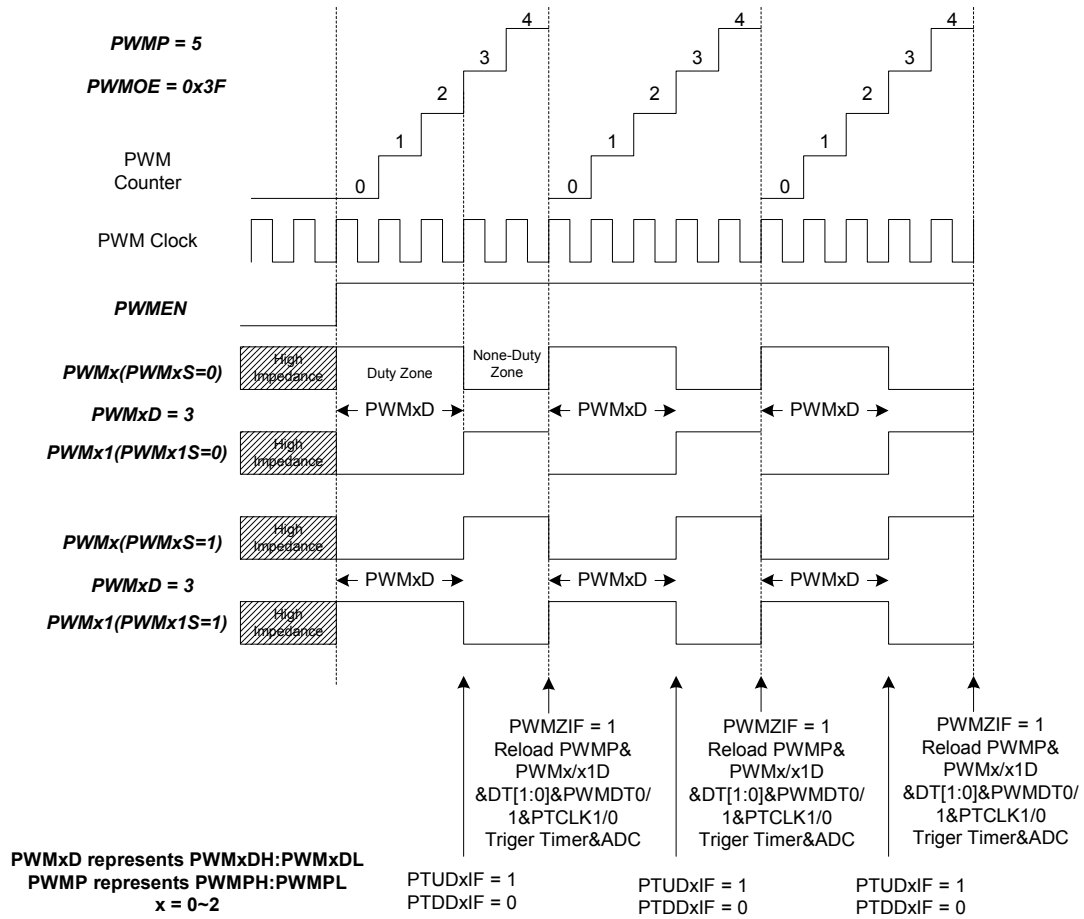


图8-1-16 边沿对齐计数、互补输出模式

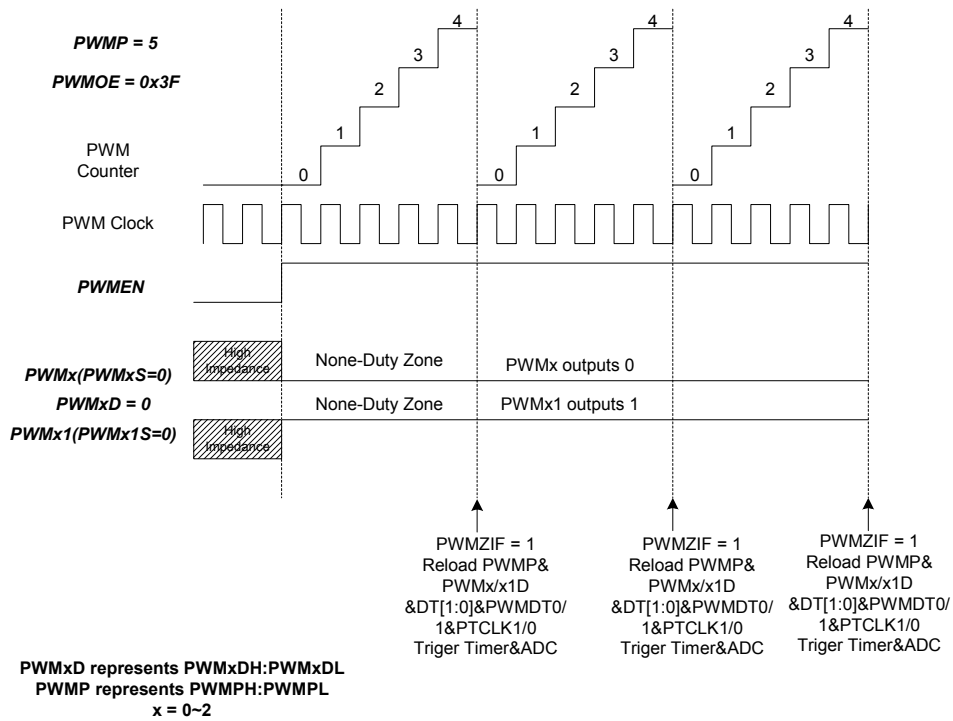


图8-1-17 边沿对齐计数、互补输出模式，占空比 = 0

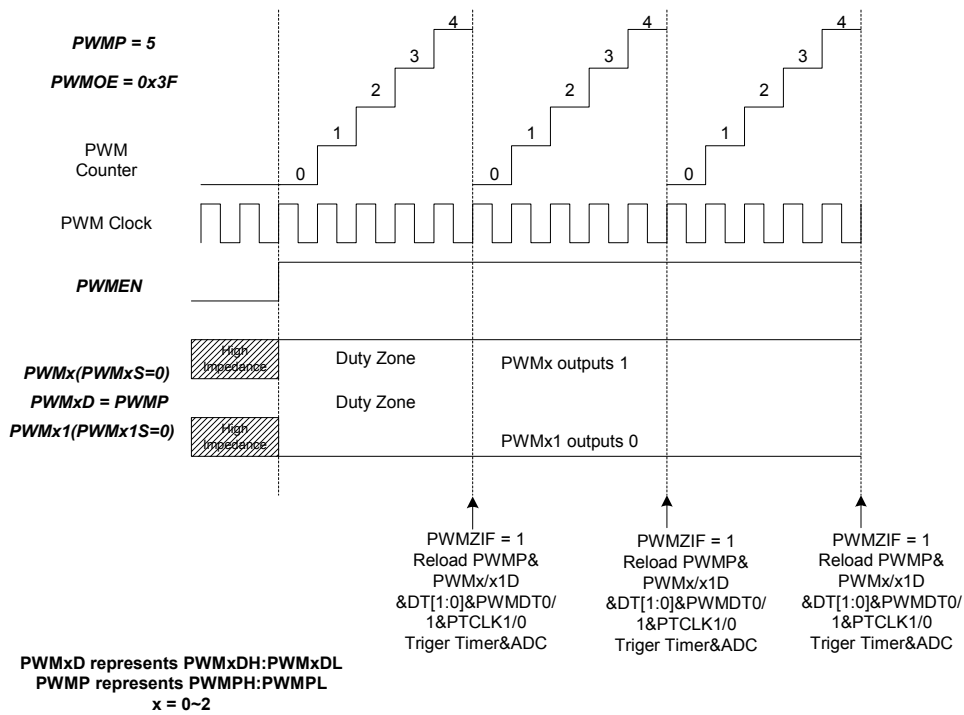


图8-1-18 边沿对齐计数、互补输出模式，占空比 = 周期寄存器

中心对齐计数模式中占空比寄存器值为0或周期值

中心对齐计数模式中，若占空比寄存器为0或周期值，则根据不同的极性设置，PWMx/PWMx1应输出占空比为0%或100%的波形。令PWMOE = 0xBF、PWMDT0L/H = 0、PWMDT1L/H = 0在边沿对齐时基计数、互补输出模式下，不同占空比值的PWM引脚输出的波形、中断标志触发位置、事件触发位置、寄存器重载位置如图8-1-19 - 图8-1-21所示。

特别注意：在中心对齐计数模式中，当占空比寄存器为0或周期值时，占空比较匹配中断标志不会置起。

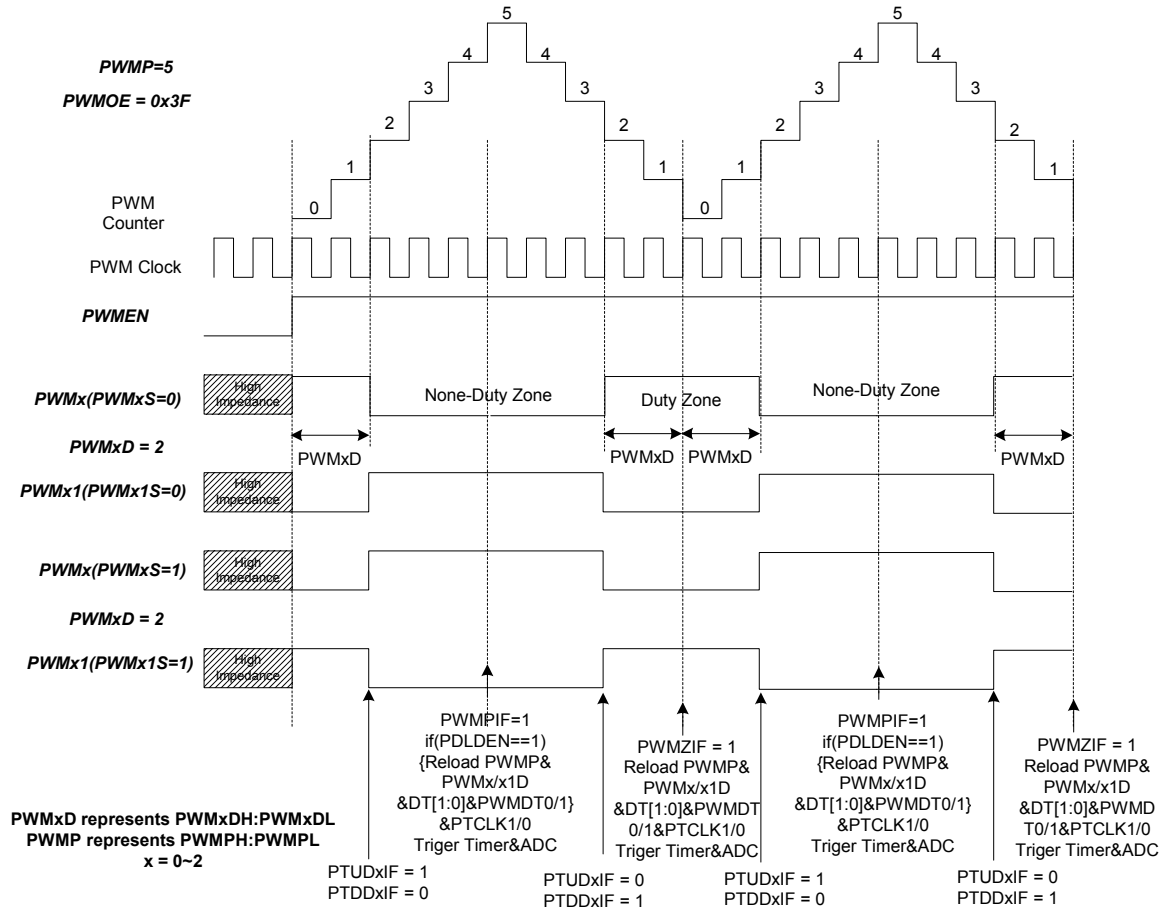


图8-1-19 中心对齐计数、互补输出模式

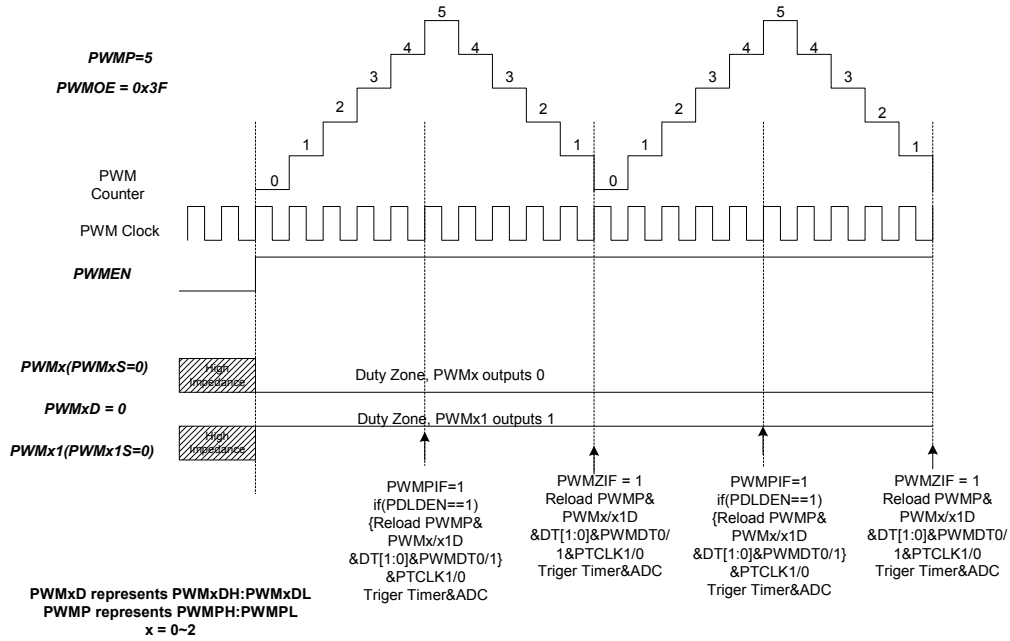


图8-1-20 中心对齐计数、互补输出模式、占空比 = 0

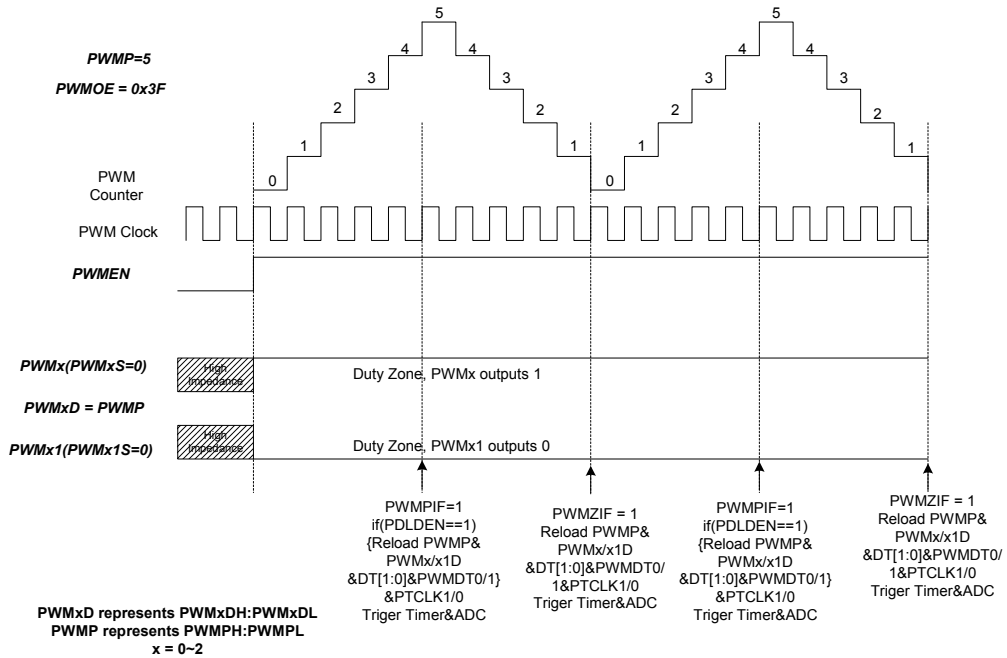


图8-1-21 中心对齐计数、互补输出模式、占空比 = 周期

周期寄存器值为0

不管时基计数器在何种模式，若周期寄存器为0，则根据不同的极性设置，PWMx/PWMx1应输出占空比为0%的波形。此时，所有中断标志都不会产生，也不会产生事件触发信号，但重载信号会产生。

仿真状态的PWM输出

G80F980支持在线仿真，在仿真状态运行和正常状态运行没有区别，如果在仿真状态停止运行或单步运行，PWM输出会切换为高阻态，保证不会误触发外部功率管。

进入IDLE模式，PWM0/1/2的波形输出状态

进入IDLE模式后，PWM输出口不受影响，比如PWM0口原先输出1K的方波，进入IDLE模式后会继续输出1K的方波，但是如果设置了PWM0/1/2中断，会唤醒IDLE模式，继续执行IDLE后面的程序。

进入STOP模式，PWM0/1/2的波形输出状态

进入STOP模式后，PWM输出口会输出高阻状态，即使设置了PWM0/1/2中断，也不能唤醒掉电模式，掉电模式必须通过其他方式唤醒，具体参考电源管理章节。

为避免在从STOP模式下恢复后PWM输出波形产生非预期效果，建议进入掉电模式前关闭PWM模块（PWMEN=0）

8.1.10 代码示例

以下是输出PWM波形的一个简单示例程序：

```
#include "G80F980.h"/*代码选项选取单时钟源内部12MHZRC*/
void Pwm_Isr(void) interrupt 12 using 1
{
    PWMINTF = 0;
    PWMRLDEN = 0x55; //每次中断将PWM0占空比加1.
    PWM0D16++;
    PWMRLDEN = 0xAA;
}
void System_Initial()
{
    CLKCON=0x00;
    EA=1;
    IEN1=0x20;
}
void Pwm_Initial()
{
    PWMRLDEN=0x55; //允许修改PWM受保护寄存器
    PWMOE=0x3f; //将6个I/O口设置为PWM输出口
    PTCON=0x00; //PWM输出为边沿对齐模式，且无分频
    PWMCON1=0x00; //PWM输出互补模式（PWM0与PWM01波形互补）
    PWMP16=6000; //PWM的周期设为6000sysclock
    PWM0D16=0; //PWM0占空比设置为0sysclock
    PWM1D16=1000; //PWM1占空比设置为1000sysclock
    PWM2D16=3000; //PWM2占空比设置为3000sysclock
    PWMINTEN=0x40; //PWM归零中断允许
    PWMOE|=0x80; //打开PWM总时基
    PWMRLDEN=0xaa; //将PWM带缓存的寄存器值载入（周期，占空比，死区等）
}

void main()
{
    System_Initial();
    Pwm_Initial();
    while(1);
}
```

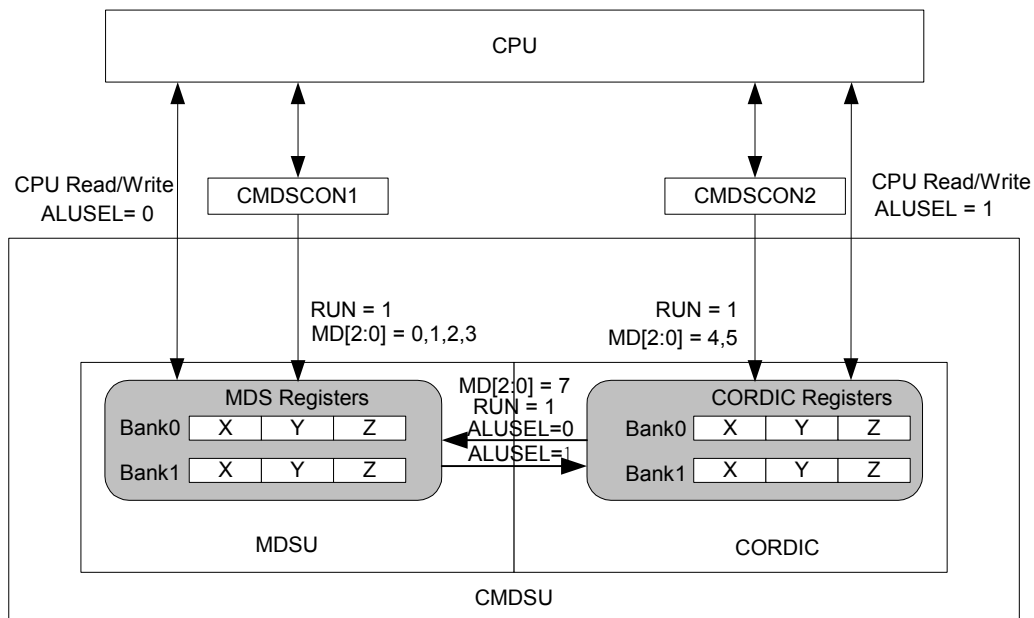
8.2 协处理器单元 (CORDIC+MDSU)

8.2.1 特性

- 包括CORDIC运算单元和乘除法移位单元，两个运算单元可同时执行运算
- 硬件实现在1周期内完成有符号/无符号16-bit X 16-bit乘法
- 硬件实现在8周期内完成有符号/无符号32-bit ÷ 16-bit除法
- 硬件实现在1周期内完成有符号/无符号32-bit数据最多31次移位
- 硬件实现基本CORDIC算法（圆函数）
- 基本CORDIC算法（圆函数）支持旋转模式和向量模式，固定17个周期完成
- CORDIC计算采用预处理及后处理逻辑，可支持圆函数的全范围取值
- CORDIC计算采用16位数据访问宽度，24位内核数据计算宽度，确保高精度
- 操作数寄存器、乘除法移位单元内核计算/结果寄存器与CORDIC内核计算/结果寄存器相互独立
- 操作数寄存器、内核计算/结果寄存器可在两套寄存器组之间切换
- 可选择读取内核计算/结果寄存器或操作数寄存器

协处理器单元包含乘/除法、移位单元及CORDIC运算单元，可完成16-bit X 16-bit数乘法、32-bit ÷ 16-bit数除法、32-bit数据左/右移位以及基本CORDIC算法（圆函数）运算。由于运算过程由硬件完成，转化速率快，为程序运行节省了大量的时间。

8.2.2 协处理器模块图



协处理器 (CORDIC+MDSU) 模块图

8.2.3 寄存器

CMDSU模块所使用寄存器如下：

功能	名称	寄存器描述
操作控制寄存器	CMDSCON1	用于选择进行乘/除法操作/移位/CORDIC/清除/更新操作数寄存器操作、启动运算、乘/除算法符号使能
	CMDSCON2	运算结果标志，清除/更新操作数寄存器设置
操作数寄存器	OPRDZH和 OPRDZL	用于存放除法操作的被除数的高16位；或者32-bit移位操作数的高16位；运算完成时作结果寄存器的高16位；或者CORDIC运算中的Z操作数

	OPRDYH和OPRDYL	用于存放16-bit乘法操作数之一或者除法操作中 被除数的低16位；或者32-bit移位操作数 的低16位；运算完成时作结果寄存器的低16位； 或者CORDIC运算中的Y操作数
	OPRDXH和OPRDXL	用于存放16-bit乘法操作数之一或者除法操作 中的除数或移位次数；运算完成时作除法的余数 寄存器；或者CORDIC运算中的X操作数

Table 8.29 协处理器操作控制寄存器1

C1H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMDSCON1	MD2	MD1	MD0	SIGNEN	OPRDY	OPRDZ	RUN	
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-5	MD[2:0]	<p>协处理器运算选择位</p> <p>000: 执行乘法操作（默认） 001: 执行除法操作 010: 执行左移操作 011: 执行右移操作 100: 执行CORDIC旋转模式操作 101: 执行CORDIC向量模式操作 110: 执行对指定操作数寄存器更新操作 111: 执行对指定操作数寄存器清0操作</p> <p><i>注：执行更新或清0操作数寄存器时，由OPRDY/OPRDZ位指定对相应操作数寄存器有效；当执行清零时由Bank及ALUSEL位确定清零相应的计算/结果寄存器；当执行更新时由Bank及ALUSEL位确定从指定运算单元的计算/结果寄存器更新到另一个运算单元的相应寄存器。</i></p>
4	SIGNEN	<p>乘除法移位算法符号使能位</p> <p>0: 进行无符号数操作（默认） 1: 进行有符号数操作</p> <p><i>注：只对乘除法及移位操作时有效，CORDIC运算操作数都按有符号格式操作</i></p>
3	OPRDY	<p>对操作数X寄存器执行更新和清0操作</p> <p>0: 无效 1: 有效</p>
2	OPRDZ	<p>对操作数Y寄存器执行更新和清0操作</p> <p>0: 无效 1: 有效</p>
1	RUN	<p>对操作数Z寄存器执行更新和清0操作</p> <p>0: 无效 1: 有效</p>
0		<p>运算启动标志位，只能硬件清0，软件置1</p> <p>0: 运算已结束或计算中 1: 启动乘/除法、移位运算、操作数寄存器更新或清0或CORDIC操作，启动后立即被清零</p>

Table 8.29 协处理器操作控制寄存器2

BBH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMDSCON2	BSYMDS	BSYCDC	DIVERR	OVF	-	XYMRS	Bank	ALUSEL
读/写	读	读	读	读	-	读/写	读/写	读/写

复位值 (POR/WDT/LVR/PIN)	0	0	0	0	-	0	0	0
位编号	位符号	说明						
7	BSYMDS	乘除法移位单元计算标志 0: 乘除法移位单元空闲 1: 乘除法移位单元运算中						
6	BSYCDC	CORDIC运算单元计算标志 0: CORDIC运算单元空闲 1: CORDIC运算单元运算中						
5	DIVERR	除法运算异常标志位, 硬件置位, 并在下一次运算启动时硬件自动清0 0: 无除数为0的情况发生 1: 有除数为0的情况发生						
4	OVF	CORDIC运算溢出标志位 0: 无溢出的情况发生 1: 有溢出的情况发生						
2	XYMRS	XY幅度调节选择位 0: 最后一次迭代后, X和Y不变作为最终结果 1: 最后一次迭代后, X和Y除以2作为最终结果 注: XYMRS设置只对CORDIC运算有效, 正确设置XYMRS位, 可避免结果数据溢出。						
1	Bank	操作数计算/结果寄存器组选择位 0: 选择寄存器组0作为操作数及计算/结果寄存器 1: 选择寄存器组1作为操作数及计算/结果寄存器						
0	ALUSEL	读/写/更新/清零操作运算单元选择位 0: 选择乘除法移位单元作为读写更新及清零的目标运算单元 1: 选择CORDIC单元作为读写更新及清零的目标运算单元 注: 执行乘除法移位及CORDIC运算操作, 直接由MD[2:0]确定执行的运算单元, 与ALUSEL设置值无关。						

Table 8.30 操作数/结果Y寄存器

C2H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDYL	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
C3H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDYH	Y15	Y14	Y13	Y12	Y11	Y10	Y9	Y8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						

7-0 7-0	Y[15:0]	<p>作16bit X 16bit乘法操作时 (MD[2:0] = 000) OPRDYL/H存放16bit乘数, Y15为MSB, Y0为LSB</p> <p>作32bit ÷ 16bit除法操作时 (MD[2:0] = 001) OPRDYL/H存放32bit被除数的低16位, Y15为MSB, Y0为LSB</p> <p>作32bit位移操作时 (MD[2:0] = 010/011) OPRDYL/H存放32bit位移操作数的低16位, Y15为MSB, Y0为LSB</p> <p>作CORDIC运算时时 (MD[2:0] = 100/101) OPRDYL/H存放16bit Y操作数, Y15为MSB, Y0为LSB</p> <p>运算完成时 (BSYMDS = 0) OPRDYL/H存放32bit结果的低16位或Y结果, Y15为MSB, Y0为LSB</p>
------------	---------	--

Table 8.31 操作数/结果Z映射寄存器

C4H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDZL	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
C5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDZH	Z15	Z14	Z13	Z12	Z11	Z10	Z9	Z8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0 7-0	Z[15:0]	<p>作32bit ÷ 16bit除法操作时 (MD[2:0] = 000) OPRDZL/H存放32bit被除数的高16位, Z15为MSB, Z0为LSB</p> <p>作32bit位移操作时 (MD[2:0] = 010/011) OPRDZL/H存放32bit位移操作数的高16位, Z15为MSB, Z0为LSB</p> <p>作CORDIC运算时时 (MD[2:0] = 100/101) OPRDZL/H存放16bit Y操作数, Z15为MSB, Z0为LSB</p> <p>运算完成时 (BSYMDS = 0) OPRDZL/H存放32bit结果的高16位或Z结果, Z15为MSB, Z0为LSB</p>

Table 8.32 操作数/结果X寄存器

C6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDXL	X7	X6	X5	X4	X3	X2	X1	X0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
C7H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPRDXH	X15	X14	X13	X12	X11	X10	X9	X8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						

7-0 7-0	X[15:0]	<p>作16bit X 16bit乘法操作时 (MD[2:0] = 000) 存放另一个16bit乘数, X15为MSB, X0为LSB</p> <p>作32bit ÷ 16bit除法操作时 (MD[2:0] = 001) 存放16bit除数, X15为MSB, X0为LSB</p> <p>作32bit位移操作时 (MD[2:0] = 010/011) 存放位移的次数 (此时只有X[4:0]位有效) 例: X[4:0] = 1; 移位1次</p> <p>作CORDIC运算时 (MD[2:0] = 100/101) OPRDXL/H存放16bit X操作数, X15为MSB, X0为LSB</p> <p>运算完成时 (BSYMDS = 0) 存放16bit余数或X结果, X15为MSB, X0为LSB</p>
------------	---------	---

8.2.4 操作数/结果寄存器组切换

CMDSDU模块两个运算单元的操作数/计算结果寄存器 (除CMDSDCON1/2以外的OPRDZH-OPRDZL, OPRDYH-OPRDYL, OPRDXH-OPRDXL) 都有两组独立寄存器。协处理器根据CMDSDCON2中Bank位指定由那一组寄存器作为操作数/计算/结果寄存器。“0”表示Bank0的操作数/内核计算结果寄存器,“1”表示访问Bank1操作数/内核计算结果寄存器。当选定Bank确定某一组寄存器时,所有读写/计算/清零/更新等操作都针对该组寄存器进行,用户可以在中断程序和其他程序中指定不同的Bank组,使得在中断和其他程序中都可以使用协处理器功能 (中断入口和出口需对Bank值进行保存和恢复)。

8.2.5 操作数/结果寄存器读写

CMDSDU模块两个运算单元的操作数/结果寄存器都映射到相同地址,用户对操作数/结果寄存器的读写可通过CMDSDCON2的ALUSEL进行指定。除了由Bank值确定读取相应组寄存器外,还由CMDSDCON2的ALUSEL位确定读写两个运算单元的其中一个的操作数/结果寄存器。当ALUSEL=0时,读写乘法移位单元的操作数/结果寄存器;当ALUSEL=1,读写CORDIC运算单元操作数/结果寄存器。

8.2.6 操作数/结果寄存器清零

CMDSDU模块带有快速清除操作数寄存器功能。将CMDSDCON1寄存器中的MD[2:0] = 111,同时选择将需要清0的寄存器 (通过CMDSDCON1中OPRDZ/OPRDY/OPRDZ选择),再置位RUN开始清0操作。将在一个系统周期时间内由硬件清0指定的操作数寄存器 (OPRDZH/OPRDZL/OPRDYH/OPRDYL,OPRDXH/OPRDXL) 的值。清零功能只对操作数寄存器有效,对内核计算/结果寄存器无法执行清零操作。清零操作同样也受CMDSDCON2中的Bank位及ALUSEL控制。当Bank=0时,清零Bank0组内相应寄存器;当Bank=1时,清零Bank1组内相应寄存器。当ALUSEL=0时,清零乘法移位单元内的相应寄存器;当ALUSEL=1时,清零CORDIC单元内的相应寄存器。

8.2.7 操作数/结果寄存器更新

CMDSDU模块带有两个运算单元操作数/结果寄存器之间相互更新功能。将CMDSDCON1寄存器中的MD[2:0] = 110,同时选择将需要更新的操作数寄存器 (通过CMDSDCON1中OPRDZ/OPRDY/OPRDZ选择),再置位RUN开始更新操作。将在一个系统周期时间内由硬件将指定的运算单元操作数/结果寄存器 (OPRDZH/OPRDZL/OPRDYH/OPRDYL,OPRDXH/OPRDXL) 更新为另一个运算单元计算/结果寄存器的值。当ALUSEL=0时,将更新乘法移位单元内的相应寄存器,也即将CORDIC单元内的寄存器值加载到乘法移位单元的相应寄存器;当ALUSEL=1时,将更新CORDIC单元内的相应寄存器,也即将乘法移位单元内的寄存器值加载到CORDIC单元的相应寄存器。更新操作同样也受CMDSDCON2中的Bank位控制,当Bank=0时,更新Bank0组内相应寄存器 (从一个运算单元的Bank0更新到另一个单元的Bank0);当Bank=1时,更新Bank1组内相应寄存器 (从一个运算单元的Bank1更新到另一个单元的Bank1)。

8.2.8 协处理器运行标志

通过置为CMDSDCON1的RUN位启动协处理器计算后RUN位立即被清零,根据执行的不同计算CMDSDCON2的BSYMDS或BSYCDC位会置1,当计算结束后相应的位被硬件清0。

8.2.9 乘法移位单元 (MDSU)

16-bit X 16-bit乘法

进行16-bit X 16-bit乘法操作时,乘数和被乘数分别存放于寄存器OPRDXH、OPRDXL和OPRDYH、OPRDYL中。由于乘法符合交换律,显然乘数和被乘数存放的寄存器也可以互相交换。

- 写被乘数和乘数到寄存器OPRDYH、OPRDYL和OPRDXH、OPRDXL
- 设置寄存器CMDSDCON1, MD[2:0] = 000, SIGNEN = 0选择无符号乘法操作,操作数用原码表示; SIGNEN = 1选择有符号乘法操作,操作数用补码表示,最高位表示符号位;置位RUN则启动运算
- 对乘法操作启动之前的寄存器设置没有顺序要求

- 当硬件完成操作后，会自动将CMDSCON2中的BSYMDS清0。此时，可以从相应寄存器中读取乘积
- 由于乘法操作只需要一个系统周期执行时间，因此可以不用判断CMDSCON2的BSYMDS标志位，直接读取乘法结果
- 16-bit X 16-bit乘积是一个32-bit数据，存放于OPRDZH - OPRDYL，其中OPRDZH是MSB，OPRDYL是LSB。当读取乘积时，对读取顺序没有要求，但是按照从MSB到LSB读取（或者倒过来）是最方便的一种。读取操作不会改变寄存器中的乘积值。乘积将一直保留在OPRDZH/ OPRDZL/OPRDYH/OPRDYL，除非被硬件/软件改写
- 对于无符号的乘法操作，操作数都是以原码的方式写入和结果以原码形式储存；对于有符号的乘法操作，操作数都是以补码的形式写入和结果以补码方式储存

32-bit ÷ 16-bit数除法

进行32-bit ÷ 16-bit除法时，OPRDZH/ OPRDZL/OPRDYH/OPRDYL，用于存放除法操作的被除数；OPRDXH和OPRDXL，用于存放除法操作的除数。32-bit ÷ 16-bit除法结果有一个32-bit商和一个16-bit余数。其中，商存放于OPRDZH/ OPRDZL/OPRDYH/OPRDYL，OPRDZH是MSB，OPRDYL是LSB。余数存放于OPRDXH/ OPRDXL，其中OPRDXH是高位，OPRDXL是低位。

- 写除数到寄存器OPRDXH和OPRDXL，写被除数到寄存器OPRDZH-OPRDYL
- 设置寄存器CMDSCON1，MD[2:0] = 001，SIGNEN = 0选择无符号除法操作，操作数用原码表示；SIGNEN = 1选择有符号除法操作，操作数用补码表示，最高位表示符号位；置位RUN启动运算
- 对除法操作启动之前的寄存器设置没有顺序要求
- 由于除法操作需要八个系统周期执行时间，启动除法后，CMDSCON2的BSYMDS为1则表示运算未完成
- 当硬件完成操作后，会自动将CMDSCON2的BSYMDS位清0。此时，可以从相应寄存器中读取商和余数
- 当读取商或者余数时，对读取顺序没有要求。读取操作不会改变寄存器中的商和余数。商和余数将一直保留在相应的寄存器中，除非被硬件/软件改写
- 对于有符号除法运算，对取余数规则描述如下：余数的符号与被除数相同，也即 余数 = 被除数 - 商 × 除数。例如：-10 ÷ 3，商为-3，余数为-1（而不是商为-4，余数为+2）；15 ÷ (-4)，商为-3，余数为3。
- 对于无符号的除法操作，操作数都是以原码的方式写入和结果以原码形式储存；对于有符号的除法操作，操作数都是以补码的形式写入和商和余数以补码方式储存

32-bit移位

32-bit数据移位结果仍然是32-bit数据，32-bit操作数存放于OPRDZH/ OPRDZL/OPRDYH/OPRDYL，位移动数存放在OPRDXL中，结果仍然存放于OPRDZH/ OPRDZL/OPRDYH/OPRDYL，OPRDZH是MSB，OPRDYL是LSB。

无符号数不论是左移操作还是右移操作，移出数据的空位均是补0。有符号数左移，空位补0，最高位可能会溢出，这点和KEIL C编译器一致；右移，空位补符号位（正数右移，最高位补0；负数右移最高位补1）操作也与KEIL C移位结果一致。

- 写待移位数据到OPRDZH/ OPRDZL/OPRDYH/OPRDYL，写待移位的次数到OPRDXL（X[4:0]的值即为移位次数）
- 设置寄存器CMDSCON1，MD[2:0] = 010，选择移位操作，置位RUN位开始移位运算
- 对移位操作启动之前的寄存器设置没有顺序要求
- 当硬件完成操作后，会自动将CMDSCON2中BSYMDS清0。此时，可以从相应的寄存器中读取移位结果
- 由于移位操作只需要一个系统周期执行时间，因此可以不用判断RUN标志位，直接读取移位操作后的结果
- 读取操作不会改变寄存器中的移位结果。移位结果将一直保留在相应的寄存器中，除非被硬件/软件改写

注意事项

(1) 在硬件执行除法操作没有结束时试图读取运算结果（即读取寄存器OPRDZH-OPRDYL和OPRDXH、OPRDXL中任意一个或者几个），会得到不可预知的结果。但是，读取动作不会影响硬件继续运算并得到正确结果。当硬件完成运算后，仍然可以读取到正确的结果

(2) 移位操作中，如果移位次数值为0（X[4:0] = 0），则不会移位，操作数寄存器OPRDY和OPRDZ保持原有值不变。硬件自动将操作控制寄存器CMDSCON1的位RUN清0，其余位保持不变，不会影响用户程序继续运行

(3) 在运算未完成时（BSYMDS = 1），任何对OPRDX、OPRDY和OPRDZ的写入操作都是无效的

异常状态

如果除法操作中除数为0 则不会执行除法操作 操作数寄存器OPRDZH/ OPRDZL/OPRDYH/OPRDYL/OPRDXH/ OPRDXL保持原有值不变 硬件自动将操作控制寄存器CMDSCON1的RUN位及CMDSCON2的BYSMDS位清0并置位DVERR标志异常情况，其余位保持不变，不会影响用户程序继续运行。DVERR异常标志位会在下一次启动任一乘法除法移位运算时硬件自动清0。

8.2.10 CORDIC计算单元

G80F980内建一个硬件的CORDIC计算单元，可通过其实现基本CORDIC算法。CORDIC算法是一个迭代过程，每次计算进行16次迭代，最多41个周期内完成，可用来解圆函数（三角函数）等。一般CORDIC算法对应如下CORDIC方程：

$$x_{i+1} = x_i - d_i \cdot y \cdot 2^{-i}$$

$$y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot e_i$$

旋转模式下 $d_i = \text{sign}(z_i), z_i \rightarrow 0$

向量模式下 $d_i = \text{sign}(y_i), y_i \rightarrow 0$

$$e_i = \arctan(2^{-i})$$

CORDIC协处理器的操作

CORDIC协处理器可工作在旋转模式或向量模式下，用来计算圆函数(三角函数)，可通过CMDSCON1控制寄存器设置旋转或向量模式。

根据需要执行的函数将相应初始值赋值到X, Y, Z数据寄存器中后，启动CORDIC计算(RUN=1)，CORDIC将数据寄存器内容加载到内核计算寄存器中，并在计算过程中CMDSCON2的BSYCDC位和RUN标志一直为1，计算结束后由硬件清零。用户启动运算后可查询CMDSCON2的BSYCDC标志，标志清零后根据需要读取计算结果。计算结束后，若有数据溢出，则OVF标志位被置位。

数据格式

CORDIC协处理器的初始数据X, Y和Z输入都是有符号数2补码格式。结果数据也是有符号数2补码格式，有一种例外是在圆函数向量模式中的X结果数据将会按无符号数存储，这样可防止潜在的X数据结果溢出，这种情况下MSB(最高位)也是数据位。只有工作在圆函数向量模式时，X操作结果始终为正。

X和Y输入数据格式可以是整数或定点数，但在任何计算中X和Y数据必须保持一致，如果是定点小数，则必须拥有相同个数的小数位；而对于Z数据始终是归一化整数值，在旋转模式下Z作为输入数据时代表角度，用 $[-2^{15}, (2^{15}-1)]$ 代表

$[-\pi, \frac{(2^{15}-1)}{2^{15}} \cdot \pi]$ ，而在向量模式下Z作为输出数据时也是代表角度，也就是可以用定点小数Q1.15格式表示整个坐标平面内

的任一角度。角度的分辨率为 $\frac{180^\circ}{32768} = 0.00549^\circ$

$$\text{输入Z的初始值(Z寄存器值)} = \text{实数Z初始值(弧度)} \times \frac{32768}{\pi}$$

$$\text{实数Z的结果数据(弧度)} = \text{Z结果数据(Z寄存器值)} \times \frac{\pi}{32768}$$

需要注意的是，CORDIC的计算结果包含一个固有的，有旋转模式或向量模式引起的增益因子K，在圆函数中 $K \approx 1.64676$

溢出标志及幅度调节

若在CORDIC计算过程中发生了溢出，则会产生OVF标志，有一种例外是在圆函数向量模式中的X结果数据将固定会按无符号数存储，如计算结果仍在 $[0, 32767]$ 范围内则不会产生溢出标志。另外CMDSCON2的XYMRS位可用来设置CORDIC结果数据的幅度调节，若XYMRS置1则CORDIC迭代结束后，将X和Y的计算结果除以2以后存入X和Y寄存器中，这样可有效防止结果数据溢出

操作模式和对应结果数据

CORDIC方程如下，协处理器操作模式和对应的结果数据见下表， $X_{\text{result}}, Y_{\text{result}}, Z_{\text{result}}$ 为最终数据，X, Y, Z表示初始值

$$x_{i+1} = x_i - d_i \cdot y \cdot 2^{-i}$$

$$y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i}$$

$$z_{i+1} = z_i - d_i \cdot e_i$$

表8-35 CORDIC操作模式和对应结果数据

圆函数	旋转模式	向量模式
	$d_i = \text{sign}(z_i), z_i \rightarrow 0$	$d_i = \text{sign}(y_i), y_i \rightarrow 0$

$e_i = \arctan(2^{-i})$	$X_{result} = K[X \cos(Z) - Y \sin(Z)]$ $Y_{result} = K[X \sin(Z) + Y \cos(Z)]$ $Z_{result} = 0$ 其中 $K \approx 1.64676$	$X_{result} = K\sqrt{X^2 + Y^2}$ $Y_{result} = 0$ $Z_{result} = Z + \arctan(Y / X)$ 其中 $K \approx 1.64676$
	设置 $X = 1/K, Y = 0$ 则 $X_{result} = \cos(Z), Y_{result} = \sin(Z)$ 由于CORDIC内部采用了预处理逻辑, 所以X,Y和Z可全范围 $([-2^{15}, (2^{15} - 1)])$ 取值	设置 $X = X/K, Y = Y/K$ 则向量幅值 $X_{result} = \sqrt{X^2 + Y^2}$ 由于CORDIC内部采用了预处理和后处理逻辑, 所以X,Y可全范围 $([-2^{15}, (2^{15} - 1)])$ 取值 设置 $Z = 0$ 则 $Z_{result} = \arctan(Y / X)$ 由于CORDIC内部采用了预处理和后处理逻辑, 所以X,Y可全范围 $([-2^{15}, (2^{15} - 1)])$ 取值, $X=0$ 除外

反正切查找表

$e_i = \arctan(2^{-i})$ 为每次迭代需要的反正切值, G80F980通过预存的查找表得到该次迭代需要的值, 该值在CORDIC内部以Q1.23格式存储

迭代次数	$e_i = \arctan(2^{-i})$ 预计算值	迭代次数	$e_i = \arctan(2^{-i})$ 预计算值
0	200000	8	28BE
1	12E405	9	145F
2	9FB38	10	A2F
3	51111	11	517
4	28B0D	12	28B
5	145D7	13	145
6	A2F6	14	A2
7	517C	15	51

CORDIC计算精度

尽管在圆函数旋转模式下, X和Y可以是整数或定点小数, 且要求X和Y保持相同的小数位数, 但考虑到结果误差及提高分辨率, 在进行计算时应尽可能选择多的小数位来提高计算精度, 也就是尽可能将幅值放大。而Z作为输入数据时是归一化的Q1.15的小数格式。如圆函数旋转模式下

$$Y_{result} = KX \sin(Z), \quad Y = 0, \quad K = 1.64676$$

为了尽可能将幅值放大, $A = KX$, 设置 $X = A/K$, 取 $A = 32768$, 则CORDIC计算的结果 $Y_{result} = KX \sin(Z)$ 相当于将角度Z的正弦值扩大了32768倍, 在16bit的数据宽度下最大化的提高了精度。

同样可以以Q格式小数的形式理解以上描述: 将X和Y都视作Q1.15格式(Q15)格式的数, 则Y的结果就是Q15格式的正弦值, 范围在[Q15(-1), Q15(1)]范围内, 而作为Y结果正弦函数的幅值, 可设置 $X = 1/K$ (实际是Q15格式小数), 即设置初值

$$X = Q15(1/K) = Q15(1/1.64676) = \frac{32768}{1.64676} = 0x4DBA, \quad Y = 0, \quad Z = angle \text{ (归一化的角度值)},$$

经过CORDIC迭代完成后, 程序即可读取结果 $Y_{result} = \sin(Z)$, 同时 $X_{result} = \cos(Z)$ 。

8.3 运算放大器和比较器模块

8.3.1 特性

- 集成 4 个通用运算放大器，放大器的输入输出端都开放，可通过外接电阻调整运放增益；也可通过设置选择内部预设的增益档位。
- 放大器输出可直接作为 ADC 转换模块的输入或者做比较器的同相输入端
- 比较器 1/2 为多输入比较器，比较器 2 集成 PWM0/1/2 同步比较功能
- 比较器输出结果滤波后可以用作 PWM 故障检测输入信号
- 比较器 1/2 反相输入端参考信号可以从外部输入，也可以采用内部参考电压（8 档）作为反向输入
- 比较器 1 内置施密特窗口和数字滤波电路，输出可触发上升沿中断
- 比较器 2 内置施密特窗口和数字滤波电路，输出可触发不同类型中断

8.3.2 寄存器

Table 8.33 比较器1控制寄存器1

FF8EH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMP1CON	CMP1EN	C1NCHS	C1PCHS1	C1PCHS0	OP1PCHS	OP1CHOP	C1DEB1	C1DEB0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	CMP1EN	比较器1使能控制位 0: 比较器1关闭 1: 比较器1开启
6	C1NCHS	比较器1反相输入端选择位 0: 选择C1N作为比较器1反相输入端 1: 选择内部参考CMP1VRS作为比较器1反相输入端 <i>注: 需提前100us将1.25V基准源打开, 即将寄存器[CMP2CON1: BGEN]位置1, 并设置CMP1VRS选择位</i>
5-4	C1PCHS[1:0]	比较器1同相输入端选择位 00: 选择C1P作为比较器1同相输入端 01: 选择OP1OUT作为比较器1的同相输入端 10: 选择OP2OUT作为比较器1的同相输入端 11: 选择OP3OUT作为比较器1的同相输入端
3	OP1PCHS	运算放大器1同相输入端选择位 0: 选择OP1P作为放大器1同相输入端 1: 选择温度传感器输出作为放大器1同相输入端 <i>注: 需提前100us将温度传感器打开, 即将寄存器[ADCH2: TPSON]位置1</i>
2	OP1CHOP	运算放大器1 CHOP功能控制位 0: 运算放大器1 CHOP功能关闭 1: 运算放大器1 CHOP功能开启 <i>注: 若选择温度传感器作为OP1同相输入, 建议开启OP1 CHOP功能</i>
1-0	C1DEB[1:0]	比较器1输出信号滤波时间 00: 无滤波 01: 滤波时间常数为1us 10: 滤波时间常数为4us 11: 滤波时间常数为16us <i>注: 上述滤波常数时间不是精确值, 仅供参考</i>

注: 当运算放大器同相输入端选择切换时, 需经过1us时间OP1OUT输出稳定, 此时可设置AD通道得到OP1输出采样结果。

Table 8.34 比较器2控制寄存器1

FF8BH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMP2CON1	CMP2EN	C2NCHS	C2PCHS2	C2PCHS1	C2PCHS0	PWMTRGS1	PWMTRGS0	TRGPOL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	CMP2EN	比较器2使能控制位 0: 比较器2关闭 1: 比较器2开启
6	C2NCHS	比较器2反相输入端选择位 0: 选择C2N作为比较器2反相输入端 1: 选择内部参考CMP2VRS作为比较器1反相输入端 <i>注: 需提前100us将1.25V基准源打开, 即将寄存器[CMP2CON1: BGEN]位置1, 并设置CMP2VRS选择位</i>
5-3	C2PCHS[1:0]	比较器2同相输入端选择位 000: 选择C2P0作为比较器2同相输入端 001: 选择C2P1作为比较器2同相输入端 010: 选择C2P2作为比较器2同相输入端 011: 选择OP1OUT作为比较器2同相输入端 100: 选择OP2OUT作为比较器2同相输入端 101: 选择OP3OUT作为比较器2同相输入端 110: 选择OP4OUT作为比较器2同相输入端 111: 选择OP4OUT作为比较器2同相输入端
2-1	PWMTRGS[1:0]	比较器2同步比较控制位 00: 连续比较模式, 无同步比较功能 01: 和PWM0输出同步 10: 和PWM1输出同步 11: 和PWM2输出同步
0	TRGPOL	同步比较功能控制位 0: 仅在PWM输出有效期间比较 1: 仅在PWM输出无效期间比较

注意: 因为比较器1负向输入端C1N是和ADC的输入通道AN7是共享的, 因此这个端口配置为模拟通道或I/O必须并且只能在ADC模块中的寄存器ADCH1配置, 分别受CH7控制。因为模拟比较器2有3个同相输入端外部端口(不包括OP2OUT), 实际应用时, 任意时刻只有一个输入端被选中, 但另外2个输入端外接的是模拟电平。此时需要将3个模拟输入端都配置为模拟输入口状态, 否则会漏电; 因为比较器2的同相输入端C2P0-2、负向输入端C2N是和ADC的输入通道AN8-AN10、AN3是共享的, 因此这四个端口配置为模拟通道或I/O必须并且只能在ADC模块中的寄存器ADCH1/2配置, 分别受CH8-CH10、CH3控制。

Table 8.35 ADC通道配置寄存器1

FF81H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH1	CH7	-	-	-	CH3	-	-	-
读/写	读/写	-	-	-	读/写	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	0	-	-	-

FF82H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH2	CH10	CH9	CH8	-	-	-	-	-
读/写	读/写	读/写	读/写	-	-	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	0	0	-	-	-	-	-

位编号	位符号	说明
ADCH2[7-4] ADCH1[7] ADCH1[3]	CH[10:8] CH7 CH3	通道配置 1: Pxy配置为模拟输入口（ADC输入或比较器1/2输入） 0: Pxy作为I/O口 <i>注：对于CMP2此控制位只是用来配置相应PORT口是否做模拟口用，具体选中那一个作为CMP2同相端输入做比较，由寄存器C2PCHS[1:0]控制</i>

Table 8.37 比较器2控制寄存器2

FF8CH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMP2CON2	C2FT1EN	C2FT1S2	C2FT1S1	C2FT1S0	C2FT2EN	C2FT2S2	C2FT2S1	C2FT2S0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	C2FT1EN	比较器2滤波器1使能位 0: 比较器2滤波器1关闭 1: 比较器2滤波器1开启 滤波器1算法说明: 用系统时钟采样滤波器输入 采用数字积分, 如果采样结果为高电平, 计数器加1, 计数器结果超出C2FT1S[2:0]设定的常数, 则滤波器输出1同时计数器置为滤波常数×2; 如果采样结果为低电平, 计数器减1, 计数器结果小于滤波器常数, 则滤波器输出0同时计数器置为0 <i>注: 1. 计数器上限设置为常数的2倍, 下限为0, 超出上下限, 不执行加减动作</i> <i>2. 滤波计数器初值设定: 若比较器2中断设为上升沿触发, 计数器初始值为0; 若设为下降沿触发, 则滤波计数器初值设为滤波常数x2。若设为双沿触发或不触发, 则滤波计数器初值设为滤波常数</i> <i>3. 滤波器输出稳定需要一个建立时间, 大致为滤波常数设置的时间, 如滤波常数设置为256, 则在256个系统时钟后, 滤波器输出稳定, 在这之前, 滤波器输出不确定, 中断标志C2IF可能会被误置1, 因此, 刚打开滤波器时, 需要在滤波器输出稳定后软件清除此中断标志位</i>
6-4	C2FT1S[2:0]	比较器2滤波器1参数设置 000: 滤波器常数为32 001: 滤波器常数为64 010: 滤波器常数为128 011: 滤波器常数为256 100: 滤波器常数为512 101: 滤波器常数为1024 110: 滤波器常数为2048 111: 滤波器常数为4096

3	C2FT2EN	<p>比较器2滤波器2使能位</p> <p>0: 比较器2滤波器2关闭 1: 比较器2滤波器2开启</p> <p>滤波器2算法说明: 用系统时钟采样滤波器输入 当滤波器2当前输出为0时, 若输入采样为1, 计数器加1; 若输入采样为0, 则将计数器清0; 当计数器超过C2FT2S[2:0]设定的滤波常数时, 滤波器输出翻转为1, 同时计数器清0 同理, 当滤波器2当前输出为1时, 若输入采样为0, 计数器加1, 若输入采样为1, 则将计数器清0, 当计数器超过C2FT2S[2:0]设定的滤波常数时, 滤波器输出翻转为0, 同时计数器清0</p> <p>注: 1. 计数器初始值为0 2. 滤波器输出稳定需要一个建立时间, 大致为滤波常数设置的时间, 如滤波常数设置为256, 则在256个系统时钟后, 滤波器输出稳定, 在这之前, 滤波器输出不确定, 中断标志C2IF可能会被误置1, 因此, 刚打开滤波器时, 需要在滤波器输出稳定后软件清除此中断标志位</p>
2-0	C2FT2S[2:0]	<p>比较器2滤波器2参数设置</p> <p>000: 滤波器常数为8 001: 滤波器常数为16 010: 滤波器常数为32 011: 滤波器常数为64 100: 滤波器常数为128 101: 滤波器常数为256 110: 滤波器常数为512 111: 滤波器常数为1024</p>

注: 1. 滤波器1和滤波器2可单独有效, 也可同时有效, 同时有效时比较器2输出先经过滤波器1滤波, 再经过滤波器2滤波。
2. 滤波器1/2打开前, 需先设置好比较器2中断模式控制位C2IFS[1:0]

Table 8.33 比较器施密特控制寄存器

FF8FH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMPSMT					C2SMT1	C2SMT0	C1SMT1	C1SMT0
读/写	读/写	读/写	读	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
3-2	C2SMT[1:0]	<p>比较器2施密特电压选择位:</p> <p>00: 没有施密特窗口特性 01: 施密特窗口为10mv 10: 施密特窗口为20mv 11: 施密特窗口为50mv</p>
1-0	C1SMT[1:0]	<p>比较器1施密特电压选择位:</p> <p>00: 没有施密特窗口特性 01: 施密特窗口为10mv 10: 施密特窗口为20mv 11: 施密特窗口为50mv</p>

Table 8.34 比较器1中断寄存器

FF8AH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMPINT1	-	-	-	-	-	-	C1IF	C1OUT
读/写	-	-	-	-	-	-	读/写	读
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0

位编号	位符号	说明
-----	-----	----

1	C1IF	比较器1输出上升沿中断标志位（经过滤波器以后） 0: 比较器1输出无上升沿，没有产生中断 1: 比较器1输出产生上升沿，发生中断请求
0	C1OUT	比较器1输出状态标志位（表示经过滤波器以后的状态，只读） 0: 比较器1输出低 1: 比较器1输出高

Table 8.35 比较器2中断寄存器

FF87H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMPINT2	-	-	C2IFS1	C2IFS0	-	-	C2IF	C2OUT
读/写	-	-	读/写	读/写	-	-	读/写	读
复位值 (POR/WDT/LVR/PIN)	-	-	0	0	-	-	0	0

位编号	位符号	说明
5-4	C2IFS[1:0]	比较器2中断模式选择位 00: 不触发中断标志 01: 下降沿触发，比较器2输出由高变低时触发中断标志 10: 上升沿触发，比较器2输出由低变高时触发中断标志 11: 双沿触发，比较器2输出由高变低时和由低变高时都触发中断标志
1	C2IF	比较器2中断标志位 0: 比较器2没有产生中断请求标志 1: 比较器2产生中断请求标志
0	C2OUT	比较器2输出状态标志位（表示经过滤波器以后的状态，只读） 0: 比较器2输出低 1: 比较器2输出高

Table 8.36 放大器控制寄存器1

FF88H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPCON1	OP2EN	OP2G2	OP2G1	OP2G0	OP1EN	OP1G2	OP1G1	OP1G0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	OP2EN	运算放大器2使能控制位 0: 运算放大器2关闭 1: 运算放大器2开启 运算放大器开启时，相应的输入输出引脚I/O功能关闭，而做为模拟引脚使用
6-4	OP2G[2: 0]	运算放大器2增益系数选择： 000: 由外部电阻网络确定，OP2N为运放反向输入脚，OP2OUT作为运放输出脚 001: 由内部增益系数确定×1，OP2N/OP2OUT为普通IO 010: 由内部增益系数确定×2，OP2N/OP2OUT为普通IO 011: 由内部增益系数确定×3，OP2N/OP2OUT为普通IO 100: 由内部增益系数确定×4，OP2N/OP2OUT为普通IO 101: 由内部增益系数确定×6，OP2N/OP2OUT为普通IO 110: 由内部增益系数确定×8，OP2N/OP2OUT为普通IO 111: 由内部增益系数确定×10，OP2N/OP2OUT为普通IO

3	OP1EN	运算放大器1使能控制位 0: 运算放大器1关闭 1: 运算放大器1开启 运算放大器开启时, 相应的输入输出引脚I/O功能关闭, 而做为模拟引脚使用
2-1	OP1G[2: 0]	运算放大器1增益系数选择: 000: 由外部电阻网络确定, OP1N为运放反向输入脚, OP1OUT作为运放输出脚 001: 由内部增益系数确定 $\times 1$, OP1N/OP1OUT为普通IO 010: 由内部增益系数确定 $\times 2$, OP1N/OP1OUT为普通IO 011: 由内部增益系数确定 $\times 3$, OP1N/OP1OUT为普通IO 100: 由内部增益系数确定 $\times 4$, OP1N/OP1OUT为普通IO 101: 由内部增益系数确定 $\times 6$, OP1N/OP1OUT为普通IO 110: 由内部增益系数确定 $\times 8$, OP1N/OP1OUT为普通IO 111: 由内部增益系数确定 $\times 10$, OP1N/OP1OUT为普通IO

注: 1. 如需改变放大器增益系数选择, 需将OPxEN关闭后(OPxEN=0)后, 重新开启指定新的增益系数

Table 8.36 放大器控制寄存器2

FF89H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
OPCON2	OP4EN	OP4G2	OP4G1	OP4G0	OP3EN	OP3G2	OP3G1	OP3G0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	1	0	0	0	0	0	0	0

位编号	位符号	说明
7	OP4EN	运算放大器4使能控制位 0: 运算放大器4关闭 1: 运算放大器4开启 运算放大器开启时, 相应的输入输出引脚I/O功能关闭, 而做为模拟引脚使用
6-4	OP4G[2: 0]	运算放大器4增益系数选择: 000: 由外部电阻网络确定, OP4N为运放反向输入脚, OP4OUT作为运放输出脚 001: 由内部电阻网络确定 $\times 1$, OP4N/OP4OUT为普通IO 010: 由内部电阻网络确定 $\times 2$, OP4N/OP4OUT为普通IO 011: 由内部电阻网络确定 $\times 3$, OP4N/OP4OUT为普通IO 100: 由内部电阻网络确定 $\times 4$, OP4N/OP4OUT为普通IO 101: 由内部电阻网络确定 $\times 6$, OP4N/OP4OUT为普通IO 110: 由内部电阻网络确定 $\times 8$, OP4N/OP4OUT为普通IO 111: 由内部电阻网络确定 $\times 10$, OP4N/OP4OUT为普通IO
3	OP3EN	运算放大器3使能控制位 0: 运算放大器3关闭 1: 运算放大器3开启 运算放大器开启时, 相应的输入输出引脚I/O功能关闭, 而做为模拟引脚使用
2-0	OP3G[2: 0]	运算放大器3增益系数选择: 000: 由外部电阻网络确定, OP3N为运放反向输入脚, OP3OUT作为运放输出脚 001: 由内部增益系数确定 $\times 1$, OP3N/OP3OUT为普通IO 010: 由内部增益系数确定 $\times 2$, OP3N/OP3OUT为普通IO 011: 由内部增益系数确定 $\times 3$, OP3N/OP3OUT为普通IO 100: 由内部增益系数确定 $\times 4$, OP3N/OP3OUT为普通IO 101: 由内部增益系数确定 $\times 6$, OP3N/OP3OUT为普通IO 110: 由内部增益系数确定 $\times 8$, OP3N/OP3OUT为普通IO 111: 由内部增益系数确定 $\times 10$, OP3N/OP3OUT为普通IO

注：1. 如需改变放大器增益系数选择，需将OPxEN关闭后（OPxEN=0）后，重新开启指定新的增益系数

Table 8.36 比较器参考电压选择控制寄存器

FF8DH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CMPVRS	-	CMP2VRS2	CMP2VRS1	CMP2VRS0	-	CMP1VRS2	CMP1VRS1	CMP1VRS0
读/写	-	读/写	读/写	读/写	-	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	0	0	0	-	0	0	0

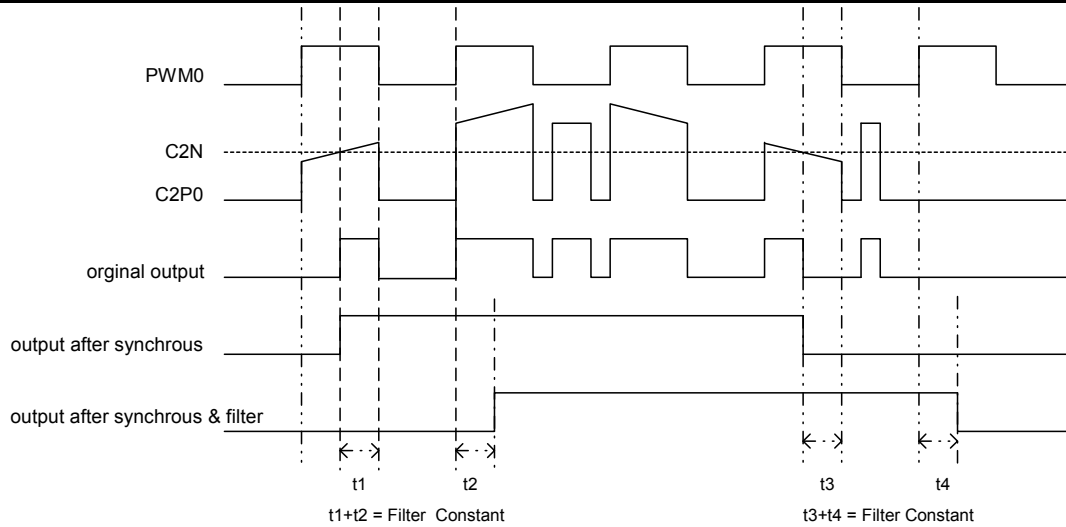
位编号	位符号	说明
6-4	CMP2VRS[2: 0]	比较器2反相端参考电压选择位 000: Vbg (1.25V) 001: $2 \times V_{bg} / 8$ 010: $3 \times V_{bg} / 8$ 011: $4 \times V_{bg} / 8$ 100: $5 \times V_{bg} / 8$ 101: $6 \times V_{bg} / 8$ 110: $7 \times V_{bg} / 8$ 111: $7 \times V_{bg} / 8$
2-0	CMP1VRS[2: 0]	比较器1反相端参考电压选择位 000: Vbg (1.25V) 001: $2 \times V_{bg} / 8$ 010: $3 \times V_{bg} / 8$ 011: $4 \times V_{bg} / 8$ 100: $5 \times V_{bg} / 8$ 101: $6 \times V_{bg} / 8$ 110: $7 \times V_{bg} / 8$ 111: $7 \times V_{bg} / 8$

注：如选择内部参考电压作为比较器反向输入，提前100us将1.25V基准源打开，即将寄存器[ADCH2:BGEN]位置1，并设置CMPxVRS选择位

8.3.3 比较器用法

G80F980置2个比较器，比较器1和比较器2，比较器1的输出可以作为PWM3模块的故障输入检测信号，自动迅速关闭PWM3输出，从而实现快速保护，也可以用在某些电压/电流闭环中。比较器2内置有可编程的施密特窗口，同时集成滤波常数可编程的数字滤波电路，同时提供中断功能，比较适合用作需要精确比较的应用场合，比如直流无刷电机的反电动势过零点检测。

比较器2集成同步比较功能，可以和PWM0/PWM1/PWM2输出同步，如果TRGPOL设为0，那么只有当MCPWM输出为有效电平期间，比较器2的输出结果才被采样，给到后级滤波电路，在MCPWM输出无效期间，不采样比较器输出，此时比较器2的输出会保持，同时滤波器也停止计数，当MCPWM输出再次有效时，恢复对比较器2的采样和滤波计数。参见下图(PWMTRGS[1:0]=01, TRGPOL=0)。比较器2的滤波是先经过滤波器1再经过滤波器2，如果只想用其中的一种滤波，可以将另一种屏蔽，也可以将两种都屏蔽。



比较器2同步比较示意图，（C2PCHS[1:0] = 0，PWMTRGS[1:0] = 01，TRGPOL = 0）

上图中的original out信号线为比较器2不滤波也不和PWM0信号同步时的C2OUT；

上图中的output after synchronous信号线为比较器2不滤波但是和PWM0信号同步时的C2OUT；

上图中的output after synchronous & filter信号线为先和PWM0同步再经滤波后的C2OUT。

8.3.4 注意事项

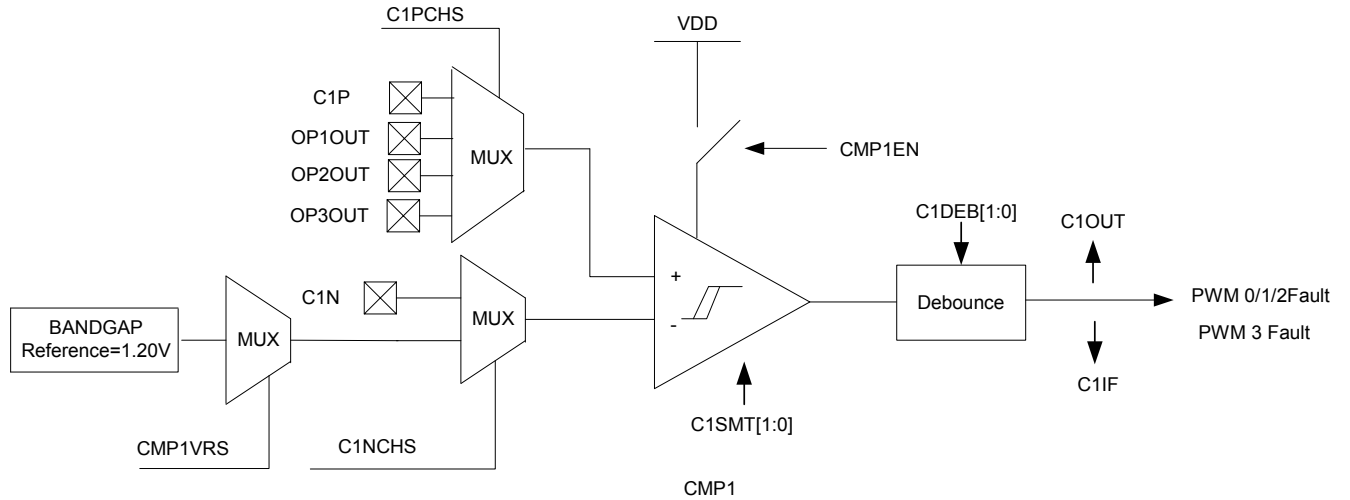
运放及比较器开启前需启动辅助电源电路，先将ADCH2的VBIAS置1后等待10us辅助电源稳定后，再开启运放或比较器。

FF82H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH2				VBIAS				
读/写				读/写				
复位值 (POR/WDT/LVR/PIN)				1				

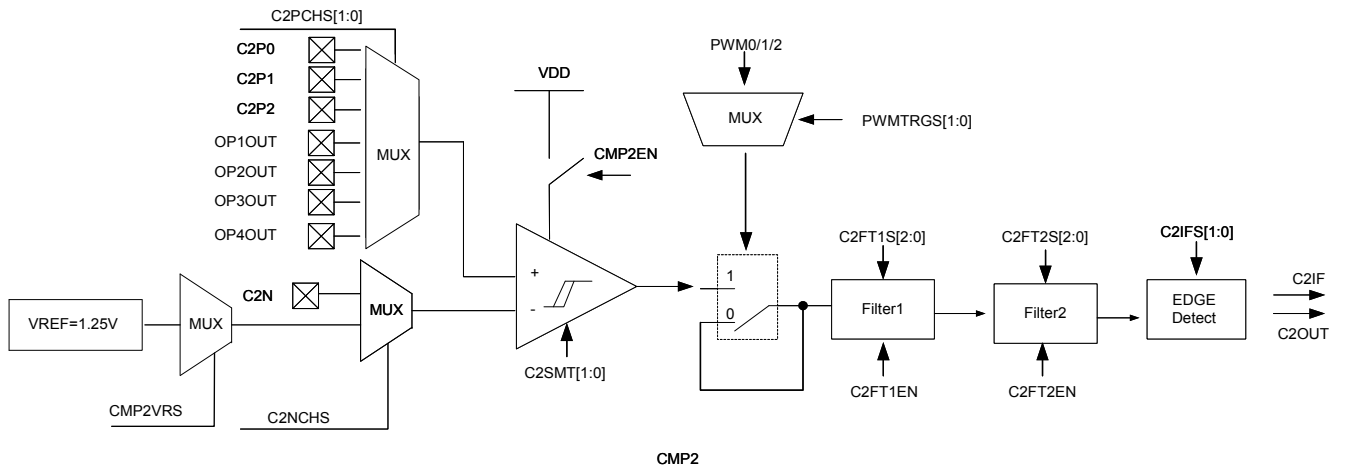
位编号	位符号	说明
4	VBIAS	<p>辅助电路电源控制位（Bias）</p> <p>0：关闭辅助电路电源</p> <p>1：打开辅助电路电源</p> <p>注：在开启ADC，运放及比较器前都应提前开启辅助电路电源，在将VBIAS位置1后等待20us后再开启ADC，运放或比较器。</p>

8.3.5 比较器框图

比较器1和2的输出波形的产生流程具体可参考下图：

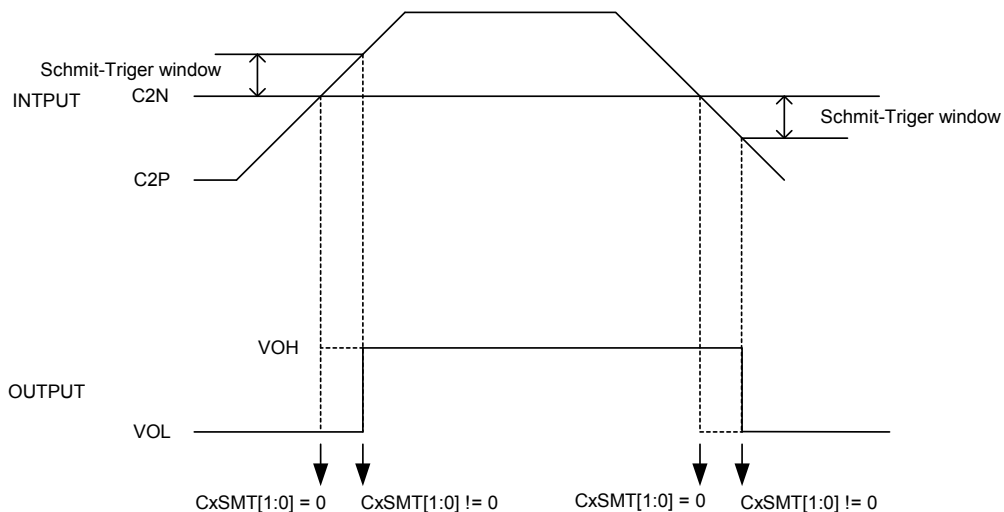


多通道输入模拟比较器1

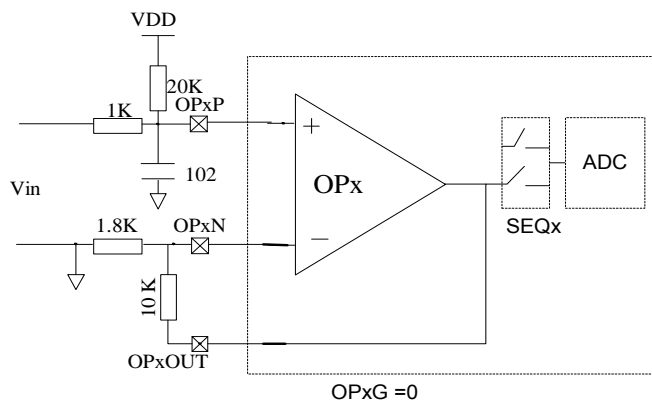


多通道输入模拟比较器2

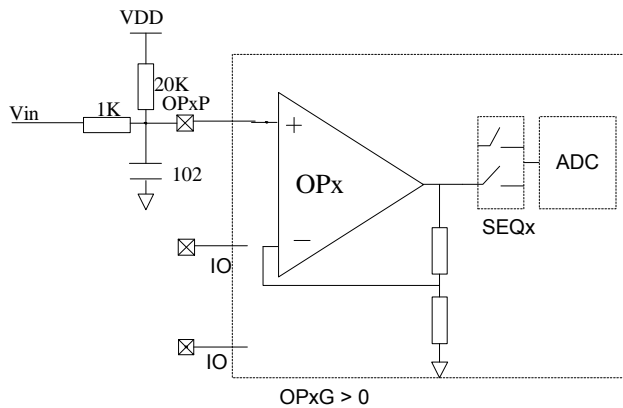
比较器1/2可以通过设置寄存器CxSMT[1:0](x=1-2)设置施密特比较器窗口（迟滞比较器），如下图所示：



8.3.6 运放典型应用图



运放典型应用图（外部增益）



运放典型应用图（内部增益）

8.4 增强型通用异步收发器 (EUART)

8.4.1 特性

- 自带波特率发生器的EUART
- 波特率发生器就是一个15位向上计数器
- 增强功能包括帧出错检测及自动地址识别
- EUART有四种工作方式

8.4.2 工作方式

EUART有4种工作方式。在通信之前用户必须先初始化SCON，选择方式和波特率。

在所有四种方式中，任何将SBUF作为目标寄存器的写操作都会启动发送。在方式0中由条件RI = 0和REN = 1初始化接收。这会在TXD引脚上产生一个时钟信号，然后在RXD引脚上移8位数据。在其它方式中由输入的起始位初始化接收（如果RI = 0和REN = 1）。外部发送器通信以发送起始位开始。

EUART工作方式列表

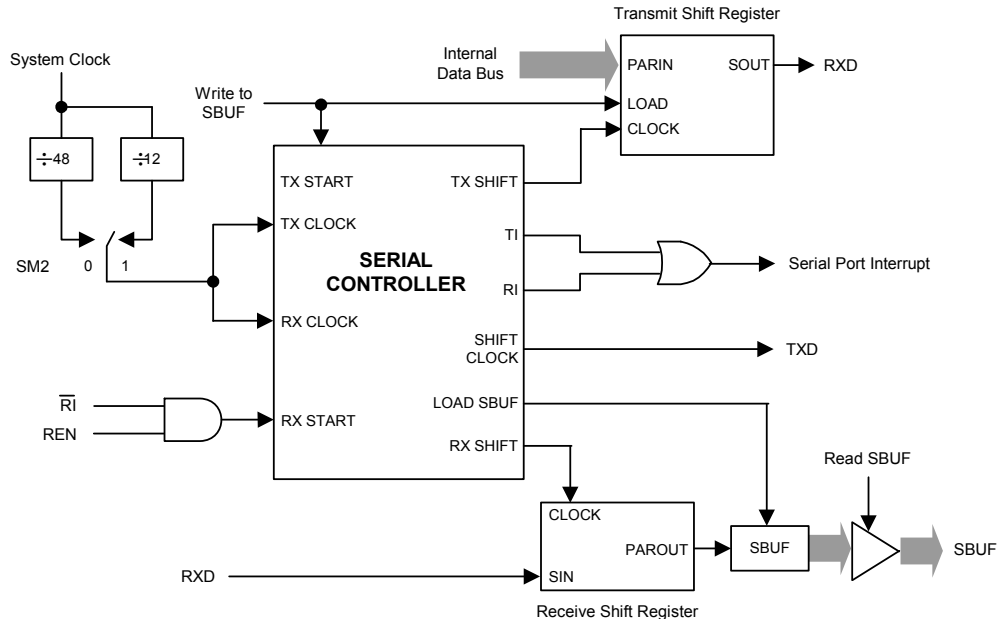
SM0	SM1	方式	类型	波特率	帧长度	起始位	停止位	第9位
0	0	0	同步	$f_{sys}/(48或12)$	8位	无	无	无
0	1	1	异步	自带波特率发生器的溢出率/16	10位	1	1	无
1	0	2	异步	$f_{sys}/(256或128)$	11位	1	1	0, 1
1	1	3	异步	自带波特率发生器的溢出率/16	11位	1	1	0, 1

方式0：同步，半双工通讯

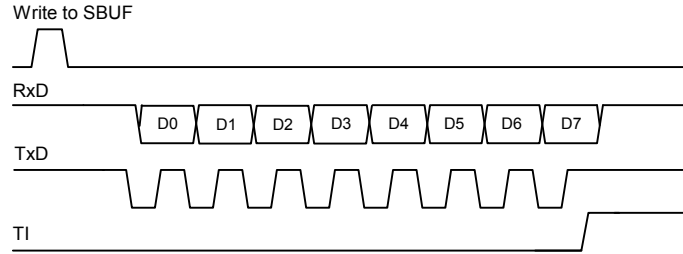
方式0支持与外部设备的同步通信。在RXD引脚上收发串行数据，TXD引脚发送移位时钟。G80F980提供TXD引脚上的移位时钟，因此这种方式是串行通信的半双工方式。在这个方式中，每帧收发8位，低位先接收或发送。

通过置SM2位（SCON.5）为0或1，波特率固定为系统时钟的1/48或1/12。当SM2位等于0时，串行端口以系统时钟的1/48运行。当SM2位等于1时，串行端口以系统时钟的1/12运行。与标准8051唯一不同的是，G80F980在方式0中有2种波特率可选。

功能块框图如下图所示。数据通过RXD引脚移入和移出串行端口，移位时钟由TXD引脚输出。

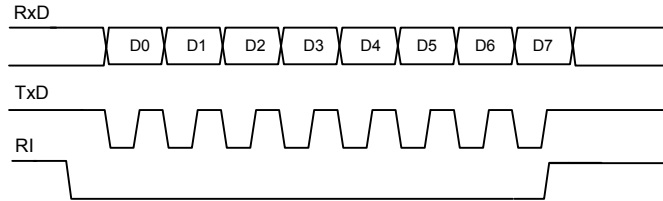


任何将SBUF作为目标寄存器的写操作都会启动发送。下一个系统时钟TX控制块开始发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置0。当移位寄存器中的所有8位都发送后，TX控制模块停止发送操作，然后在下一个系统时钟的上升沿将TI置位（SCON.1）。



Send Timing of Mode 0

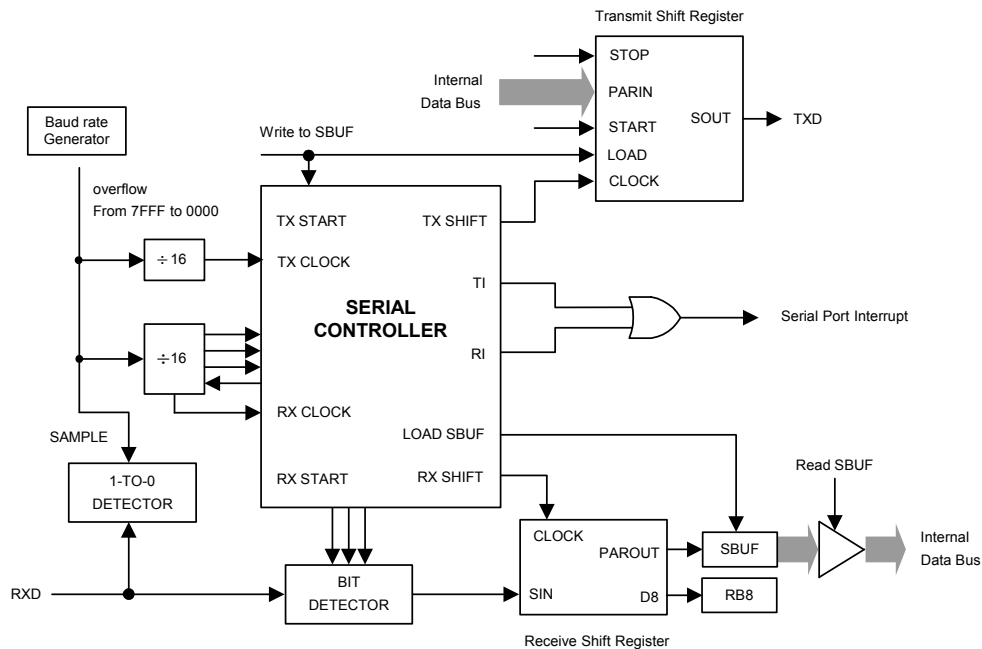
REN（SCON.4）置1和RI（SCON.0）清0初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有8位数据都移到移位寄存器中后，RX控制块停止接收，在下一个系统时钟的上升沿RI置位，直到被软件清零才允许下一次接收。



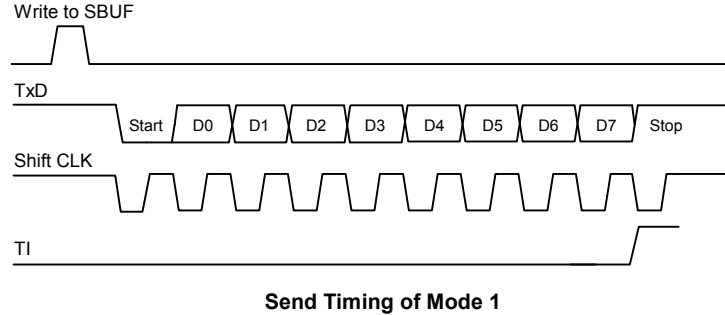
Receive Timing of Mode 0

方式1：8位EUART，可变波特率，异步全双工

方式1提供10位全双工异步通信，10位由一个起始位（逻辑0），8个数据位（低位为第一位）和一个停止位（逻辑1）组成。在接收时，这8个数据位存储在SBUF中而停止位存储在RB8（SCON.2）中。方式1中的波特率固定为自带波特率发生器溢出率的1/16。功能块框图如下图所示：



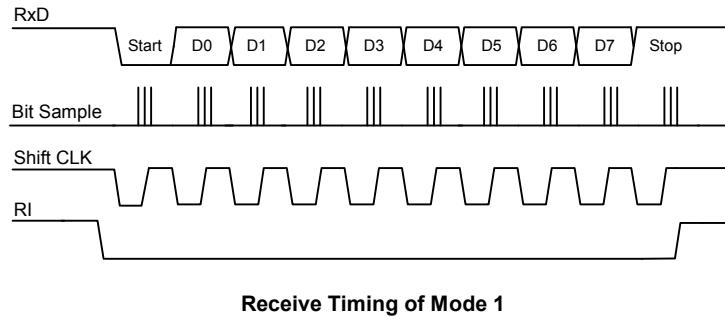
任何将SBUF作为目标寄存器的写操作都会启动发送，实际上发送是从16分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与16分频计数器是同步的，与对SBUF的写操作不同步。起始位首先在TXD引脚上移出，然后是8位数据位。在发送移位寄存器中的所有8位数据都发送完后，停止位在TXD引脚上移出，在停止位发出的同时TI标志置位。



只有REN置位时才允许接收。当RXD引脚检测到下降沿时串行口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位，这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8个数据位和1个停止位移入之后，移位寄存器的内容被分别装入SBUF和RB8中，RI置位，但必须满足下列条件：

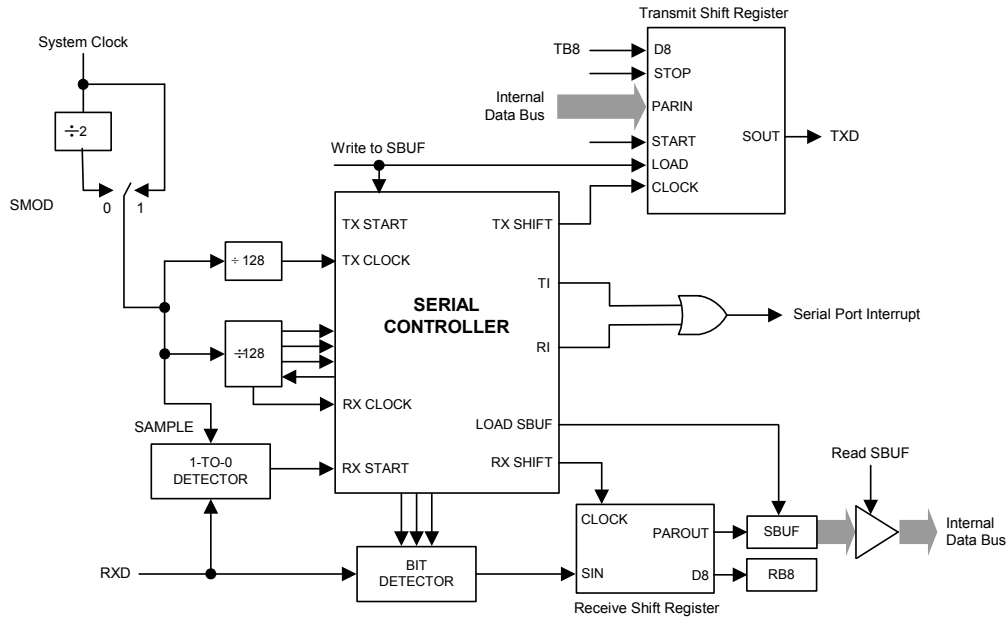
1. RI = 0
2. SM2 = 0或者接收的停止位 = 1

如果这些条件被满足，那么停止位装入RB8，8个数据位装入SBUF，RI被置位。否则接收的帧会丢失。这时，接收器将重新去探测RXD端是否另一个下降沿。用户必须用软件清零RI，然后才能再次接收。

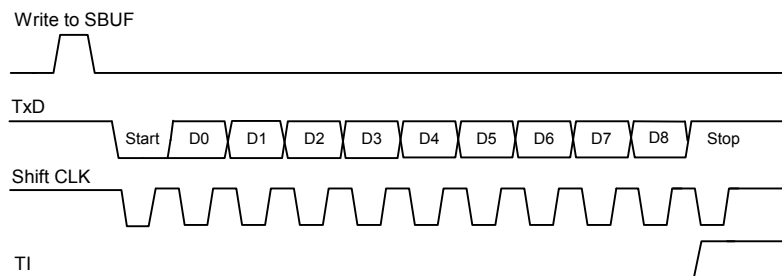


方式2: 9位EUART, 固定波特率, 异步全双工

这个方式使用异步全双工通信中的11位。一帧由一个起始位(逻辑0), 8个数据位(低位为第一位), 一个可编程的第9数据位和 一个停止位(逻辑1)组成。方式2支持多机通信和硬件地址识别(详见**多机通讯**章节)。在数据传送时, 第9数据位(SCON中的TB8)可以写0或1, 例如, 可写入PSW中的奇偶位P, 或用作多机通信中的数据/地址标志位。当接收到数据时, 第9数据位移入RB8而停止位不保存。PCON中的SMOD位选择波特率为系统工作频率的1/256或1/128。功能块框图如下所示:



任何将SBUF作为目标寄存器的写操作都会启动发送, 同时也将TB8载入到发送移位寄存器的第9位中。实际上发送是从16分频计数器中的下一次跳变之后的系统时钟开始的, 因此位时间与16分频计数器是同步的, 与对SBUF的写操作不同步。起始位首先在TXD引脚上移出, 然后是9位数据。在发送转换寄存器中的所有9位数据都发送完后, 停止位在TXD引脚上移出, 在停止位开始发送时TI标志置位。

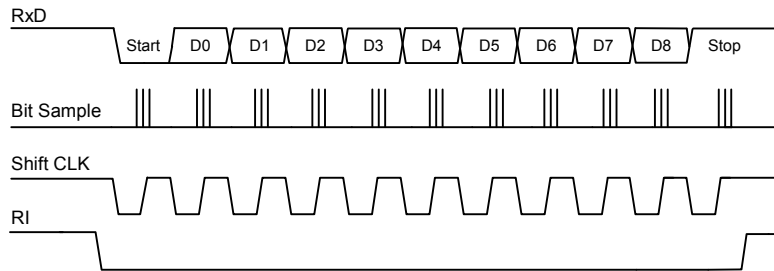
**Send Timing of Mode 2**

只有REN置位时才允许接收。当RXD引脚检测到下降沿时串口开始接收串行数据。为此，CPU对RXD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位。这有助于16分频计数器与RXD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RXD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RXD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9个数据位和1个停止位移入之后，移位寄存器的内容被分别装入SBUF和RB8中，RI置位，但必须满足下列条件：

1. RI = 0
2. SM2 = 0或者接收的第9位 = 1，且接收的字节符合约定从机地址

如果这些条件被满足，那么第9位移入RB8，8位数据移入SBUF，RI被置位。否则接收的数据帧会丢失。

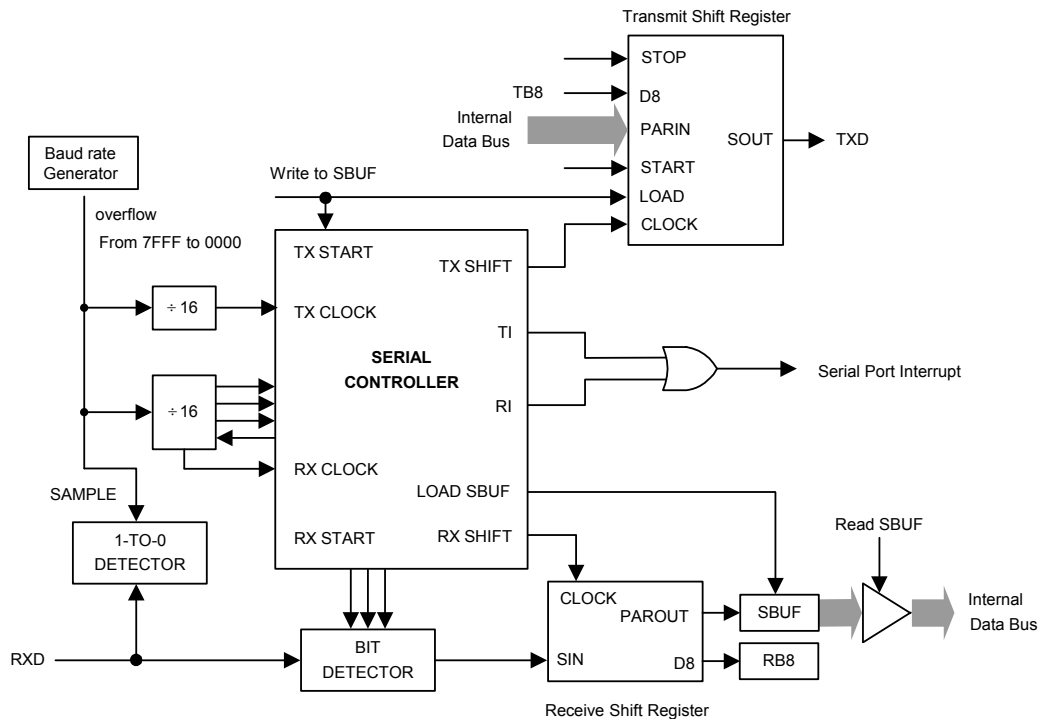
在停止位的当中，接收器回到寻找RXD引脚上的另一个下降沿。用户必须用软件清除RI，然后才能再次接收。



Receive Timing of Mode 2

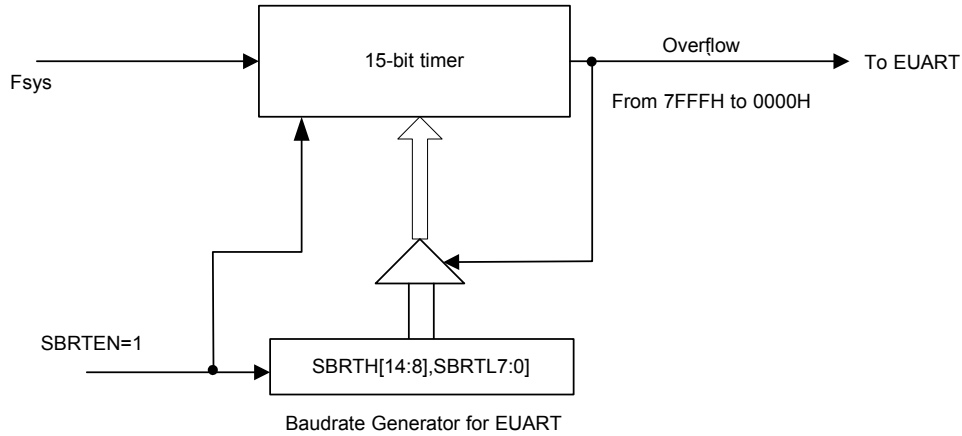
方式3：9位EUART，可变波特率，异步全双工

方式3使用方式2的传输协议以及方式1的波特率产生方式。



8.4.3 可微调波特率

EUART自带一个波特率发生器，它实质上就是一个15位递增计数器。



由图得到，波特率发生器的溢出率为

$$SBRT_{overflowrate} = \frac{F_{sys}}{32768 - SBRT}, \quad SBRT = [SBRTH, SBRTL]$$

因此，EUART在各模式下的波特率计算公式如下。

在方式0中，波特率可编程为系统时钟的1/48或1/12，由SM2位决定。当SM2为0时，串行端口在系统时钟的1/48下运行。当SM2为1时，串行端口在系统时钟的1/12下运行。

在方式1和方式3中，波特率可微调，精度为一个系统时钟，公式如下：

$$BaudRate = \frac{F_{sys}}{16 \times (32768 - SBRT) + BFINE}$$

例如：F_{sys} = 96MHz，需要得到256000Hz的波特率，SBRT和SFINE值计算方法如下：

$$96000000/16/256000 = 23.44$$

$$SBRT = 32768 - 23 = 32745$$

$$\text{由BaudRate计算公式：} 256000 = 96000000/(16 \times 23 + BFINE)$$

$$\text{得到：} BFINE = 7$$

此微调方式计算出的实际波特率为256000，误差为0.00%；以往方式计算出的波特率误差为1.90%。

在方式2中，波特率固定为系统时钟的1/256或1/128，由SMOD位（PCON.7）中决定。当SMOD位为0时，EUART以系统时钟的1/256运行。当SMOD位为1时，EUART以系统时钟的1/128运行。

$$BaudRate = 2^{SMOD} \times \left(\frac{f_{SYS}}{256} \right)$$

8.4.4 多机通讯

软件地址识别

方式2和方式3具有适用于多机通讯功能。在这两个方式下，接收的是9位数据，第9位移入RB8中，之后是停止位。可以这样设定EUART：当接收到停止位，且RB8 = 1时，串行口中断有效（请求标志RI置位）。此时置位SCON寄存器的SM2，EUART工作在多机通讯模式。

在多机通讯系统中，按如下所述来使用这一功能。当主机要发送一数据块给几个从机中的一个时，先发送一地址字节，以寻址目标从机。地址字节与数据字节可用第9数据位来区别，地址字节的第9位为1，数据字节的第9位为0。

如果从机SM2为1，则不会响应数据字节中断。地址字节可以使所有从机产生中断，每一个从机都检查所接收到的地址字节，以判别本机是不是目标从机。被寻到的从机对SM2位执行清零操作，并准备接收即将到来的数据字节。当接收完毕时，从机再一次将SM2置位。没有被寻址的从机，则保持SM2位为1，不响应数据字节。

注意：在方式0中，SM2用来2倍频波特率。在方式1中，SM2用来检测停止位是否有效，如果SM2 = 1，接收中断不会响应直到接收到一个有效的停止位。

自动（硬件）地址识别

在方式2和方式3中，SM2置位，EUART运行状态如下：接收到停止位，RB8的第9位为1（地址字节），且接收到的数据字节符合EUART的从机地址，EUART产生一个中断。从机将SM2清零，接收后续数据字节。

第9位为1表明该字节是地址而非数据。当主机要发送一组数据给几个从机中的一个时，必须先发送目标从机地址。所有从机等待接收地址字节，为了确保仅在接收地址字节时产生中断，SM2位必须置位。自动地址识别的特点是只有地址匹配的从机才能产生中断，硬件完成地址比较。

中断产生后，地址匹配的从机清零SM2，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。全部信息接收完毕后，地址匹配的从机应该再次把SM2置位，忽略所有传送的非地址字节，直到接收到下一个地址字节。

使用自动地址识别功能时，主机可以通过调用给定的从机地址选择与一个或多个从机通信。主机使用广播地址可以寻址所有从机。有两个特殊功能寄存器，从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个8位的字节，存于SADDR寄存器中。SADEN用于定义SADDR各位的有效与否，如果SADEN中某一位为0，则SADDR中相应位被忽略，如果SADEN中某一位置位，则SADDR中相应位将用于产生约定地址。这可以使用户在不改变SADDR寄存器中的从机地址的情况下灵活地寻址多个从机。

	从机1	从机2
SADDR	10100100	10100111
SADEN（为0的位被忽略）	11111010	11111001
约定地址	10100x0x	10100xx1
广播地址（SADDR或SADEN）	1111111x	11111111

从机1和从机2的约定地址最低位是不同的。从机1忽略了最低位，而从机2的最低位是1。因此只与从机1通讯时，主机必须发送最低位为0的地址（10100000）。类似地，从机1的第1位为0，从机2的第1位被忽略。因此，只与从机2通讯时，主机必须发送第1位为1的地址（10100011）。如果主机需要同时与两从机通讯，则第0位为1，第1位为0，第2位被两从机都忽略，两个不同的地址用于选定两个从机（1010 0001和1010 0101）。

主机可以通过广播地址与所有从机同时通讯。这个地址等于SADDR和SADEN的位或，结果中的0表示该位被忽略。多数情况下，广播地址为0xFFh，该地址可被所有从机应答。

系统复位后，SADDR和SADEN两个寄存器初始化为0，这两个结果设定了约定地址和广播地址为XXXXXXXX（所有位都被忽略）。这有效地去除了多从机通讯的特性，禁止了自动寻址方式。这样的EUART将对任何地址都产生应答，兼容了不支持自动地址识别的8051控制器。用户可以按照上面提到的方法实现软件地址识别的多机通讯。

8.4.5 帧出错检测

当寄存器PCON中的SSTAT位为逻辑1时，帧出错检测功能才有效。3个错误标志位被置位后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

注意：SSTAT位必须为逻辑1是访问状态位（FE，RXOVR和TXCOL），SSTAT位为逻辑0时是访问方式选择位（SM0，SM1和SM2）。

发送冲突

如果在一个发送正在进行时，用户软件写数据到SBUF寄存器时，发送冲突位（SCON寄存器中的TXCOL位）置位。如果发生了冲突，新数据会被忽略，不能被写入发送缓冲器。

接收溢出

如果在接收缓冲器中的数据未被读取之前，RI清零，又有新的数据存入接收缓冲器，那么接收溢出位（SCON寄存器中的RXOV位）置位。如果发生了接收溢出，接收缓冲器中原来的数据将丢失。

帧出错

如果检测到一个无效（低）停止位，那么帧出错位（寄存器SCON中的FE）置位。

注意：在发送之前TXD，引脚必须被设置为输出高电平。

8.4.6 寄存器

Table 8.38 电源控制寄存器

87H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PCON	SMOD	SSTAT	IT41	IT40	GF1	GF0	PD	IDL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	SMOD	波特率加倍器 0: 在方式2中，波特率为系统时钟的1/256 1: 在方式2中，波特率为系统时钟的1/128
6	SSTAT	SCON[7:5]功能选择 0: SCON[7:5]工作方式作为SM0, SM1, SM2 1: SCON[7:5]工作方式作为FE, RXOV, TXCOL
5-4	IT4[1:0]	外部中断4模式设置位
3-2	GF[1:0]	用于软件的通用标志位
1	PD	掉电模式控制位
0	IDL	空闲模式控制位

Table 8.39 EUART数据缓冲器寄存器

99H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SBUF	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	SBUF[7:0]	这个寄存器寻址两个寄存器：一个移位寄存器和一个接收锁存寄存器 SBUF的写入将发送字节到移位寄存器中，然后开始传输 SBUF的读取返回接收锁存器中的内容

Table 8.40 EUART控制及状态寄存器

98H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SCON	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-6	SM[0:1]	EUART串行方式控制位, SSTAT = 0 00: 方式0, 同步方式, 固定波特率 01: 方式1, 8位异步方式, 可变波特率 10: 方式2, 9位异步方式, 固定波特率 11: 方式3, 9位异步方式, 可变波特率
7	FE	EUART帧出错标志位, 当FE位被读时, SSTAT位必须被置位 0: 无帧出错, 由软件清零 1: 帧出错, 由硬件置位
6	RXOV	EUART接收溢出标志位, 当RXOV位被读时, SSTAT位必须被置位 0: 无接收溢出, 由软件清零 1: 接收溢出, 由硬件置位
5	SM2	EUART多处理机通讯允许位 (第9位“1”校验器), SSTAT = 0 0: 在方式0下, 波特率是系统时钟的1/48 在方式1下, 禁止停止位确认检验, 任何停止位都会置位RI 在方式2和3下, 任何字节都会置位RI 1: 在方式0下, 波特率是系统时钟的1/12 在方式1下, 允许停止位确认检验, 只有有效的停止位 (1) 才能置位RI 在方式2和3下, 只有地址字节 (第9位 = 1) 才能置位RI
5	TXCOL	EUART发送冲突标志位, 当TXCOL位被读时, SSTAT位必须被置位 0: 无发送冲突, 由软件清零 1: 发送冲突, 由硬件置位
4	REN	EUART接收器允许位 0: 接收禁止 1: 接收允许
3	TB8	在EUART的方式2和3下发送的第9位, 由软件置位或清零
2	RB8	在EUART的方式1, 2和3下接收的第9位 在方式0下, 不使用RB8 在方式1下, 如果接收中断发生, 停止位移入RB8 在方式2和3下, 接收第9位
1	TI	EUART的传送中断标志位 0: 由软件清零 1: 由硬件置位
0	RI	EUART的接收中断标志位 0: 由软件清零 1: 由硬件置位

Table 8.41 EUART从机地址及地址屏蔽寄存器

9AH-9BH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SADDR (9AH)	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SADEN (9BH)	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0 SA	DDR[7:0]	寄存器SADDR定义了EUART的从机地址
7-0 SA	DEN[7:0]	寄存器SADEN是一个位屏蔽寄存器，决定SADDR的哪些位被检验 0: SADDR中的相应位被忽略 1: SADDR中的相应位对照接收到的地址被检验

Table 8.42 EUART波特率发生器寄存器

FDH-FEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SBRTH (FEH)	SBRTEN	SBRT.14	SBRT.13	SBRT.12	SBRT.11	SBRT.10	SBRT.9	SBRT.8
SBRTL (FDH)	SBRT.7	SBRT.6	SBRT.5	SBRT.4	SBRT.3	SBRT.2	SBRT.1	SBRT.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	SBRTEN	EUART波特率发生器使能控制位 0: 关闭 (默认) 1: 打开
6-0 7-0	SBRT[14:0]	EUART波特率发生器计数器高7位和低8位寄存器

Table 8.43 EUART波特率发生器微调寄存器

A4H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
BFINE	-	-	-	-	BFINE.3	BFINE.2	BFINE.1	BFINE.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	BFINE[3:0]	EUART波特率发生器微调数据寄存器

8.5 模/数转换模块（ADC）

8.5.1 特性

- 12位分辨率/10位分辨率可选
- 参考电压可选外接 V_{REF} 或 V_{DD}
- 最多15路模拟输入
- 启动一次ADC可以自动完成多通道转换（序列），而且每个通道都可以配置为多路模拟输入中的任意一路
- 序列可配置为单通道或多通道，一个序列最多可包含16个通道，转换结果保存在16个结果寄存器中
- 存在两种工作模式：序列转换模式和连续转换模式
- 序列转换时相邻通道转换之间的时间间隔可由软件设定
- 可由PWM或Timer2的溢出信号自动触发AD转换
- 单通道转换速率最高可达1MSPS
- 序列转换模式和连续转换模式都带比较功能

G80F980包含一个单端型、12位逐次逼近型模数转换器（ADC，Analog-to-Digit Converter），模块图如图8-5-1所示。ADC基准电压在芯片复位后默认使用 V_{DD} ，用户也可以选择外部 V_{REF} 端口的输入作为基准电压。

该模块中有15路模拟输入通道（CH0 - CH11，OP1OUT，OP2OUT， V_{BG} ），都可以编入序列中自动进行转换。结果储存在对应的结果寄存器ADDxH，ADDxL（ $x = 0 - 15$ ）中，每转换一次序列，结果寄存器的值更新一次，并作一次比较。结果寄存器和模拟输入通道之间的映射关系可以随意编程，从而组成一个转换序列，而且可以将某一模拟输入通道在序列中重复编程，从而在结果寄存器中获得此模拟输入通道连续多次转换的结果。

当启动连续转换功能时，序列自动循环启动转换，结果寄存器也不断更新，并将每次的更新值与极限值作比较。

对于单个通道，转换速率最高可达1MSPS，可由寄存器设置ADC时钟速率以及采样时间。序列中相邻通道之间的时间间隔亦可通过寄存器设置（TGAP[2:0]）。

8.5.2 ADC模块图

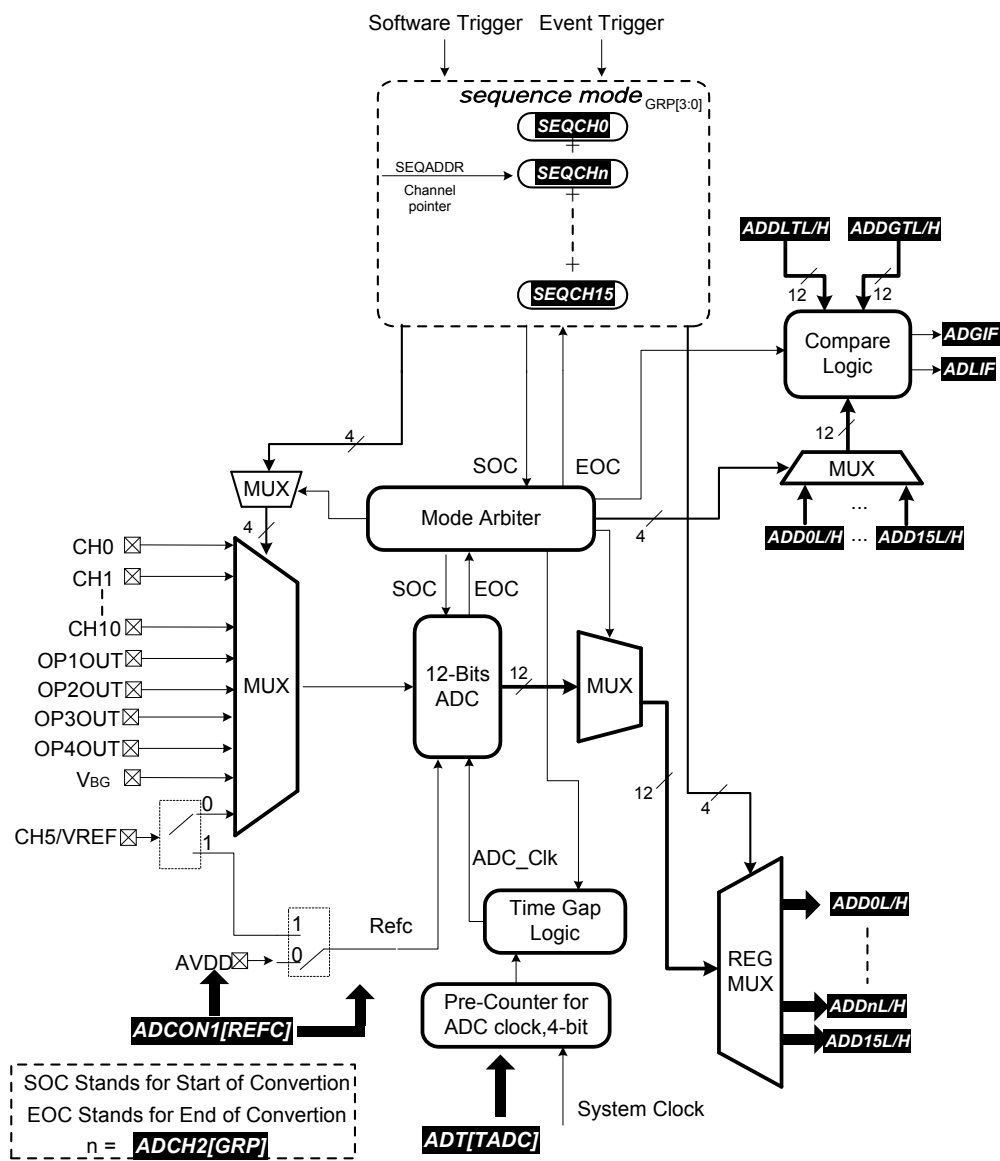


图8-5-1 ADC模块图

8.5.3 ADC寄存器

ADC模块所使用寄存器如下:

功能	名称	寄存器描述
ADC时钟设置	ADT	设置ADC时钟与采样时间、ADC精度模式选择
ADC控制	ADCON1	AD模块使能、启动、参考电压的选择、及序列转换完成中断标志、事件触发设置、转换完成中断允许、连续转换功能使能位
	ADCON2	序列通道总数设置、相邻通道之间时间间隔设置
ADC采样通道指针	SEQADDR	ADC当前采样通道指针
序列控制及映射AD通道配置	SEQCON	序列采样间隔及通道映像控制
AD通道配置1	ADCH1	设置AD通道引脚为AD通道功能或I/O功能
AD通道配置2	ADCH2	设置AD通道引脚为AD通道功能或I/O功能
通道和转换顺序设置	SEQCHx	指定序列中的通道以及转换顺序, x = 0 - 15
ADC结果寄存器	ADDxL	SEQCHx中指定通道转换值的低位, x = 0 - 15
	ADDxH	SEQCHx中指定通道转换值的高位, x = 0 - 15
上限比较寄存器	ADDGTL	上限比较寄存器的低位
	ADDGTH	上限比较寄存器的高位
下限比较寄存器	ADDLTL	下限比较寄存器的低位
	ADDLTH	下限比较寄存器的高位
比较控制寄存器	ADCMPCON	设置比较中断以及用于比较的结果寄存器

Table 8.44 ADC时钟控制寄存器

FF80H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADT	TADC3	TADC2	TADC1	TADC0	TS3	TS2	TS1	TS0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-4	TADC [3:0]	ADC时钟周期选择 0000: ADC时钟周期 $t_{AD} = 1 t_{SYS}$ 0001: ADC时钟周期 $t_{AD} = 2 t_{SYS}$ 0010: ADC时钟周期 $t_{AD} = 3 t_{SYS}$ 0011: ADC时钟周期 $t_{AD} = 4 t_{SYS}$ 0100: ADC时钟周期 $t_{AD} = 5 t_{SYS}$ 0101: ADC时钟周期 $t_{AD} = 6 t_{SYS}$ 0110: ADC时钟周期 $t_{AD} = 8 t_{SYS}$ 0111: ADC时钟周期 $t_{AD} = 12 t_{SYS}$ 1000: ADC时钟周期 $t_{AD} = 16 t_{SYS}$ 1001: ADC时钟周期 $t_{AD} = 24 t_{SYS}$ 1010: ADC时钟周期 $t_{AD} = 32 t_{SYS}$ 1011: ADC时钟周期 $t_{AD} = 48 t_{SYS}$ 1100: ADC时钟周期 $t_{AD} = 64 t_{SYS}$ 1101: ADC时钟周期 $t_{AD} = 96 t_{SYS}$ 1110: ADC时钟周期 $t_{AD} = 128 t_{SYS}$ 1111: ADC时钟周期 $t_{AD} = 192 t_{SYS}$
3-0	TS[3:0]	采样时间选择 $2 t_{AD} \leq \text{采样时间} = (\text{TS} [3:0]+1) * t_{AD} \leq 15 t_{AD}$

注意:

1. 即使TS[3:0] = 0000, 最小采样时间为 $2t_{AD}$; 即使TS[3:0] = 1111, 最大采样时间为 $15t_{AD}$;
2. 在设置TS[3:0]前, 请估算连接到ADC输入引脚的串联电阻;
3. 选择 $2 * t_{AD}$ 为采样时间时, 请确保连接到ADC输入引脚的串联电阻满足输入阻抗需求, 详见8.5.10;
4. 12bit模式下一个通道的总共转换时间 = $14t_{AD} + \text{采样时间}$; 10bit模式下一个通道的总共转换时间 = $12t_{AD} + \text{采样时间}$;
5. ADC工作在10BIT模式下, 可以提高ADC的转换效率, 在10BIT模式下一个通道的转换时间比12BIT模式下能节省2个 t_{AD} 。
6. TADC设置需保证ADC时钟周期 $T_{adc} \geq 50ns$

Table 8.45 ADC控制寄存器1

93H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCON1	ADON	ADCIF	SC	REFC	ADCIE	PWM TRGEN	TIM TRGEN	GO/DONE
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	ADON	ADC允许位 0: 禁止ADC模块 1: 允许ADC模块
6	ADCIF	序列转换完成中断标志位 0: 无序列转换完成标志, 软件清0 1: 由硬件置1表示已完成序列转换 (在连续转换时, 每次序列转换完成都会置1)
5	SC	连续转换控制位 0: 连续转换功能禁止, 将GO/DONE置1启动一次序列转换并自动将制定结果进行比较 (单次序列转换) 1: 连续转换功能打开, 将GO/DONE置1后, 完成一次序列转换后自动启动下一次序列转换。两次序列转换的间隔时间也受TGAP[2:0]控制, 并且自动将转换结果进行比较, 参见ADC比较章节 (循环序列转换)
4	REZC	基准电压选位 0: 选择V _{DD} 为基准电压 1: 选择外部V _{REF} 端口输入为基准电压
3	ADCIE	序列转换完成中断允许位 0: 禁止ADC转换中断, 但是ADCIF标志不受影响 1: 允许ADC转换中断
2	PWMTRGEN	PWM0/1/2模块溢出信号启动序列转换控制位 0: 不允许此功能 1: 允许PWM0/1/2模块的触发信号启动一次AD转换序列
1	TIMTRGEN	TIMER2溢出信号启动序列转换控制位 0: 不允许此功能 1: 允许Timer2溢出信号会启动一次AD转换序列
0	GO/DONE	序列转换启动控制/状态标志位 0: 未进行序列转换/连续转换或已完成 1: SC = 0时, 置1开始序列转换, 在转换过程中保持为1, 转换完成后硬件自动清0 (若转换期间清0此位会立即终止序列转换。) SC = 1时, 置1开始连续转换, 则该位不会由硬件清0, 可以由软件清0立即终止转换

注意: 当选择外部V_{REF}端口作为基准电压时 (REFC = 1), P1.1作为V_{REF}输入引脚而非AN5输入口。

Table 8.46 ADC控制寄存器2

92H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCON2	GRP3	GRP2	GRP1	GRP0	MODE	ADDRST	-	ALR
读/写	读/写	读/写	读/写	读/写	读/写	读/写	-	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	-	0

位编号	位符号	说明
7-4	GRP[3:0]	ADC序列的通道转换个数设置位 0: 待转换序列的通道个数为1 1: 待转换序列的通道个数为2 2: 待转换序列的通道个数为3 3: 待转换序列的通道个数为4 4: 待转换序列的通道个数为5 5: 待转换序列的通道个数为6 6: 待转换序列的通道个数为7 7: 待转换序列的通道个数为8 8: 待转换序列的通道个数为9 9: 待转换序列的通道个数为10 10: 待转换序列的通道个数为11 11: 待转换序列的通道个数为12 12: 待转换序列的通道个数为13 13: 待转换序列的通道个数为14 14: 待转换序列的通道个数为15 15: 待转换序列的通道个数为16
3	MODE	ADC转换模式选择 0: 12BIT 模式 1: 10BIT模式
2	ADDRST	采样通道指针复位选择 0: 自动复位 1: 手动复位 注意: 此标志位只能在单次序列转换时有效, 循环序列转换时固定是自动复位模式
0	ALR	ADC结果左右对齐选择位 0: 结果寄存器ADDxL/H (x = 0 - 15) 中存储的12-bit结果按照左对齐存放, 高8位存放在ADDxH中, 低4位存放在ADDxL中的高4位 1: 结果寄存器ADDxL/H (x = 0 - 15) 中存储的12-bit结果按照右对齐存放, 高4位存放在ADDxH的低4位中, 低8位存放在ADDxL中 注意: 此标志位只能在GO/DONE为0时修改, GO/DONE为1 (即ADC在转换中或连续转换功能开启) 时, 此标志位无法修改

Table 8.48 ADC采样通道指针寄存器

94H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SEQADDR	-	-	-	-	ADDR3	ADDR2	ADDR1	ADDR0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	ADDR [3:0]	<p>ADC当前采样通道指针</p> <p>0000: ADC当前采样通道为SEQCH0 0001: ADC当前采样通道为SEQCH1 0010: ADC当前采样通道为SEQCH2 0011: ADC当前采样通道为SEQCH3 0100: ADC当前采样通道为SEQCH4 0101: ADC当前采样通道为SEQCH5 0110: ADC当前采样通道为SEQCH6 0111: ADC当前采样通道为SEQCH7 1000: ADC当前采样通道为SEQCH8 1001: ADC当前采样通道为SEQCH9 1010: ADC当前采样通道为SEQCH10 1011: ADC当前采样通道为SEQCH11 1100: ADC当前采样通道为SEQCH12 1101: ADC当前采样通道为SEQCH13 1110: ADC当前采样通道为SEQCH14 1111: ADC当前采样通道为SEQCH15</p>

Table 8.49 序列映射控制寄存器

A6H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SEQCON	TGAP2	TGAP1	TGAP0	REGSEL	REG3	REG2	REG1	REG0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-5	TGAP[2:0]	<p>序列中时相邻通道之间时间间隔设置位段</p> <p>TGAP[2:0]位段定义了在一次序列转换中,一个通道转换完毕到下一个通道开始采样之间的时间间隔,同时在连续转换模式中,一次序列转换完成到启动下一次序列的等待时间也受此寄存器控制</p> <p>000: 无等待时间 001: 2ADC时钟周期 010: 4ADC时钟周期 011: 8ADC时钟周期 100: 16ADC时钟周期 101: 32ADC时钟周期 110: 64ADC时钟周期 111: 128ADC时钟周期</p>
4	REGSEL	<p>ADC结果寄存器与通道寄存器映射切换控制位</p> <p>0: 直接寻址区中96H地址和97H地址作为ADC结果寄存器使用,软件读写该地址即操作寄存器ADDxL/H (x = 0 - 15) 1: 直接寻址区中96H地址作为通道寄存器使用,软件读写该地址即操作寄存器SEQCHx (x = 0 - 15)</p>

3-0	REG[3:0]	<p>结果寄存器/通道寄存器选择位</p> <p>当REGSEL = 0时:</p> <ul style="list-style-type: none"> 0: 直接寻址区中96H和97H地址映射为ADD0L和ADD0H 1: 直接寻址区中96H和97H地址映射为ADD1L和ADD1H 2: 直接寻址区中96H和97H地址映射为ADD2L和ADD2H 3: 直接寻址区中96H和97H地址映射为ADD3L和ADD3H 4: 直接寻址区中96H和97H地址映射为ADD4L和ADD4H 5: 直接寻址区中96H和97H地址映射为ADD5L和ADD5H 6: 直接寻址区中96H和97H地址映射为ADD6L和ADD6H 7: 直接寻址区中96H和97H地址映射为ADD7L和ADD7H 8: 直接寻址区中96H和97H地址映射为ADD8L和ADD8H 9: 直接寻址区中96H和97H地址映射为ADD9L和ADD9H 10: 直接寻址区中96H和97H地址映射为ADD10L和ADD10H 11: 直接寻址区中96H和97H地址映射为ADD11L和ADD11H 12: 直接寻址区中96H和97H地址映射为ADD12L和ADD12H 13: 直接寻址区中96H和97H地址映射为ADD13L和ADD13H 14: 直接寻址区中96H和97H地址映射为ADD14L和ADD14H 15: 直接寻址区中96H和97H地址映射为ADD15L和ADD15H <p>当REGSEL = 1时:</p> <ul style="list-style-type: none"> 0: 直接寻址区中96H地址映射为SEQCH0 1: 直接寻址区中96H地址映射为SEQCH1 2: 直接寻址区中96H地址映射为SEQCH2 3: 直接寻址区中96H地址映射为SEQCH3 4: 直接寻址区中96H地址映射为SEQCH4 5: 直接寻址区中96H地址映射为SEQCH5 6: 直接寻址区中96H地址映射为SEQCH6 7: 直接寻址区中96H地址映射为SEQCH7 8: 直接寻址区中96H地址映射为SEQCH8 9: 直接寻址区中96H地址映射为SEQCH9 10: 直接寻址区中96H地址映射为SEQCH10 11: 直接寻址区中96H地址映射为SEQCH11 12: 直接寻址区中96H地址映射为SEQCH12 13: 直接寻址区中96H地址映射为SEQCH13 14: 直接寻址区中96H地址映射为SEQCH14 15: 直接寻址区中96H地址映射为SEQCH15
-----	----------	--

Table 8.50 ADC通道配置寄存器1

FF81H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH1	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	CH[7:0]	通道配置 1: P4x (x = 0, 1, 3, 4), P0.7, P1x (x = 0 - 1) 和P0.3配置为模拟输入口 (ADC输入或比较器2同相输入) 0: P4x (x = 0, 1, 3, 4), P0.7, P1x (x = 0 - 1) 和P0.3作为I/O口

Table 8.51 ADC通道配置寄存器2

FF82H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH2	CH10	CH9	CH8	VBIAS	TPSEN	TPSCHOP	BGEN	BGCHOP
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	1	0	0	0	0

位编号	位符号	说明
7-5	CH[10:8]	通道配置 1: P0x (x = 4 - 6) 配置为模拟输入口 0: P0x (x = 4 - 6) 作为I/O口
4	VBIAS	辅助电路电源控制位 (Bias) 0: 关闭辅助电路电源 1: 打开辅助电路电源 <i>注: 在开启ADC, 运放及比较器前都应提前开启辅助电路电源, 在将VBIAS位置1后等待20us后再开启ADC, 运放或比较器。</i>
3	TPSEN	温度传感器开关 0: 关闭温度传感器 1: 开启温度传感器
2	TPSCHOP	温度传感器CHOP开关 0: 温度传感器CHOP功能关闭 1: 温度传感器CHOP功能开启
1	BGEN	基准源 (1.25V) 功能控制位 0: 关闭1.25V基准源V _{BG} 1: 开启1.25V基准源V _{BG} <i>注意: 基准源开启需要建立时间, 最长不超过100us, 因此当需要用到1.25V基准源时, 需要提前100us将此标志位置1</i>
0	BGCHOP	基准源CHOP功能开关 0: 基准源V _{BG} CHOP功能关闭 1: 基准源V _{BG} CHOP功能开启

注意:

V_{BG}为内部基准源, 电平为1.25V, 误差+/-20mV。

当AD通路选取V_{BG}为待转换值时, 由于V_{BG}驱动能力有限, 需要加一级CHOP开关才能精确转换, 即AD转换V_{BG}时, 需要同时打开V_{BG}和BGCHOP。

V_{BG} 主要是查看外部 V_{DD} 稳定否, V_{BG} 为内部基准电源, 即使芯片供电 V_{DD} 有抖动, V_{BG} 电压也很稳定, 所以当ADC选取 V_{DD} 作为参考源 V_{REF} 转换 V_{BG} 时, 用抖动的 V_{DD} 参考源去采样稳定的 V_{BG} , 得到的转换值也是抖动, 由此可以判断此时的 V_{DD} 供电不稳定。

下面是利用 V_{BG} 电压倒推出目前 V_{REF} 电压的公式: ($V_{BG} = 1.25V$; DV_{BG} 为 V_{BG} 的AD出码值)

$$V_{REF} = 1.25V * 4096 / DV_{BG};$$

如果选择 V_{REF} 为 V_{DD} , 则可推算出当前的 V_{DD} 电压。由于 V_{BG} 精度 $\pm 1.67\%$, 所以得出的 V_{DD} 精度也是 $\pm 1.67\%$ 。

Table 8.51 通道寄存器x (x = 0 - 15)

96H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SEQCHx	-	-	-	-	SEQx3	SEQx2	SEQx1	SEQx0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-0	SEQx[3:0]	<p>通道选择位段</p> <p>0000: 通道0 (CH0)</p> <p>0001: 通道1 (CH1)</p> <p>0010: 通道2 (CH2)</p> <p>0011: 通道3 (CH3)</p> <p>0100: 通道4 (CH4)</p> <p>0101: 通道5 (CH5)</p> <p>0110: 通道6 (CH6)</p> <p>0111: 通道7 (CH7)</p> <p>1000: 通道8 (CH8)</p> <p>1001: 通道9 (CH9)</p> <p>1010: 通道10 (CH10)</p> <p>1011: 通道选择OP1OUT</p> <p>1100: 通道选择OP2OUT</p> <p>1101: 通道选择OP3OUT</p> <p>1110: 通道选择OP4OUT</p> <p>1111: V_{BG}</p> <p>注意: 若转换通道选中V_{BG}, 需提前100us将1.25V基准源打开, 即将寄存器[ADCH2:BGEN]位置1</p> <p>若转换OP_{OUT}, 也需要提前打开OP功能, OP需要一定的建立时间(约100us)才能输出稳定的结果。</p>

注意: SEQCH0 - SEQCH15都映像在同一个地址96H上。令SEQCON寄存器中REGSEL位等于1, 同时改变SEQCON寄存器中的位段REG[3:0]的值, 可以由软件读写SEQCH0 - SEQCH15。如令REGSEL = 1, REG[3:0] = 3, 则读写直接寻址区中96H地址即读写寄存器SEQCH3。

Table 8.52 ADC结果寄存器x (x = 0 - 15)

左对齐模式:

96H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDxL	A3	A2	A1	A0	-	-	-	-
读/写	只读	只读	只读	只读	-	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	-	-	-	-
97H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDxH	A11	A10	A9	A8	A7	A6	A5	A4
读/写	只读	只读	只读	只读	只读	只读	只读	只读
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

右对齐模式:

96H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDxL	A7	A6	A5	A4	A3	A2	A1	A0
读/写	只读	只读	只读	只读	只读	只读	只读	只读
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
97H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDxH	-	-	-	-	A11	A10	A9	A8
读/写	-	-	-	-	只读	只读	只读	只读
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
7-0, 7-4/ 3-0, 7-0	A11-A0	<p>左对齐方式 (ALR = 0)</p> <p>一个通道转换完毕后, 数据立即更新并存放在ADDxL/H (x = 0 - 15) 中高8位存放在ADDxH中, 低4位存放在ADDxL的高4位, 序列转换一次所有结果寄存器也都更新了一次</p> <p>右对齐方式 (ALR = 1)</p> <p>一个通道转换完毕后, 数据立即更新并存放在ADDxL/H (x = 0 - 15) 中高4位存放在ADDxH的低4位, 低8位存放在ADDxL中, 序列转换一次所有结果寄存器也都更新了一次</p>

注意:

ADD0L - ADD15L都映像在同一个地址96H上; ADD0H - ADD15H也都映像在同一个地址97H上, 并且都为只读寄存器。令SEQCON寄存器中REGSEL位等于0, 同时改变SEQCON寄存器中的位段REG[3:0]的值, 可以由软件读ADD0L/H - ADD15L/H。如令REGSEL = 0, REG[3:0] = 5, 则读直接寻址区中96H地址即对读寄存器ADD5L, 读直接寻址区中97H地址即读寄存器ADD5H。

在10BIT模式下, 结果的存储也是按上表的方式存储, 但有效位为A11 - A2, 而A1 - A0自动填充为0, 这样做主要是考虑ADC的转换速率的提高, 10bit模式下一个通道转换时间比12BIT模式下节省2个ADC时间周期。

Table 8.53 比较控制寄存器

91H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCMPCON	ADLIE	ADGIE	ADLIF	ADGIF	CSEL3	CSEL2	CSEL1	CSEL0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	ADLIE	下限比较中断使能位 0: 禁止下限比较中断 1: 使能下限比较中断
6	ADGIE	上限比较中断使能位 0: 禁止上限比较中断 1: 使能上限比较中断
5	ADLIF	下限比较中断标志位 0: 无下限中断发生, CSEL位段所指定结果寄存器中的最近一次更新值大于ADDLTH/L中的值 1: 发生下限中断, CSEL位段所指定结果寄存器中的最近一次更新值小于等于ADDLTH/L中的值 该标志置起后需由软件清0
4	ADGIF	上限比较中断标志位 0: 无上限中断发生, CSEL位段所指定结果寄存器中的最近一次更新值小于ADDGTH/L中的值 1: 发生上限中断, CSEL位段所指定结果寄存器中的最近一次更新值大于等于ADDGTH/L中的值 该标志置起后需由软件清0
3-0	CSEL[3:0]	用于比较的结果寄存器选择位段 0: 选择ADD0L/H中的值与ADDGTL/H、ADDLTL/H比较 1: 选择ADD1L/H中的值与ADDGTL/H、ADDLTL/H比较 2: 选择ADD2L/H中的值与ADDGTL/H、ADDLTL/H比较 3: 选择ADD3L/H中的值与ADDGTL/H、ADDLTL/H比较 4: 选择ADD4L/H中的值与ADDGTL/H、ADDLTL/H比较 5: 选择ADD5L/H中的值与ADDGTL/H、ADDLTL/H比较 6: 选择ADD6L/H中的值与ADDGTL/H、ADDLTL/H比较 7: 选择ADD7L/H中的值与ADDGTL/H、ADDLTL/H比较 8: 选择ADD8L/H中的值与ADDGTL/H、ADDLTL/H比较 9: 选择ADD9L/H中的值与ADDGTL/H、ADDLTL/H比较 10: 选择ADD10L/H中的值与ADDGTL/H、ADDLTL/H比较 11: 选择ADD11L/H中的值与ADDGTL/H、ADDLTL/H比较 12: 选择ADD12L/H中的值与ADDGTL/H、ADDLTL/H比较 13: 选择ADD13L/H中的值与ADDGTL/H、ADDLTL/H比较 14: 选择ADD14L/H中的值与ADDGTL/H、ADDLTL/H比较 15: 选择ADD15L/H中的值与ADDGTL/H、ADDLTL/H比较

Table 8.54 ADC上限比较寄存器

FF86H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDGTL	GT7	GT6	GT5	GT4	GT3	GT2	GT1	GT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
FF85H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDGTH	GT15	GT14	GT13	GT12	GT11	GT10	GT9	GT8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0 7-0	GT15 - GT0	当一次序列转换完成时，ADCMPCON寄存器中CSEL位段指定结果寄存器的值立刻与ADDGTL/H中的值比较，若大于等于ADDGTL/H中的值则令ADCMPCON中的ADGIF位置1，ADGIF位将保持为1直到软件将其清除

Table 8.55 ADC下限比较寄存器

FF84H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDLTL	LT7	LT6	LT5	LT4	LT3	LT2	LT1	LT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
FF83H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADDLTH	LT15	LT14	LT13	LT12	LT11	LT10	LT9	LT8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0 7-0	LT15- LT0	当一次序列转换完成时，ADCMPCON寄存器中CSEL位段指定结果寄存器的值立刻与ADDLTL/H中的值比较，若小于等于ADDLTL/H中的值则令ADCMPCON中的ADLIF位置1，ADLIF位将保持为1直到软件将其清除

8.5.3 单序列转换工作模式

SC = 0, ADC配置为单序列转换模式, 一次转换一个序列。序列由单个通道或多个通道组成, 对序列进行转换即对序列中的通道进行逐个转换。在硬件上, 使得多个信号在同一时间点上转换得以实现(2通道间最短采样间隔1us, 可近似看作同时)。

转换的结果储存在对应的结果寄存器中, 序列转换完成的同时, 把指定的某个结果寄存器的值与ADDGTL/H和ADDLTL/H的值比较, 并以标志指示比较的结果。

单序列转换模式下, 当前转换通道由SEQADDR寄存器指定, 每次单通道转换结束后该寄存器自动加1, 随后进行下一通道转换, 当单序列转换通道个数达到设定值(GRP[3:0])时, 本次序列转换结束, GO/DONE位清零, 表明本次序列转换完成; 同时ADCON1中的ADCIF位将由硬件置起, 此时若ADCON1中ADCIE位为1, 同时IEN0寄存器中的EADC和EA位为1, 则将触发序列转换完成中断, ADCIF标志需要软件清除。

每次启动ADC序列转换(GO/DONE位置1)时, 若采样通道指针复位选择为0(ADDRST=0), 则在开始转换前先自动复位SEQADDR寄存器为0, 然后再开始从通道0开始转换; 例: 有4路通道需要转换, 按优先级排序为CH0, CH2, CH4, CH9, 则设置为GRP[3:0]=3, SEQCH0=0, SEQCH1=2, SEQCH2=4, SEQCH3=9, 则每次启动AD转换(GO/DONE=1)时, SEQADDR都被复位为0, 随后4个通道将依次转换, 结果将依次存在ADD0L/H, ADD1 L/H, ADD2 L/H, ADD3 L/H中。

若采样通道指针复位选择为1(ADDRST=1), 则ADC直接从由SEQADDR寄存器指定的通道开始转换, 当转换个数达到(GRP[3:0])设定值时, 本次序列转换结束。例: 有4路通道需要分两次转换, 按优先级排序为CH0, CH2, CH4, CH9, 则设置SEQCH0=0, SEQCH1=2, SEQCH2=4, SEQCH3=9, 而设置SEQADDR=0, GRP[3:0]=1, 则第一次启动AD转换(GO/DONE=1)时, CH0和CH2将依次转换, 结果依次存在ADD0L/H, ADD1 L/H中, 转换结束后SEQADDR值应该为2。而再次启动AD转换时, CH4和CH9将依次转换, 结果依次存在ADD2 L/H, ADD3 L/H中, 转换结束后SEQADDR值应该为4, 以此类推。如果此时将SEQADDR改写为0, 则下次转换恢复从通道0开始; 若不改写SEQADDR, 则下次转换从通道4开始, 若通道4的预设模拟通道未进行过设置, 则可能得到非预期结果, 因此程序对该寄存器的改写要小心处理(ADC转换期间(GO/DONE=1), SEQADDR无法修改)。

ADC采样通道序号最大为15, 因此当转换通道指针寄存器SEQADDR超过15时, 会自动归0。例: 若设置为GRP[3:0]=3, SEQCH0=0, SEQCH1=2...SEQCH14=4, SEQCH15=9, SEQADDR=14, ADDRST=1, 则启动AD转换(GO/DONE=1)后, 当前转换通道由SEQADDR寄存器指定, 将依次转换CH4, CH9, CH0, CH2, 结果将依次存在ADD14L/H, ADD15 L/H, ADD0 L/H, ADD1 L/H中。

序列转换中通道的选择设置

一个序列可包含单个或多个通道, 将待转换的通道号分别存放在通道寄存器SEQCHx中(x=0-15), 寄存器SEQCHx有16组, 所以序列转换时, 一次最多可转换16路通道。而每次转换的通道数量由寄存器ADCH2中的GRP[3:0]的值决定。例: GRP[3:0]=0, 即单通道转换, 将SEQCH0中存放的通道进行转换; GRP[3:0]=3, 则序列中有4路通道, 将SEQCH0到SEQCH3中存放的通道依次转换。

通道寄存器SEQCHx(x=0-15)中存放的即是待转换的通道号。例: 有3路通道需要转换, 按优先级排序为OP1OUT, CH2, CH9, 则设置为GRP[3:0]=2, SEQCH0=12, SEQCH1=2, SEQCH2=9, 将依次转换。(注意: 需要转换的通道必须配置为AD通道功能。SEQCHx中也可设置相同的通道号, 比如将SEQCHx中的值全设置为CH3, 结果寄存器中储存的将是不同时间段的CH3的转换值。)

寄存器SEQCHx(x=0-15)与ADDxL共享一个地址96H, 访问时需要由SEQCON寄存器中的REGSEL位和REG[3:0]位区分, REGSEL=1时, 读写直接寻址寄存器96H即访问通道转换寄存器SEQCHx, 而REG[3:0]的值确定x的值。这里REG[3:0]=0即访问SEQCH0寄存器, 依次类推, 可设置SEQCH0-SEQCH15中的通道号。

序列转换模式ADC结果寄存器

序列转换的结果依次储存在结果寄存器ADDxL/H(x=0-15)中, 结果寄存器为只读寄存器。由寄存器SEQCON中的ALR位决定转换结果的存储方式, ALR=0(默认)时结果按左对齐存储在ADDxL/H中, ALR=1时结果按右对齐存储在ADDxL/H中, 如图8-5-2。一个序列转换完毕, 结果寄存器ADDxL/H中的值也更新了一次。由于ADDxL和SEQCHx(通道寄存器)共享一个地址96H, 需要由SEQCON寄存器中的REGSEL位和REG[3:0]位区分。REGSEL=0时, 读直接寻址区的97H, 96H即访问结果寄存器ADDxL/H, 而REG[3:0]的值确定x的值。比如REG[3:0]=0, 访问直接寻址区97H, 96H即访问ADD0L/H。以上节中的3路通道转换为例, 按优先级排序为OP1OUT, CH2, CH9, 结果依次储存在ADDxL/H(x=0-2), 如图8-5-3。(注意: 如果想转换上述的3个通道, 而GRP[3:0]=1, 则最后一个SEQCH2中的通道9则不转换, 读取ADD2L/H的值是无效值; 相反如果设置了GRP[3:0]=3, 则多转换了一路通道, 第4路通道是SEQCH3中的值对应的通道。)

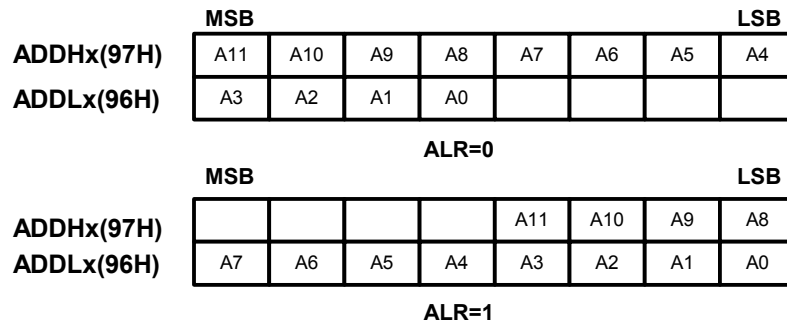


图8-5-2

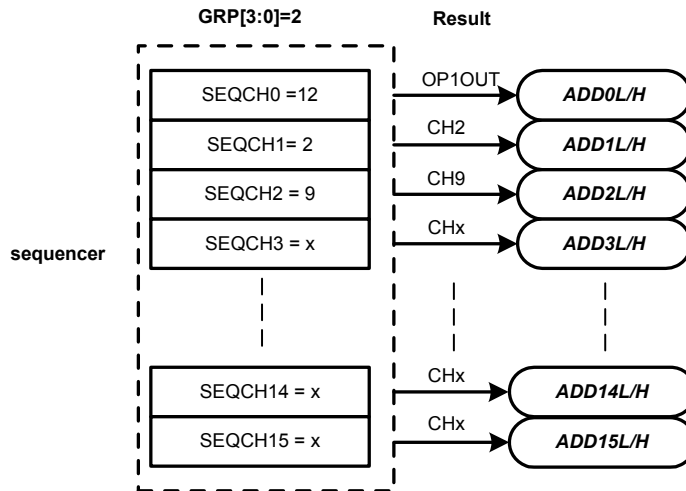


图8-5-3

序列转换时通道之间时间间隔（Gap Time）的设置

序列转换时，从上一个通道转换完毕时刻到下一个通道开始采样时刻之间的时间可以通过寄存器ADCON2中的TGAP位段来设置。当TGAP[2: 0] = 0时，一个通道转换完毕后立即开始下一个通道的采样，之间没有等待时间。

序列转换模式的启动和停止

序列转换模式的启动可以分为软件启动和硬件启动。

软件启动，通过将ADCON1中的ADON位置1，这样可以使能ADC模块的时钟ADC_CLK，同时令ADON位置1可以令ADC模块中的模拟电路上电。当ADCON1寄存器中的GO/DONE位置1，则启动序列转换。当一次序列转换完成，由硬件将GO/DONE清0，同时进行比较（在8.5.5节中详细阐述）。当读取GO/DONE为1时，则标志转换进行未完成。如果在转换过程中清0 GO/DONE位，则终止转换。

硬件启动有2种方式，TIMER2的溢出触发启动一次序列转换和PWM0/1/2模块触发信号启动一次序列转换。具体通过设置ADCON1寄存器中的PWMTRGEN位和TIMETRGEN位以及PWM0/1/2模块的相关寄存器。

硬件触发信号优先于软件触发信号。当软件已经处于序列转换中时，硬件触发会终止先前的序列转换而重新启动一次序列转换，转换值将会覆盖之前的结果。另外如果在连续转换期间，有硬件触发信号过来，会清0SC终止连续转换，并重新触发一次序列转换。建议在使用硬件触发功能时，将SC清0。

PWM0/1/2模块的触发序列转换过程中，又进来一次TIMER2的触发信号，则序列将会终止先前的转换重新启动一次，转换值将覆盖之前的结果；反之TIMER2触发序列的转换过程中，又进来一次PWM0/1/2的触发信号，则也会终止先前的转换，重新启动一次序列，转换值覆盖之前的结果。

注意：在序列转换过程中，即GO/DONE为1时，对除了ADCON1、ADDGTL/H、ADDLTL/H以外的所有ADC寄存器的写入都将视为无效操作；ADCMPCON寄存器的CSEL[3:0]位在GO/DONE为1时，只在序列循环转换模式下可修改，单次序列转换过程中不可修改。

序列转换完成中断

序列转换完毕后，ADCON1中的ADCIF位将由硬件置起，此时若ADCON1中ADCIE位为1，同时IEN0寄存器中的EADC和EA位为1，则将触发序列转换完成中断，ADCIF位只能由软件清除。

比较中断

序列转换完的同时，会将指定结果寄存器的值去比较，若超限，也可以产生一个比较中断，因为比较和序列转换完成是同时发生的，所以中断的产生也是同时，并且共用一个中断入口。（在8.5.5节中详细阐述）

8.5.4 连续转换工作模式

SC = 1选择连续转换工作模式，序列循环转换，转换完一个序列便进行此序列的下一序列的转换。连续转换的一些寄存器设置可参考“序列转换工作模式”，和序列转换不同的是该模式将循环转换序列，一个序列转换完成到下个序列开始转换的时间也是由TGAP[2:0]控制的。每次序列转换完成的同时，把指定的某个结果寄存器的值与ADDGTL/H和ADDLTL/H的值比较，并以标志指示比较的结果。

连续转换功能的启动和停止

连续转换功能只有通过软件启动，在SC = 1时，通过置位GO/DONE来启动，通过软件清0 SC位来终止。当清0SC位时，AD转换将在当前序列转换结束时停止，并且GO/DONE自动清0，同时进行比较；而手动清GO/DONE会立刻终止转换。

启动连续转换功能后，ADCIF会在每次序列转换完成时置起，比较功能也会在每次序列转换完成时进行。如果发生指定的结果寄存器值超限，会置位相应的标志位ADGIF、ADLIF，如果相应的中断允许位开启，则会产生中断。

连续转换启动时，SEQADDR采样通道指针会自动复位为0，在此模式下采样通道指针手动复位（ADDRST设为1）无效。每次单通道转换结束后SEQADDR寄存器自动加1，随后进行下一通道转换，当转换通道个数达到设定值(GRP[3:0])时，SEQADDR采样通道指针会自动复位为0，随后立即启动下一序列转换。

注意：在连续转换过程中，即GO/DONE为1时，对除了ADCON1、SEQCON、ADCMPCON、ADDGTL/H、ADDLTL/H以外的所有ADC寄存器的写入都将视为无效操作。

序列转换完成中断

每次序列转换完毕后，ADCON1中的ADCIF位都会置起，此时若ADCON1中ADCIE位为1，同时IEN0寄存器中的EADC和EA位为1，则将触发序列转换完成中断，此时ADCIF位必须在中断程序中由软件清除。

比较中断

每次序列转换完成时，都会将指定结果寄存器的值去作比较，如果超限，也可以产生一个比较中断。因为比较和序列转换完成是同时发生的，所以中断的产生也是同时，并且共用一个中断入口。（在8.5.5节中详细阐述）

8.5.5 比较功能

比较用结果寄存器的指定

由寄存器ADCMPCON中的CSEL位段来指定被比较的结果寄存器，若CSEL = n，则结果寄存器ADDnL/H将与ADDGTL/H和ADDLTL/H的值比较。以图8-5-3的配置为例，若设置CSEL = 2，每次序列转换完成的同时将ADD2L/H的值与ADDGTL/H和ADDLTL/H的值比较。

需要注意的是，若指定一个序列转换不使用的结果寄存器，则不会发生任何比较动作。如上例中令CSEL = 3，转换后未使用结果寄存器ADD3L/H，故没有任何比较动作发生，标志位ADGIF和ADLIF的值不会发生变化。

注意：在写入比较值ADDGTL/H和ADDLTL/H时，应注意结果寄存器的存储方式是左对齐还是右对齐，数据写入格式应和结果寄存器的存储格式一致，如果是在10BIT模式下比较，则写入的值只比较A11 - A2前10位，剩下2位不作比较。ADDGTL/H和ADDLTL/H写入值有即时生效性，比较时会采用ADDGTL/H和ADDLTL/H最近一次的更新值。

比较过程

设CSEL = n，ADDGTL/H = Max，ADDLTL/H = Min，当序列转换完成时，立即将指定结果寄存器ADDnL/H的值与Max和Min比较。若ADDnL/H >= Max，则ADCMPCON寄存器中ADGIF位将置1，若ADGIF位为1，则可以触发一个比较中断（与序列转换完成共享一个中断向量），ADGIF位将保持为1直到软件将其清除；若ADDnL/H <= Min，则ADCMPCON寄存器中ADLIF位将置1，若ADLIF位为1，也可以触发一个比较中断（与序列转换完成共享一个中断向量），ADLIF位将保持为1直到软件将其清除。

设ADDGTL/H = 0x3DF，ADDLTL/H = 0x8FE，CSEL = n，则ADDnL/H的值与ADGIF、ADLIF位中值的关系如图8-5-4所示；设ADDGTL/H = 0x8FE，ADDLTL/H = 0x3DF，CSEL = n，则ADDnL/H的值与ADGIF、ADLIF位中值的关系如图8-5-5所示。

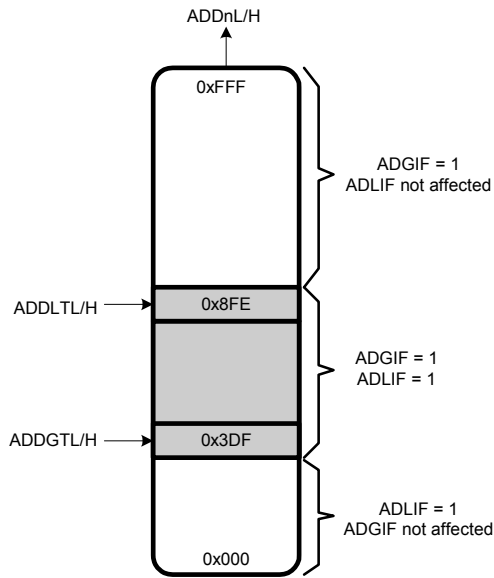


图8-5-4

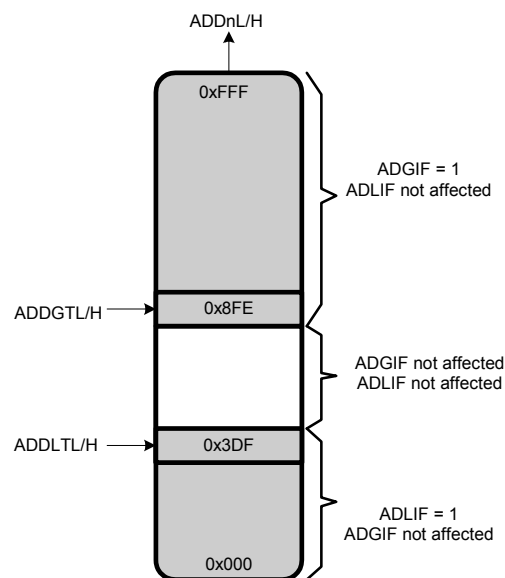


图8-5-5

8.5.6 ADC转换时间设置

通过寄存器ADT可以设置ADC的时钟以及采样时间。通过设置ADT中的TADC[3:0]位段可以设置ADC的时钟。

寄存器ADT中的TS[3:0]位段可以设置每个通道的采样时间 t_{SAMP} ， $t_{SAMP} = (TS[3:0]+1) * t_{AD}$ ，具体见ADC寄存器章节。需要注意的是，无论TS[3:0]取何值，采样时间的最小值不会小于 $2 * t_{AD}$ ，最大值不会大于 $15 * t_{AD}$ 。即若TS[3:0] = 0，采样时间亦为 $2 * t_{AD}$ ；若TS[3:0] = 15，采样时间亦为 $15 * t_{AD}$ 。

对于12BIT模式AD转换每个通道的AD转换时间固定为 $14 * t_{AD}$ 。因此每个通道的总共转换时间 = $t_{SAMP} + 14 * t_{AD}$ 。对于一个序列，若寄存器ADCON1中的位段TGAP[2:0]不为零，则序列中相邻的两个通道转换之间将加入一段时间 T_G ，这样若一个序列总共转换的通道数为n，则完成整个序列的转换所需时间为： $n * (t_{SAMP} + 14 * t_{AD}) + (n - 1) * T_G$ 。如图8-5-3中序列的转换时间为 $3 * (t_{SAMP} + 14 * t_{AD}) + 2 * T_G$ 。（注意：10BIT模式下AD转换每个通道的固定时间为 $12 t_{AD}$ ，其他时间计算以此为基础）

8.5.7 ADC模块参考电压的设置

可以选择芯片 V_{DD} 或者 V_{REF} 引脚上的电压作为ADC模块的参考电压。芯片复位后，ADC模块的参考电压为芯片的 V_{DD} 。令寄存器ADCON1中的REFC位置1将使ADC模块使用 V_{REF} 引脚上的电压作为参考电压（此时CH3通道无效，若转换CH3的电平，转换出来的值没有意义）。

8.5.8 ADC通道与IO口功能设置

15个AD通道中12路都与IO功能复用（CH0 - CH11），芯片复位后，12路AD通道都是作为IO功能使用。使用ADC模块之前应该通过寄存器ADCH1、ADCH2将需要使用的通道设置为AD通道功能。具体设置参见“ADC通道配置寄存器”。

另外，此12路中有4路（AN0 - AN2, AN4）与比较器2的端口共用，它们可以在作AD输入模拟通道的同时，作比较器2的输入（详见端口共用章节）。

8.5.9 ADC结果寄存器/通道寄存器之间的映射关系

ADDxL寄存器和SEQCHx寄存器都映像在直接寻址区的96H地址上， $x = 0 - 15$ 。通过寄存器SEQCON中的位REGSEL和位段REG[3:0]可以具体制定操作ADDxL/H、SEQCHx中的任一个。

若REGSEL = 0、REG[3:0] = y，则直接寻址区中的96H映像至ADDyL寄存器，97H映像至ADDyH寄存器，对直接寻址区96H和97H的读操作即对ADDyL/H寄存器的读操作。

若REGSEL = 1、REG[3:0] = y，则直接寻址区中的96H映像至SEQCHy寄存器，对直接寻址区96H的读写操作即对SEQCHy寄存器的读写操作。

8.5.10 AD转换时对传感器输出阻抗的要求

如图8-5-6所示，为了保证AD转换在一定的转换速率的条件下能转换出精确的AD结果，需要在一定时间内（采样时间TS）对G80F980内部采样电容C完成充电。如果在规定的采样时间内，采样电容C未能充满电，即 $V_0 < V_1$ ，AD转换结果则会产生精度误差。因为 R_0 （G80F980内部电阻）和 C （内部采样电容）是常量，已经由芯片内定，能否在转换成允许精度范围内的AD结果就取决于传感器内部等效电阻 R_x 。设AD转换结果的分辨率为 M （12bit结果的分辨率为4096，10bit结果的分辨率为1024），如果AD转换的精度要求为 N ，则对传感器电阻的选择有如下要求：

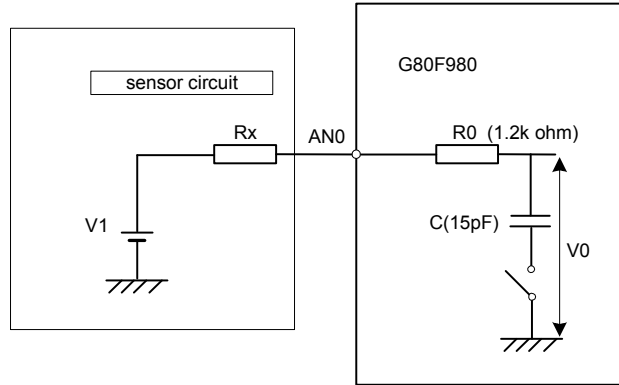


图8-5-6

$$R_x \leq -\frac{T}{C * \ln \frac{V_1 - V_0}{V_1}} - R_0$$

上式也可表示为：

$$R_x \leq -\frac{T}{C * \ln \frac{N}{M}} - R_0$$

对于10bit模式下即1024分辨率，标准采样时间为0.5us，结果精度误差控制在0.1LSB内时，按上述公式带入值有

$$R_x \leq -\frac{0.5 \times 10^{-6}}{15 \times 10^{-12} \ln \frac{0.1}{1024}} - 1.2 \times 10^3 \approx 2.4 K\Omega$$

即传感器选型时的内部等效电阻必须小于2.4KΩ。

对于12bit模式下即4096分辨率，标准采样时间为2us，结果精度误差控制在0.1LSB内时，按上述公式带入有

$$R_x \leq -\frac{2 \times 10^{-6}}{15 \times 10^{-12} \ln \frac{0.1}{4096}} - 1.2 \times 10^3 \approx 11.3 K\Omega$$

即传感器选型时的内部等效电阻必须小于11.3KΩ。

8.5.11 注意事项

ADC开启前需启动辅助电源电路，先将VBIAS置1后等待10us辅助电源稳定后，再开启ADC。

ADC转换，最好将ADON打开10us后再将GO/DONE位置1，进行转换。因为ADC模块打开后有个稳定时间。

如需要转换增益放大通路，需要先打开OP功能，OP需要建立时间（约为100us）才能稳定输出稳定，再进行AD转换，转换时选取OPUT1或OPOUT2通路。

转换模拟通路时要先将端口设置为AD通道功能才能顺利转换。

转换单个通道时，只要将GRP[3:0] = 0，并设置SEQCH0为需要转换的通道号，即可进行转换。

当ADCIE，ADLIE，ADGIE的都开启时，ADCIF，ADLIF，ADGIF任意一个置位都能引起中断，且共享一个中断向量，通过判位ADCIF，ADLIF，ADGIF哪个为1，来确定具体的中断源并定制特定的操作。

8.5.12 程序示例

ADC示例:

/*连续转换AN3, AN1, AN11三个通道的模拟电压值, 并将AN11作比较, 如果AN11超出(0x123~0x678)范围则报警, 并将此时AN3, AN1, AN11此时的值取出保存并停止连续转换*/

```
#include"G80F980.h"
unsigned int ad_AN1, ad_AN3, ad_AN11;
bit alarm_flag;
void ADC_inter () interrupt 6
{
    if(ADCMPCON&0x30)
    {
        ADCON1&=0xfe; //停止转换
        ADCMPCON&=0xcf; //清比较FLAG
        alarm_flag=1;
        SEQCON=0x80; //映射寄存器切换至结果寄存器第0通道右对齐方式取出
        ad_AN3=ADDx_16;
        SEQCON=0x81; //映射寄存器切换至结果寄存器第1通道右对齐方式取出
        ad_AN1=ADDx_16;
        SEQCON=0x82; //映射寄存器切换至结果寄存器第2通道右对齐方式取出
        ad_AN11=ADDx_16;
    }
}
ADC_initial()
{
    ADCON1=0xa0; //连续转换模式
    ADCH1=0x0a;
    ADCH2=0x80;
    ADT=0x27;
    IEN0=0xc0;
    ADCON2=0x27; //选取12bit转换模式, 总的转换通道数为3,
    SEQCON=0x10; //映射寄存器切换至通道寄存器第0通道
    SEQCHx=0x03; //选取AN3
    SEQCON=0x11; //映射寄存器切换至通道寄存器第1通道
    SEQCHx=0x01; //选取AN1
    SEQCON=0x12; //映射寄存器切换至通道寄存器第2通道
    SEQCHx=0x0B; //选取AN11
    SEQCON|=0x80; //右对齐方式
    ADCMPCON=0xc2; //选取通道2中的AN11作比较, 允许比较中断
    ADDG_16=0x0678; //设置上限值
    ADDL_16=0x0123; //设置下限值
    ADCON1|=0x01; //开启转换
}
void main()
{
    System_initial();
    ADC_initail();
    while (1)
    {
        if(alarm_flag)
        {
            ;
        }
    }
}
```

8.6 温度传感器 (Temperature Sensor)

8.6.1 特性

- 内置温度传感器，测量范围 -40~150℃
- 可作为OP1输入并通过AD读取结果

G80F980含一个温度传感器转换器，可通过TPSEN位控制，温度传感器可选择与内部OP1同相端相连，可通过选择OP1内部增益系数以及AD序列配置将温度通道（OP1OUT通道）转换为数字值。

为了更准确读取温度传感器数值，在选择温度传感器作为内部OP1同相输入端同时，建议同时使用温度传感器和OP1的CHOP功能，即在每次转换时同时切换温度传感器与OP1的CHOP的开启与关闭，在CHOP开启/关闭状态下分别得到温度传感器AD转换数值，取平均后即可得到更准确的温度转换结果。

$$\text{温度 } T = (V_{25} - V_{\text{TPS}}) / T_{\text{slope}} + 25$$

V_{25} 为25度时温度传感器输出电压

V_{TPS} 为温度传感器输出电压

T_{slope} 为温度传感器斜率

8.6.2 温度传感器校准

G80F980每颗芯片出厂后都校准温度传感器（分别将生产时实际温度及TPSCHOP关闭和开启时采样的AD值存放在地址信息存储区0x1275 – 127A），可以由程序或编程工具读出。将实时AD转换温度与校准时存储AD值相减后加上校准时温度值可以得到真实的温度。（AD值为将温度传感器接入OP1同相端后，选择内部增益系数2倍，再将OP1OUT通道做AD转换得到的数值）

校准时温度传感器实际温度值存于1275H – 1276H，校准时G80F980温度传感器TPSCHOP关闭和开启时AD值分别1277H – 127AH。校准时实际温度保留1位小数，按Q4格式存放（即储存值=温度×2⁴，16进制）；采样AD值存放格式为12bit模式下右对齐值。

程序读出示例：

```
unsigned char TempH, TempL, TempADH0, TEMPADL0, TempADH1, TEMPADL1
FLASHCON = 0x01;
TempH = CBYTE[0x1275]; //校准时温度值
TempL = CBYTE[0x1276];
TempADH0 = CBYTE[0x1277]; //校准时存储的AD值 CHOP=0
TempADL0 = CBYTE[0x1278];
TempADH1 = CBYTE[0x1279]; //校准时存储的AD值 CHOP=1
TempADL1 = CBYTE[0x127A];
FLASHCON = 0x00;
```

温度传感器输出电压随温度线性变化，但不同芯片的曲线偏移会有不同，尽管已包含了上述温度传感器的校准方法，但实际温度也因不同芯片不同温度下误差的差异以及芯片本身功耗引起发热造成的温升与环境温度的差异，因此温度传感器适合于检测温度的变化，而不是测量绝对温度，如需要测量精确的温度，应该使用外置的温度传感器。

8.6.3 寄存器

Table 8.51 ADC通道配置寄存器2

FF82H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
ADCH2	-	-	-	-	TPSEN	TPSCHOP	-	-
读/写	-	-	-	-	读/写	读/写	-	-
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	-	-

位编号	位符号	说明
3	TPSEN	温度传感器开关 0: 温度传感器关闭 1: 温度传感器开启

2	TPSCHOP	温度传感器CHOP开关 0: 温度传感器CHOP功能关闭 1: 温度传感器CHOP功能开启
---	---------	---

注：温度传感器从关闭到开启，有一个建立过程，从TPSEN置为1开始，至少等待100us再开始AD转换读取转换结果

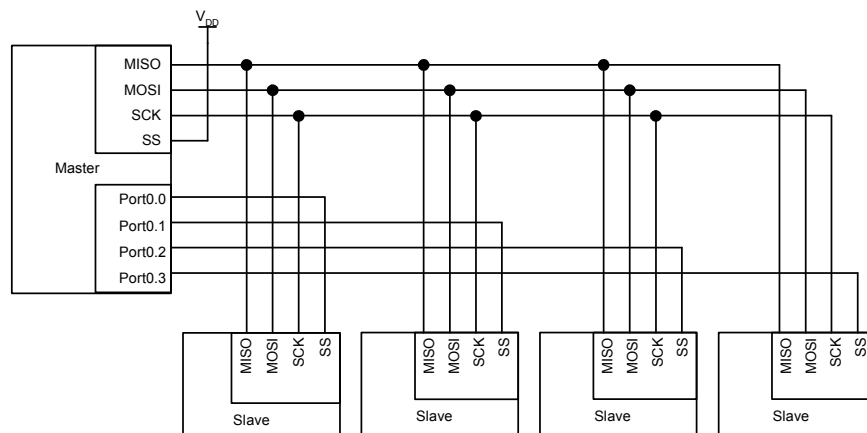
8.7 串行外部设备接口 (SPI)

8.7.1 特性

- 全双工，三线同步传输
- 主从机操作
- 8个可编程主时钟频率
- 极性相位可编程的串行时钟
- 带MCU中断的主模式故障出错标志
- 写入冲突标志保护
- 可选择LSB或MSB传输

串行外部设备接口 (SPI) 是一种高速串行通信接口，允许MCU与外围设备（包括其它MCU）进行全双工，同步串行通讯。

下图所示即为典型的由一个主设备和若干从属外部设备组成的SPI总线网络，主设备通过3条线连接所有从设备，主设备控制连接从属设备SS引脚的4个并行端口来选中其中一个从属设备进行通讯。



8.7.2 信号描述

主输出从输入 (MOSI)

该路信号连接主设备和从设备。数据通过MOSI从主设备串行传送到从设备，主设备输出，从设备输入。

主输入从输出 (MISO)

该路信号连接从设备和主设备。数据通过MISO从从设备串行传送到主设备，从设备输出，主设备输入。当SPI配置为从设备并未被选中 (SS引脚为高电平)，从设备的MISO引脚处于高阻状态。

SPI串行时钟 (SCK)

SCK信号用作控制MOSI和MISO线上输入输出数据的同步移动。每8时钟周期线上传送一个字节。如果从设备未被选中 (SS引脚为高电平)，SCK信号被此从设备忽略。

从设备选择引脚 (SS)

每个从属外围设备由一个从选择引脚 (SS引脚) 选择，当引脚信号为低电平时，表明该从设备被选中。主设备可以通过软件控制连接于从设备SS引脚的端口电平选择每个从设备，很明显，只有一个主设备可以驱动通讯网络。为了防止MISO总线冲突，同一时间只允许一个从设备与主设备通讯。在主设备模式中，SS引脚状态关联SPI状态寄存器SPSTA中MODF标志位以防止多个主设备驱动MOSI和SCK。

下列情况，SS引脚可以作为普通端口或其它功能使用：

(1) 设备作为主设备，SPI控制寄存器SPCON寄存器的SSDIS位置1。这种配置仅仅存在于通讯网络中只有一个主设备的情况，因此，SPI状态寄存器SPSTA中MODF标志位不会被置1。

(2) 设备配置为从设备，SPI控制寄存器SPCON的CPHA位和SSDIS位置1。这种配置情况存在于只有一个主设备一个从设备的通讯网络中，因此，设备总是被选中的，主设备也不需要控制从设备的SS引脚选择其作为通讯目标。

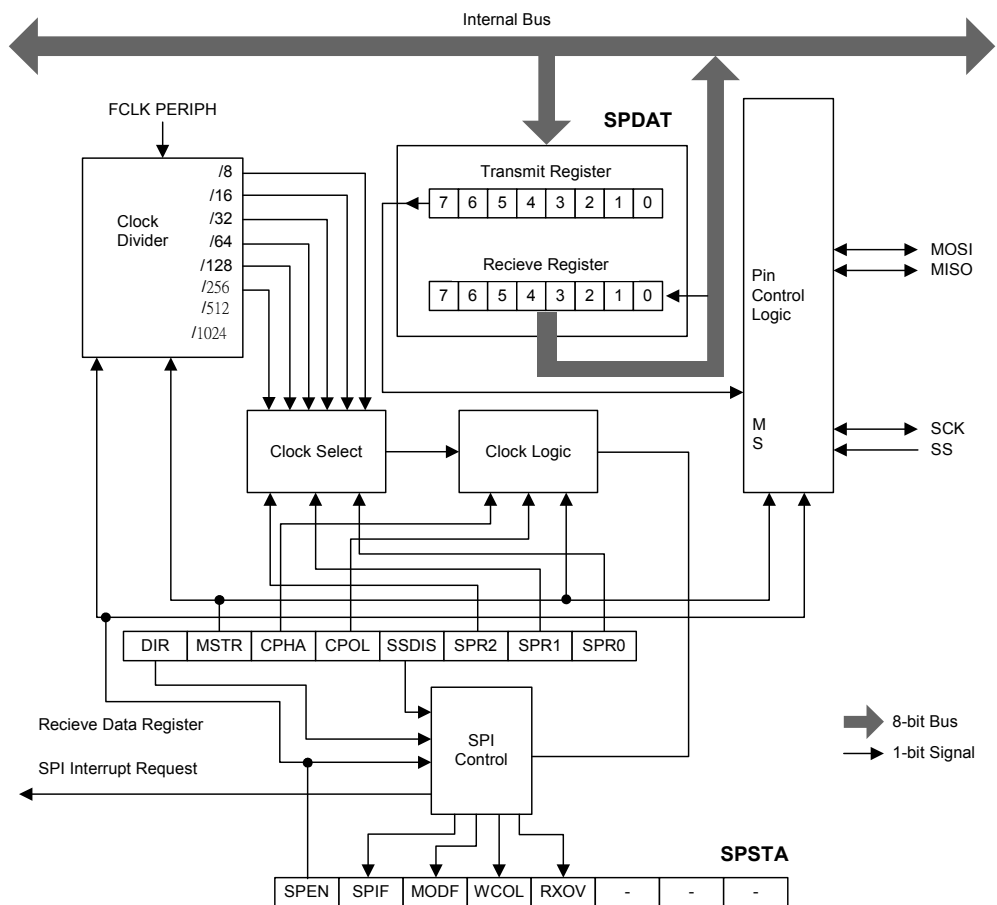
注意：当CPHA = '0'时，SS引脚产生下降沿表示启动发送。

8.7.3 波特率

在主模式下，SPI的波特率有八种可选择的频率，分别是内部时钟的8，16，32，64，128，256，512，1024分频，可以通过设定SPCON寄存器的SPR[2:0]位进行选择。

8.7.4 功能描述

下图所示是SPI模块的详细结构。



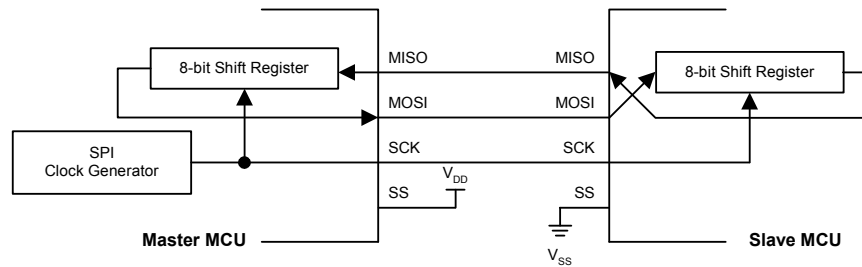
SPI模块框图

8.7.5 工作模式

SPI可配置为主模式或从模式中的一种。SPI模块的配置和初始化通过设置SPCON寄存器（串行外围设备控制寄存器）和SPSTA（串行外围设备状态寄存器）来完成。配置完成后，通过设置SPCON，SPSTA，SPDAT（串行外围设备数据寄存器）来完成数据传送。

在SPI通讯期间，数据同步地被串行的移进移出。串行时钟线（SCK）使两条串行数据线（MOSI和MISO）上数据的移动和采样保持同步。从设备选择线（ \overline{SS} ）可以独立地选择SPI从属设备；如果从设备没有被选中，则不能参与SPI总线上的活动。

当SPI主设备通过MOSI线传送数据到从设备时，从设备通过MISO线发送数据到主设备作为响应，这就实现了在同一时钟下数据发送和接收的同步全双工传输。发送移位寄存器和接收移位寄存器使用相同的特殊功能器地址，对SPI数据寄存器SPDAT进行写操作将写入发送移位寄存器，对SPDAT寄存器进行读操作将获得接收移位寄存器的数据。



全双工主从互联图

主模式

(1) 模式启动

SPI主设备控制SPI总线上所有数据传送的启动。当SPCON寄存器中的MSTR位置1时，SPI在主模式下运行，只有一个主设备可以启动传送。

(2) 发送

在SPI主模式下，写一个字节数据到SPI数据寄存器SPDAT，数据将会写入发送移位缓冲器。如果发送移位寄存器已经存在一个数据，那么主SPI产生一个WCOL信号以表明写入太快。但是在发送移位寄存器中的数据不会受到影响，发送也不会中断。另外如果发送移位寄存器为空，那么主设备立即按照SCK上的SPI时钟频率串行地移出发送移位寄存器中的数据到MOSI线上。当传送完毕，SPSTA寄存器中的SPIF位被置1。如果SPI中断被允许，当SPIF位置1时，也会产生一个中断。

(3) 接收

当主设备通过MOSI线传送数据给从设备时，相对应的从设备同时也通过MISO线将其发送移位寄存器的内容传送给主设备的接收移位寄存器，实现全双工操作。因此，SPIF标志位置1即表示传送完成也表示接收数据完毕。从设备接收的数据按照MSB或LSB优先的传送方向存入主设备的接收移位寄存器。当一个字节的数据完全被移入接收寄存器时，处理器可以通过读SPDAT寄存器获得该数据。如果发生超限（SPIF标志未被清0，就试图开始下一次传送），RXOV位置1，表示发生数据超限，此时接收移位寄存器保持原有数据并且SPIF位置1，这样直到SPIF位被清0，SPI主设备将不会接收任何数据。

从模式

(1) 模式启动

当SPCON寄存器中的MSTR位清0，SPI在从模式下运行。在数据传送之前，从设备的 \overline{SS} 引脚必须被置低，而且必须保持低电平直到一个字节数据传送完毕。

(2) 发送与接收

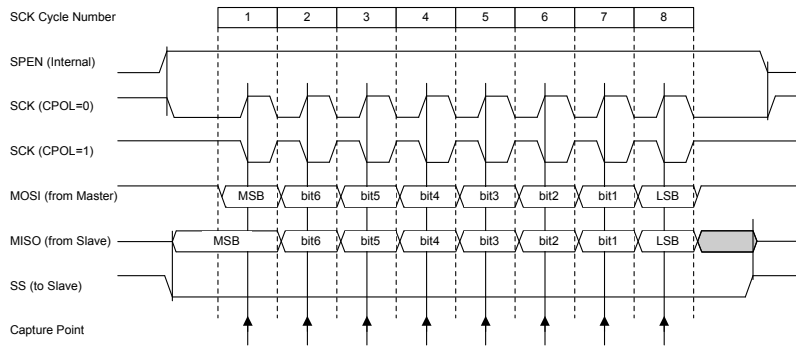
从属模式下，按照主设备控制的SCK信号，数据通过MOSI引脚移入，MISO引脚移出。一个位计数器记录SCK的边沿数，当接收移位寄存器移入8位数据（一个字节）同时发送移位寄存器移出8位数据（一个字节），SPIF标志位被置1。数据可以通过读取SPDAT寄存器获得。如果SPI中断被允许，当SPIF置1时，也会产生一个中断。

为防止超限，SPI从设备在向接收移位寄存器移入数据之前也必须软件清零SPIF标志位，否则RXOV位置1，表示发生数据超限。此时接收移位寄存器保持原有数据并且SPIF位置1，这样SPI从设备将不会接收任何数据直到SPIF清0。

SPI从设备不能启动数据传送，所以SPI从设备必须在主设备开始一次新的数据传送之前将要传送的数据写入发送移位寄存器。如果从设备在第一次开始发送之前未写入数据，从设备将传送“0x00”字节给主设备。如果写SPDAT操作发生在传送过程中，那么SPI从设备的WCOL标志位置1，即如果传送移位寄存器已经含有数据，SPI从设备的WCOL位置1，表示写SPDAT冲突。但是移位寄存器的数据不受影响，传送也不会被中断。

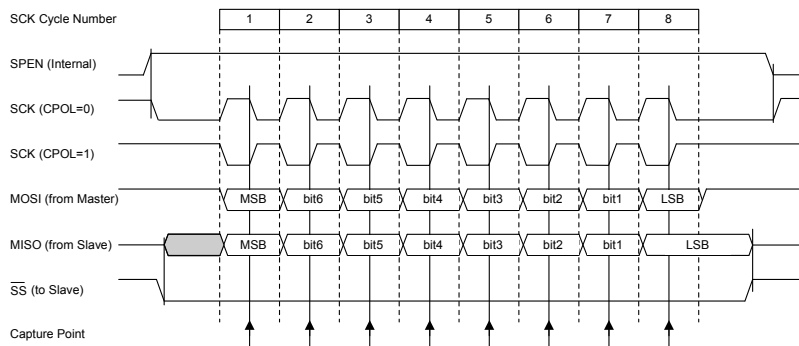
8.7.6 传送形式

通过软件设置SPCON寄存器的CPOL位和CPHA位，用户可以选择SPI时钟极性和相位的四种组合方式。CPOL位定义时钟的极性，即空闲时的电平状态，它对SPI传输格式影响不大。CPHA位定义时钟的相位，即定义允许数据采样移位的时钟边沿。在主从通讯的两个设备中，时钟极性相位的设置应一致。



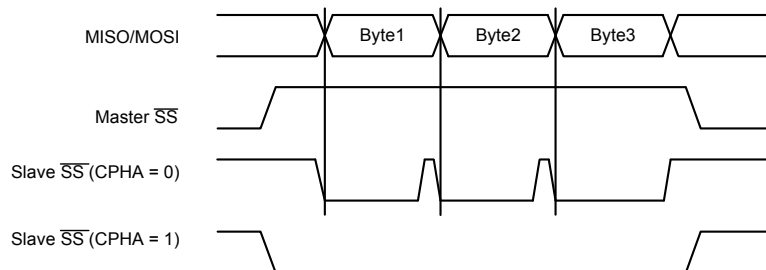
数据传送形式 (CPHA = 0)

如果CPHA = 0，SCK的第一个沿捕获数据，从设备必须在SCK的第一个沿之前将数据准备好，因此， \overline{SS} 引脚的下降沿从设备开始发送数据。 \overline{SS} 引脚在每次传送完一个字节之后必须被拉高，在发送下一个字节之前重新设置为低电平，因此当CPHA = 0，SSDIS不起作用。



数据发送形式 (CPHA = 1)

如果CPHA = 1，主设备在SCK的第一个沿将数据输出到MOSI线上，从设备把SCK的第一个沿作为开始发送信号。用户必须在第一个SCK的第二个沿之前完成写SPDAT的操作。 \overline{SS} 引脚在每个字节数据的传送过程始终保持低电平。这种数据传输形式是一个主设备一个从设备之间通信的首选形式。



CPHA/SS时序

注意：当SPI用作从设备模式，且SPCON寄存器的CPOL位清0，P5.2/SCK端口必须设置为输入模式，并在SPEN位置1前打开上拉电阻。

8.7.7 SPSTA异常情况

SPSTA寄存器中的标志位表示在SPI通讯中的出错情况：

(1) 模式故障 (MODF)

SPI主模式下的模式故障出错表明 \overline{SS} 引脚上的电平状态与实际的设备模式不一致。SPSTA寄存器中MODF位置1后，表明系统控制存在多主设备冲突的问题。这种情况下，SPI系统受到如下影响：

- 产生 SPI 接收/错误 CPU 中断请求；
- SPSTA 寄存器的 SPEN 位清 0，SPI 被禁止；
- SPCON 寄存器的 MSTR 位清 0。

当SPCON寄存器的 \overline{SS} 引脚禁止位 (SSDIS) 清0， \overline{SS} 引脚信号为低时，MODF标志位置1。然而，对于只有一个主设备的系统来说，主设备的 \overline{SS} 引脚被拉低，那决不是另外一个主设备试图驱动网络。这种情况下，为防止MODF置1，可使SPCON寄存器中的SSDIS位置1， \overline{SS} 引脚作为普通I/O口或是其它功能引脚。

重新启动串行通信时，用户必须将MODF位软件清0，将SPCON寄存器中的MSTR位和SPSTA寄存器的SPEN位置1，重新启动主模式。

(2) 写冲突 (WCOL)

在发送数据序列期间写入SPDAT寄存器而引起的写冲突，SPSTA寄存器中的WCOL位置1。WCOL位置1不会引起中断，发送也不会中止。WCOL位需由软件清0。

(3) 超限情况 (RXOV)

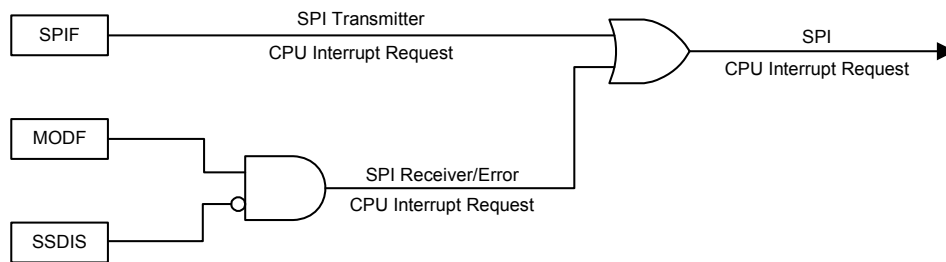
主设备或从设备尚未清除SPIF位，主或从设备又试图发送几个数据字节时，超限情况发生。在这种情况下，接收移位寄存器保持原有数据，SPIF置1，同样SPI设备直到SPIF被清除后才会再接收数据。在SPIF位被清除之前继续调用中断，发送也不会中止。RXOV位置1不会引起中断，RXOV位需由软件清0。

8.7.8 中断

两种SPI状态标志SPIF & MODF能产生一个CPU中断请求。

串行外围设备数据发送标志，SPIF：完成一个字节发送后由硬件置1。

模式故障标志，MODF：该位被置1表示 \overline{SS} 引脚上的电平与SPI模式不一致的。SSDIS位为0并且MODF置1将产生SPI接收器/出错CPU中断请求。当SSDIS置1时，无MODF中断请求产生。



SPI中断请求的产生

8.7.9 寄存器

Table 8.56 SPI控制寄存器

A2H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SPCON	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	DIR	传送方向选择位 0: MSB优先发送 1: LSB优先发送
6	MSTR	SP设备选择位 0: 配置SPI作为从属设备 1: 配置SPI作为主设备
5	CPHA	时钟相位控制位 0: SCK周期的第一沿采集数据 1: SCK周期的第二沿采集数据
4	CPOL	时钟极性控制位 0: 在Idle状态下SCK处于低电平 1: 在Idle状态下SCK处于高电平
3	SSDIS	SS引脚控制位 0: 在主和从属模式下, 打开 \overline{SS} 引脚 1: 在主和从属模式下, 关闭 \overline{SS} 引脚 如果SSDIS置1, 不产生MODF中断请求 在从属模式下, 如果CPHA = 0, 该位不起作用
2-0	SPR[2:0]	串行外部设备时钟速率选择位 000: $f_{SYS}/8$ 001: $f_{SYS}/16$ 010: $f_{SYS}/32$ 011: $f_{SYS}/64$ 100: $f_{SYS}/128$ 101: $f_{SYS}/256$ 110: $f_{SYS}/512$ 111: $f_{SYS}/1024$

Table 8.57 SPI状态寄存器

A1H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SPSTA	SPEN	SPIF	MODF	WCOL	RXOV	-	-	-
读/写	读/写	读/写	读/写	读/写	读/写	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	-	-	-

位编号	位符号	说明
7	SPEN	SPI控制位 0: 关闭SPI 1: 打开SPI接口
6	SPIF	串行外部设备数据传送标志位 0: 由软件清0 1: 表明已完成数据传输, 由硬件置1
5	MODF	模式故障位 0: 由软件清0 1: 表明SS引脚电平与SPI模式不一致, 由硬件置1
4	WCOL	写入冲突标志位 0: 由软件清0, 表明已处理写入冲突 1: 由硬件置1, 表明检测到一个冲突
3	RXOV	接收超限位 0: 表明已处理接收超限, 由软件清0 1: 表明已检测到接收超限, 由硬件置1

Table 8.58 SPI数据寄存器

A3H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	SPDAT[7:0]	写入SPDAT的数据被放置到发送移位寄存器中 读取SPDAT时将获得接收移位寄存器的数据

注意: 当关闭SPI功能后, 读取SPI数据寄存器SPDAT的数据无效。

8.8 输入捕捉模块

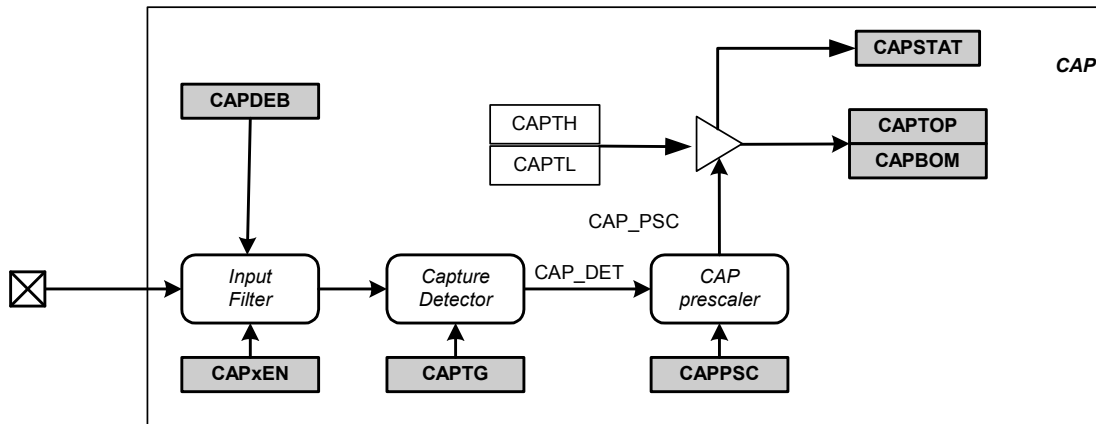
8.8.1 特性

- 3路施密特输入捕捉引脚
- 引脚滤波时间可设置
- 捕捉预分频系数可设置
- 自带捕捉定时器作为捕捉时基
- 捕捉事件可设置为上升沿、下降沿、双沿
- 2级捕捉存储栈及状态指示

G80F980内部含有一个输入捕捉单元，对应三个输入捕捉引脚CAPx(x=0~2)，当该引脚设置为输入捕获功能时，对应的输入捕获功能有效，捕捉单元的三个输入引脚共用边沿检测方式，引脚滤波时间，预分频系数以及捕捉存储栈。

当输入捕获引脚出现符合设置要求的捕捉事件，并经过内部检测电路最终产生输入捕获信号后，硬件会自动抓取捕捉定时器的计数值存储到其对应的2级捕捉存储栈，并给出相应的中断标志和存储栈状态。用户可以从存储栈读取捕捉到计数值，以实现相关的应用功能。

8.8.2 模块图



8.8.3 寄存器

捕捉模块所使用寄存器如下：

功能	名称	寄存器描述
捕捉控制寄存器	CAPCON1 CAPCON2	用于控制捕捉功能开启，预分频系数，引脚滤波时间，存储栈状态等
捕捉状态寄存器	CAPSTAT	用于查询捕捉堆栈及捕捉口状态
捕捉中断寄存器	CAPINT	用于设置捕捉中断使能及捕捉中断标志
捕捉定时器计数/重载寄存器	CAPTL/H	捕捉定时器计数/重载寄存器
捕捉存储栈顶/栈底寄存器	CAPTOL/H	捕捉存储栈顶寄存器
	CAPBOTL/H	捕捉存储栈底寄存器

Table 8.33 输入捕捉控制寄存器1

FFB8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPCON1	CAP2EN	CAP1EN	CAP0EN	CAPTG1	CAPTG0	CAPDEB2	CAPDEB1	CAPDEB0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0
位编号	位符号	说明						
7	CAP2EN	捕捉输入0使能控制位 0: 捕捉输入2关闭, 做普通IO用 1: 捕捉输入2开启						
6	CAP1EN	捕捉输入1使能控制位 0: 捕捉输入1关闭, 做普通IO用 1: 捕捉输入1开启						
5	CAP0EN	捕捉输入0使能控制位 0: 捕捉输入0关闭, 做普通IO用 1: 捕捉输入0开启						
4-3	CAPTG[1:0]	捕捉引脚触发模式位 00: 低电平触发 01: 上升沿触发 10: 下降沿触发 11: 双沿触发 CAP2~CAP0采用同一种触发方式						
2-0	CAPDEB[2:0]	CAP输入信号信号滤波时间 000: 无滤波 001: 滤波时间常数为1us 010: 滤波时间常数为2us 011: 滤波时间常数为4us 100: 滤波时间常数为6us 101: 滤波时间常数为8us 110: 滤波时间常数为12us 111: 滤波时间常数为16us 注: 上述滤波常数时间不是精确值, 仅供参考						

Table 8.34 输入捕捉控制寄存器2

FFB9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPCON2	CAPPSC2	CAPPSC1	CAPPSC0	-	CAPTPS2	CAPTPS1	CAPTPS0	CAPTR
读/写	读/写	读/写	读/写	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

位编号	位符号	说明
7-5	CAPPSC[2:0]	捕捉输入信号预分频系数 000: 1/1 001: 1/2 010: 1/4 011: 1/6 100: 1/8 101: 1/16 110: 1/32 111: 1/64

3-1	CAPTPS1 [2:0]	捕捉定时器预分频比选择位 000: 1/1 001: 1/4 010: 1/16 011: 1/64 100: 1/256 101: 1/1024 其他值无意义
0	CAPTR	捕捉定时器允许控制位 0: 关闭捕捉定时器 1: 打开捕捉定时器

TABLE 8.34 输入捕捉状态寄存器

A5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPSTAT	CAP0IE	-	CAPSTAT	CAPSTAT	CAP0IF	CAP2	CAP1	CAP0
读/写	读/写	-	读	读	读/写	读	读	读
复位值 (POR/WDT/LVR/PIN)	0	-	0	0	0	0	0	0

位编号	位符号	说明
7	CAP0IE	捕捉引脚0中断使能位 0: 禁止输入捕捉中断 1: 允许输入捕捉中断
5 - 4	CAPSTAT[1:0]	捕捉存储栈状态 00: 存储栈为空 01: 有一个捕捉值 (CAPTOP) 10: 有两个捕捉值 (CAPTOP, CAPBOT都有捕捉值) 11: 存储栈有溢出 (CAPTOP, CAPBOT都有捕捉值, 前一个捕捉值未被读取而溢出)
3	CAP0IF	捕捉引脚0中断标志位 0: 无捕捉中断发生 1: 有捕捉中断发生
2 - 0	CAPx (x = 0 - 2)	捕捉口电平状态 0: 低电平 1: 高电平 注: 若捕捉口开启了捕捉功能, 电平状态为滤波后的结果; 若未开启捕捉功能, 则电平状态直接为相应IO口状态 (相应IO需配置为输入, 且打开上拉功能)

Table 8.34 输入捕捉中断寄存器

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPINT1 (8FH)	CAP1IE	-	-	-	CAP1IF	-	-	-
CAPINT2 (FFBCH)	CAP2IE	-	-	-	CAP2IF	-	-	-
读/写	读/写	-	-	-	读/写	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	0	-	-	-

位编号	位符号	说明
-----	-----	----

7	CAPxIE (x = 1 - 2)	捕捉引脚中断使能位 0: 禁止输入捕捉中断 1: 允许输入捕捉中断
3	CAPxIF (x = 1 - 2)	捕捉引脚中断标志位 0: 无捕捉中断发生 1: 有捕捉中断发生

Table 8.34 输入捕捉定时器中断寄存器

FFBDH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPTINT	CAPTIE	-	-	-	CAPTIF	-	-	-
读/写	读/写	-	-	-	读/写	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	0	-	-	-

位编号	位符号	说明
7	CAPTIE	捕捉定时器溢出中断允许位 0: 禁止捕捉定时器溢出中断 1: 允许捕捉定时器溢出中断
3	CAPTIF	捕捉定时器溢出标志位 0: 无溢出 (软件清0) 1: 溢出 (硬件置1)

Table 8.33 捕捉定时器重载/计数数据寄存器

FFBAH - FFBBH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPTL (FFBBH)	CAPTL.7	CAPTL.6	CAPTL.5	CAPTL.4	CAPTL.3	CAPTL.2	CAPTL.1	CAPTL.0
CAPTH (FFBAH)	CAPTH.7	CAPTH.6	CAPTH.5	CAPTH.4	CAPTH.3	CAPTH.2	CAPTH.1	CAPTH.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	CAPTL.x CAPTH.x	捕捉定时器低位高位计数器, x = 0 - 7

Table 8.33 捕捉存储栈栈顶寄存器

9CH,9DH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPTOPL(9CH)	CAPTOPL.7	CAPTOPL.6	CAPTOPL.5	CAPTOPL.4	CAPTOPL. 3	CAPTOPL. 2	CAPTOPL. 1	CAPTOPL. 0
CAPTOPH(9DH)	CAPTOPH.7	CAPTOPH.6	CAPTOPH.5	CAPTOPH.4	CAPTOPH. .3	CAPTOPH. 2	CAPTOPH. 1	CAPTOPH. 0
读/写	读	读	读	读	读	读	读	读
复位值 (POR/WDT/LVR/ PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
-----	-----	----

7-0	CAPTOPL.x	捕捉存储栈栈顶寄存器, x = 0 - 7
	CAPTOPH.x	

Table 8.33 捕捉存储栈栈底寄存器

9EH,9FH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CAPBOTL(9EH)	CAPBOTL.7	CAPBOTL.6	CAPBOTL.5	CAPBOTL.4	CAPBOTL. 3	CAPBOTL. 2	CAPBOTL. 1	CAPBOTL. 0
CAPBOTH(9FH)	CAPBOTH.7	CAPBOTH.6	CAPBOTH.5	CAPBOTH.4	CAPBOTH. .3	CAPBOTH. 2	CAPBOTH. 1	CAPBOTH. 0
读/写	读	读	读	读	读	读	读	读
复位值 (POR/WDT/LVR/ PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	CAPBOTL.x	捕捉存储栈栈底寄存器, x = 0 - 7
	CAPBOTH.x	

8.8.4 捕捉过程与存储栈操作

每个输入捕捉单元CAPx (x = 0-3) 共用一个专用的2级存储栈。存储栈的操作满足以下两个基本原则:

(1) 若存储栈为空 (CAPTOP和CAPBOM都没有捕获值), 则最先进栈的数据存放于栈顶寄存器CAPTOP中; 若存储栈底为空, 则下一次数据进栈时存放于栈底, 若存储栈底不为空, 则下一次数据进栈时, CAPBOM值被压入到CAPTOP, 新数据被存放于栈底寄存器CAPBOM。

(2) 每次读取栈顶寄存器CAPTOP后, 都会使CAPBOM值被压入到CAPTOP, 同时CAPBOM被清0; 每次读取CAPBOM寄存器后, 都会使CAPBOM被清0, 但CAPTOP寄存器值保持不变。

三个输入捕捉引脚CAPx (x = 0-3) 共用CAPSTAT[1:0]状态位表示当前存储栈的状态, 输入捕捉和读取过程的状态机转换与存储栈操作示意图, 如图9-7-2所示 (图中CAPTOP和CAPBOM为灰底时表示该寄存器中有捕捉数据)。

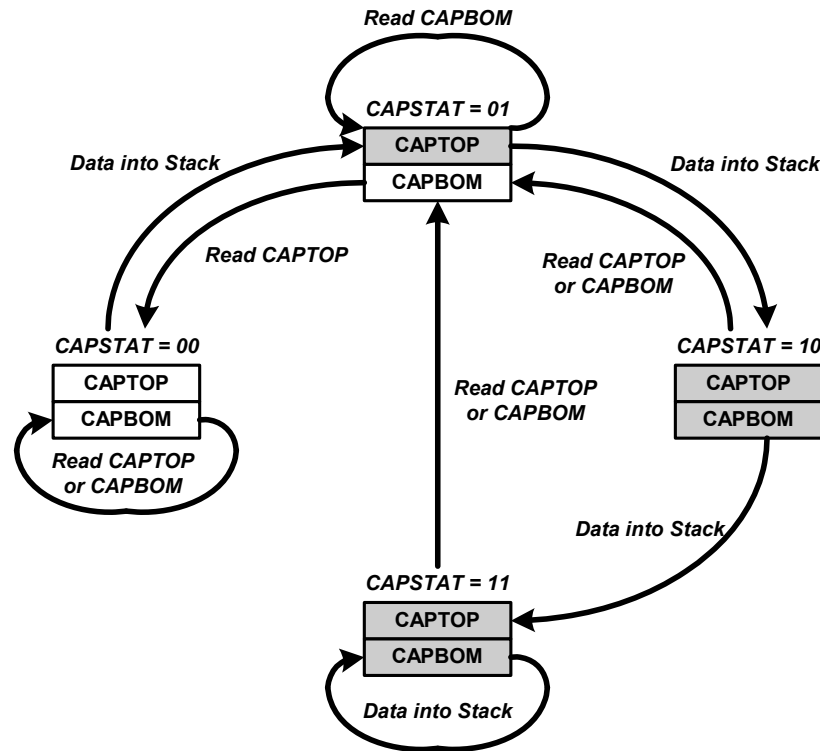


图9-7-2 输入捕捉过程状态机转换与存储栈操作图

第一次捕捉：此时存储栈为空，CAPSTAT[1:0] = 00。当CAPx引脚上发生了指定的跳变后，捕捉定时器的计数值被压入栈顶寄存器CAPTOP中，CAPSTAT[1:0]变为01。若在下次捕捉发生之前对栈顶寄存器CAPTOP进行了读取操作，则读取后CAPTOP寄存器被清0，且CAPSTAT[1:0]变为00；若在下次捕捉发生之前对栈底寄存器CAPBOM进行了读取操作，由于CAPBOM中没有捕捉值，因此固定读到0，且不会改变CAPSTAT值。

第二次捕捉：如果在前次捕捉计数值被读取之前发生了另一次捕捉，则新的捕捉值将存放在栈底寄存器CAPBOM中，同时CAPSTAT[1:0]变为10。若在下次捕捉发生之前对栈顶寄存器CAPTOP进行了读取操作，则CAPTOP的值被读取之后，栈底寄存器CAPBOM值将压入栈顶寄存器CAPTOP中，且CAPSTAT[1:0]变为01；若在下次捕捉发生之前对栈底寄存器CAPBOM进行了读取操作，则读取后CAPBOM寄存器被清0，且CAPSTAT[1:0]变为01。

第三次捕捉：如果在捕捉发生时，存储栈已经有两个捕捉到的计数值，则栈顶寄存器CAPTOP中数据将被弹出并丢弃，而栈底寄存器CAPBOM值将压入栈顶寄存器CAPTOP中，新捕捉到的值被压入栈底寄存器CAPBOM中，并且CAPSTAT[1:0]变为11。若在下次捕捉发生之前对栈顶寄存器CAPTOP进行了读取操作，则CAPTOP的值被读取之后，栈底寄存器CAPBOM的值将压入栈顶寄存器CAPTOP中，且CAPSTAT[1:0]变为01；若在下次捕捉发生之前对栈底寄存器CAPBOM进行了读取操作，则读取后CAPBOM寄存器被清0，且CAPSTAT[1:0]变为01。

CAPSTAT[1:0] = 11，表明存储栈发生了溢出，有数据已经丢失。

CAPTOPL/H和CAPBOTL/H读操作需要遵循以下顺序：

读操作：先高位，后低位。

8.8.5 输入捕捉引脚滤波设置

3个输入捕捉引脚CAPx (x = 0-2) 的输入信号都可以通过内部滤波电路保证检测的可靠性。

CAPDEB[2:0]位段来调整对此信号的滤波时间。CAPDEB [2:0] = 0时，无滤波作用。CAPDEB [2:0] = 1 - 7时，可以将滤波时间设置为1us~16us共8档，捕捉输入信号至少保持CAPDEB [2:0]定义的时间长度，捕捉检测模块才认为捕捉检测输入信号有效，从而触发捕捉事件。

注：捕捉功能开启前，需设置对应IO管脚为输入功能。

8.8.6 输入捕捉预分频设置

外部输入的捕捉信号经过预分频器分频后才真正触发捕捉过程，用户使用CAPPSC控制位可以设置每个捕捉单元的预分频参数。

每次对CAPPSC写入与原值不同的设置值时，硬件会自动对预分频计数器清0，然后从0开始增计数到CAPPSC指定值时归零。当分频计数值为0时，会产生一次分频后的捕捉信号，并触发一次捕捉过程。

以CAP0为例，当CAPPSC[2:0] = 010时，实际触发捕捉的信号波形如下图所示：

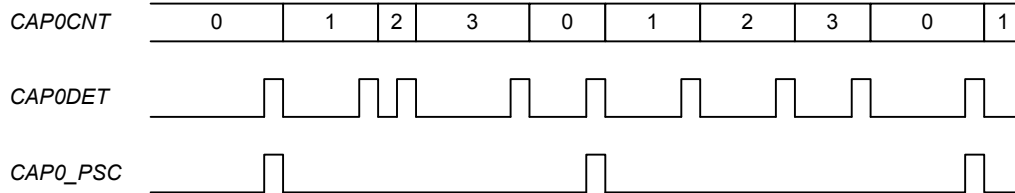


图9-7-3 预分频设置的工作波形

8.8.7 输入捕捉中断

每次捕捉后都会置位CAPxIF标志，如果此时CAPxIE = 1则产生中断。这种设置适用于开关霍尔信号的检测。

当CPU进入空闲模式或掉电模式，可通过输入捕捉中断退出。如果捕捉引脚设置为下降沿触发，外部输入从高电平变为低电平，然后至少保持输入信号信号滤波时间的低电平后，立即产生中断。在中断发生后，CPU时钟和外部设备时钟立即恢复，SUSLO寄存器和PCON寄存器中的PD位会被硬件清除，然后程序运行中断服务程序。在完成中断服务程序之后，跳转到进入掉电模式之后的指令继续运行。

注意：掉电模式下，只要外部输入保持超过滤波时间的电平后便立即产生中断，此时的预分频系数无效，捕捉事件不会发生，同时堆栈状态也不会发生改变，也就是说掉电模式下有效的边沿只能起到唤醒CPU的作用，而无法捕捉到正确的定时器值。

8.8.8 捕捉定时器

捕捉定时器是16位自动重载定时器，时钟源固定为系统时钟。两个数据寄存器CAPTH和CAPTL可作为一个16位寄存器来访问。由CAPCON2寄存器控制。CAPINT寄存器的CAPTIE位置1允许捕捉定时器中断。

当CAPTH和CAPTL被写时，用作定时器重载寄存器，当被读时，被用作计数寄存器。CAPTR位置1使定时器开始递增计数。定时器在0xFFFF到0x0000溢出并置CAPTIF位为1。溢出同时，定时器重载寄存器的16位数据重新载入计数寄存器中，对CAPTH的写操作也导致重载寄存器的数据重新载入计数寄存器。

CAPTH和CAPTL读写操作遵循以下顺序：可在C语言中使用unsigned int访问

写操作：先高位，后低位。

读操作：先高位，后低位。

捕捉定时器启动由CAPTR控制，当CAPTR=1时，定时器开始计数，当CAPTR=0时，定时器停止计数。当捕捉发生时，捕捉定时器值被存入堆栈。

当捕捉功能禁止时，也即CAPxEN = 0 (x = 0 - 2) 时，捕捉定时器功能仍然有效，可当作普通定时器使用。

8.9 实时监控模块 (RM)

8.9.1 特性

- 支持仿真环境下全速运行时的实时监控
- 支持最多6个字节的监控数据

8.9.2 寄存器

FFC0~FFC5	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
RMDAT1 (FFC0H)	RMDAT1.7	RMDAT1.6	RMDAT1.5	RMDAT1.4	RMDAT1.3	RMDAT1.2	RMDAT1.1	RMDAT1.0
RMDAT2 (FFC1H)	RMDAT2.7	RMDAT2.6	RMDAT2.5	RMDAT2.4	RMDAT2.3	RMDAT2.2	RMDAT2.1	RMDAT2.0
RMDAT3 (FFC2H)	RMDAT3.7	RMDAT3.6	RMDAT3.5	RMDAT3.4	RMDAT3.3	RMDAT3.2	RMDAT3.1	RMDAT3.0
RMDAT4 (FFC3H)	RMDAT4.7	RMDAT4.6	RMDAT4.5	RMDAT4.4	RMDAT4.3	RMDAT4.2	RMDAT4.1	RMDAT4.0
RMDAT5 (FFC4H)	RMDAT5.7	RMDAT5.6	RMDAT5.5	RMDAT5.4	RMDAT5.3	RMDAT5.2	RMDAT5.1	RMDAT5.0
RMDAT6 (FFC5H)	RMDAT6.7	RMDAT6.6	RMDAT6.5	RMDAT6.4	RMDAT6.3	RMDAT6.2	RMDAT6.1	RMDAT6.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

8.10 电压复位 (LVR)

8.10.1 特性

- 通过代码选项选择，LVR 设定电压 V_{LVR} 可为 4.1V,3.7V 或 2.8V
- LVR 去抖动时间 T_{LVR} 为 30 μ s
- 当供电电压低于设定电压 V_{LVR} 时，将产生内部复位

低电压复位 (LVR) 功能是为了监测供电电压，当供电电压低于设定电压 V_{LVR} 时，MCU 将产生内部复位。LVR 去抖动时间 T_{LVR} 大约为 30 μ s。

LVR 功能打开后，具有以下特性 (t 表示电压低于设定电压 V_{LVR} 的时间)：

当 $V_{DD} \leq V_{LVR}$ 且 $t \geq T_{LVR}$ 时产生系统复位。

当 $V_{DD} > V_{LVR}$ 或 $V_{DD} < V_{LVR}$ ，但持续时间 $< T_{LVR}$ 时不会产生系统复位。

通过代码选项，可以选择 LVR 功能的打开与关闭。

在交流电或大容量电池应用中，接通大负载后容易导致 MCU 供电暂时低于定义的工作电压。低电压复位可以应用于此，保护系统在低于设定电压下产生有效复位。

8.11看门狗定时器（WDT），程式超范围溢出（OVL）复位及其它复位状态

8.11.1 特性

- 程式超范围溢出后硬件自动检测，并产生 OVL 复位
- 看门狗可以工作在掉电模式下
- 看门狗溢出频率可选

程序超范围溢出复位

G80F980为进一步增强CPU运行可靠性，内建程式超范围溢出检测电路，一旦检测到程式计数器的值超出ROM最大值，或者发现指令操作码（不检测操作数）为8051指令集中不存在的A5H，便认为程式跑飞，产生CPU复位信号，同时将WDOF标志位置1。为应用这个特性，用户应该将未使用的Flash ROM用0xA5填满。

看门狗

看门狗定时器是一个递减计数器，独立内建RC振荡器作为其时钟源，因此可以通过代码选项选择在掉电模式下仍持续运行。当定时器溢出时，将芯片复位。通过代码选项可以打开或关闭该功能。

WDT控制位（第2 - 0位）用来选择不同的溢出时间。定时器溢出后，WDT溢出标志（WDOF）将由硬件自动置1。通过读写RSTSTAT寄存器，看门狗定时器在溢出前重新开始计数。

其它一些复位标志列举如下：

8.11.2 寄存器

Table 8.60 复位状态寄存器

B1H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
RSTSTAT	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR)	0	-	1	0	0	0	0	0
复位值 (WDT)	1	-	u	u	u	0	0	0
复位值 (LVR)	u	-	u	1	u	0	0	0
复位值 (PIN)	u	-	u	u	1	0	0	0

位编号	位符号	说明
7	WDOF	看门狗溢出或程式超范围溢出标志位 看门狗溢出时由硬件置1，可由软件或上电复位清0 0: 未发生WDT溢出或程式超范围溢出 1: 发生WDT溢出或程式超范围溢出
5	PORF	上电复位标志位 上电复位后硬件置1，只能由软件清0 0: 没有发生上电复位 1: 发生过上电复位
4	LVRF	低压复位标志位 低压复位后置1，可由软件或上电复位清0 0: 没有发生低压复位 1: 发生过低压复位
3	CLRF	Reset引脚复位标志位 引脚复位后置1，由软件或上电复位清0 0: 没有发生引脚复位 1: 发生过引脚复位
2-0	WDT[2:0]	WDT溢出周期控制位 000: 溢出周期最小值 = 4096ms 001: 溢出周期最小值 = 1024ms 010: 溢出周期最小值 = 256ms 011: 溢出周期最小值 = 128ms 100: 溢出周期最小值 = 64ms 101: 溢出周期最小值 = 16ms 110: 溢出周期最小值 = 4ms 111: 溢出周期最小值 = 1ms 注意: 应用中如果看门狗打开，程式清看门狗的最大间隔时间不能大于以上所列最小值

8.12 电源管理

8.12.1 特性

- 空闲模式和掉电模式两种省电模式
- 发生中断和复位可退出空闲（Idle）、掉电（Power-Down）模式

为减少功耗，G80F980提供两种低功耗省电模式：空闲模式和掉电模式，这两种模式都由PCON和SUSLO寄存器控制。

8.12.2 空闲模式

空闲模式能够降低系统功耗，在此模式下，程序中止运行，CPU时钟停止，但外部设备时钟继续运行。空闲模式下，CPU在确定的状态下停止，并在进入空闲模式前所有CPU的状态都被保存，如PC，PSW，SFR，RAM等。

两条连续指令：先设置SUSLO寄存器为55H，随即将PCON寄存器中的IDL位置1，使G80F980进入空闲模式。如果按顺序要求的连续指令不被满足，CPU在下一个机器周期清除SUSLO寄存器或IDL位，CPU也不会进入空闲模式。

IDL位置1是CPU进入空闲模式之前执行的最后一条指令。

两种方式可以退出空闲模式：

(1) 产生一个中断。立即恢复CPU时钟，硬件清除SUSLO寄存器和PCON寄存器的IDL位。然后执行中断服务程序，随后跳转到进入空闲模式指令之后的指令。

(2) 复位信号产生后（复位引脚上出现低电平，WDT复位，LVR复位）。在预热定时结束之后，CPU恢复时钟，SUSLO寄存器和在PCON寄存器中的IDL位被硬件清除，最后G80F980复位。然后程序从地址位0000H开始执行。RAM保持不变而SFR的值根据不同功能模块改变。

8.12.3 掉电模式

掉电模式可以使G80F980进入功耗非常低的状态。掉电模式将停止CPU和外围设备的所有时钟信号(掉电模式会自动关闭内部ADC模块供电，但不会关闭OP和CMP模块供电，如需切断相应模块供电，须软件手动关闭该模块)。如果WDT使能，WDT模块将继续工作。在进入掉电模式前所有CPU的状态都被保存，如PC，PSW，SFR，RAM等。

两条连续指令：先设置SUSLO寄存器为55H，随即将PCON寄存器中的PD位置1，使G80F980进入掉电模式。如果按顺序要求的连续指令不被满足，CPU在下一个机器周期清除SUSLO寄存器或IDL位，CPU也不会进入掉电模式。

PD位置1是CPU进入掉电模式之前执行的最后一条指令。

当PLL振荡器开启时，无法进入掉电模式，PD位无法置1。如需进入掉电模式，若当前选择PLL为系统时钟则必须先将系统时钟选择为OSC1或OSC2，然后关闭PLL振荡器后，执行上述两条连续指令进入掉电模式

注意：如果同时设置IDL位和PD位，G80F980进入掉电模式。退出掉电模式后，CPU也不会进入空闲模式，从掉电模式退出后硬件会清除IDL及PD位。

有两种方式可以退出掉电模式：

(1) 有效外部中断（如INT0，INT1，INT4，CAP0~CAP2）和一定情况下的Timer3中断（详见定时器3章节）能使G80F980退出掉电模式。在中断发生后振荡器启动，CPU时钟和外部设备时钟立即恢复，SUSLO寄存器和PCON寄存器中的PD位会被硬件清除，然后程序运行中断服务程序。在完成中断服务程序之后，跳转到进入掉电模式之后的指令继续运行。

(2) 复位信号（复位引脚上出现低电平，WDT复位如果被允许，LVR复位如果被允许）。在预热计时之后会恢复CPU时钟，SUSLO寄存器和PCON寄存器中的PD位会被硬件清除，最后G80F980会被复位。然后程序会从0000H地址位开始运行。RAM将保持不变，而根据不同功能模块SFR的值可能改变。

注意：如要进入这两种低功耗模式，必须在置位PCON中的IDL/PD位后增加3个空操作指令。

8.12.4 寄存器

Table 8.61 电源控制寄存器

87H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PCON	SMOD	SSTAT	IT41	IT40	GF1	GF0	PD	IDL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	SMOD	UART波特率加倍器
6	SSTAT	SCON[7:5]功能选择位
5-4	IT4[1:0]	外部中断4模式选择控制位
3-2	GF[1:0]	用于软件的通用标志
1	PD	掉电模式控制位 0: 当一个中断或复位产生时由硬件清0 1: 由软件置1激活掉电模式
0	IDL	空闲模式控制位 0: 当一个中断或复位产生时由硬件清0 1: 由软件置1激活空闲模式

Table 8.62 省电模式控制寄存器

8EH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
SUSLO	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	SUSLO[7:0]	此寄存器用来控制CPU进入省电模式（空闲或掉电）。只有像下面的连续指令才能使CPU进入省电模式，否则在下一个周期中SUSLO，IDL或PD位将被硬件清0

程序举例：

```

IDLE_MODE:
    MOV     SUSLO, #55H
    ORL     PCON, #01H
    NOP
    NOP
    NOP

```

```

POWERDOWN_MODE:
    MOV     SUSLO, #55H
    ORL     PCON, #02H
    NOP
    NOP
    NOP

```


8.13 预热计数器

8.13.1 特性

- 内建电源预热计数器消除电源的上电的不稳定状态
- 内建振荡器预热计数器消除振荡器起振时的不稳定状态

G80F980内建有电源上电预热计数器，主要是用来消除上电电压建立时的不稳定态，同时完成内部一些初始化序列，如读取内部客户代码选项等。

G80F980还内建振荡器预热计数器，它能消除振荡器在下列情况下起振时的不稳定状态：上电复位，引脚复位，从低功耗模式中唤醒，看门狗复位和LVR复位。

上电后，G80F980先经过电源上电预热计数过程，等待溢出后再进行振荡器的预热计数过程过程，溢出后开始运行程式。

电源上电预热计数时间

上电复位/ 按键复位/低压复位		WDT复位 (不在掉电情况下)		WDT复位 (从掉电情况唤醒)		从掉电情况下唤醒 (只是中断唤醒)	
电源上电 预热计数时间	振荡器上电 预热计数	电源上电 预热计数时间	振荡器上电 预热计数	电源上电 预热计数时间	振荡器上电 预热计数	电源上电 预热计数时间	振荡器上电 预热计数
11ms	有	≈1ms	无	≈1ms	有	≈800us	有

振荡器上电预热计数时间

Option: OP_WMT 振荡器类型	00 01		10	11
陶振/晶振	$2^{17} \times T_{osc}$	$2^{14} \times T_{osc}$	$2^{11} \times T_{osc}$	$2^8 \times T_{osc}$
内部RC	$2^7 \times T_{osc}$			

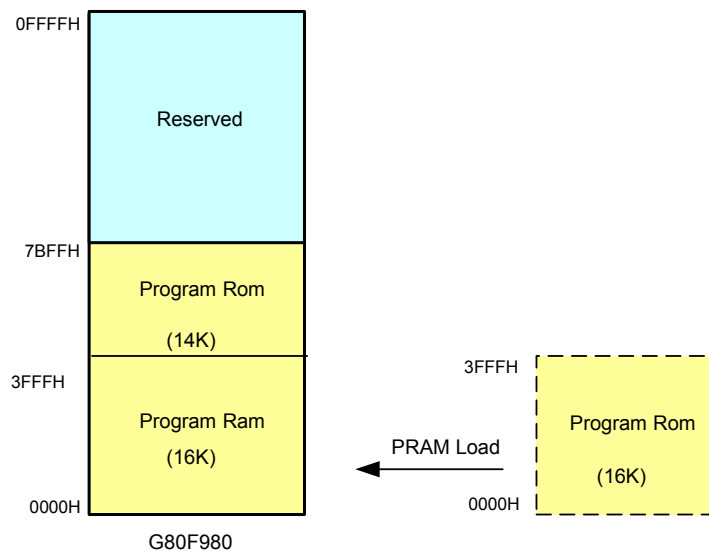
8.14 高速程序RAM (PRAM)

8.14.1 特性

- 可选择从Flash加载PRAM区代码
- PRAM加载完成后自动校验
- 可选择程序指针运行区域Flash或PRAM

G80F980内建一块16KByte的高速程序RAM，逻辑地址与Flash低16K区域相对应；用户可通过寄存器PRAMCON中的Load位控制启动一次从Flash加载低16KByte空间的数据到高速程序RAM中的过程，加载期间CPU进入空闲状态，加载完成后产生校验标志，用户可根据校验结果选择再次加载或将通过将EXEC置1将程序区切换为高速RAM（程序执行默认是在Flash空间进行）。MOVC指令也受EXEC位控制，当EXEC=0时，低16K地址程序空间为Flash，MOVC指令取到也是Flash内容；当EXEC=1时，低16K地址程序空间为PRAM，MOVC指令取到也是PRAM内容。

KEIL仿真模式下CODE区显示的低16K地址内容也与EXEC位有关，EXEC=0时，低16K地址程序空间显示为Flash内容；EXEC=1时，低16K地址程序空间显示为PRAM内容。



当程序空间的低16K地址（0000H~3FFFH）选择为Program RAM区域时（默认是Flash），数据存取速度相比其他Program Rom（Flash，4000H~7BFFH）要快得多，因此RAM中程序可以以更高速度执行，用户可以将对执行速度要求高的代码通过连接器定位在此范围内（0000H~3FFFH），在PRAM加载完成后，通过设置PLL将系统频率倍频到更高速度（最大11MHz）来执行此部分代码；而在其他区域，由于此时系统时钟超出Flash的存储速度限制，因此CPU会自动以降频的方式在Flash中执行Flash中代码（无需用户参与）。

8.14.2 寄存器

Table 8.63 PRAM控制锁定寄存器

FFC8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PRAMLOCK	LOCK.7	LOCK.6	LOCK.5	LOCK.4	LOCK.3	LOCK.2	LOCK.1	LOCK.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7 - 0	LOCK[7: 0]	PRAMCON寄存器修改允许寄存器 0x55: 允许软件对PRAMCON寄存器的修改 其他值: 禁止软件对PRAMCON寄存器的修改

Table 8.64 PRAM控制寄存器

FFC9H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PRAMCON	PRAMLDF	-	-	-	-	-	EXEC	LOAD
读/写	读	-	-	-	-	-	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	-	-	0	0

位编号	位符号	说明
7	PRAMLDF	PRAM加载校验标志 0: PRAM加载校验正确 1: PRAM加载校验出错 <i>注: 启动PRAM加载过程后, CPU进入空闲状态, 加载后自动校验, 校验完成后CPU重新运行, 软件可通过此标志判断加载正确性, 用户可选择重新加载或切换到PRAM执行</i>
1	EXEC	程序空间低16Kbyte切换选择 0: 低16K程序空间为Flash 1: 低16K程序空间为PRAM <i>注: PRAM加载正确后, 可设置EXEC=1将程序空间切换到PRAM执行, 切换后需插入一条NOP指令</i>
0 L	OAD	启动PRAM加载 0: 未启动或启动完成 1: 启动PRAM加载过程 <i>注: 启动PRAM加载过程后, CPU进入空闲状态, 因此软件将LOAD设为1后需插入三条NOP指令;只能在EXEC=0的情况下启动PRAM加载过程。</i>

8.14.3 程序举例

PRAM加载及切换:

```

PRAM_LOADRUN:
CLR    EA
MOV    DPTR, #PRAMLOCK
MOV    A, #0x55
MOVX   @DPTR, A      ;unlock PRAMCON
MOV    DPTR, #PRAMCON
MOVX   A, @DPTR
ORL    A, #0x01
MOVX   @DPTR, A      ;PRAM Load start
NOP
NOP
NOP
MOVX   A, @DPTR
JB     ACC.7, PRAMLOAD ;load error,reload

SETB   EA
MOVX   A, @DPTR
ORL    A, #0x02      ;ANL A, #FD (if Flash Run)
MOVX   @DPTR, A      ;PRAM run
NOP

MOV    DPTR, #PRAMLOCK
MOV    A, #0xAA
MOVX   @DPTR, A      ; lock PRAMCON
...

```

8.15 CRC校验模块

8.15.1 特性

- 对 PRAM 内容 CRC 校验，可设置初始值
- CRC 生成多项式采用 CRC-CCITT 标准： $X^{16}+X^{12}+X^5+1$ ，高位优先
- 2 种模式：高速 CRC 模式和普通 CRC 模式

为提高系统可靠性，G80F980内建1个CRC校验模块，可用来实时生成code的CRC校验码，采用的生成多项式采用CRC-CCITT标准： $X^{16}+X^{12}+X^5+1$ 。用户可利用此校验码和理论值比较，监测PRAM内容是否有变化。CRC理论值可存放Flash区域或其他位置（比如类EEPROM区，序列号区，用户识别码区等）用于软件对比。

当程序在Flash中运行时，可设置CRC_GO位启动CRC模块，校验完成后，CRC_GO自动清0，并置起CRCIF标志，如果中断允许位ECRC为1，则CPU产生CRC中断，程序响应中断后此标志位被硬件自动清0。

普通CRC模式：在CPU运行程序指令过程中，利用一些特殊指令的执行间隙，进行CRC校验，不会影响CPU的指令执行时间，但校验时间较长，且不可控。

高速CRC模式：为提高CRC的校验时间，可以打开CRC模块，让CPU进入IDLE模式，将执行高速CRC模式，校验时间较短，CRC中断可唤醒IDLE模式。

STOP模式下，CRC校验将停止，唤醒STOP后，CRC校验继续运行。

注：CRC校验模块只对PRAM的区域（16K）做CRC校验，且只有程序空间低16Kbyte切换选择为Flash时（EXEC=0）才可启动CRC校验

8.15.2 寄存器

Table 8.50 CRC控制寄存器

FFF8H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CRCCON	CRC_GO	CRCIF	-	-	-	-	-	-
读/写	读/写	读/写	-	-	-	-	-	-
复位值 (POR/WDT/LVR/PIN)	0	0	-	-	-	-	-	-

位编号	位符号	说明
7	CRC_GO	CRC启动控制位 0: 关闭CRC模块 1: 启动CRC模块,CRC校验完成后自动清0
6	CRCIF	CRC完成中断请求标志位 0: 未启动或启动后未完成。 1: CRC校验完成，向CPU请求中断，如果中断允许位ECRC为1，则CPU响应中断，程序响应中断后此标志位软件清0。

Table 8.51 CRC校验结果寄存器

FFF9H - FFFAH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
CRCDL(FFFAH)	CRCD7	CRCD6	CRCD5	CRCD4	CRCD3	CRCD2	CRCD1	CRCD0
CRCDH(FFF9H)	CRCD15	CRCD14	CRCD13	CRCD12	CRCD11	CRCD10	CRCD9	CRCD8
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

位编号	位符号	说明
7-0	CRCD[15:0]	CRC校验结果寄存器 启动CRC模块前，此寄存器值作为CRC校验初值，校验完成后，校验结果存储于此寄存器中，覆盖之前的数据。

注意：

1. CRC启动之前，需要对CRC结果寄存器赋初始值。
2. CRC_GO置为1启动CRC校验，在校验过程中，不允许对CRC结果寄存器CRCDL/CRCDH有修改动作。

8.16 代码选项**OP_LVREN:**

- 0: 禁止低电压复位功能（默认）
- 1: 允许低电压复位功能

OP_LVRLE:

- 00: 低电压复位设定电压为4.1V（默认）
- 01: 低电压复位设定电压为3.7V
- 1x: 低电压复位设定电压为2.5V

OP_RST:

- 0: 引脚P1.3用作复位引脚（默认）
- 1: 引脚P1.3用作I/O口

OP_WDT:

- 0: 允许WDT功能
- 1: 禁止WDT功能（默认）

OP_WDTPD:

- 0: 禁止掉电模式下WDT功能（默认）
- 1: 允许掉电模式下WDT功能

OP_WMT:（除了RC）

- 00: 振荡器采用最长预热计数时间（默认）
- 01: 振荡器采用比较长预热计数时间
- 10: 振荡器采用比较短预热计数时间
- 11: 振荡器采用最短预热计数时间

OP_SCM:

- 0: 在预热期间禁止时钟单元检测功能（默认）
- 1: 在预热期间允许时钟单元检测功能

OP_OSC:

- 0011: 双振荡器模式，内部128KHz RC振荡器作为振荡器1，12MHz 内部RC作为振荡器2（默认）
- 0110: 双振荡器模式，内部128KHz RC振荡器作为振荡器1，400k-16MHz晶体/陶瓷谐振器作为振荡器2

P3 N buffer enlarge Sink current:

- 0: P3口灌电流能力增强功能关闭（默认）
- 1: P3口灌电流能力增强功能开启

P1/P0 P buffer enlarge Driving current:

- 0: P1/P0口拉电流能力增强功能关闭（默认）
- 1: P1/P0口拉电流能力增强功能开启

OP_PWMDRV:

- 0: PWM输出引脚驱动能力减弱功能关闭（默认）
- 1: PWM输出引脚驱动能力减弱功能开启

OP_PWMPIN:

- 000: 管脚17~22按序号由低至高PWM0/PWM1/PWM2/PWM01/PWM11/PWM21排列（默认）（FNB41060）
- 001: 管脚17~22按序号由低至高PWM0/PWM01/PWM1/PWM11/PWM2/PWM21排列（SD05M50,FSB50550（B））
- 010: 管脚17~22按序号由低至高PWM21/PWM11/PWM01/PWM2/PWM1/PWM0排列（FNB41060（B）/IR213x）
- 011: 管脚17~22按序号由低至高PWM21/PWM2/PWM11/PWM1/PWM01/PWM0排列（SD05M50,FSB50550 / FAN7388）
- 1xx: 管脚17~22按序号由低至高PWM2/PWM21/PWM11/PWM1/PWM01/PWM0排列（TM2132）

9. 指令集

算术操作指令				
指令	功能描述	代码	字节	周期
ADD A, Rn	累加器加寄存器	0x28-0x2F	1	1
ADD A, direct	累加器加直接寻址字节	0x25	2	2
ADD A, @Ri	累加器加内部RAM	0x26-0x27	1	2
ADD A, #data	累加器加立即数	0x24	2	2
ADDC A, Rn	累加器加寄存器和进位位	0x38-0x3F	1	1
ADDC A, direct	累加器加直接寻址字节和进位位	0x35	2	2
ADDC A, @Ri	累加器加内部RAM和进位位	0x36-0x37	1	2
ADDC A, #data	累加器加立即数和进位位	0x34	2	2
SUBB A, Rn	累加器减寄存器和借位位	0x98-0x9F	1	1
SUBB A, direct	累加器减直接寻址字节和借位位	0x95	2	2
SUBB A, @Ri	累加器减内部RAM和借位位	0x96-0x97	1	2
SUBB A, #data	累加器减立即数和借位位	0x94	2	2
INC A	累加器加1	0x04	1	1
INC Rn	寄存器加1	0x08-0x0F	1	2
INC direct	直接寻址字节加1	0x05	2	3
INC @Ri	内部RAM加1	0x06-0x07	1	3
DEC A	累加器减1	0x14	1	1
DEC Rn	寄存器减1	0x18-0x1F	1	2
DEC direct	直接寻址字节减1	0x15	2	3
DEC @Ri	内部RAM减1	0x16-0x17	1	3
INC DPTR	数据指针加1	0xA3	1	4
MUL AB	累加器乘寄存器B	0xA4	1	11 20
DIV AB	累加器除以寄存器B	0x84	1	11 20
DA A	十进制调整	0xD4	1	1

逻辑操作指令				
指令	功能描述	代码	字节	周期
ANL A, Rn	累加器与寄存器	0x58-0x5F	1	1
ANL A, direct	累加器与直接寻址字节	0x55	2	2
ANL A, @Ri	累加器与内部RAM	0x56-0x57	1	2
ANL A, #data	累加器与立即数	0x54	2	2
ANL direct, A	直接寻址字节与累加器	0x52	2	3
ANL direct, #data	直接寻址字节与立即数	0x53	3	3
ORL A, Rn	累加器或寄存器	0x48-0x4F	1	1
ORL A, direct	累加器或直接寻址字节	0x45	2	2
ORL A, @Ri	累加器或内部RAM	0x46-0x47	1	2
ORL A, #data	累加器或立即数	0x44	2	2
ORL direct, A	直接寻址字节或累加器	0x42	2	3
ORL direct, #data	直接寻址字节或立即数	0x43	3	3
XRL A, Rn	累加器异或寄存器	0x68-0x6F	1	1
XRL A, direct	累加器异或直接寻址字节	0x65	2	2
XRL A, @Ri	累加器异或内部RAM	0x66-0x67	1	2
XRL A, #data	累加器异或立即数	0x64	2	2
XRL direct, A	直接寻址字节异或累加器	0x62	2	3
XRL direct, #data	直接寻址字节异或立即数	0x63	3	3
CLR A	累加器清零	0xE4	1	1
CPL A	累加器取反	0xF4	1	1
RL A	累加器左环移位	0x23	1	1
RLC A	累加器连进位标志左环移位	0x33	1	1
RR A	累加器右环移位	0x03	1	1
RRC A	累加器连进位标志右环移位	0x13	1	1
SWAP A	累加器高4位与低4位交换	0xC4	1	4

数据传送指令				
指令	功能描述	代码	字节	周期
MOV A, Rn	寄存器送累加器	0xE8-0xEf	1	1
MOV A, direct	直接寻址字节送累加器	0xE5	2	2
MOV A, @Ri	内部RAM送累加器	0xE6-0xE7	1	2
MOV A, #data	立即数送累加器	0x74	2	2
MOV Rn, A	累加器送寄存器	0xF8-0xFF	1	2
MOV Rn, direct	直接寻址字节送寄存器	0xA8-0xAF	2	3
MOV Rn, #data	立即数送寄存器	0x78-0x7F	2	2
MOV direct, A	累加器送直接寻址字节	0xF5	2	2
MOV direct, Rn	寄存器送直接寻址字节	0x88-0x8F	2	2
MOV direct1, direct2	直接寻址字节送直接寻址字节	0x85	3	3
MOV direct, @Ri	内部RAM送直接寻址字节	0x86-0x87	2	3
MOV direct, #data	立即数送直接寻址字节	0x75	3	3
MOV @Ri, A	累加器送内部RAM	0xF6-0xF7	1	2
MOV @Ri, direct	直接寻址字节送内部RAM	0xA6-0xA7	2	3
MOV @Ri, #data	立即数送内部RAM	0x76-0x77	2	2
MOV DPTR, #data16	16位立即数送数据指针	0x90	3	3
MOVC A, @A+DPTR	程序代码送累加器（相对数据指针）	0x93	1	7
MOVC A, @A+PC	程序代码送累加器（相对程序计数器）	0x83	1	8
MOVX A, @Ri	外部RAM送累加器（8位地址）	0xE2-0xE3	1	5
MOVX A, @DPTR	外部RAM送累加器（16位地址）	0xE0	1	6
MOVX @Ri, A	累加器送外部RAM（8位地址）	0xF2-F3	1	4
MOVX @DPTR, A	累加器送外部RAM（16位地址）	0xF0	1	5
PUSH direct	直接寻址字节压入栈顶	0xC0	2	5
POP direct	栈顶弹至直接寻址字节	0xD0	2	4
XCH A, Rn	累加器与寄存器交换	0xC8-0xCF	1	3
XCH A, direct	累加器与直接寻址字节交换	0xC5	2	4
XCH A, @Ri	累加器与内部RAM交换	0xC6-0xC7	1	4
XCHD A, @Ri	累加器低4位与内部RAM低4位交换	0xD6-0xD7	1	4

控制程序转移指令				
指令	功能描述	代码	字节	周期
ACALL addr11	2KB内绝对调用	0x11-0xF1	2	7
LCALL addr16	64KB内长调用	0x12	3	7
RET	子程序返回	0x22	1	8
RETI	中断返回	0x32	1	8
AJMP addr11	2KB内绝对转移	0x01-0xE1	2	4
LJMP addr16	64KB内长转移	0x02	3	5
SJMP rel	相对短转移	0x80	2	4
JMP @A+DPTR	相对长转移	0x73	1	6
JZ rel (不发生转移) (发生转移)	累加器为零转移	0x60	2	3 5
JNZ rel (不发生转移) (发生转移)	累加器为非零转移	0x70	2	3 5
JC rel (不发生转移) (发生转移)	C置位转移	0x40	2	2 4
JNC rel (不发生转移) (发生转移)	C清零转移	0x50	2	2 4
JB bit, rel (不发生转移) (发生转移)	直接寻址位置位转移	0x20	3	4 6
JNB bit, rel (不发生转移) (发生转移)	直接寻址位清零转移	0x30	3	4 6
JBC bit, rel (不发生转移) (发生转移)	直接寻址位置位转移并清该位	0x10	3	4 6
CJNE A, direct, rel (不发生转移) (发生转移)	累加器与直接寻址字节不等转移	0xB5	3	4 6
CJNE A, #data, rel (不发生转移) (发生转移)	累加器与立即数不等转移	0xB4	3	4 6
CJNE Rn, #data, rel (不发生转移) (发生转移)	寄存器与立即数不等转移	0xB8-0xBf	3	4 6
CJNE @Ri, #data, rel (不发生转移) (发生转移)	内部RAM与立即数不等转移	0xB6-0xB7	3	4 6
DJNZ Rn, rel (不发生转移) (发生转移)	寄存器减1不为零转移	0xD8-0xDf	2	3 5
DJNZ direct, rel (不发生转移) (发生转移)	直接寻址字节减1不为零转移	0xD5	3	4 6
NOP	空操作	0	1	1

位操作指令				
指令	功能描述	代码	字节	周期
CLR C	C清零	0xC3	1	1
CLR bit	直接寻址位清零	0xC2	2	3
SETB C	C置位	0xD3	1	1
SETB bit	直接寻址位置位	0xD2	2	3
CPL C	C取反	0xB3	1	1
CPL bit	直接寻址位取反	0xB2	2	3
ANL C, bit	C逻辑与直接寻址位	0x82	2	2
ANL C, /bit	C逻辑与直接寻址位的反	0xB0	2	2
ORL C, bit	C逻辑或直接寻址位	0x72	2	2
ORL C, /bit	C逻辑或直接寻址位的反	0xA0	2	2
MOV C, bit	直接寻址位送C	0xA2	2	2
MOV bit, C	C送直接寻址位	0x92	2	3

10. 电气特性

极限参数*

直流供电电压.....	-0.3V to +6.0V
输入/输出电压.....	GND-0.3V to $V_{DD}+0.3V$
工作环境温度.....	-40°C to +105°C
存储温度.....	-55°C to +125°C
FLASH存储器写/擦除操作.....	0°C to +85°C

*注释

如果器件的工作条件超过左列“极限参数”的范围，将造成器件永久性破坏。只有当器件工作在说明书所规定的范围内时功能才能得到保障。器件在极限参数列举的条件下工作将会影响到器件工作的可靠性。

直流电气特性 ($V_{DD} = 2.4 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值*	最大值	单位	条件
工作电压	V_{DD}	2.4	5.0	5.5	V	$400k \leq f_{OSC} \leq 11 MHz$
工作电流	I_{OP1}	-	40	40	mA	$f_{sys} = 11 MHz$, $V_{DD} = 5.0V$ 所有输出引脚无负载（所有数字输入引脚不浮动）； CPU打开（执行NOP指令）；WDT打开；PLL打开 所有功能模块关闭
	I_{OP2}	-	40	80	μA	$f_{OSC} = 128kHz$, $V_{DD} = 5.0V$ 所有输出引脚无负载（所有数字输入引脚不浮动）； CPU打开（执行NOP指令）；WDT打开；PLL关闭； 所有功能模块关闭
待机电流 (空闲模式)	I_{SB1}	-	30	30	mA	$f_{sys} = 11 MHz$, $V_{DD} = 5.0V$ 所有输出引脚无负载（所有数字输入引脚不浮动）； CPU停止工作 (IDLE)；WDT关；PLL打开 LVR打开，关闭其它所有功能
	I_{SB2}	-	20	10	μA	$f_{OSC} = 128kHz$, $V_{DD} = 5.0V$ 所有输出引脚无负载（所有数字输入引脚不浮动）； CPU停止工作 (IDLE)；WDT关；PLL关闭 LVR打开，关闭其它所有功能
待机电流 (掉电模式)	I_{SB3}	-	15	30	μA	$f_{OSC} = 128kHz$, $V_{DD} = 5.0V$ 所有输出引脚无负载；CPU停止（掉电模式）； WDT关闭；LVR关闭；PLL关闭；关闭其它所有功能
	I_{SB4}	-	10	30	μA	Osc off, $V_{DD} = 5.0V$ 所有输出引脚无负载；CPU停止（掉电模式）； WDT关闭；LVR关闭；PLL关闭；关闭其它所有功能
LVR电流	I_{LVR1}	-	1	-	μA	LVR打开，LVR电平 = 4.1V/3.7V/2.8V
WDT电流	I_{WDT}	-	1	3	μA	所有输出引脚无负载；看门狗打开 $V_{DD} = 5.0V$
输入低电压1	V_{IL1}	GND	-	$0.3 \times V_{DD}$	V	I/O端口
输入高电压1	V_{IH1}	$0.7 \times V_{DD}$	-	V_{DD}	V	I/O端口
输入低电压2	V_{IL2}	GND	-	$0.2 \times V_{DD}$	V	\overline{RST} , T0, T1, T2, T3, T4, INT0/1/4, RXD, FLT, CAP0/1/2
输入高电压2	V_{IH2}	$0.8 \times V_{DD}$	-	V_{DD}	V	\overline{RST} , T0, T1, T2, T3, T4, INT0/1/4, RXD, FLT, CAP0/1/2
输入漏电流	I_{IL}	-1	-	1	μA	输入口, $V_{IN} = V_{DD}$ 或者GND
上拉电阻	R_{PH1}	-	30	-	k Ω	$V_{DD} = 5.0V$, $V_{IN} = GND$

续上表

输出高电压1	V_{OH1}	$V_{DD} - 0.7$	-	-	V	I/O端口, $I_{OH} = -10mA$, $V_{DD} = 5.0V$ 除P0, P1端口以外
输出高电压2	V_{OH2}	$V_{DD} - 0.7$	-	-	V	I/O端口, $I_{OH} = -20mA$, $V_{DD} = 5.0V$, P0, P1端口 (Note 4)
输出高电压3	V_{OH3}	$V_{DD} - 0.7$			V	P2.0~P2.5, $I_{OH} = -4mA$, $V_{DD} = 5.0V$ (Note 5)
输出低电压1	V_{OL1}	-	-	GND + 0.6	V	I/O端口, $I_{OL} = 15mA$, $V_{DD} = 5.0V$, 除P3端口以外
输出低电压2	V_{OL2}	-	-	GND + 1.5	V	I/O端口, $I_{OL} = 120mA$, $V_{DD} = 5.0V$, P3端口 (Note 4)
输出低电压3	V_{OL3}	-	-	GND + 0.6	V	P2.0~P2.5, $I_{OL} = 4mA$, $V_{DD} = 5.0V$ (Note 5)
电压基准源精度	V_{BG}	1.23	1.25	1.27	V	$V_{DD} = 5.0V$, $T_A = -40 \sim +105^\circ C$ (Note3)

注意:

1. “*”表示典型值下的数据是在5.0V, 25°C下测得的, 除非另有说明。
2. 流过 V_{DD} 的最大电流值须小于150mA。流过GND的最大电流值须小于200mA。
3. 设计保证, 实际测试时未包含温度。
4. 驱动口能力加大受代码选项控制。(可增加P3口灌电流能力或P0,P1口拉电流能力)
5. 驱动口能力减小受代码选项控制(由OP_PWMDRV控制P2.0到P2.5减小驱动能力)。

高速转换模式下10BIT模/数转换器电气特性1 (1LSB = $V_{DD}/1024$)转换速率为1M sps。($V_{DD} = 2.7 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
工作电压范围	V_{AD}	2.7	5.0	5.5	V	
精度	N_R	-	10	-	bit	$V_{REF} = 5.0V$
A/D输入电压	V_{AIN}	GND	-	V_{REF}	V	
A/D输入电阻	R_{AIN}	2	-	-	M Ω	$V_{IN} = 5.0V$
外部模拟参考电压	V_{REF}	2.5	-	V_{DD}	V	
模拟输入电压源推荐阻抗	Z_{AIN}	-	-	10	k Ω	
A/D转换电流	I_{AD}	-	1.5	3	mA	ADC模块工作, $V_{DD} = 5.0V$
微分非线性误差	D_{LE}	-	-	± 1	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 20MHz$
积分非线性误差	I_{LE}	-	-	± 2	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 20MHz$
满刻度误差	E_F	-	-	± 3	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 20MHz$
偏移量误差	E_Z	-	-	± 3	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 20MHz$
总绝对误差	E_{AD}	-	-	± 3	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 20MHz$
ADC时钟周期	t_{AD}	0.05	-	80	μs	
ADC采样时间	t_{SAMP}	0.4	0.5	-	μs	
总转换时间	T_{CON}	14	-	27	t_{AD}	

低速转换模式下12BIT模/数转换器电气特性2 (1LSB = $V_{DD}/4096$)转换速率为100K sps。 ($V_{DD} = 2.7 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
工作电压范围	V_{AD}	2.7	5.0	5.5	V	
精度	N_R	-	12	-	bit	$V_{REF} = 5.0V$
A/D输入电压	V_{AIN}	GND	-	V_{REF}	V	
A/D输入电阻	R_{AIN}	2	-	-	M Ω	$V_{IN} = 5.0V$
外部模拟参考电压	V_{REF}	2.5	-	V_{DD}	V	
模拟电压源推荐阻抗	Z_{AIN}	-	-	5	k Ω	
A/D转换电流	I_{AD}	-	1.5	3	mA	ADC模块工作, $V_{DD} = 5.0V$
微分非线性误差	D_{LE}	-	-	± 1	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 2.7MHz$
积分非线性误差	I_{LE}	-	-	± 2	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 2.7MHz$
满刻度误差	E_F	-	-	± 8	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 2.7MHz$
偏移量误差	E_Z	-	-	± 8	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 2.7MHz$
总绝对误差	E_{AD}	-	-	± 8	LSB	$V_{DD} = 5.0V$, $V_{REF} = 5.0V$, $ADC\ CLK \leq 2.7MHz$
ADC时钟周期	t_{AD}	0.3	-	80	μs	
ADC采样时间	t_{SAMP}	2	4	-	μs	
总转换时间	T_{CON}	16	-	29	t_{AD}	

温度传感器特性 ($V_{DD} = 2.7 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
工作温度	T	-40		150	$^\circ C$	
温度传感器线性度	T_L		± 3	± 5	$^\circ C$	$V_{DD} = 5V$, $T_A = -40 \sim +105^\circ C$
温度传感器斜率	T_{Slope}		2.5085		mV/ $^\circ C$	
25度时输出电压	V_{25}		0.75		V	

比较器1/2电气特性 ($V_{DD} = 2.7 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
输入失调电压	V_{IO}	-	2	3	mV	
输入共模电压范围	V_{ICM}	0	-	$V_{DD} - 1.2$	V	
小信号响应时间	T_{RS1}	-	1	2	μs	$V_{DD} = 5V$, $C1/2N = 1.20V$, $C1/2P = 1.15V$ Step to 1.25V
大信号响应时间	T_{RS2}	-	0.3	0.5	μs	$V_{DD} = 5V$, $C1/2N = 1V$, $C1/2P = 0V$ Step to 2V
比较器施密特窗口1	V_{SMT1}	-	0	-	mv	$V_{DD} = 5V$, $C1/2SMT[1:0] = 00$, 下降沿窗口
比较器施密特窗口2	V_{SMT2}	-	10	16	mv	$V_{DD} = 5V$, $C1/2SMT[1:0] = 01$, 下降沿窗口
比较器施密特窗口3	V_{SMT3}	-	20	30	mv	$V_{DD} = 5V$, $C1/2SMT[1:0] = 10$, 下降沿窗口
比较器施密特窗口4	V_{SMT4}	-	50	70	mv	$V_{DD} = 5V$, $C1/2SMT[1:0] = 11$, 下降沿窗口
内部反相端参考源精度	V_{N1}	-	-	$\pm 1\%$		$V_{DD} = 5V$

放大器1/2/3/4电气特性 ($V_{DD} = 2.7 - 5.5V$, $GND = 0V$, $T_A = +25^\circ C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
输入失调电压	V_{IO}	-	2	3	mV	$T_A = 25^\circ C$

G80F980

集成MDU+CORDIC协处理器、ADC和PWM的增强型8051微控制器

输入共模电压	V_{ICM}	0	-	$V_{DD} - 1.2$	V	
温度系数		-	0.015%	-		$V_{OUT} = 2.5V$, 温度变每化 $1^{\circ}C$, 输出电压的变化率
转换速率	SR	5	-	-	V/us	内部增益电压跟随器模式
输出电压范围	V_{OUR}	0.1	-	$V_{DD} - 0.5$	V	
电压抑制比	SVR	65	80	-	dB	直流特性
共模抑制比	CMRR	60	80	-	dB	直流特性
增益带宽	BW	2	8	-	MHz	电压跟随器模式
内部增益系数精度	Gain		$\pm 0.5\%$	$\pm 1\%$		$V_{DD} = 5V, T_A = -40 \sim +105^{\circ}C$

交流电气特性 ($V_{DD} = 2.4 - 5.5V$, $GND = 0V$, $T_A = +25^{\circ}C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
复位脉冲宽度	t_{RESET}	10	-	-	μs	低电平有效
复位引脚上拉电阻	R_{RPH}	-	30	-	$k\Omega$	$V_{DD} = 5.0V, V_{IN} = GND$
频率稳定性 (RC)	$ \Delta F /F$	-	-	0.5	%	内部RC振荡器频率精度: (1024个周期求平均值) $ F_{RC} - 12MHz /12MHz \times 100\%$ ($V_{DD} = 2.4 - 5.5V, T_A = 25^{\circ}C$)
	$ \Delta F /F$	-	-	± 2	%	内部RC振荡器频率精度: (1024个周期求平均值) $ F_{RC} - 12MHz /12MHz \times 100\%$ ($V_{DD} = 2.4 - 5.5V, T_A = -40^{\circ}C$ 至 $+105^{\circ}C$, 设计保证, 不在生产中测试)

PLL特性 ($V_{DD} = 2.4 - 5.5V$, $GND = 0V$, $T_A = +25^{\circ}C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
PLL输入时钟频率	f_{PLL_IN}	16	-	16	MHz	
PLL输入时钟Duty	D_{PLL_IN}	40	-	60	%	
PLL输出时钟频率	f_{PLL_OUT}	32		100	MHz	
PLL建立时间	T_{PLL}		2		ms	
Jitter	Jitter			300	ps	
PLL频率稳定性			0.5	± 1	%	(1024个周期求平均值 $V_{DD} = 2.4 - 5.5V, T_A = 25^{\circ}C$)

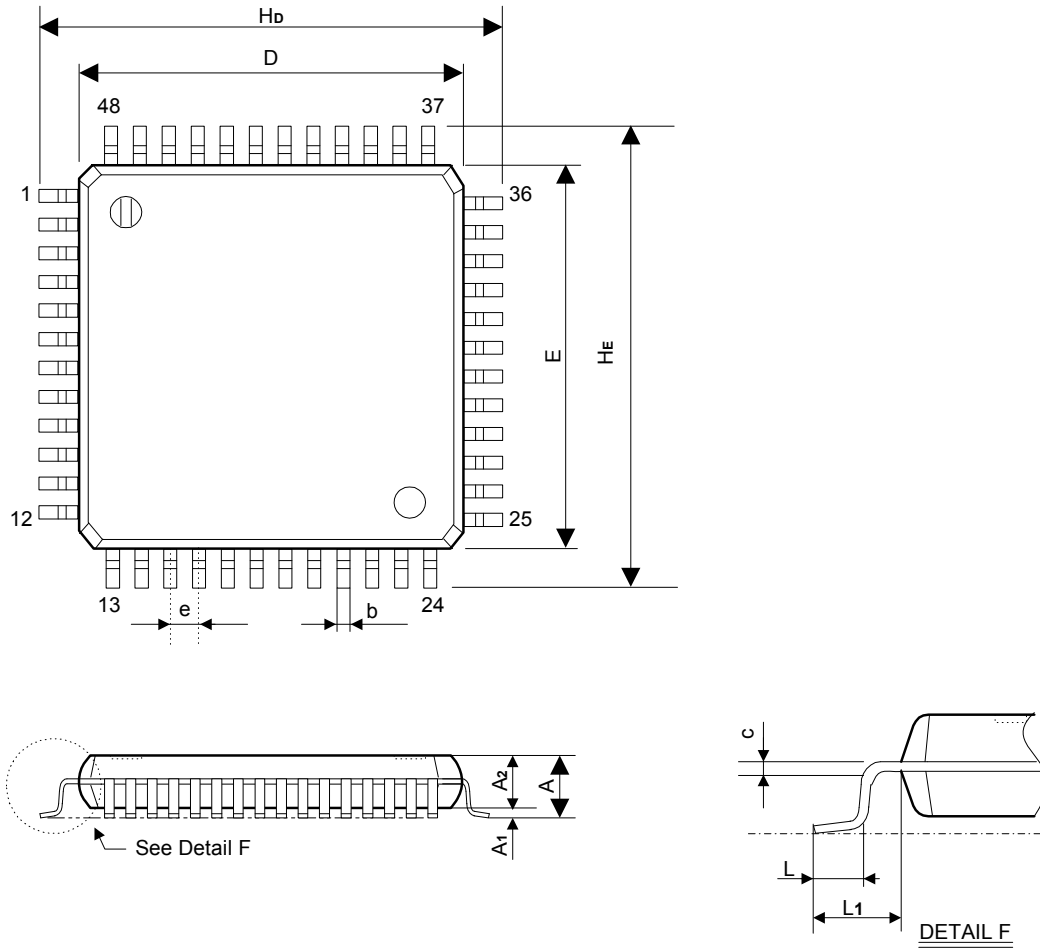
低电压复位电气特性 ($V_{DD} = 2.4 - 5.5V$, $GND = 0V$, $T_A = +25^{\circ}C$, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
LVR电压1	V_{LVR1}	4.0	4.1	4.2	V	LVR1使能
LVR电压2	V_{LVR2}	3.6	3.7	3.8	V	LVR2使能
LVR电压3	V_{LVR2}	2.4	2.5	2.6	V	LVR3使能
辅助LVR	V_{LVR}	2.1	2.2	2.3	V	内部辅助LVR (始终有效)
LVR电压检测迟滞窗口	V_{SMTLV}	-	50	80	mv	
LVR低电压复位宽度	T_{LVR}	-	30	-	μs	

11. 封装信息

TQFP48外形尺寸

单位: 英寸/毫米



Symbol	Dimensions in inches		Dimensions in mm	
	MIN	MA	MIN	MA
A	---	0.047	---	1.2
A1	0.002	0.006	0.05	0.15
A2	0.035	0.041	0.9	1.05
D	0.270	0.281	6.85	7.15
E	0.270	0.281	6.85	7.15
H_D	0.346	0.362	8.8	9.2
H_E	0.346	0.362	8.8	9.2
b	0.007	0.010	0.19	0.26
e	0.020 TYP		0.500 TYP	
c	0.004	0.008	0.090	0.200
L	0.018	0.030	0.45	0.75
L1	0.033	0.045	0.85	1.15
θ	0°	10°	0°	10°

12. 规格更改记录

G80F980 规格书版本记录		
版本号	记录	日期
V1.00	初始版本	2016.04
V1.01	PLL 最高输出改为 84MH	2016.06
V1.10	1. 工作电压典型值改为 18mA 2. 空闲模式电流典型值 16mA 3. 修改 I_{SB2} I_{SB4} 电流最大值 4. 修改 PLL 输入频率范围	2016.09

13 业务联络

南京立超电子科技有限公司
中国南京市和燕路 251 号金港大厦 A 幢 2406 室
ZIP:210028
Tel: 0086-25-83306839/83310926
Fax: 0086-25-83737785
Email: Yunchao.Ding@sykee.net
Website:Http://www.dycmcu.com

14 免责声明

规格书中所出现的信息在出版当时相信是正确的，然而本公司对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明，本公司不保证或表示这些没有进一步修改的应用将是适当的，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本产品不授权使用于救生、维生器件或系统中做为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考我们的网址 <http://www.dycmcu.com>