



## 1. 特性

双核结构：**16位增强型DSP+8位MCU**

**DSP及其外设：**

- CPU
  - 最大100MIPS，16位宽度数据总线
  - 单周期同步执行16位MAC和存取操作
  - 32级桶形移位器
  - 单周期执行“取双操作数”和“取指令”
- 片上存储器
  - 64K X 8bits Flash作为DSP的非易失性存储器
  - 16K X 24bits SRAM作为DSP运行程序的Code Memory
- 脉冲宽度调制（PWM）模块
  - 6路（3对）16位PWM输出
  - 3种时基计数模式：单事件模式、边沿对齐模式与中心对齐模式
  - 每对PWM提供2种输出模式：互补输出和独立输出
  - PWM输出有效极性选择
  - 死区产生逻辑
  - 自动更新周期、占空比、输出极性以及ADC触发时刻
  - 5个保护输入引脚
  - 每个保护输入引脚上的Debounce时间可设置为八档（0 $\mu$ s - 16 $\mu$ s）
  - 后分频系数
- 正交编码器接口（QEI）
  - QEA, QEB, Index三路信号输入
  - 16位增/减双向位置计数器
  - 5种可屏蔽中断：计数器重载/Index信号中断；Index信号发生错误中断；QEA/B输入信号相位错误中断；测速定时器溢出中断；测速定时器捕捉中断
- 2 X 8通道14位模数转换器（ADC）
  - 10MSPS转换速率
  - 两路同步并行采样模式、两路独立顺序采样模式与单路顺序采样模式
- 4个16位定时器DTimer0/1/2/3

- 3个外部中断DXINT0/1/2
- 1个JTAG接口实现DSP的仿真调试
- 4个输入捕捉功能引脚
  - 使用DTimer2/3之一作为时基
  - 2级FIFO缓冲器
  - 捕捉事件可设置为上升沿、下降沿、上升/下降沿
  - 输入分频/滤波设置
- 3路输出比较引脚
  - 使用DTimer3/2之一作为时基
  - 单次比较模式、双比较模式、边沿对齐PWM输出模式
- 2个运算放大器

**MCU及其外设：**

- CPU
  - 基于8051指令流水线结构的8位单片机
  - 最大工作频率16MHz，CISC指令集
- 片上存储器
  - 64KBytes Flash
  - 4KBytes SRAM
  - 1K X 8bits Flash作为ISP程序存储器
- 56个CMOS双向I/O管脚（功能复用），内建输入上拉电阻
- SPI接口（主/从模式）
- 2个增强型UART
- 3个16位定时器Timer0/1/2
- 3个外部中断INT0/1/2
- 8通道10位逐次逼近型模数转换器（ADC）
- 3路12位脉冲宽度调制（PWM）输出
- 1个JTAG接口实现MCU的仿真调试

**其它：**

- 工作电压：V<sub>DD</sub> = 2.7 - 5.5V
- 停振检测
  - 外部晶振失效后切换至内部RC作为系统频率并产生中断
- ISP功能
- 看门狗定时器（WDT）
- 低电压复位LVR
- 封装：LQFP100



### 2. 概述

SH99F100是一款双核（DSP和MCU）芯片，适用于工业控制场合，如电机调速、变频电源等。MCU可以用来处理开关量、输入输出以及通讯，而DSP只需专注于控制算法的运算。

DSP的架构由三个并行的计算单元（ALU, MAC, SHIFTER）、两个独立的地址发生器（DAG）和一个强大的程序序列发生器（PSQ）组成。计算单元提供了所有处理器能够提供的计算功能，包括单周期乘/累加，位操作，移位操作等。双数据地址发生器使得处理器能够在一个周期内从两个数据存储区（Data Memory和Program Memory）同时取出两个数据。程序序列发生器实现了所有指令的单周期操作，快速的中断响应以及嵌套的非CPU干预的循环。16位的PWM模块、正交编码接口（QEI）与14位的流水线型ADC与运算放大器（OP）作为DSP核的内部外设，能够方便地实现多种工业控制算法。

MCU核是一种高速高效率8051兼容核，在同样的振荡频率下，较之传统的8051芯片具有运行更快速，性能更优越的特性，其资源有内置256字节SRAM，外部3840字节SRAM，3个16位定时器和3个外部中断，同时还集成了2个增强型UART，1个SPI接口。

### 3. 方框图

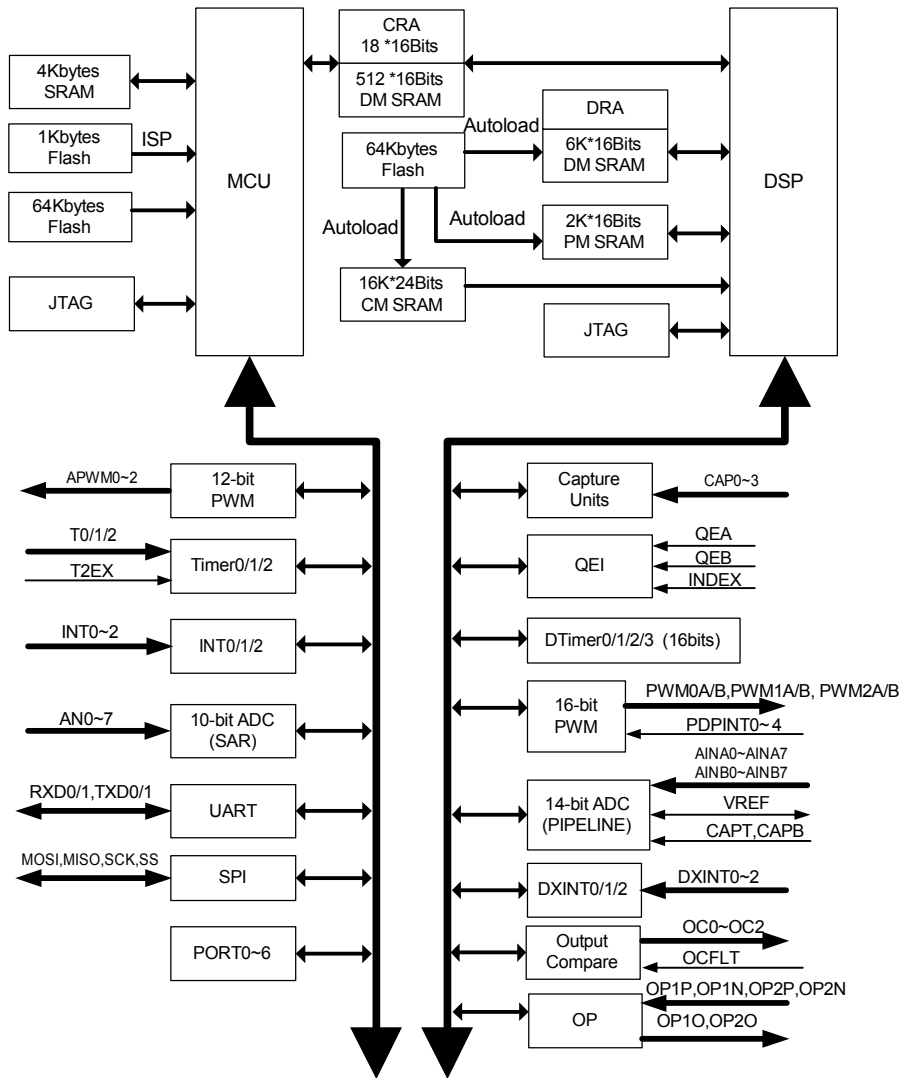


图3-1



4. 引脚配置

LQFP100

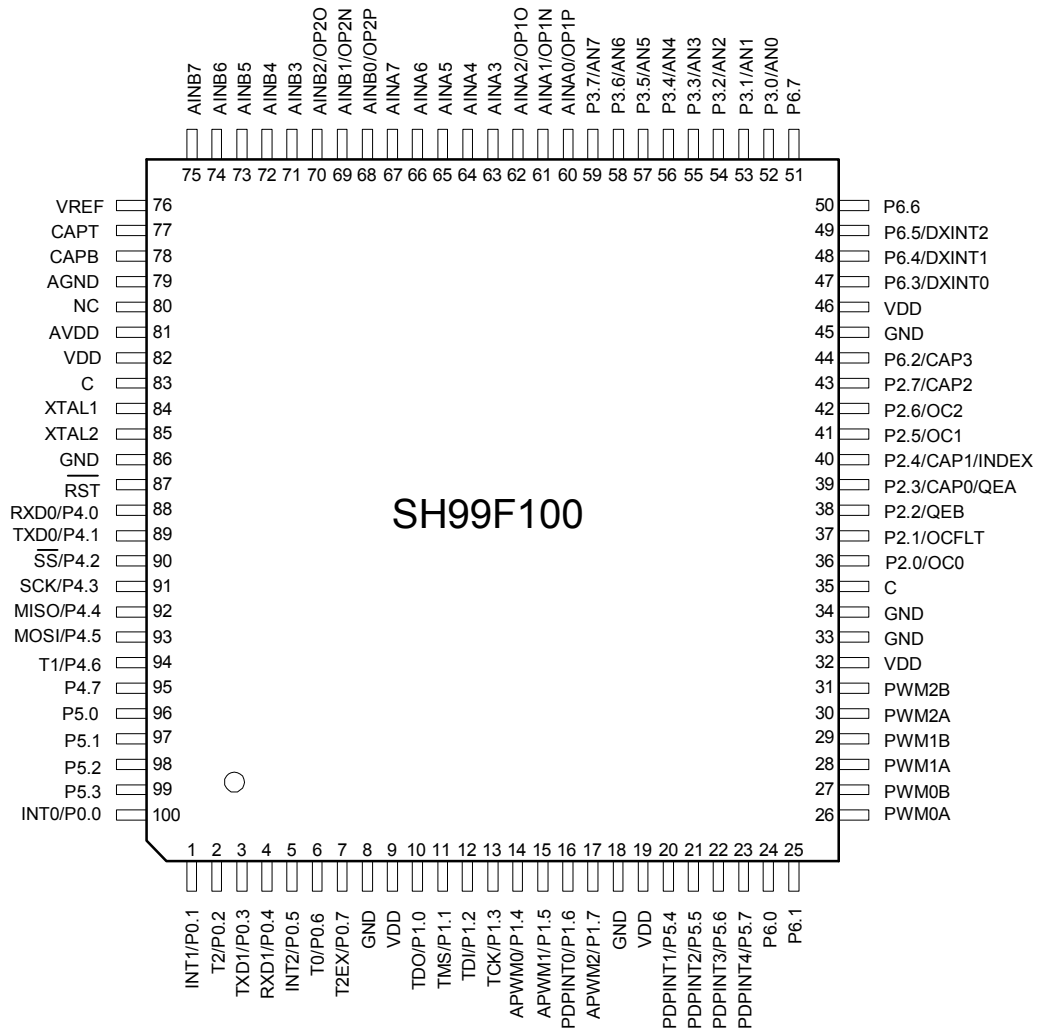


图4-1

共计100个管脚。



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	INT1/P0.1	P0.1	51	P6.7	-----
2	T2/P0.2	P0.2	52	P3.0/AN0	P3.0
3	TXD1/P0.3	P0.3	53	P3.1/AN1	P3.1
4	RXD1/P0.4	P0.4	54	P3.2/AN2	P3.2
5	INT2/P0.5	P0.5	55	P3.3/AN3	P3.3
6	T0/P0.6	P0.6	56	P3.4/AN4	P3.4
7	T2EX/P0.7	P0.7	57	P3.5/AN5	P3.5
8	GND	-----	58	P3.6/AN6	P3.6
9	V <sub>DD</sub>	-----	59	P3.7/AN7	P3.7
10	TDO/P1.0	P1.0	60	AINA0/OP1P	AINA0
11	TMS/P1.1	P1.1	61	AINA1/OP1N	AINA1
12	TDI/P1.2	P1.2	62	AINA2/OP1O	AINA2
13	TCK/P1.3	P1.3	63	AINA3	-----
14	APWM0/P1.4	P1.4	64	AINA4	-----
15	APWM1/P1.5	P1.5	65	AINA5	-----
16	PDPINT0/P1.6	P1.6	66	AINA6	-----
17	APWM2/P1.7	P1.7	67	AINA7	-----
18	GND	-----	68	AINB0/OP2P	AINB0
19	V <sub>DD</sub>	-----	69	AINB1/OP2N	AINB1
20	PDPINT1/P5.4	P5.4	70	AINB2/OP2O	AINB2
21	PDPINT2/P5.5	P5.5	71	AINB3	-----
22	PDPINT3/P5.6	P5.6	72	AINB4	-----
23	PDPINT4/P5.7	P5.7	73	AINB5	-----
24	P6.0	P6.0	74	AINB6	-----
25	P6.1	P6.1	75	AINB7	-----
26	PWM0A	-----	76	VREF	-----
27	PWM0B	-----	77	CAPT	-----
28	PWM1A	-----	78	CAPB	-----
29	PWM1B	-----	79	AGND	-----
30	PWM2A	-----	80	AGND	-----
31	PWM2B	-----	81	AV <sub>DD</sub>	-----
32	V <sub>DD</sub>	-----	82	V <sub>DD</sub>	-----
33	GND	-----	83	C	-----
34	GND	-----	84	XTAL1	-----
35	C	-----	85	XTAL2	-----
36	P2.0/OC0	P2.0	86	GND	-----
37	P2.1/OCFLT	P2.1	87	RST	-----
38	P2.2/QEB	P2.2	88	RXD0/P4.0	P4.0
39	P2.3/CAP0/QEA	P2.3	89	TXD0/P4.1	P4.1
40	P2.4/CAP1/INDEX	P2.4	90	SS/P4.2	P4.2
41	P2.5/OC1	P2.5	91	SCK/P4.3	P4.3
42	P2.6/OC2	P2.6	92	MISO/P4.4	P4.4
43	P2.7/CAP2	P2.7	93	MOSI/P4.5	P4.5
44	P6.2/CAP3	P6.2	94	T1/P4.6	P4.6
45	GND	-----	95	P4.7	-----
46	V <sub>DD</sub>	-----	96	P5.0	-----
47	P6.3/DXINT0	P6.3	97	P5.1	-----
48	P6.4/DXINT1	P6.4	98	P5.2	-----
49	P6.5/DXINT2	P6.5	99	P5.3	-----
50	P6.6	-----	100	INT0/P0.0	P0.0



5. 引脚描述

引脚编号	类型	说明
<b>PORT</b>		
P6.0 - P6.7	I/O	8位双向I/O端口
P5.0 - P5.7	I/O	8位双向I/O端口
P4.0 - P4.7	I/O	8位双向I/O端口
P3.0 - P3.7	I/O	8位双向I/O端口
P2.0 - P2.7	I/O	8位双向I/O端口
P1.0 - P1.7	I/O	8位双向I/O端口
P0.0 - P0.7	I/O	8位双向I/O端口
<b>Timer (MCU)</b>		
T0	I/O	Timer0外部输入或比较输出
T1	I/O	Timer1外部输入或比较输出
T2	I/O	Timer2外部输入/波特率时钟输出
T2EX	I	Timer2重载/捕捉/方向控制
<b>PWM控制器 (MCU)</b>		
APWM2	O	12位辅助PWM模块2的输出通道
APWM1	O	12位辅助PWM模块1的输出通道
APWM0	O	12位辅助PWM模块0的输出通道
<b>EUART (MCU)</b>		
RXD0	I/O	EUART0数据输入/输出引脚
TXD0	O	EUART0数据输出引脚
RXD1	I/O	EUART1数据输入/输出引脚
TXD1	O	EUART1数据输出引脚
<b>SPI (MCU)</b>		
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从属选择引脚
<b>SAR ADC (MCU)</b>		
AN0 - AN7	I	MCU中SAR型ADC输入通道
<b>外部中断 (MCU)</b>		
INT0 - INT2	I	MCU外部中断0, 1, 2
<b>PWM控制器 (DSP)</b>		
PWM0A	O	16位PWM模块的PWM0A通道输出
PWM0B	O	16位PWM模块的PWM0B通道输出
PWM1A	O	16位PWM模块的PWM1A通道输出
PWM1B	O	16位PWM模块的PWM1B通道输出
PWM2A	O	16位PWM模块的PWM2A通道输出
PWM2B	O	16位PWM模块的PWM2B通道输出
PDPINT0 - PDPINT4	I	16位PWM模块的功率保护输入引脚



续上表

引脚编号	类型	说明
<b>PIPELINE ADC (DSP)</b>		
AINA0 - 7、AINB0 - 7	I	Pipeline型ADC输入通道
VREF	I/O	Pipeline型ADC外部基准电压输入或内部基准电压输出
CAPT	O	去耦电容T
CAPB	O	去耦电容B
<b>运算放大器 (OP)</b>		
OP1P	I	运算放大器1正相输入
OP1N	I	运算放大器1反相输入
OP1O	O	运算放大器1输出
OP2P	I	运算放大器2正相输入
OP2N	I	运算放大器2反相输入
OP2O	O	运算放大器2输出
<b>正交编码器接口QEI (DSP)</b>		
QEA	I	编码器A相输入
QEB	I	编码器B相输入
INDEX	I	编码器索引信号输入
<b>输入捕捉CAP (DSP)</b>		
CAP0 - CAP3	I	输入捕捉输入引脚
<b>输出比较CMPOUT (DSP)</b>		
OC0 - OC2	O	输出比较引脚
OCFLT	I	输出比较引脚在带保护PWM模式下的保护输入引脚
<b>外部中断 (DSP)</b>		
DXINT0 - DXINT2	I	DSP外部中断0, 1, 2
<b>复位、复位、时钟、电源</b>		
$\overline{\text{RST}}$	I	该引脚上保持10us以上的低电平，CPU将复位。由于有内建100k上拉电阻连接到V <sub>DD</sub> ，所以仅接一个外部电容即可实现上电复位。
XTAL1	I	振荡器输入
XTAL2	O	振荡器输出
GND	P	数字地
V <sub>DD</sub>	P	数字电源
AGND	P	模拟地
AV <sub>DD</sub>	P	模拟电源
<b>稳压源</b>		
C	O	内建稳压源滤波电容引脚，推荐选用47μF钽电容接地，LQFP100封装中两个引脚必须有外部PCB连接
<b>编程接口 (MCU): JTAG</b>		
TDO	O	调试接口：测试数据输出
TMS	I	调试接口：测试模式选择
TDI	I	调试接口：测试数据输入
TCK	I	调试接口：测试时钟输入



## 6. 寄存器映像

### 6.1 MCU寄存器映像

MCU内置256字节的直接寻址寄存器，包括通用数据存储器 and 特殊功能存储器（SFR），MCU的SFR有以下几种：

<b>CPU内核寄存器：</b>	ACC, B, PSW, SP, DPL, DPH
<b>CPU增强内核寄存器：</b>	AUXC, DPL1, DPH1, INSCON, XPAGE1, XPAGE
<b>电源和时钟控制寄存器：</b>	PCON, SUSLO
<b>Flash寄存器：</b>	IB_OFFSET, XPAGE1, XPAGE, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5, FLASHCON
<b>数据存储页寄存器：</b>	XPAGE1, XPAGE
<b>系统时钟控制寄存器：</b>	CLKCON
<b>硬件看门狗定时器寄存器：</b>	RSTSTAT
<b>软件复位控制寄存器：</b>	SWRCON, DSPRST
<b>通讯寄存器：</b>	MCOMIEH, MCOMIEL, MTDCH, MTDC, MTDIFH, MTDIFL, MTDDxH (x = 0-5), MTDDxL (x = 0-5), DTMCH, DTMCL, DTMIFH, DTMIFL, DTMDxH (x = 0-5), DTMDxL (x = 0-5)
<b>中断系统寄存器：</b>	IEN0, IEN1, IPH0, IPL0, IPH1, IPL1, EXF0
<b>I/O端口寄存器：</b>	P0, P1, P2, P3, P4, P5, P6, P0CR, P1CR, P2CR, P3CR, P4CR, P5CR, P6CR, P0PCR, P1PCR, P2PCR, P3PCR, P4PCR, P5PCR, P6PCR
<b>定时器寄存器：</b>	TCON, TMOD, TL0, TH0, TL1, TH1, T2CON, T2MOD, TL2, TH2, RCAP2L, RCAP2H, TCON1
<b>EUART0寄存器：</b>	SCON, SBUF, SADEN, SADDR, PCON
<b>EUART1寄存器：</b>	SCON1, SBUF1, SADEN1, SADDR1, SBRT1, SBRT0
<b>SPI寄存器：</b>	SPCON, SPSTA, SPDAT
<b>ADC寄存器：</b>	ADCON, ADT, ADCH, ADDL, ADDH
<b>辅助PWM寄存器：</b>	APWMCON0, APWMCON1, APWMCON2, APWMPL0, APWMPL1, APWMPL2, APWMPH0, APWMPH1, APWMPH2, APWMDL0, APWMDL1, APWMDL2, APWMDH0, APWMDH1, APWMDH2



**Bank0**

	可位寻址		不可位寻址						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H	SPSTA			IB_OFFSET	IB_DATA			<b>(Reserved)</b>	FFH
F0H	B	AUXC	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7H
E8H	EXF0	P0PCR	P1PCR	P2PCR	P3PCR	P4PCR		XPAGE1	EFH
E0H	ACC	P0CR	P1CR	P2CR	P3CR	P4CR			E7H
D8H	SCON1	SBUF1	SADDR1	SADEN1	MCOMIEL	MCOMIEH	MTDCL	MTDCH	DFH
D0H	PSW		MTDIFL	MTDIFH	MTDD0L	MTDD0H	MTDD1L	MTDD1H	D7H
C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TCON1		CFH
C0H	P4		MTDD2L	MTDD2H	MTDD3L	MTDD3H	MTDD4L	MTDD4H	C7H
B8H	IPL0	IPL1	MTDD5L	MTDD5H	DTMIFL	DTMIFH	DTMD0L	DTMD0H	BFH
B0H	P3	RSTSTAT	CLKCON	DSRST	IPH0	IPH1	DTMD1L	DTMD1H	B7H
A8H	IEN0	IEN1	DTMD2L	DTMD2H	DTMD3L	DTMD3H	APWMDL0	APWMDH0	AFH
A0H	P2		SPCON	SPDAT	SBRT1		SWRCON	FLASHCON	A7H
98H	SCON	SBUF	SADDR	SADEN	SBRT0	APWMCON0	APWMPL0	APWMPH0	9FH
90H	P1	DTMCL	DTMCH	ADCON	ADT	ADCH	ADDL	ADDH	97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	SUSLO		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	

**Bank1**

	可位寻址		不可位寻址						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H								<b>(Reserved)</b>	FFH
F0H	B	C						XPAGE	F7H
E8H		P5PCR	P6PCR						EFH
E0H	ACC	P5CR	P6CR						E7H
D8H									DFH
D0H	PSW				DTMD4L	DTMD4H	DTMD5L	DTMD5H	D7H
C8H				APWMCON1	APWMPL1	APWMPH1	APWMDL1	APWMDH1	CFH
C0H				APWMCON2	APWMPL2	APWMPH2	APWMDL2	APWMDH2	C7H
B8H	IPL0	IPL1							BFH
B0H					IPH0	IPH1			B7H
A8H	IEN0	IEN1							AFH
A0H									A7H
98H									9FH
90H	P6								97H
88H							SUSLO		8FH
80H	P5	SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	





## 6.2 DSP寄存器映像

DSP寄存器包括非存储器映射的（non-memory mapped）核寄存器和存储器映射（memory mapped）的寄存器。

### 6.2.1 核寄存器（non-memory mapped）

索引寄存器:	I0, I1, I2, I3, I4, I5, I6, I7
长度寄存器:	L0, L1, L2, L3, L4, L5, L6, L7
修改寄存器:	M0, M1, M2, M3, M4, M5, M6, M7
中断控制寄存器:	ICNTL, IMASK
循环控制寄存器:	CNTR, OWRCNTR
状态与模式控制寄存器:	SSTAT, ASTAT, MSTAT
栈顶寄存器:	TOPPCSTACK
计算单元相关寄存器:	AX0/1, AY0/1, AR, AF, MX0/1, MY0/1, MR2/1/0, MF, SI, SE, SB, SR1/0

### 6.2.2 存储器映射寄存器（memory-mapped）

<b>DSP系统控制寄存器:</b>	DSYSR
<b>DSP堆栈状态寄存器:</b>	DCCSD, DCLSD, DCSSD, DCPSD
<b>DSP时钟与看门狗控制寄存器:</b>	DCLKCON
<b>与MCU通讯寄存器:</b>	DCOMIE, DTMC, DTMIF, DTMDx (x = 0-5), MTDC, MTDIF, MTDDx (x = 0-5)
<b>DSP外部中断寄存器:</b>	DXINTCON, DXINTIF, INTPR0, INTPR1
<b>PWM模块寄存器:</b>	PWMLO, PWMTCON, PWMTPRD, PWMTCNT, PWMWCON, PWMWDxA (x = 0-2), PWMWDxB (x = 0-2), PWMxDTR (x = 0-2), PWMxDTF (x = 0-2), PWMECON, PWMECMPx (x = 0-2), PWMFCON, PWMFSTAT, PWMFDEB, PWMFMAPx (x = 0-2), PWMOCON, PWMOE, PWMIE, PWMIF
<b>QEI模块寄存器:</b>	QEICON, QFLTCON, QPOSCNT, QPOSLAT, QTMCON, QTMCNT, QTMLAT, QCNTMAX, QEIINT
<b>ADC寄存器:</b>	ADCON1, ADCON2, ADPTR, ADPCH, ADINT, ADCHSELx (x = 0-3), ADDRx (x = 0-15)
<b>DSP定时器寄存器:</b>	TMRxPRD (x = 0-3), TMRxPBUF (x = 0-3), TMRxCNT (x = 0-3), TMRMOD, TMRCON, TMRPSC0, TMRPSC1, TMRINT
<b>输入捕捉寄存器:</b>	CAPxCON (x = 0-3), CAPxTOP (x = 0-3), CAPxBOM (x = 0-3), CAPINT
<b>输出比较寄存器:</b>	OCxCON (x = 0-2), OCxR (x = 0-2), OCxRS (x = 0-2), OCINT
<b>运算放大器:</b>	OPCON



# SH99F100

寄存器名	地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	复位值	
<b>DSP系统控制</b>																			
DSYSR	3FF7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BIASRND		0000 0000 0000 0000	
DCLKCON	3FFE	DSCMIE	DSCMST	-	-	-	-	DWDT			-	-	-	PLL_LOCK	PLL_EN	PLL_MUL		0000 0000 0000 0000	
DCCSD	3FFD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCCSD		0000 0000 0000 0000	
DCLSD	3FFC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCLSD		0000 0000 0000 0000	
DCSSD	3FFB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCSSD		0000 0000 0000 0000	
DCPSD	3FFA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DCPSD		0000 0000 0000 0000	
<b>DSP中断</b>																			
INTPR1	3FF7	-	-	-	-	-	-	-	-	-	-	-	-	-	IPR_PWM		IPR_TMR0	0000 0000 0000 0000	
INTPR0	3FF6	IPR_ADC		IPR_QEI		IPR_DXIOC		IPR_TMR1		IPR_MR2		IPR_TMR3		IPR_CAP		IPR_COMM		0000 0000 0000 0000	
DXINTCON	3FF5	-	-	DXINT2MOD		DXINT1MOD		DXINT0MOD		-	-	-	-	-	DXINT2IE	DXINT1IE	DXINT0IE	0000 0000 0000 0000	
DXINTIF	3FF4	-	-	-	-	-	-	-	-	-	-	-	-	-	DXINT2IF	DXINT1IF	DXINT0IF	0000 0000 0000 0000	
<b>PWM模块</b>																			
PWMTCON	3FEF	PWMEN	PWMEMU		PTDIR	PPOST					-	-	-	PTMOD		PTCLKPS		0000 0000 0000 0000	
PWMTPRD	3FEE	PWM时基周期寄存器																	0000 0000 0000 0000
PWMTCNT	3FED	PWM时基计数寄存器																	0000 0000 0000 0000
PWMWCON	3FEC	-	-	-	PWDL	PWSYM	PWMOD2	PWMOD1	PWMOD0	PDMOD		PWPLR2A	PWPLR2B	PWPLR1A	PWPLR1B	PWPLR0A	PWPLR0B	0000 0000 0010 1010	
PWMWD2A	3FEB	PWM2A占空比寄存器																	0000 0000 0000 0000
PWMWD2B	3FEA	PWM2B占空比寄存器																	0000 0000 0000 0000
PWMWD1A	3FE9	PWM1A占空比寄存器																	0000 0000 0000 0000
PWMWD1B	3FE8	PWM1B占空比寄存器																	0000 0000 0000 0000
PWMWD0A	3FE7	PWM0A占空比寄存器																	0000 0000 0000 0000
PWMWD0B	3FE6	PWM0B占空比寄存器																	0000 0000 0000 0000
PWM2DTR	3FE5	PWM2A/B上升沿死区延迟时间寄存器																	0000 0000 0000 0000
PWM2DTF	3FE4	PWM2A/B下降沿死区延迟时间寄存器																	0000 0000 0000 0000
PWM1DTR	3FE3	PWM1A/B上升沿死区延迟时间寄存器																	0000 0000 0000 0000
PWM1DTF	3FE2	PWM1A/B下降沿死区延迟时间寄存器																	0000 0000 0000 0000
PWM0DTR	3FE1	PWM0A/B上升沿死区延迟时间寄存器																	0000 0000 0000 0000
PWM0DTF	3FE0	PWM0A/B下降沿死区延迟时间寄存器																	0000 0000 0000 0000
PWMECON	3FDF	-	-	-	-	-	PEMOD			PECMP0DIR			PECMP2EN		PECMP1EN		PECMP0EN		0000 0000 0000 0000
PWMECMP2	3FDE	PWM事件触发比较寄存器2																	0000 0000 0000 0000
PWMECMP1	3FDD	PWM事件触发比较寄存器1																	0000 0000 0000 0000
PWMECMP0	3FDC	PWM事件触发比较寄存器0																	0000 0000 0000 0000
PWMFCON	3FDB	-	-	-	PFLT4RM	PFLT3RM	PFLT2RM	PFLT1RM	PFLT0RM	-	-	-	PFLT4EN	PFLT3EN	PFLT2EN	PFLT1EN	PFLT0EN	0000 0000 0000 0000	
PWMFSTAT	3FDA	-	-	-	PFLT4TG	PFLT3TG	PFLT2TG	PFLT1TG	PFLT0TG	-	-	-	PFLT4ST	PFLT3ST	PFLT2ST	PFLT1ST	PFLT0ST	0000 0000 000u uuuu	
PWMFDEB	3FD9	-	PFLT4DEB			PFLT3DEB			PFLT2DEB			PFLT1DEB			PFLT0DEB			0000 0000 0000 0000	
PWMFMAP2	3FD8	-	-	-	PF4MAP2A	PF3MAP2A	PF2MAP2A	PF1MAP2A	PF0MAP2A	-	-	-	PF4MAP2B	PF3MAP2B	PF2MAP2B	PF1MAP2B	PF0MAP2B	0000 0000 0000 0000	
PWMFMAP1	3FD7	-	-	-	PF4MAP1A	PF3MAP1A	PF2MAP1A	PF1MAP1A	PF0MAP1A	-	-	-	PF4MAP1B	PF3MAP1B	PF2MAP1B	PF1MAP1B	PF0MAP1B	0000 0000 0000 0000	
PWMFMAP0	3FD6	-	-	-	PF4MAP0A	PF3MAP0A	PF2MAP0A	PF1MAP0A	PF0MAP0A	-	-	-	PF4MAP0B	PF3MAP0B	PF2MAP0B	PF1MAP0B	PF0MAP0B	0000 0000 0000 0000	
PWMOCON	3FD5	POSYNC	-	POCON2A	POCON2B	POCON1A	POCON1B	POCON0A	POCON0B	-	-	POUT2A	POUT2B	POUT1A	POUT1B	POUT0A	POUT0B	0000 0000 0000 0000	
PWMOE	3FD4	-	-	-	-	-	-	-	-	-	-	PWMOE2A	PWMOE2B	PWMOE1A	PWMOE1B	PWMOE0A	PWMOE0B	0000 0000 0000 0000	
PWMIE	3FD3	-	-	-	-	-	-	-	-	PZRIE	PFLIE	-	PDP4IE	PDP3IE	PDP2IE	PDP1IE	PDP0IE	0000 0000 0000 0000	
PWMIF	3FD2	-	-	-	-	-	-	-	-	PZRIF	PFLIF	-	PDP4IF	PDP3IF	PDP2IF	PDP1IF	PDP0IF	0000 0000 0000 0000	
PWMLO	3FD1	PWM模块锁定寄存器																	0000 0000 0000 0000
<b>QEI模块</b>																			
QEICON	3FCF	QPEMU		-	-	-	QPDIR	QIDXS	-	-	QSWAP	QIDXEN	QEIEN	QPCAPEN	QPMOD			0000 0000 0000 0000	
QFLTCON	3FCE	-	-	-	-	-	-	-	QIDXFEN	QIDXCPS			QABFEN	QABCPS			0000 0000 0000 0000		
QPOSCNT	3FCD	QEI位置计数器																	0000 0000 0000 0000
QPOSLAT	3FCC	QEI位置计数器锁存寄存器																	0000 0000 0000 0000
QTMCON	3FCB	QTEMU		-	-	QTIE	QTCAPIE	QTIF	QTCAPIF	QTEN	-	QTCPS		QTEPS			0000 0000 0000 0000		



# SH99F100

QTMCNT	3FCA	QTimer计数器														0000 0000 0000 0000					
QTMLAT	3FC9	QTimer计数值锁存寄存器														0000 0000 0000 0000					
QCNTMAX	3FC8	最大计数值寄存器														1111 1111 1111 1111					
QEINT	3FC7	-	-	-	-	-	-	QEIE	QCEIE	QPEIE	-	-	-	-	-	QEIF	QCEIF	QPEIF	0000 0000 0000 0000		
<b>ADC模块</b>																					
ADCON1	3FBF	ADEMU				ADMOD				-	-	-	-	ADEN	ADREN	ADVRS	-	ADCPS			0000 0000 0000 0000
ADCON2	3FBE	-	ADMAXCH2				ADEVTEN2	ADCTU2	ADRST2	ADSOC2	ADMAXCH1				ADEVTEN1	ADCTU1	ADRST1	ADSOC1	0000 0000 0000 0000		
ADPTR	3FBD	-	ADCNT2				-	ADPRR2				ADCNT1				ADPRR1				0000 0000 0000 0000	
ADPCH	3FBC	-	-	-	-	-	ADPCH2				-	-	-	-	ADPCH1				0000 0000 0000 0000		
ADINT	3FBB	ADIE2	-	-	-	-	-	ADBSY2	ADIF2	ADIE1	-	-	-	-	-	-	ADBSY1	ADIF1	0000 0000 0000 0000		
ADCHSEL3	3FB3	ADCHS15				ADCHS14				ADCHS13				ADCHS12				0000 0000 0000 0000			
ADCHSEL2	3FB2	ADCHS11				ADCHS10				ADCHS9				ADCHS8				0000 0000 0000 0000			
ADCHSEL1	3FB1	ADCHS7				ADCHS6				ADCHS5				ADCHS4				0000 0000 0000 0000			
ADCHSEL0	3FB0	ADCHS3				ADCHS2				ADCHS1				ADCHS0				0000 0000 0000 0000			
ADRR15	3FAF	AD转换结果寄存器15														0000 0000 0000 0000					
ADRR14	3FAE	AD转换结果寄存器14														0000 0000 0000 0000					
ADRR13	3FAD	AD转换结果寄存器13														0000 0000 0000 0000					
ADRR12	3FAC	AD转换结果寄存器12														0000 0000 0000 0000					
ADRR11	3FAB	AD转换结果寄存器11														0000 0000 0000 0000					
ADRR10	3FAA	AD转换结果寄存器10														0000 0000 0000 0000					
ADRR9	3FA9	AD转换结果寄存器9														0000 0000 0000 0000					
ADRR8	3FA8	AD转换结果寄存器8														0000 0000 0000 0000					
ADRR7	3FA7	AD转换结果寄存器7														0000 0000 0000 0000					
ADRR6	3FA6	AD转换结果寄存器6														0000 0000 0000 0000					
ADRR5	3FA5	AD转换结果寄存器5														0000 0000 0000 0000					
ADRR4	3FA4	AD转换结果寄存器4														0000 0000 0000 0000					
ADRR3	3FA3	AD转换结果寄存器3														0000 0000 0000 0000					
ADRR2	3FA2	AD转换结果寄存器2														0000 0000 0000 0000					
ADRR1	3FA1	AD转换结果寄存器1														0000 0000 0000 0000					
ADRR0	3FA0	AD转换结果寄存器0														0000 0000 0000 0000					
<b>DSP定时器</b>																					
TMRMOD	3F9F	TMR3ADT	TMR2ADT	TMR1ADT	TMR0ADT	-	-	-	-	TMR3MOD	TMR2MOD	TMR1MOD	TMR0MOD	0000 0000 0000 0000							
TMRCON	3F9E	TMR3EMU	TMR2EMU	TMR1EMU	TMR0EMU	-	-	-	-	TMR3TR	TMR2TR	TMR1TR	TMR0TR	0000 0000 0000 0000							
TMRPSC1	3F9D	TMR3PSC							TMR2PSC							0000 0000 0000 0000					
TMRPSC0	3F9C	TMR1PSC							TMR0PSC							0000 0000 0000 0000					
TMR3CNT	3F9B	DTimer3计数寄存器														0000 0000 0000 0000					
TMR2CNT	3F9A	DTimer2计数寄存器														0000 0000 0000 0000					
TMR1CNT	3F99	DTimer1计数寄存器														0000 0000 0000 0000					
TMR0CNT	3F98	DTimer0计数寄存器														0000 0000 0000 0000					
TMR3PRD	3F97	DTimer3周期寄存器														0000 0000 0000 0000					
TMR2PRD	3F96	DTimer2周期寄存器														0000 0000 0000 0000					
TMR1PRD	3F95	DTimer1周期寄存器														0000 0000 0000 0000					
TMR0PRD	3F94	DTimer0周期寄存器														0000 0000 0000 0000					
TMR3PBUF	3F93	DTimer3周期缓存寄存器														0000 0000 0000 0000					
TMR2PBUF	3F92	DTimer2周期缓存寄存器														0000 0000 0000 0000					
TMR1PBUF	3F91	DTimer1周期缓存寄存器														0000 0000 0000 0000					
TMR0PBUF	3F90	DTimer0周期缓存寄存器														0000 0000 0000 0000					
TMRINT	3F8F	-	-	-	-	TMR3IE	TMR2IE	TMR1IE	TMR0IE	-	-	-	-	TMR3IF	TMR2IF	TMR1IF	TMR0IF	0000 0000 0000 0000			
<b>输入捕获(CAP)模块</b>																					
CAP3CON	3F7F	CAP3TS	CAP3TG	CAP3PSC				-	-	CAP3STAT	CAP3FEN	CAP3CPS				0000 0000 0000 0000					
CAP2CON	3F7E	CAP2TS	CAP2TG	CAP2PSC				-	-	CAP2STAT	CAP2FEN	CAP2CPS				0000 0000 0000 0000					
CAP1CON	3F7D	CAP1TS	CAP1TG	CAP1PSC				-	-	CAP1STAT	CAP1FEN	CAP1CPS				0000 0000 0000 0000					



# SH99F100

CAP0CON	3F7C	CAP0TS	CAP0TG			CAP0PSC					-	-	CAP0STAT		CAP0FEN	CAP0CPS			0000 0000 0000 0000	
CAPINT	3F7B	CAP3IM	CAP2IM	CAP1IM	CAP0IM	CAP3IE	CAP2IE	CAP1IE	CAP0IE	-	-	-	-	CAP3IF	CAP2IF	CAP1IF	CAP0IF	0000 0000 0000 0000		
CAP3TOP	3F77	CAP3存储栈顶寄存器																0000 0000 0000 0000		
CAP3BOM	3F76	CAP3存储栈底寄存器																0000 0000 0000 0000		
CAP2TOP	3F75	CAP2存储栈顶寄存器																0000 0000 0000 0000		
CAP2BOM	3F74	CAP2存储栈底寄存器																0000 0000 0000 0000		
CAP1TOP	3F73	CAP1存储栈顶寄存器																0000 0000 0000 0000		
CAP1BOM	3F72	CAP1存储栈底寄存器																0000 0000 0000 0000		
CAP0TOP	3F71	CAP0存储栈顶寄存器																0000 0000 0000 0000		
CAP0BOM	3F70	CAP0存储栈底寄存器																0000 0000 0000 0000		
<b>输出比较(OC)模块</b>																				
OC2CON	3F6F	-	-	-	-	-	-	-	-	-	-	-	-	OC2TS	OC2MOD			0000 0000 0000 0000		
OC1CON	3F6E	-	-	-	-	-	-	-	-	-	-	-	-	OC1TS	OC1MOD			0000 0000 0000 0000		
OC0CON	3F6D	-	-	-	-	-	-	-	-	-	-	-	-	OC0TS	OC0MOD			0000 0000 0000 0000		
OCINT	3F6C	-	-	-	-	OCFIE	OC2IE	OC1IE	OC0IE	OCFCPS			OCFST	OCFIF	OC2IF	OC1IF	OC0IF	0000 0000 0000 0000		
OC2R	3F6B	通道2比较寄存器																0000 0000 0000 0000		
OC2RS	3F6A	通道2比较辅助寄存器 <sup>(2)</sup>																0000 0000 0000 0000		
OC1R	3F69	通道1比较寄存器																0000 0000 0000 0000		
OC1RS	3F68	通道1比较辅助寄存器 <sup>(2)</sup>																0000 0000 0000 0000		
OC0R	3F67	通道0比较寄存器																0000 0000 0000 0000		
OC0RS	3F66	通道0比较辅助寄存器 <sup>(2)</sup>																0000 0000 0000 0000		
<b>运算放大器(OP)模块</b>																				
OPCON	3F64																	OP2EN	OP1EN	0000 0000 0000 0000
<b>通信控制(COMM)模块</b>																				
DCOMIE	3F60	MCMDIE	MFG14IE	MFG13IE	MFG12IE	MFG11IE	MFG10IE	MFG9IE	MFG8IE	MFG7IE	MFG6IE	MFG5IE	MFG4IE	MFG3IE	MFG2IE	MFG1IE	MFG0IE	0000 0000 0000 0000		
DTMC	3F5F	DSP至MCU的命令寄存器																0000 0000 0000 0000		
DTMIF	3F5E	DCMDIF	DFG14IF	DFG13IF	DFG12IF	DFG11IF	DFG10IF	DFG9IF	DFG8IF	DFG7IF	DFG6IF	DFG5IF	DFG4IF	DFG3IF	DFG2IF	DFG1IF	DFG0IF	0000 0000 0000 0000		
DTMD5	3F5D	DSP至MCU的数据寄存器5																0000 0000 0000 0000		
DTMD4	3F5C	DSP至MCU的数据寄存器4																0000 0000 0000 0000		
DTMD3	3F5B	DSP至MCU的数据寄存器3																0000 0000 0000 0000		
DTMD2	3F5A	DSP至MCU的数据寄存器2																0000 0000 0000 0000		
DTMD1	3F59	DSP至MCU的数据寄存器1																0000 0000 0000 0000		
DTMD0	3F58	DSP至MCU的数据寄存器0																0000 0000 0000 0000		
MTDC	3F57	MCU至DSP的命令寄存器																0000 0000 0000 0000		
MTDIF	3F56	MCMDIF	MFG14IF	MFG13IF	MFG12IF	MFG11IF	MFG10IF	MFG9IF	MFG8IF	MFG7IF	MFG6IF	MFG5IF	MFG4IF	MFG3IF	MFG2IF	MFG1IF	MFG0IF	0000 0000 0000 0000		
MTDD5	3F55	MCU至DSP的数据寄存器5																0000 0000 0000 0000		
MTDD4	3F54	MCU至DSP的数据寄存器4																0000 0000 0000 0000		
MTDD3	3F53	MCU至DSP的数据寄存器3																0000 0000 0000 0000		
MTDD2	3F52	MCU至DSP的数据寄存器2																0000 0000 0000 0000		
MTDD1	3F51	MCU至DSP的数据寄存器1																0000 0000 0000 0000		
MTDD0	3F50	MCU至DSP的数据寄存器0																0000 0000 0000 0000		

**注意:**

- (1) 灰底的位置表示只读特性;
- (2) 当OCxMOD = 111时, OCxRS寄存器也为只读特性。



## 7. 系统集成模块SIM (System Integrated Module)

### 7.1 存储器架构

SH99F100的片上存储器包括128KBytes Flash、1KBytes BootRom (for ISP)、(12K+4K+48K+1K) Bytes 高速SRAM (12Kbytes在DDM空间中, 4Kbytes在DPM空间中, 48Kbytes在DCM空间中, 1KBytes在DDM空间中) 以及4KBytes低速SRAM。这些存储器和寄存器分为以下几个部分:

**MIDM** - MCU Internal Data Memory

**MCM** - MCU Code Memory

**DDM** - DSP Data Memory

**DRA** - DSP Registers Area

**SFR** - Special Function Registers for MCU

**DCSM** - DSP Code Storage Memory

**MEDM** - MCU External Data Memory

**DCRM** - DSP Code Running Memory

**DPM** - DSP Program Data Memory

**CRA** - Communication Registers and Data Area

**BOOTROM** - ISP Code Running Memory

- MCM:** 64K bytes的Flash, 位于MCU代码空间。存放MCU的代码, 并在其中执行代码。
- MIDM:** 256 bytes的低速SRAM, 位于MCU Internal数据存储器空间。作为MCU的数据存储器。其中低128字节可以直接与间接寻址, 高128字节只能间接寻址。
- MEDM:** 3840 bytes的低速SRAM, 位于MCU External数据存储器空间。作为MCU的数据存储器, 只能间接寻址。
- SFR:** 位于MCU的特殊功能寄存器空间。寻址空间大小是128 bytes。作为MCU的寄存器, 其地址与MIDM的高128字节重合, 只能通过直接寻址访问。
- DCRM:** 16K X 24bits的高速SRAM, 位于DSP代码存储器空间。DSP在其中执行代码。
- DDM:** 6K X 16bits的高速SRAM, 位于DSP的Data Memory空间中 (DSP的Data Memory空间寻址大小是16K X 16bits)。是DSP的一个数据存储器。
- DPM:** 2K X 16bits的高速SRAM, 位于DSP的Program Memory空间中 (DSP的Program Memory空间寻址大小是16K X 16bits)。是DSP的另一个数据存储器。DSP可以同时DDM与DPM区中取操作数, 以实现Multifunction指令。
- DRA:** 由114 X 16bits个寄存器组成, 位于DSP的Data Memory空间中。作为DSP的memory-mapped寄存器。
- CRA:** 由512 X 16bits的高速SRAM和18 X 16bits的寄存器组成, 位于DSP的Data Memory空间中。用于DSP与MCU之间的通信。对于DSP, CRA区中的SRAM位于其Data Memory空间的0x1800 - 0x19FF地址区域中, CRA区中的寄存器位于Data Memory空间的0x3F50 - 0x3F60地址区域中; 对于MCU, CRA区中的SRAM位于其外部存储空间中的0xF000 - 0xF3FF地址区域中, CRA区中的寄存器位于SFR (Bank0 & Bank1) 区中。
- DCSM:** 64 Kbytes的Flash, 作为非易失性存储器, 用于存放DSP代码以及变量初始化值。
- BOOTROM:** 1Kbytes的Flash, 储存了ISP代码。

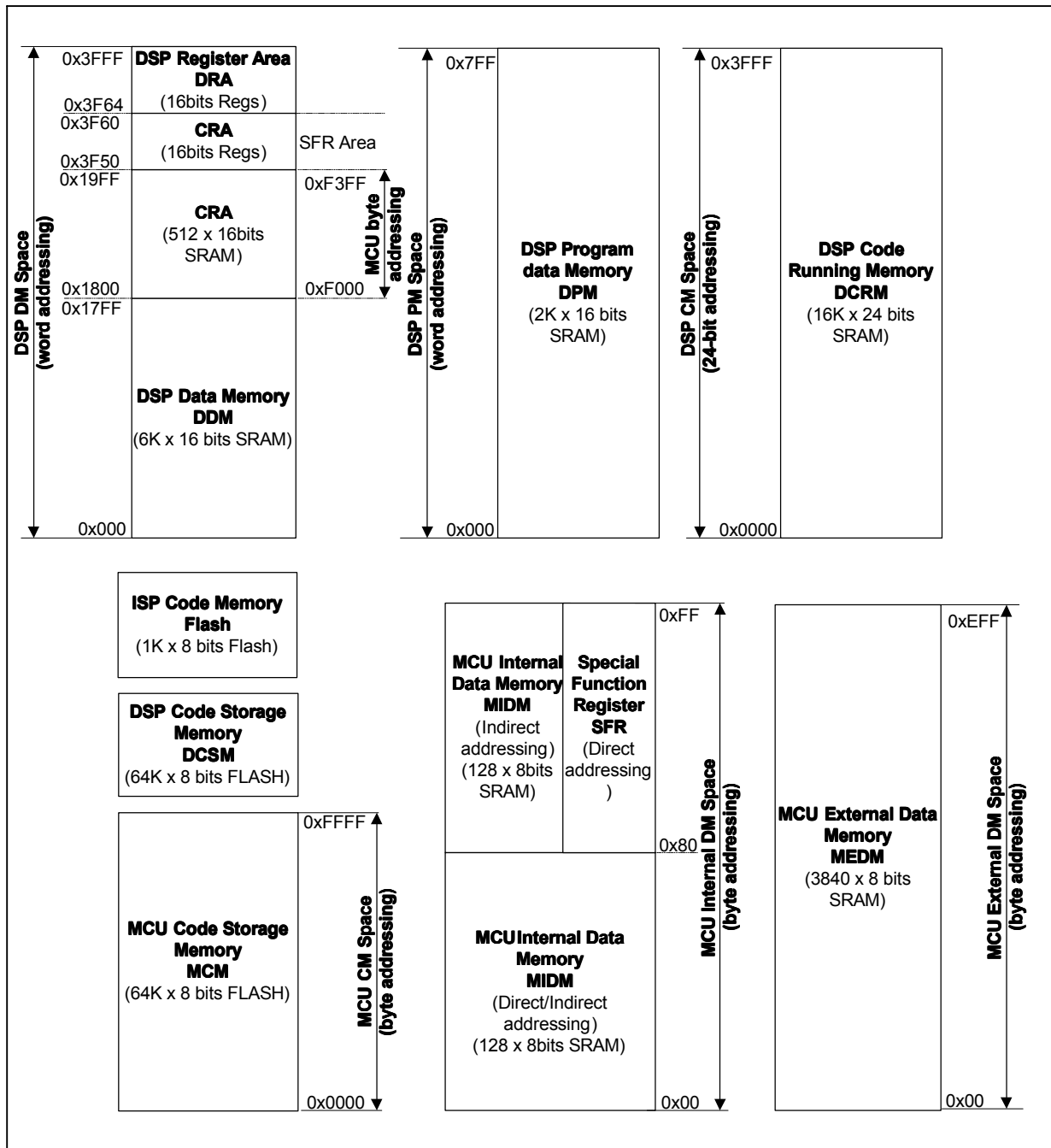


图7-1



## 7.2 复位

一共有五种复位源：上电复位POR、按键复位PIN、看门狗复位WDT、低电压复位LVR和软件复位。前四种复位都将触发将数据从DCSM区加载至DCRM区、DDM区和DPM区的Autoload过程。

**Table 7.1** 复位控制寄存器

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	DWDOF	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	读/写
复位值 (POR)	0	0	1	0	0	0	0	0
复位值 (WDT)	1	u	u	u	u	0	0	0
复位值 (DWDT)	u	1	u	u	u	0	0	0
复位值 (LVR)	u	u	u	1	u	0	0	0
复位值 (PIN)	u	u	u	u	1	0	0	0

位编号	位符号	说明
7	WDOF	<b>看门狗溢出或程式超范围溢出标志位</b> 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溢出或程序超范围溢出 1: 发生WDT溢出或程序超范围溢出
6	DWDOF	<b>DSP看门狗定时器溢出标志</b> 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DSP的WDT溢出 1: 发生DSP的WDT溢出
5	PORF	<b>上电复位标志位</b> 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	<b>低压复位标志位</b> 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	<b>Reset引脚复位标志位</b> 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]	<b>WDT溢出周期控制位</b> 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 <b>注意:</b> 应用中如果看门狗打开，程序清看门狗的最大间隔时间不能大于以上所列最小值。



### 7.2.1 上电复位POR (Power On Reset)

系统工作电压是2.7 - 5.5V。上电复位发生之后，经过的电源预热时间和振荡器预热时间，判断是否执行ISP（详见7.8节）。RSTSTAT寄存器中的PORF位将置1。在振荡器预热的同时进行Autoload过程。

### 7.2.2 按键复位PIN (Pin Reset)

Reset引脚不与其它功能复用。低电压有效。Pin Reset发生后，经过电源预热时间和振荡器预热时间，判断是否执行ISP（详见7.8节）。RSTSTAT寄存器中的CLRf位将置1。在振荡器预热的同时进行Autoload过程。

### 7.2.3 看门狗复位WDT (WatchDogTimer Reset)

DSP和MCU都有各自的看门狗，可以分别设置看门狗溢出时间。任何一个看门狗溢出后都将导致系统复位。WDT Reset发生后，经过1000系统时钟的延时和振荡器预热时间后，系统开始运行。RSTSTAT寄存器中WDOF位（MCU中的看门狗溢出）或DWDOF位（DSP中的看门狗溢出）将置1。在振荡器预热的同时进行Autoload过程。注意，寄存器DSPRST = 0xA5时，DSP的看门狗不工作，详见MCU使DSP复位章节。

### 7.2.4 低电压复位LVR (Low Voltage Reset)

通过代码选项选择，LVR设定电压 $V_{LVR}$ 可为2.8V或4.3V（SH99F100的工作电压范围是2.7 - 5.5V）。LVR去抖动时间为30-60 $\mu$ s。当供电电压低于设定电压 $V_{LVR}$ 时，将产生内部复位。LVR发生之后，经过电源预热时间和振荡器预热时间，系统开始运行。RSTSTAT寄存器中的LVRf位将置1。在振荡器预热的同时进行Autoload过程。

### 7.2.5 软复位SWR (Software Reset)

系统软件复位

Table 7.2 软件复位控制寄存器SWRCON (SoftWare Reset Control)

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
SWRCON	SWRCON.7	SWRCON.6	SWRCON.5	SWRCON.4	SWRCON.3	SWRCON.2	SWRCON.1	SWRCON.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	SWRCON[7:0]	软件复位控制 5AH: 设置后将发生软件复位 其它: 软件复位不会发生 软件复位只能在BootRom中使用

Table 7.3 软件复位标志和指令访问控制寄存器

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	SWRF	-	-	-	-	-	-	FAC
读/写	读/写	-	-	-	-	-	-	读/写
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	-	-	-	0

位编号	位符号	说明
7	SWRF	软件复位标志 只能在发生软件复位后由硬件自动置1； 由软件清除或者其它复位清0。
0	FAC	执行区域选择位（详见SSP章节）

BootRom是一块大小为1Kbytes的Flash，其中固化了ISP程序。其程序地址空间为0x0000 - 0x03FF段地址。软件复位只能通过MCU在BootRom中执行程序，写寄存器SWRCON为5AH时发生。

用户只需要令SWRCON = 5AH，即可发生软件复位。软件复位发生后，MCU和DSP的所有寄存器都将复位，MCU和DSP都停止运行，并开始执行Autoload过程。Autoload过程结束后，MCU从MCM的0x0000地址开始运行，DSP的PC指针指向DCRM空间的0x0000地址并仍处于停止状态，且FLASHCON寄存器中的SWRF位将置1。

注意：软件复位不清除RSTSTAT寄存器。



**MCU使DSP复位****Table 7.4** 复位DSP控制寄存器DSPRST

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
<b>DSPRST</b>	DSPRST.7	DSPRST.6	DSPRST.5	DSPRST.4	DSPRST.3	DSPRST.2	DSPRST.1	DSPRST.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)	1	0	1	0	0	1	0	1

位编号	位符号	说明
<b>7-0</b>	<b>DSPRST [7:0]</b>	<b>MCU复位DSP控制寄存器</b> A5H: DSP所有寄存器复位, DSP停止运行 5AH: DSP开始运行 其它值: 无效

MCU的SFR区中有一个寄存器DSPRST。向此寄存器写0xA5，将使得DSP的所有寄存器复位，DSP系统时钟被停止；向DSPRST写0x5A将使得DSP脱离上述复位状态，并开始运行。向DSPRST写其它值无意义。上电后该寄存器的值为0xA5，即DSP处于复位状态。此外，DSPRST = 0xA5时，DSP的看门狗计数器将复位，并停止运行；DSPRST = 0x5A时，DSP的看门狗是否工作取决于代码选项OP\_DWDT位（详见10章 代码选项）。



## 7.2.6 上电Autoload过程

SH99F100中有一块硬件逻辑，可以将DCSM区中指定区域中的内容自动加载至DCRM、DDM以及DPM区域中。对应关系如图7-2所示。

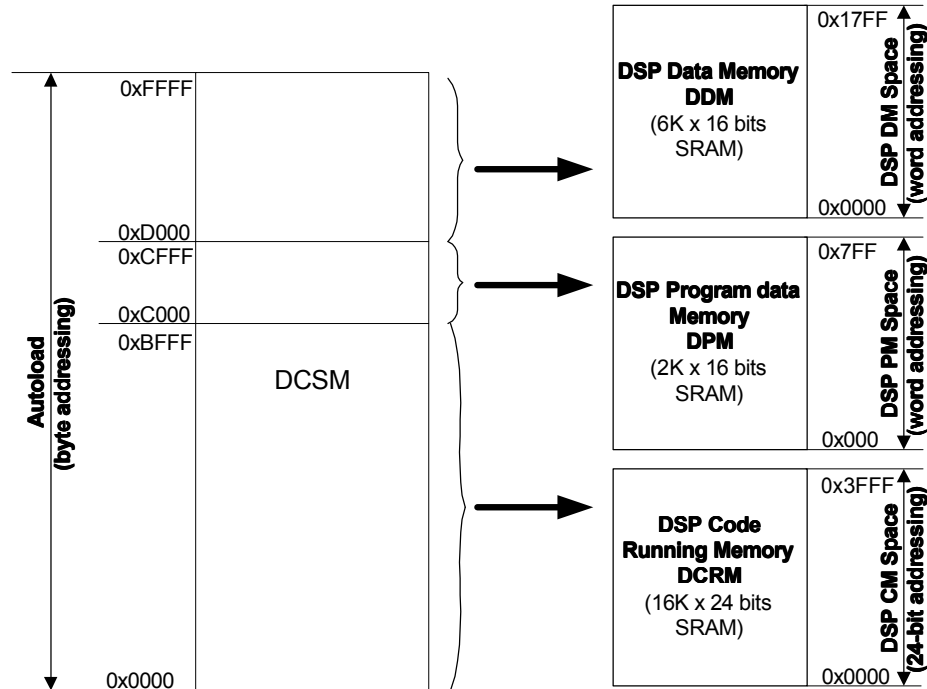


图7-2

## 7.2.7 复位过程综述

SH99F100发生上电复位、按键复位、看门狗复位和低电压复位之后：

- DSP和MCU均复位其寄存器（RSTSTAT寄存器的状态位会根据不同的复位给出指示），并停止运行（MCU和DSP不执行程序）。
- 若是上电复位、按键复位或低电压复位则将经历一段上电预热时间（详见7.5节），长短约11ms；若是看门狗复位，则将经历一段8000CKs（1CKs指1个16MHz RC振荡器时钟周期）的时间。
- 上电预热或8000CKs时间之后，再经历振荡器预热时间（详见7.5节），在这段时间内同时执行Autoload过程：MCU与DSP停止运行，DCSM中的数据按照图7-2中的对应关系加载至DCRM、DDM和DPM区中。
- 所有预热时间结束后，如果是上电复位或按键复位，则MCU通过代码选项中OP\_ISP的值判断是否运行引导扇区中的ISP程序，若OP\_ISP = 0则要执行ISP程序，接下来通过代码选项中OP\_ISPPIN的值判断是否需要判断P2.2和P2.3的电平以确定是否执行ISP程序；若OP\_ISPPIN = 1，则不判断pin的电平而直接执行ISP程序，即只要发生了上电复位/按键复位即执行ISP程序；反之，需要判断P2.2和P2.3的电平是否保持了100us以上的低电平时间，若是则执行ISP程序，否则不执行ISP程序，即发生上电复位/按键复位之后需要用外部的P2.2和P2.3来确定是否执行ISP程序。ISP程序存储在一块BootRom中，由MCU执行。
- ISP程序执行完毕后，将触发Autoload过程（Autoload过程实际上是由BootRom中的程序触发，程序中通过设置寄存器SWRCON = 0x5A实现（详见7.7节）），Autoload过程结束后，系统即开始运行：MCU从MCM空间0x0000地址开始执行程序；DSP的PC指针指向DCRM空间的0x0000地址并仍处于停止状态。若没有发生过ISP，则预热结束后系统即开始运行。



### 7.3 系统时钟

系统内部有三个时钟源：2KHz的RC，2MHz的RC，16MHz的RC以及晶振/外部时钟。2KHz的RC供MCU看门狗使用，16MHz的RC在全温度、全电压范围内是±2%的精度，在外部晶振失效后供系统使用（同时作为DSP看门狗时钟使用）。2MHz的RC用于计时电源预热时间。

系统正常工作时，使用晶振（Crystal）结合芯片内部振荡电路来产生时钟信号 $f_c$ 或直接使用与TTL电平相匹配的时钟源作为时钟信号 $f_c$ 。整体结构如图7-2所示。

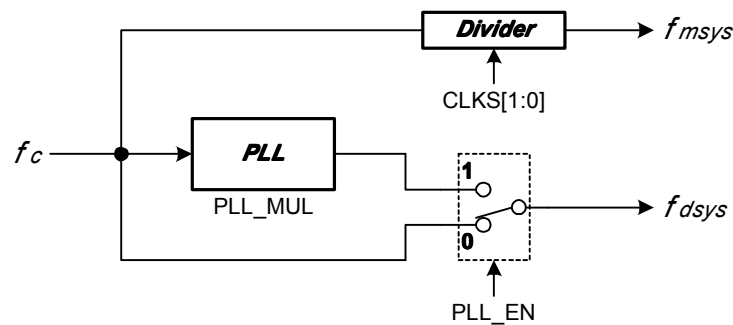


图7-3

#### 7.3.1 DSP系统时钟

DSP系统时钟记作 $f_{DSYS}$ ，DSP的最大系统时钟为100MHz。使能内部PLL时（ $PLL\_EN = 1$ ），用户可以通过 $PLL\_MUL$ 设置位选择不同的倍频系数，在 $f_c$ 基础上产生需要的系统时钟，DSP系统时钟由下式决定：

$$f_{DSYS} = f_c \times PLL\_MUL$$

第一次使能PLL或改变PLL的倍频设置值 $PLL\_MUL$ 后，用户程序需要等待PLL输出稳定。用户程序可以查询PLL输出稳定标志 $PLL\_LOCK$ 来确认PLL的稳定状态。DSP的时钟只由DSP系统时钟控制寄存器 $DCLKCON$ 控制，与MCU无关。



**Table 7.5** DSP系统时钟控制寄存器 DCLKCON (DSP Clock Control Register)

15	14	13 - 11	10 - 8
R/W-0	R-0	R-0	R/W-0
<b>DSCMIE</b>	<b>DSCMST</b>	<b>Reserved</b>	<b>DWDT</b>
7 - 5		4	3
R-0		R-0	R/W-0
<b>Reserved</b>		<b>PLL_LOCK</b>	<b>PLL_EN</b>
			<b>PLL_MUL</b>

位编号	位符号	说明
<b>15</b>	<b>DSCMIE</b>	<b>DSP中SCM中断使能位</b> 0: 禁止发生SCM时触发SCM中断 1: 允许发生SCM时触发SCM中断
<b>14</b>	<b>DSCMST</b>	<b>DSP中SCM中断标志位</b> 0: 没有发生时钟故障 1: 已发生时钟故障 该标志位只能由硬件置位和清除
<b>13 - 11</b>	<b>Reserved</b>	保留位
<b>10 - 8</b>	<b>DWDT</b>	<b>DWDT溢出频率控制</b> 000: 约16384μs 001: 约8192μs 010: 约4096μs 011: 约1024μs 100: 约256μs 101: 约64μs 110: 约16μs 111: 约4μs
<b>7 - 5</b>	<b>Reserved</b>	保留位
<b>4</b>	<b>PLL_LOCK</b>	<b>DSP锁相环输出稳定指示位</b> 0: PLL输出未稳定, 此时 $f_{DSYS} = f_c$ 1: PLL输出已稳定, 此时 $f_{DSYS} = f_c \times PLL\_MUL$
<b>3</b>	<b>PLL_EN</b>	<b>DSP锁相环 (PLL) 使能位</b> 0: 禁止PLL, DSP直接使用时钟信号 $f_c$ 作为 $f_{DSYS}$ 1: 使能PLL, $f_{DSYS} = f_c \times PLL\_MUL$
<b>2 - 0</b>	<b>PLL_MUL</b>	<b>PLL倍频系数选择位</b> 000: X 2 001: X 3 010: X 4 011: X 5 100: X 6.25 (for 16MHz) 101: X 8.33 (for 12MHz) 110: X 10 111: X 12.5 (for 8MHz)



## 7.3.2 MCU系统时钟

Table 7.6 系统时钟控制寄存器CLKCON

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	-	-	-	-
读/写	-	读/写	读/写	只读	-	-	-	-
复位值 (POR/WDT/LVR/PIN)	-	0	0	0	-	-	-	-

位编号	位符号	说明
6-5	CLKS[1:0]	系统时钟频率分频器 00: $f_{MSYS} = f_C$ 01: $f_{MSYS} = f_C/2$ 10: $f_{MSYS} = f_C/4$ 11: $f_{MSYS} = f_C/12$
4	SCMIF	系统时钟故障MCU标志 0: 系统时钟无故障 1: 发生系统时钟故障, 由硬件清0

MCU的系统时钟记作 $f_{MSYS}$ 。MCU的最大系统时钟为16MHz。在寄存器CLKCON中可以通过设置CLKS[1:0]将 $f_{MSYS}$ 设置为 $f_C$ 的1/1, 1/2, 1/4, 1/12倍。

## 7.3.3 系统时钟监控模块SCM (System Clock Monitor)

为了增强系统的可靠性, SH99F100中含有一个系统时钟监控(SCM)模块。如果系统时钟 $f_C$ 出现故障(例如:外部振荡器停振等),内建SCM模块会将 $f_C$ 自动切换到内部RC时钟(16MHz)。在代码选项中可以选择在振荡器预热期间是否使能SCM模块。

**检测到时钟故障后,  $f_C$ 自动切换至内部16MHz的RC时钟。**

MCU模块中,使用16M RC产生的 $f_C$ 在原来的分频系数上产生 $f_{MSYS}$ 运行,同时系统时钟监控中断标志位(SCMIF)被置1。SCMIF位反映了振荡器的工作状态,由硬件控制,SCMIF = 1说明处于时钟故障中、SCMIF = 0说明无时钟故障。当EA和ESCM位均被置1时,SCM模块将会产生中断并跳向处于0073H位置的中断向量。如果外部振荡器恢复工作,SCM将会切换 $f_C$ 到外部振荡器,然后SCMIF位自动清0。

DSP模块中,若寄存器DCLKCON中的位DSCMIE = 1时,则发生时钟故障时将触发SCM中断,进入0x002C处的中断向量,相应的中断标志将被硬件自动清0(注意该标志没有开放出来,用户无法软件操作)。DSCMST位指示时钟是否有故障,1表示有故障、0表示无故障,该位由硬件控制。发生时钟故障后,DSP的系统时钟 $f_{OSYS}$ 就是16M RC产生的 $f_C$ 。DSCMST等于1时,PLLEN无法置1,即外部晶振或信号源失效时,不能使用PLL。DSCMST位只能硬件清除,如果外部振荡器恢复工作,SCM将会切换 $f_C$ 到外部振荡器,然后DSCMST位自动清0,此时如果要使用PLL必须要手动设置。



7.4 看门狗

Table 7.7 SH99F100内部含有两个看门狗定时器，分别给DSP和MCU使用。这两个定时器结构上一样。

<b>15</b>	<b>14</b>	<b>13 - 11</b>	<b>10 - 8</b>
<i>R/W-0</i>	<i>R-0</i>	<i>R-0</i>	<i>R/W-0</i>
<b>DSCMIE</b>	<b>DSCMST</b>	<b>Reserved</b>	<b>DWDT</b>
<b>7 - 5</b>		<b>4</b>	<b>3</b>
<i>R-0</i>		<i>R-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>PLL_LOCK</b>	<b>PLL_EN</b>
			<b>PLL_MUL</b>

位编号	位符号	说明
<b>15</b>	<b>DSCMIE</b>	<b>DSP中SCM中断使能位</b> 0: 禁止发生SCM时触发SCM中断 1: 允许发生SCM时触发SCM中断
<b>14</b>	<b>DSCMST</b>	<b>DSP中SCM中断标志位</b> 0: 没有发生时钟故障 1: 已发生时钟故障 该标志位只能由硬件置位和清除
<b>13 - 11</b>	<b>Reserved</b>	保留位
<b>10 - 8</b>	<b>DWDT</b>	<b>DWDT溢出频率控制</b> 000: 约16384μs 001: 约8192μs 010: 约4096μs 011: 约1024μs 100: 约256μs 101: 约64μs 110: 约16μs 111: 约4μs
<b>7 - 5</b>	<b>Reserved</b>	保留位
<b>4</b>	<b>PLL_LOCK</b>	<b>DSP锁相环输出稳定指示位</b> 0: PLL输出未稳定, 此时 $f_{DSYS} = f_c$ 1: PLL输出已稳定, 此时 $f_{DSYS} = f_c \times PLL\_MUL$
<b>3</b>	<b>PLL_EN</b>	<b>DSP锁相环 (PLL) 使能位</b> 0: 禁止PLL, DSP直接使用时钟信号 $f_c$ 作为 $f_{DSYS}$ 1: 使能PLL, $f_{DSYS} = f_c \times PLL\_MUL$
<b>2 - 0</b>	<b>PLL_MUL</b>	<b>PLL倍频系数选择位</b> 000: X 2 001: X 3 010: X 4 011: X 5 100: X 6.25 (for 16MHz) 101: X 8.33 (for 12MHz) 110: X 10 111: X 12.5 (for 8MHz)

看门狗定时器DWDT (DSP Watch Dog Timer) /WDT (MCU Watch Dog Timer) 是一个递减计数器，由内部独立的RC振荡器作为时钟源。当DWDT/WDT溢出时，DSP/MCU将发生看门狗复位。通过代码选项可以允许或禁止该功能。看门狗复位过程见7.2.3节。

DSP和MCU的看门狗溢出标志都在MCU的寄存器RSTSTAT中，即无论DSP还是MCU的看门狗溢出，都将被MCU得知。DSP中，寄存器DCLKCON中的位DWDT用来选择不同的溢出时间。读或者写DCLKCON寄存器，WDT会重新开始计数。DWDT溢出后，MCU寄存器RSTSTAT中的DWDT溢出标志 (DWDOF) 将由硬件自动设置为1。

RSTSTAT中的位WDT用来选择不同的溢出时间。通过读写RSTSTAT寄存器，看门狗定时器在溢出前重新开始计数。WDT溢出后，RSTSTAT中的WDT溢出标志 (WDOF) 将由硬件自动置1。

**7.5 预热计数器**

SH99F100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。

SH99F100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，看门狗复位和低电压（LVR）复位。

上电后，SH99F100会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，溢出后开始运行程式。

**7.5.1 电源上电预热计数时间**

上电复位POR/ 引脚复位 CLR/低电压复位 LVR		看门狗复位WDR (不包含掉电模式)	
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	1000CKs	有

*注意：1CKs指1个2MHz RC振荡器时钟周期*

**7.5.2 振荡器上电预热计数时间**

振荡器上电预热时间约为 $2^{17} \times T_c$ ， $T_c$ 是外部振荡器的周期时间。



### 7.6 DSP核与MCU核之间的通讯

DSP核与MCU核之间的双核通讯，由通讯控制/数据寄存器和通讯数据单元通过中断的方式实现。通讯控制/数据寄存器与数据单元总共占据了530个字的存储空间。

#### 7.6.1 通讯存储器结构

CRA是DSP核和MCU核进行通讯的存储器区，由512个字的SRAM和18个字的寄存器组成，其对DSP和MCU的地址分配如下图所示：

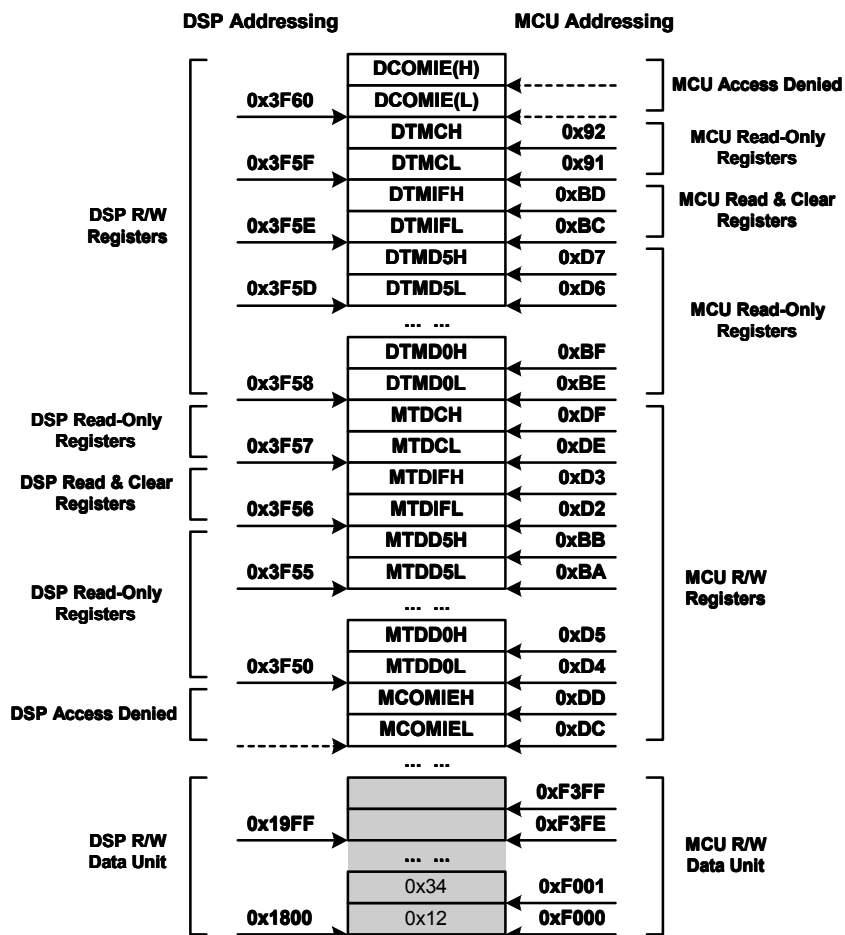


图7-6-1 CRA存储器分配和寻址示意图

图中灰底部分存储区是512个字的通讯数据单元。对于DSP，通讯数据单元映射到DM空间，寻址范围是0x1800 - 0x19FF；对于MCU，通讯数据单元映射到External数据存储器空间，寻址范围是0xF000 - 0xF3FF。DSP寻址的16位数据，高8位对应MCU的偶地址，低8位对应MCU的奇地址，例如DSP地址0x1800的存储数据为0x1234，则MCU地址0xF000存储数据为0x12，MCU地址0xF001存储数据为0x34。

DSP和MCU都可以读写所有的通讯数据单元，但同一时刻只能由一方操作。通讯数据单元的操作权由INSCON寄存器中DUINT位决定。DUINT = 0时，由MCU操作；DUINT = 1时，由DSP操作。上电复位之后，DUINT为0，即通讯数据单元由MCU控制。

图中白底部分为18个字的通讯控制/数据寄存器。对于DSP，通讯控制/数据寄存器映射到DM空间，其中，0x3F58 - 0x3F60是DSP可读可写的寄存器，0x3F50 - 0x3F57是DSP只读的寄存器。对于MCU，通讯控制/数据寄存器映射到SFR空间，其中地址为C2H - C7H，D2H - D7H，BAH - BBH，DCH - DFH共18个字节是MCU可读可写的寄存器，地址为91H - 92H，AAH - ADH，B6H - B7H，BCH - BFH，D4H - D7H共16个字节是MCU只读的寄存器。





### 7.6.2 通讯控制/数据寄存器

与双核通讯相关的所有控制/数据寄存器，如下表所示：

类别	缩写符号		功能说明
	DSP使用的缩写符号	MCU使用的缩写符号	
通讯控制寄存器	DCOMIE	-	DSP通讯中断使能寄存器
	-	MCOMIEH MCOMIEL	MCU通讯中断使能寄存器
	DTMC	DTMCH DTMCL	DSP至MCU的命令寄存器
	DTMIF	DTMIFH DTMIFL	DSP至MCU的标志寄存器
	MTDC	MTDCH MTDCL	MCU至DSP的命令寄存器
	MTDIF	MTDIFH MTDIFL	MCU至DSP的标志寄存器
通讯数据寄存器	DTMDx (x = 0-5)	DTMDxH (x = 0-5) DTMDxL (x = 0-5)	DSP至MCU的数据寄存器
	MTDDx (x = 0-5)	MTDDxH (x = 0-5) MTDDxL (x = 0-5)	MCU至DSP的数据寄存器

### 7.6.3 DSP使用的通讯寄存器

**Table 7.8** DSP通讯中断使能寄存器DCOMIE (DSP Communication Interrupt Enable Register)

15	14	13	12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>MCMDIE</b>	<b>MFG14IE</b>	<b>MFG13IE</b>	<b>MFG12IE</b>	<b>MFG11IE</b>	<b>MFG10IE</b>	<b>MFG9IE</b>	<b>MFG8IE</b>
7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
<b>MFG7IE</b>	<b>MFG6IE</b>	<b>MFG5IE</b>	<b>MFG4IE</b>	<b>MFG3IE</b>	<b>MFG2IE</b>	<b>MFG1IE</b>	<b>MFG0IE</b>

位编号	位符号	说明
15	<b>MCMDIE</b>	<b>DSP响应MCU通讯命令的中断使能位</b> 0: 禁止DSP响应MCU的通讯命令中断 1: 允许DSP响应MCU的通讯命令中断
14-0	<b>MFGxIE (x = 14-0)</b>	<b>DSP响应MCU通讯标志的中断使能位</b> 0: 禁止DSP响应MCU的通讯标志x (x = 14-0) 中断 1: 允许DSP响应MCU的通讯标志x (x = 14-0) 中断



**Table 7.9** DSP至MCU的命令寄存器DTMC (DSP to MCU Command Register)

<b>15-0</b>
<i>R/W-0</i>
<b>DTMC</b>

位编号	位符号	说明
<b>15-0</b>	<b>DTMC</b>	<b>DSP至MCU的命令寄存器</b> DSP向此寄存器写数据将使得DTMIF中的DCMDIF标志置1，若MCU允许响应DSP的通讯命令中断使能位DCMDIE = 1，则将触发MCU的通讯中断。MCU程序可以通过读取DTMCH/L寄存器得到DSP写入的命令值。

**Table 7.10** DSP至MCU的标志寄存器DTMIF (DSP to MCU Interrupt Flag Register)

15	14	13	12	11	10	9	8
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>DCMDIF</b>	<b>DFG14IF</b>	<b>DFG13IF</b>	<b>DFG12IF</b>	<b>DFG11IF</b>	<b>DFG10IF</b>	<b>DFG9IF</b>	<b>DFG8IF</b>
7	6	5	4	3	2	1	0
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>DFG7IF</b>	<b>DFG6IF</b>	<b>DFG5IF</b>	<b>DFG4IF</b>	<b>DFG3IF</b>	<b>DFG2IF</b>	<b>DFG1IF</b>	<b>DFG0IF</b>

位编号	位符号	说明
<b>15</b>	<b>DCMDIF</b>	<b>DSP向MCU写命令的标志位</b> DSP向DTMC寄存器写值后，硬件将自动置位该标志为1；DSP软件也可以直接对该标志位写1置位，写0清除
<b>14-0</b>	<b>DFGxIF (x = 14-0)</b>	<b>DSP向MCU发出的通讯标志</b> 0: DSP清0，取消对MCU发出通讯标志x (x = 14-0) 1: DSP置1，立即向MCU发出通讯标志x (x = 14-0)

**Table 7.11** DSP至MCU的数据寄存器DTMDx (x = 0-5) (DSP to MCU Data Register)

<b>15-0</b>
<i>R/W-0</i>
<b>DTMDx (x = 0-5)</b>

位编号	位符号	说明
<b>15-0</b>	<b>DTMDx (x = 0-5)</b>	<b>DSP至MCU的数据寄存器</b> DSP向此寄存器写入的数据，可以被MCU程序读取到

**Table 7.12** MCU至DSP的命令寄存器MTDC (MCU to DSP Command Register)

<b>15-0</b>
<i>R-0</i>
<b>MTDC</b>

位编号	位符号	说明
<b>15-0</b>	<b>MTDC</b>	<b>MCU至DSP的命令寄存器</b> DSP程序可以通过读取MTDC寄存器得到MCU写入的命令值



**Table 7.13** MCU至DSP的标志寄存器MTDIF (MCU to DSP Interrupt Flag Register)

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>MCMDIF</b>	<b>MFG14IF</b>	<b>MFG13IF</b>	<b>MFG12IF</b>	<b>MFG11IF</b>	<b>MFG10IF</b>	<b>MFG9IF</b>	<b>MFG8IF</b>
<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>MFG7IF</b>	<b>MFG6IF</b>	<b>MFG5IF</b>	<b>MFG4IF</b>	<b>MFG3IF</b>	<b>MFG2IF</b>	<b>MFG1IF</b>	<b>MFG0IF</b>

位编号	位符号	说明
<b>15</b>	<b>MCMDIF</b>	<b>MCU向DSP写命令的中断标志位</b> 0: MCU未向MTDC寄存器写过值 1: MCU已向MTDC寄存器写过值 写0清除该标志, 写1无效
<b>14-0</b>	<b>MFGxIF (x = 14-0)</b>	<b>MCU向DSP发出通讯标志x的中断标志位</b> 0: MCU未发出通讯标志x (x = 14-0) 1: MCU已发出通讯标志x (x = 14-0) 写0清除该标志, 写1无效

**Table 7.14** MCU至DSP的数据寄存器MTDDx (x = 0-5) (MCU to DSP Data Register)

<b>15-0</b>
<i>R-0</i>
<b>MTDDx (x = 0-5)</b>

位编号	位符号	说明
<b>15-0</b>	<b>MTDDx (x = 0-5)</b>	<b>MCU至DSP的数据寄存器</b> DSP程序可以通过读取MTDDx (x = 0-5) 寄存器得到MCU写入的数据



### 7.6.4 MCU使用的通讯寄存器

**Table 7.15** MCU通讯中断使能寄存器MCOMIEH/L (MCU Communication Interrupt Enable Register)

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
<b>MCOMIEH</b>	DCMDIE	DFG14IE	DFG13IE	DFG12IE	DFG11IE	DFG10IE	DFG9IE	DFG8IE
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

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
<b>MCOMIEL</b>	DFG7IE	DFG6IE	DFG5IE	DFG4IE	DFG3IE	DFG2IE	DFG1IE	DFG0IE
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
<b>H.7</b>	<b>DCMDIE</b>	<b>MCU响应DSP通讯命令的中断使能位</b> 0: 禁止MCU响应DSP的通讯命令中断 1: 允许MCU响应DSP的通讯命令中断
<b>H.6-H.0 L.7-L.0</b>	<b>DFGxIE (x = 14-0)</b>	<b>MCU响应DSP通讯标志的中断使能位</b> 0: 禁止MCU响应DSP的通讯标志x (x = 14-0) 中断 1: 允许MCU响应DSP的通讯标志x (x = 14-0) 中断

**Table 7.16** MCU至DSP的命令寄存器MTDCH/L (MCU to DSP Command Register)

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
<b>MTDCH</b>	MTDC.15	MTDC.14	MTDC.13	MTDC.12	MTDC.11	MTDC.10	MTDC.9	MTDC.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

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
<b>MTDCL</b>	MTDC.7	MTDC.6	MTDC.5	MTDC.4	MTDC.3	MTDC.2	MTDC.1	MTDC.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
<b>H.7-H.0 L.7-L.0</b>	<b>MTDC</b>	<b>MCU至DSP的命令寄存器</b> MCU依次向MTDCH和MTDCL寄存器写数据,将使得MTDIF中的MCMDIF标志置1,若DSP允许响应MCU的通讯命令中断使能位MCMDIE = 1,则将触发DSP的通讯中断。DSP程序可以通过读取MTDC寄存器得到MCU写入的命令值。 注意, DSP通讯中断由MCU写MTDCL寄存器的动作触发,因此必须先写MTDCH再写MTDCL,才能保证DSP通讯中断发生时, DSP程序读取DTMC值的正确性。



**Table7.17** MCU至DSP的标志寄存器MTDIFH/L (MCU to DSP Interrupt Flag Register)

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
<b>MTDIFH</b>	MCMDIF	MFG14IF	MFG13IF	MFG12IF	MFG11IF	MFG10IF	MFG9IF	MFG8IF
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

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
<b>MCOMIEL</b>	MFG7IF	MFG6IF	MFG5IF	MFG4IF	MFG3IF	MFG2IF	MFG1IF	MFG0IF
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
<b>H.7</b>	<b>MCMDIF</b>	<b>MCU向DSP写命令的标志位</b> MCU向MTDCH/L寄存器写值后，硬件将自动置位该标志为1；MCU软件也可以直接对该标志位写1置位，写0清除
<b>H.6-H.0 L.7-L.0</b>	<b>MFGxIF (x = 14-0)</b>	<b>MCU向DSP发出的通讯标志</b> 0: MCU清0，取消对DSP发出通讯标志x (x = 14-0) 1: MCU置1，立即向DSP发出通讯标志x (x = 14-0)

**Table 7.18** MCU至DSP的数据寄存器MTDDxH/L (x = 0-5) (MCU to DSP Data Register)

D5H/D7H/C3H/ C5H/C7H/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
<b>MTDDxH (0-5)</b>	MTDDx.15	MTDDx.14	MTDDx.13	MTDDx.12	MTDDx.11	MTDDx.10	MTDDx.9	MTDDx.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

D4H/D6H/C2H/ C4H/C6H/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
<b>MTDDxL (0-5)</b>	MTDDx.7	MTDDx.6	MTDDx.5	MTDDx.4	MTDDx.3	MTDDx.2	MTDDx.1	MTDDx.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
<b>H.7-H.0 L.7-L.0</b>	<b>MTDDx (x = 0-5)</b>	<b>MCU至DSP的数据寄存器</b> MCU向此寄存器写入的数据，可以被DSP程序读取到



**Table 7.19** DSP至MCU的命令寄存器DTMCH/L (DSP to MCU Command Register)

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
<b>DTMCH</b>	DTMC.15	DTMC.14	DTMC.13	DTMC.12	DTMC.11	DTMC.10	DTMC.9	DTMC.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

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
<b>DTMCL</b>	DTMC.7	DTMC.6	DTMC.5	DTMC.4	DTMC.3	DTMC.2	DTMC.1	DTMC.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
H.7-H.0 L.7-L.0	<b>DTMC</b>	<b>DSP至MCU的命令寄存器</b> MCU程序可以通过读取DTMCH/L寄存器得到DSP写入的命令值

**Table 7.20** DSP至MCU的标志寄存器DTMIFH/L (DSP to MCU Interrupt Flag Register)

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
<b>DTMIFH</b>	DCMDIF	DFG14IF	DFG13IF	DFG12IF	DFG11IF	DFG10IF	DFG9IF	DFG8IF
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

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
<b>DTMIFL</b>	DFG7IF	DFG6IF	DFG5IF	DFG4IF	DFG3IF	DFG2IF	DFG1IF	DFG0IF
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
<b>H.7</b>	<b>DCMDIF</b>	<b>DSP向MCU写命令的中断标志位</b> 0: DSP未向DTMC寄存器写过值 1: DSP已向DTMC寄存器写过值 写0清除该标志, 写1无效
<b>H.6-H.0 L.7-L.0</b>	<b>DFGxIF (x = 14-0)</b>	<b>DSP向MCU发出通讯标志x的中断标志位</b> 0: DSP未发出通讯标志x (x = 14-0) 1: DSP已发出通讯标志x (x = 14-0) 写0清除该标志, 写1无效



**Table 7.21** DSP至MCU的数据寄存器DTMDxH/L (x = 0-5) (DSP to MCU Data Register)

BFH/B7H/ABH/ ADH/D5H/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
<b>DTMDxH (0-5)</b>	DTMDx.15	DTMDx.14	DTMDx.13	DTMDx.12	DTMDx.11	DTMDx.10	DTMDx.9	DTMDx.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

BEH/B6H/AAH/ ACH/D4H/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
<b>DTMDxL (0-5)</b>	DTMDx.7	DTMDx.6	DTMDx.5	DTMDx.4	DTMDx.3	DTMDx.2	DTMDx.1	DTMDx.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
H.7-H.0 L.7-L.0	<b>DTMDx (x = 0-5)</b>	<b>DSP至MCU的数据寄存器</b> DSP向此寄存器写入的数据, 可以被MCU程序读取到

\*说明: 对于MCU使用的通讯寄存器, DTMD4H (D5H)、DTMD4L (D4H)、DTMD5H (D7H)、DTMD5L (D6H) 位于SFR Bank1中, 其余通讯寄存器均位于SFR Bank0。

### 7.6.5 通信中断

DSP和MCU之间通过中断的方式传递数据。通讯中断分为两种: 命令中断和标志中断。

#### 通讯命令中断

DSP写DTMC寄存器, 将使DTMIF寄存器中的DCMDIF置1, 若MCU的通讯中断使能寄存器MCOMIEH中的DCMDIE = 1, IEN1中的EDCMD = 1且EA = 1, 则MCU的通讯中断将被触发, MCU可以在通讯中断服务程序中通过DTMCH/L寄存器读取DSP写入的命令值, 即实现DSP传递命令信息给MCU的过程。

MCU依次写MTDCH和MTDCL寄存器, 将使MTDIF寄存器中的MCMDIF置1, 若DSP的通讯中断使能寄存器DCOMIE中的DCMDIE = 1, 且IM\_COMM = 1, 则DSP的通讯中断将被触发, DSP可以在通讯中断服务程序中通过MTDC寄存器读取MCU写入的命令值, 即实现MCU传递命令信息给DSP的过程。注意, DSP通讯中断由MCU写MTDCL寄存器的动作触发, 因此必须先写MTDCH再写MTDCL, 才能保证DSP通讯中断发生时, DSP程序读取MTDC值的正确性。

DTMC和MTDC在DSP和MCU之间起到邮箱的作用。

#### 通讯标志中断

SH99F100提供了30个通讯标志位供DSP与MCU通讯使用。

对于DSP, 将DTMIF寄存器中DFGxIF (x = 0-14) 任何一位置1, 若MCU的通讯中断使能寄存器MCOMIEH/L中对应的DFGxIE位为1, IEN1中的EDFLAG = 1且EA = 1, 则MCU的通讯中断将被触发。

对于MCU, 将MTDIF寄存器中MFGxIF (x = 0-14) 任何一位置1, 若DSP的通讯中断使能寄存器DCOMIE中对应的MFGxIE位为1, 且IM\_COMM = 1, 则DSP的通讯中断将被触发。



### 7.7 在系统可编程功能ISP (In System Program)

SH99F100使用了一块128Kbytes的Flash，如图7-8所示。其中64Kbytes被作为DCSM (DSP Code Storage Memory) 区域，用于存储DSP代码和变量初始化值；另64Kbytes作为MCM (MCU Code Memory) 区域，用于存放MCU的代码，并且MCU在其中运行程序。

SH99F100支持在系统编程(ISP)模式。芯片内部含有一块BootRom，其大小为1Kbytes，占据程序存储空间0x0000 - 0x03FF的地址段。BootRom内部固化了ISP程序，可对Flash存储器中的MCM区和DCSM区进行擦除和读写操作。通过串口实现烧写芯片。

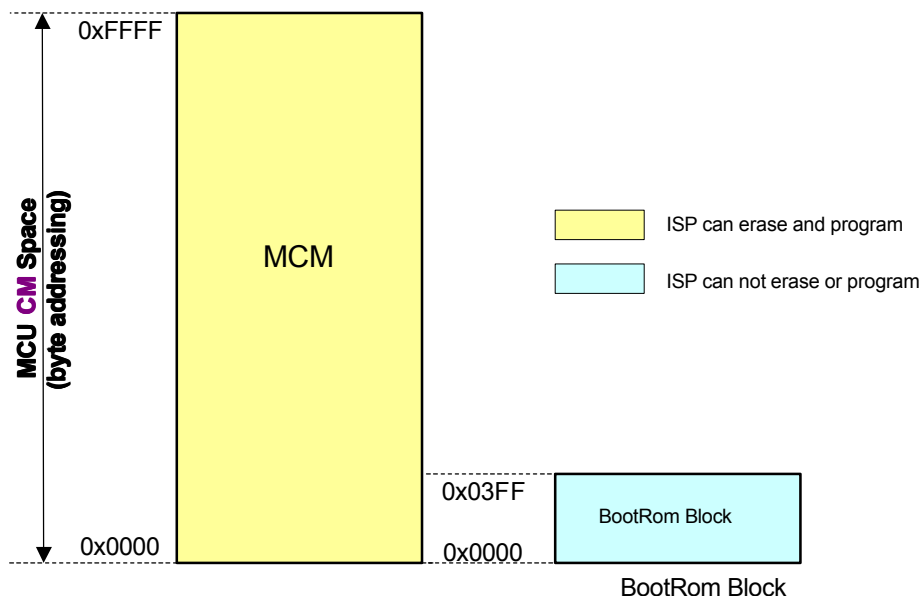


图7-8 Flash存储空间分配示意图

SH99F100通过MCU核支持ISP模式。有两种进入ISP的方式。

#### 7.7.1 上电复位触发ISP

如果要使用上电复位触发ISP，则用户必须通过代码选项选择使能ISP功能（代码选项OP\_ISP清0）。此外代码选项OP\_ISPPIN若设置为1，MCU将直接执行BootRom中的程序（此时DSP停止运行），从BootRom的0000H地址开始运行。代码选项OP\_ISPPIN若设置为0，则复位时检测P2.2和P2.3状态。当P2.2和P2.3同时为低且持续时间超过100μs时，MCU将执行BootRom中的程序（此时DSP停止运行），从BootRom的0000H地址开始运行。若不满足P2.2和P2.3同时为低，或P2.2和P2.3同时为低但持续时间小于100μs时，则MCU将不会执行BootRom中的程序。

BootRom中的ISP程序执行完毕时会执行一条软件复位指令SWRCON = 5AH，这样系统将执行Autoload过程，Autoload过程结束后，MCU从MCM区的0x0000地址开始运行，DSP的PC指针指向DCRM空间的0x0000地址并处于停止状态。





### 7.7.2 按键复位触发ISP

如果要使用按键复位触发ISP，则用户必须通过代码选项选择使能ISP功能（代码选项OP\_ISP清0）。此外代码选项OP\_ISPPIN若设置为1，MCU将直接执行BootRom中的程序（此时DSP停止运行），从BootRom的000H地址开始运行。代码选项OP\_ISPPIN若设置为0，则复位时检测P2.2和P2.3状态。当P2.2和P2.3同时为低且持续时间超过100μs时，MCU将执行BootRom中的程序（此时DSP停止运行），从BootRom的000H地址开始运行。若不满足P2.2和P2.3同时为低，或P2.2和P2.3同时为低但持续时间小于100μs时，则MCU将不会执行BootRom中的程序。

BootRom中的ISP程序执行完毕时会执行一条软件复位指令SWRCON = 5AH，这样系统将执行Autoload过程，Autoload过程结束后，MCU从MCM区的0x0000地址开始运行，DSP的PC指针指向DCRM空间的0x0000地址并处于停止状态。

上电复位和按键复位触发ISP的流程如图7-9所示。

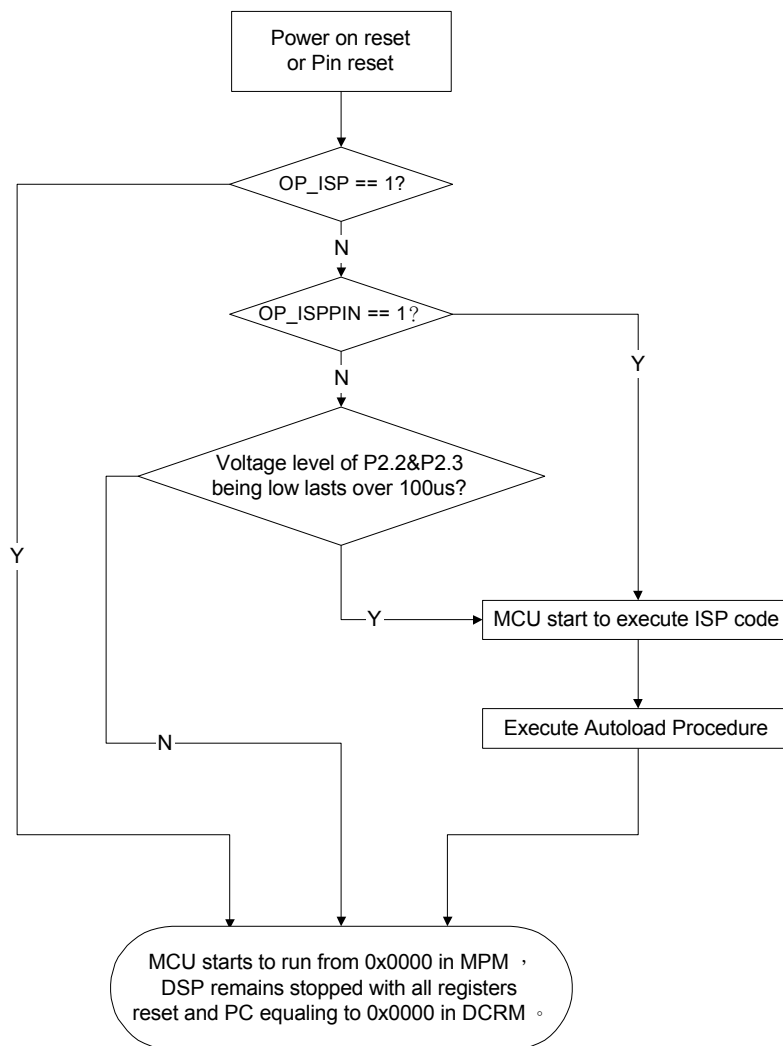


图7-9



## 7.8 电源管理

### 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，SH99F100提供两种低功耗省电模式：空闲（Idle）模式和掉电（Power-Down）模式，这两种模式都由PCON和SUSLO两个寄存器控制。

### 7.8.1 空闲模式（Idle）

空闲模式能够降低系统功耗，在此模式下，程序中止运行，MCU核的时钟停止，此时DSP核继续运行，不受MCU空闲模式的影响。但外部设备时钟继续运行。空闲模式下，MCU在确定的状态下停止，并在进入空闲模式前所有MCU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0x55，随即将PCON寄存器中的IDL位置1，使SH99F100进入空闲模式。如果不满足上述的两条连续指令，MCU在下一个机器周期清除SUSLO寄存器或IDL位，MCU也不会进入空闲模式。

IDL位置1是MCU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) 中断产生。在预热定时结束后，恢复MCU核的时钟，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）。MCU核的恢复时钟，SUSLO寄存器和在PCON寄存器中的IDL位被硬件清除，最后SH99F100复位，复位结果见7.2节。此时，MCU的RAM保持不变而SFR的值根据不同功能模块改变。

### 7.8.2 掉电模式（Power-Down）

掉电模式可以使SH99F100进入功耗非常低的状态。掉电模式将停止MCU核和外围设备的所有时钟信号，DSP时钟亦停止且处于复位态。如果WDT使能，WDT模块将继续工作。在进入掉电模式前所有MCU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0x55，随即将PCON寄存器中的PD位置1，使SH99F100进入掉电模式。如果不满足上述的两条连续指令MCU在下一个机器周期清除SUSLO寄存器或的PD位，MCU也不会进入掉电模式。

PD位置1是MCU进入掉电模式之前执行的最后一条指令。

**注意：**如果同时设置IDL位和PD位，SH99F100进入掉电模式。退出掉电模式后，MCU也不会进入空闲模式，从掉电模式退出后硬件清0 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和INT2）使SH99F100退出掉电模式。在中断发生后振荡器启动，在预热计时结束之后MCU核时钟和外部设备时钟恢复，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。DSP仍处于复位状态。

(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MCU核时钟。SUSLO寄存器和PCON寄存器中的PD位会被硬件清除，最后SH99F100会被复位，复位结果见7.2节。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（NOP）。



7.8.3 寄存器

Table 7.22 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
<b>PCON</b>	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
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	<b>SMOD</b>	<b>UART</b> 波特率加倍器
6	<b>SSTAT</b>	<b>SCON[7:5]</b> 功能选择位
5	<b>SSTAT1</b>	<b>SCON1[7:5]</b> 功能选择位
3-2	<b>GF[1:0]</b>	用于软件的通用标志
1	<b>PD</b>	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	<b>IDL</b>	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 7.23 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
<b>SUSLO</b>	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSL O.3	SUSL O.2	SUSL O.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	<b>SUSLO[7:0]</b>	此寄存器用来控制MCU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. MCU部分

### 8.1 MCU核特性

#### 8.1.1 CPU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 8.1** 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
<b>PSW</b>	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	<b>进位标志位</b> 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	<b>辅助进位标志位</b> 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	<b>F0标志位</b> 用户自定义标志位
4-3	RS[1:0]	<b>R0-R7寄存器页选择位</b> 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	<b>溢出标志位</b> 0: 没有溢出发生 1: 有溢出发生
1	F1	<b>F1标志位</b> 用户自定义标志位
0	P	<b>奇偶校验位</b> 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。



### 8.1.2 CPU增强内核特殊功能寄存器

#### 特性

- 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

MCU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		
			A	B	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)	商低位字节	余数	---
	INSCON.3 = 1; 16位模式	(AUXC A)/(B)	商低位字节	余数	商高位字节

#### 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。

### 8.1.3 寄存器

Table 8.2 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	-	BANKS	DUINT	-	DIV	MUL	-	DPS
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
6	BANKS	<b>SFR Bank选择位</b> 0: 选择bank0 1: 选择bank1
5	DUINT	<b>通讯数据单元控制权分配位</b> 0: MCU享有通讯数据单元控制权 1: DSP享有通讯数据单元控制权
3	DIV	<b>16位/8位除法选择位</b> 0: 8位除法 1: 16位除法
2	MUL	<b>16位/8位乘法选择位</b> 0: 8位乘法 1: 16位乘法
0	DPS	<b>数据指针选择位</b> 0: 数据指针 1: 数据指针1

**8.2 低速RAM区（MIDM/MEDM区）****8.2.1 特性**

SH99F100为MCU的数据存储提供了256bytes的内部RAM（MIDM区）和3840bytes的外部RAM（MEDM区）。下列为存储器空间分配：

- MIDM区中低位128字节的RAM（地址从00H到7FH）可直接或间接寻址
- MIDM区中高位128字节的RAM（地址从80H到FFH）只能间接寻址
- MIDM区中特殊功能寄存器（SFR，地址从80H到FFH）只能直接寻址
- MEDM区中外部RAM字节可通过MOVX指令间接寻址

MIDM区中高位128字节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。当一个指令访问地址高于7FH的内部位置时，MCU可以根据指令的寻址方式来区分是访问高位128字节数据RAM还是访问SFR。

**注意：**未使用的SFR地址禁止读写。

MCU支持传统的访问外部RAM方法。使用 *MOVXA, @Ri*或*MOVX@Ri, A*来访问外部低位256字节RAM；用 *MOVX A, @DPTR*或*MOVX@DPTR, A*来访问外部3840字节RAM。

用户也能用XPAGE寄存器来访问外部RAM，使用 *MOVX A, @Ri*或*MOVX@Ri, A*指令即可，此时用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）。

**8.2.2 寄存器****Table 8.3** 数据存储页寄存器

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

<b>位编号</b>	<b>位符号</b>	<b>说明</b>
<b>7-0</b>	<b>XPAGE[7:0]</b>	RAM页选择控制位



### 8.3 Flash程序存储器（MCM区与DCSM区）

#### 8.3.1 特性

- Flash 存储器包括 128 X 1KB 扇区，总共 128KBytes
- 在工作电压范围内都能进行编程和擦除操作
- 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：至少 100000 次
- 数据保存年限：至少 10 年
- 低功耗

SH99F100 为存储程序代码内置128Kbytes可编程Flash，其中64K作为MCM区给MCU存放和运行代码；另64K作为DCSM区用于存放DSP代码和DSP变量初始化值。可以通过在线编程（ICP）模式对MCM和DCSM进行擦写。而扇区自编程（SSP）模式只能对MCM存储器操作。

在ICP模式中，编程器能操作所有Flash，例如擦除或写入。Flash的读取或写入操作以字节为单位，擦除以扇区（1Kbytes）为单位，或者整体擦除。

在 ICP 模式下还可以进行整体擦除，这个操作会擦除整个 Flash 存储器（包括 MCM 区和 DCSM 区）。整体擦除操作仅能在 ICP 模式下使用，SSP 模式下不支持整体擦除操作。

#### 8.3.2 ICP模式下的Flash操作

ICP模式即在线编程模式，可以在CPU焊在用户板上以后编程。ICP模式下，用户系统必须关机后编程器才能通过ICP编程接口刷新Flash存储器。ICP编程接口包括6个引脚（V<sub>DD</sub>，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**。

**ICP模式支持以下操作：**

##### (1) 代码保护控制模式编程

SH99F100的代码保护功能为用户代码提供了高性能的安全措施。每个分区有两种模式可用。

代码保护模式0：允许/禁止任何编程器的写入/读取/扇区擦除操作（不包括整体擦除）。

代码保护模式1：允许/禁止在其他分区中通过MOVC指令进行读取操作，或通过SSP模式进行擦除/写入操作。

用户必须应用**Flash编程器**设置相应的保护位，以进入所需的保护模式。

##### (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，代码保护位和自定义ID码的内容。（Flash编程器为用户提供自定义ID码设置功能以区别他们的产品）。

在用户程序区，整体擦除只能由Flash编程器操作，不能通过程序指令完成。

##### (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。

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。

**注意：SSP程序所在扇区自身无法通过用户程序执行扇区擦除功能。**

##### (4) 写/读代码

读/写代码操作可以将代码、数据从Flash存储器中读出或写入Flash存储器。编程器或用户程式都能执行该操作。

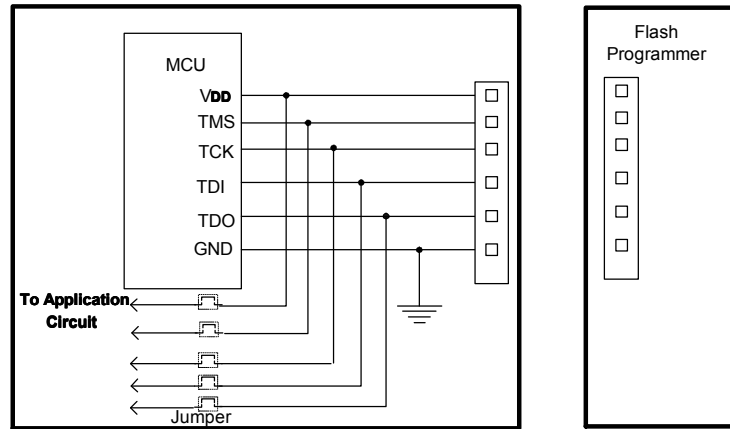
若需用户程式执行该操作，必须禁止所选扇区的代码保护控制模式1。不管安全位设置与否，用户程序都能读/写程式自身所在扇区。

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。

操作	ICP	SSP	ISP
代码保护	支持	不支持	支持
扇区擦除	支持（无安全位）	支持（无安全位）	支持（无安全位）
整体擦除	支持	不支持	支持
写/读	支持（无安全位）	支持（无安全位或固有分区）	支持（无安全位）



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用户需要先用6个跳线将芯片的编程引脚（V<sub>DD</sub>，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编程器编程接口后，开始编程；
- (3) 编程结束后断开编程器接口，连接跳线恢复应用电路。





### 8.4 Flash的扇区自编程（SSP）功能

SH99F100中的Flash支持SSP功能，由MCU核来控制SSP的运行。如果所选扇区未被保护，用户代码可以擦除所有扇区或对整个扇区执行编程操作。一旦该扇区被编程，则在该扇区被擦除之前不能被再次编程。

MCU内建一个复杂控制流程以避免误入SSP模式导致代码被误修改。为进入SSP模式，寄存器IB\_CON2（位于MCU的SFR区中）必须满足特定条件。若IB\_CON2-5不满足特定条件，则无法进入SSP模式。

**注意：**XPAGE1的第0位与XPAGE的7-2位组成扇区号，一共能表示128个扇区（包括MCM和DCSM区），当MCU程序运行在BootRom中时寄存器XPAGE1有效，当MCU程序运行在MCM区时寄存器XPAGE1无效。

#### 8.4.1 SSP寄存器

**Table 8.4** 编程用地址选择寄存器XPAGE1

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
XPAGE1	-	-	-	-	-	-	-	XPAGE.8
读/写	-	-	-	-	-	-	-	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	-	-	0

编程用地址选择寄存器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
XPAGE1[0]-XPAGE[7:2]	XPAGE[8:2]	被编程的存储单元扇区号，000000代表扇区0，以此类推
1-0	XPAGE[1:0]	被编程的存储单元高2位地址

**Table 8.5** 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

**Table 8.6** 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 8.7 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	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_CON1[7:0]	SSP操作选择 0xE6: 扇区擦除 0x6E: 存储单元编程

Table 8.8 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 8.9 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 8.10 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 8.11 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

Table 8.12 软件复位标志和指令访问控制寄存器

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	SWRF	-	-	-	-	-	-	FAC
读/写	读/写	-	-	-	-	-	-	读/写
复位值 (POR/WDT/LVR/PIN)	0	-	-	-	-	-	-	0

位编号	位符号	说明
7	SWRF	软件复位标志 只能在发生软件复位后由硬件自动置1； 由软件清除或者其它复位清0。
0	FAC	执行区域选择位（详见SSP章节）

### 8.4.2 SSP编程注意事项

为确保顺利完成SSP编程，MCU程序必须按以下步骤设置：

#### (1) 用于代码/数据编程：

1. 关闭MCU中断（EA = 0）；
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（EA = 1）。

#### (2) 用于扇区擦除：

1. 关闭MCU中断（EA = 0）；
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（EA = 1）。

注意：数据访问可通过指令“MOVC A, @A+DPTR”或“MOVC A, @A+PC”实现。



## 8.5 I/O端口

### 8.5.1 特性

- 56个双向I/O端口
- I/O端口可与其它功能共用

SH99F100提供8组8位可编程双向I/O端口。I/O口只能由MCU控制。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（对于x = 0-6, y = 0-7）。

SH99F100的有些I/O引脚能与选择功能共用。当所有功能都允许时，在CPU中存在优先权以避免功能冲突。（详见端口共用章节）。

### 8.5.2 寄存器

Table 8.13 端口控制寄存器

E1H-E5H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>P0CR (E1H)</b>	P0CR.7	P0CR.6	P0CR.5	P0CR.4	P0CR.3	P0CR.2	P0CR.1	P0CR.0
<b>P1CR (E2H)</b>	P1CR.7	P1CR.6	P1CR.5	P1CR.4	P1CR.3	P1CR.2	P1CR.1	P1CR.0
<b>P2CR (E3H)</b>	P2CR.7	P2CR.6	P2CR.5	P2CR.4	P2CR.3	P2CR.2	P2CR.1	P2CR.0
<b>P3CR (E4H)</b>	P3CR.7	P3CR.6	P3CR.5	P3CR.4	P3CR.3	P3CR.2	P3CR.1	P3CR.0
<b>P4CR (E5H)</b>	P4CR.7	P4CR.6	P4CR.5	P4CR.4	P4CR.3	P4CR.2	P4CR.1	P4CR.0
<b>P5CR (E1H) (bank1)</b>	P5CR.7	P5CR.6	P5CR.5	P5CR.4	P5CR.3	P5CR.2	P5CR.1	P5CR.0
<b>P6CR (E2H) (bank1)</b>	P6CR.7	P6CR.6	P6CR.5	P6CR.4	P6CR.3	P6CR.2	P6CR.1	P6CR.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	<b>PxCR.y</b> x = 0-6, y = 0-7	端口输入/输出控制寄存器 0: 输入模式 1: 输出模式

Table 8.14 端口上拉电阻控制寄存器

E9H-EDH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>P0PCR (E9H)</b>	P0PCR.7	P0PCR.6	P0PCR.5	P0PCR.4	P0PCR.3	P0PCR.2	P0PCR.1	P0PCR.0
<b>P1PCR (EAH)</b>	P1PCR.7	P1PCR.6	P1PCR.5	P1PCR.4	P1PCR.3	P1PCR.2	P1PCR.1	P1PCR.0
<b>P2PCR (EBH)</b>	P2PCR.7	P2PCR.6	P2PCR.5	P2PCR.4	P2PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
<b>P3PCR (ECH)</b>	P3PCR.7	P3PCR.6	P3PCR.5	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	P3PCR.0
<b>P4PCR (EDH)</b>	P4PCR.7	P4PCR.6	P4PCR.5	P4PCR.4	P4PCR.3	P4PCR.2	P4PCR.1	P4PCR.0
<b>P5PCR (E9H) (bank1)</b>	P5PCR.7	P5PCR.6	P5PCR.5	P5PCR.4	P5PCR.3	P5PCR.2	P5PCR.1	P5PCR.0
<b>P6PCR (EAH) (bank1)</b>	P6PCR.7	P6PCR.6	P6PCR.5	P6PCR.4	P6PCR.3	P6PCR.2	P6PCR.1	P6PCR.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	<b>PxPCR.y</b> x = 0-6, 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

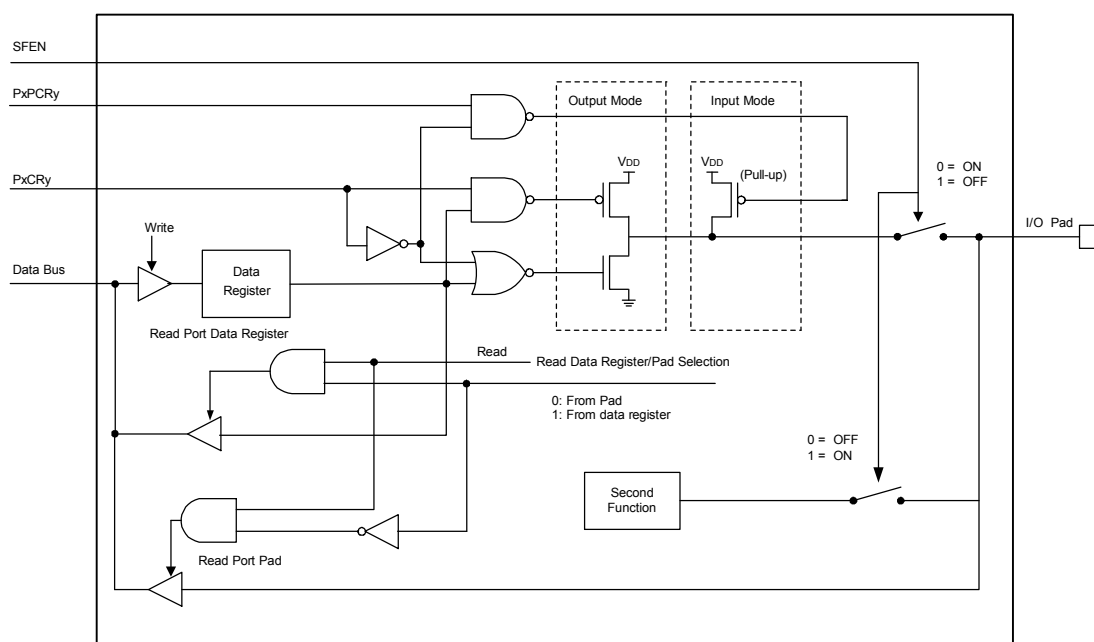


**Table 8.15** 端口数据寄存器

80H-C0H	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>P0 (80H)</b>	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
<b>P1 (90H)</b>	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
<b>P2 (A0H)</b>	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
<b>P3 (B0H)</b>	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
<b>P4 (C0H)</b>	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
<b>P5 (80H) (bank1)</b>	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
<b>P6 (90H) (bank1)</b>	P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0
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
<b>7-0</b>	<b>Px.y</b> <b>x = 0-6, y = 0-7</b>	端口数据寄存器

**8.5.3 端口模块图**



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-修改-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它功能，对端口写操作都是针对端口数据寄存器。

**8.5.4 端口共用**

56个双向I/O端口也能共用作第二种特殊功能。在**引脚配置图**中引脚名称标识为粗体的功能享有最高优先级。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 = 0-6），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，任何对端口的读写操作只会影响到数据寄存器的值，端口引脚值保持不变，直到复用的其它功能关闭。

**PORT0:**

PORT0共用列表

引脚编号	优先级	功能	允许位
100	1	INT0	TCON寄存器的TR0位和TMOD寄存器的GATE0位都置1（自动上拉）
	2	P0.0	无上述情况
1	1	INT1	TCON寄存器的TR1位和TMOD寄存器的GATE1位都置1（自动上拉）
	2	P0.1	无上述情况
2	1	T2	T2CON寄存器的TR2位和C/T2位都置1（自动上拉）或T2CON寄存器的TR2位和T2MOD寄存器的T2OE位都置1
	2	P0.2	无上述情况
3	1	TXD1	对SBUF1寄存器写操作
	2	P0.3	无上述情况
4	1	RXD1	在模式0下对SBUF1寄存器写操作或SCON1寄存器的REN位置1（自动上拉）
	2	P0.4	无上述情况
5	1	INT2	
	2	P0.5	始终作为I/O
6	1	T0	TCON寄存器的TR1位和TMOD寄存器的C/T0位都置1（自动上拉）或TCON寄存器的TR0位和TCON1寄存器的TC0位都置1
	2	P0.6	无上述情况
7	1	T2EX	T2CON寄存器的EXEN2位置1（自动上拉）或在模式1下T2CON寄存器的EXEN2位清0和T2MOD寄存器的DCEN位置1（自动上拉）
	2	P0.7	无上述情况



**PORT1:**

PORT1共用列表

引脚编号	优先级	功能	允许位
10	1	TDO	使用JTAG功能
	2	P1.0	不使用JTAG功能时
11	1	TMS	使用JTAG功能
	2	P1.1	不使用JTAG功能时
12	1	TDI	使用JTAG功能
	2	P1.2	不使用JTAG功能时
13	1	TCK	使用JTAG功能
	2	P1.3	不使用JTAG功能时
14	1	APWM0	APWMCON0寄存器中APWMCH0置1
	2	P1.4	APWMCON0寄存器中APWMCH0置0
15	1	APWM1	APWMCON1寄存器中APWMCH1置1
	2	P1.5	APWMCON1寄存器中APWMCH1置0
16	1	PDPINT0	PWMFCON寄存器中PFLT0EN位置1
	2	P1.6	无上述情况
17	1	APWM2	APWMCON2寄存器中APWMCH2置1
	2	P1.7	APWMCON2寄存器中APWMCH2置0

**PORT2:**

PORT2共用列表

引脚编号	优先级	功能	允许位
36	1	OC0	OC0CON中OC0MOD = 1/2/3/4/5/6/7
	2	P2.0	OC0CON中OC0MOD = 0
37	1	OCFLT	OCxCON中OCxMOD = 7 (x = 0-2)
	2	P2.1	无上述情况
38	1	QEB	QEICON寄存器中QEIEN = 1
	2	P2.2	QEICON寄存器中QEIEN = 0
39	1	QEA	QEICON寄存器中QEIEN = 1
	2	CAP0	CAP0CON中CAP0TG = 1/2/3
	3	P2.3	QEICON寄存器中QEIEN = 0且CAP0CON中CAP0TG = 0
40	1	INDEX	QEICON寄存器中QEIEN = 1
	2	CAP1	CAP1CON中CAP1TG = 1/2/3
	3	P2.4	QEICON寄存器中QEIEN = 0且CAP1CON中CAP1TG = 0
41	1	OC1	OC1CON中OC1MOD = 1/2/3/4/5/6/7
	2	P2.5	OC1CON中OC1MOD = 0
42	1	OC2	OC2CON中OC2MOD = 1/2/3/4/5/6/7
	2	P2.6	OC2CON中OC2MOD = 0
43	1	CAP2	CAP2CON中CAP2TG = 1/2/3
	2	P2.7	CAP2CON中CAP2TG = 0



**PORT3:**

PORT3共用列表

引脚编号	优先级	功能	允许位
52	1	AN0	ADCH寄存器中CH0位置1
	2	P3.0	ADCH寄存器中CH0位置0
53	1	AN1	ADCH寄存器中CH1位置1
	2	P3.1	ADCH寄存器中CH1位置0
54	1	AN2	ADCH寄存器中CH2位置1
	2	P3.2	ADCH寄存器中CH2位置0
55	1	AN3	ADCH寄存器中CH3位置1
	2	P3.3	ADCH寄存器中CH3位置0
56	1	AN4	ADCH寄存器中CH4位置1
	2	P3.4	ADCH寄存器中CH4位置0
57	1	AN5	ADCH寄存器中CH5位置1
	2	P3.5	ADCH寄存器中CH5位置0
58	1	AN6	ADCH寄存器中CH6位置1
	2	P3.6	ADCH寄存器中CH6位置0
59	1	AN7	ADCH寄存器中CH7位置1
	2	P3.7	ADCH寄存器中CH7位置0

**PORT4:**

PORT4共用列表

引脚编号	优先级	功能	允许位
88	1	RXD0	在模式0下对SBUF0寄存器写操作或SCON寄存器的REN位置1（自动上拉）
	2	P4.0	无上述情况
89	1	TXD0	对SBUF0寄存器写操作
	2	P4.1	无上述情况
90	1	SS	当SPEN = 1时，在SPI主模式下将SPCON寄存器的SSDIS位清0， 或者在SPI从模式下当CPHA = 1时将SPCON寄存器的SSDIS位清0， 或者在SPI从模式下将SPCON寄存器的CPHA位清0 (当SPEN = 1 & Master = 1 & SSDIS = 0时，或当SPEN = 1 & Master = 0时，自动上拉)
	2	P4.2	无上述情况
91	1	SCK	SPSTA寄存器的SPEN位置1 (当SPEN, CPHA, SSDIS位在从属模式下都置1时，自动上拉)
	2	P4.3	SPSTA寄存器的SPEN位清0
92	1	MISO	将SPSTA寄存器的SPEN位置1 (在主模式下将SPSTA寄存器的SPEN位置1时，自动上拉)
	2	P4.4	SPSTA寄存器的SPEN位清0
93	1	MOSI	在从属模式下将SPSTA寄存器的SPEN位置1 (当SPEN, CPHA, SSDIS位在从属模式下都置为1时，自动上拉)
	2	P4.5	SPSTA寄存器的SPEN位清0
94	1	T1	TCON寄存器的TR1位和TMOD寄存器的C/T1 $\bar{1}$ 位都置1（自动上拉） 或TCON寄存器的TR1位和TCON1寄存器的TC1位都置1
	2	P4.6	无上述情况



**PORT5:**

PORT5共用列表

引脚编号	优先级	功能	允许位
20	1	PDPINT1	PWMFCON寄存器中PFLT1EN位置1
	2	P5.4	无上述情况
21	1	PDPINT2	PWMFCON寄存器中PFLT2EN位置1
	2	P5.5	无上述情况
22	1	PDPINT3	PWMFCON寄存器中PFLT3EN位置1
	2	P5.6	无上述情况
23	1	PDPINT4	PWMFCON寄存器中PFLT4EN位置1
	2	P5.7	无上述情况

**PORT6:**

PORT6共用列表

引脚编号	优先级	功能	允许位
44	1	CAP3	CAP3CON中CAP3TG = 1/2/3
	2	P6.2	CAP3CON中CAP3TG = 0
47	1	DXINT0	
	2	P6.3	始终作为I/O
48	1	DXINT1	
	2	P6.4	始终作为I/O
49	1	DXINT2	
	2	P6.5	始终作为I/O



### 8.6 定时器Timer0/1/2

#### 8.6.1 特性

- SH99F100有3个定时器（定时器0，1，2）
- 定时器0兼容标准的8051
- 定时器1兼容标准的8051
- 定时器2兼容标准的8052，且有递增递减计数和可编程输出功能
- 定时器0/1增加了比较输出功能
- 定时器0/1增加了时钟源分频功能

#### 8.6.2 定时器0和定时器1

每个定时的两个数据寄存器（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MCU中断章节）。

#### 定时器x的工作方式 (x = 0, 1)

通过计数器/定时器模式寄存器（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

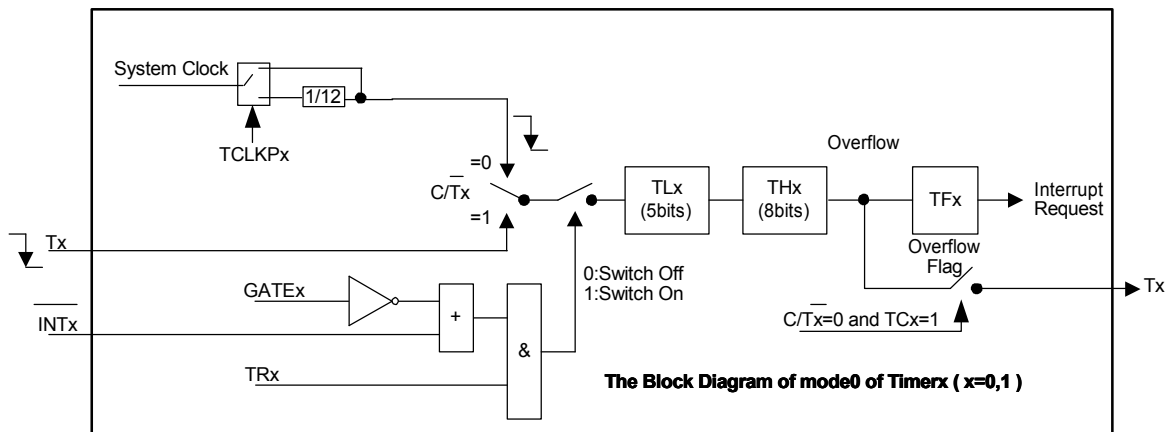
在模式0中，定时器x为13位计数器/定时器。THx寄存器存放13位计数器/定时器的高8位，TLx存放低5位（TLx.4-TLx.0）。TLx的高三位（TLx.7-TLx.5）是不确定的，在读取时应该被忽略。当13位定时器寄存器递增，溢出时，系统置起定时器溢出标志TFx。如果定时器x中断被允许，将会产生一个中断。C/Tx位选择计数器/定时器的时钟源。

如果C/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/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INTx控制，便于测量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。

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。

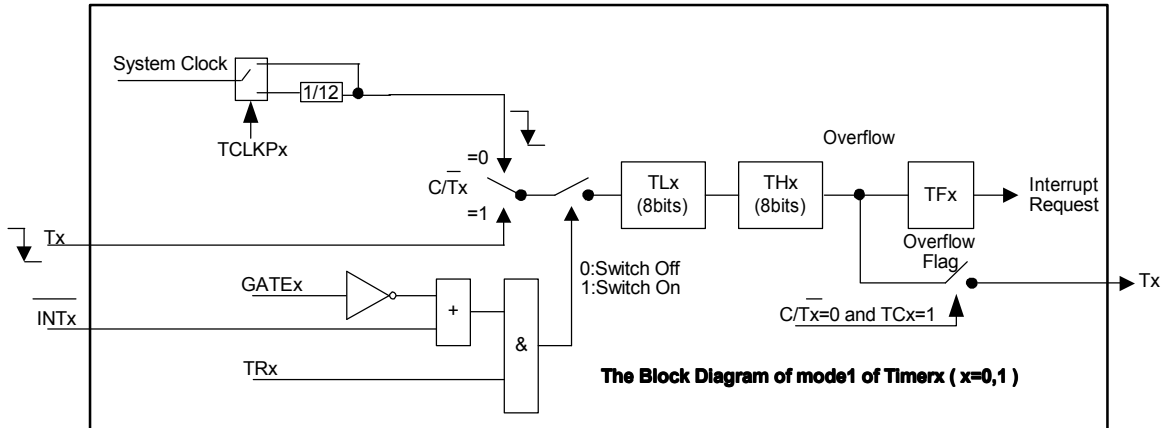
当作为定时器应用时，可配置寄存器TCON1中的TC0/1位使定时器0/1溢出时T0/T1脚自动翻转。如果TC0/1被置1，T0/T1引脚自动设置为输出。





### 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。



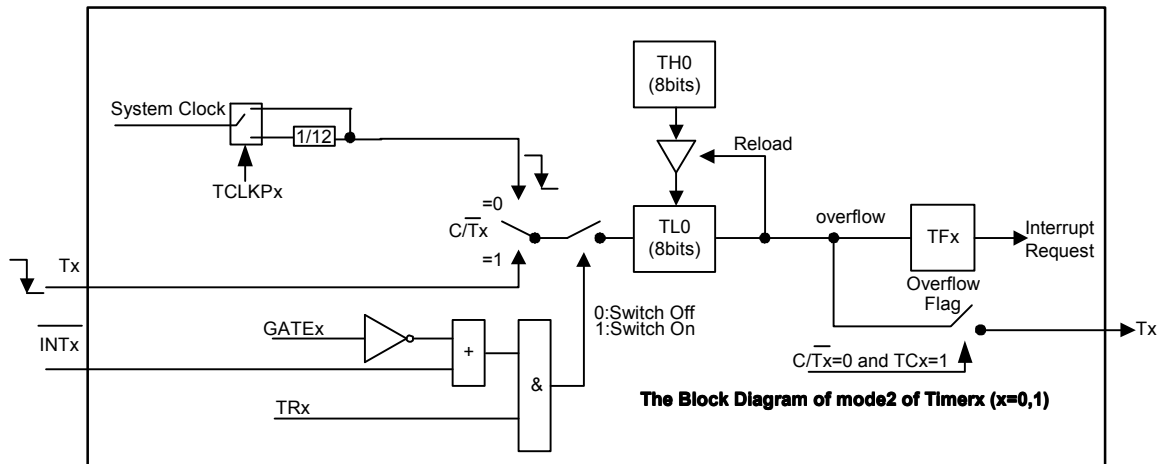
### 方式2：8位自动重载计数器/定时器

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THx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。

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）**

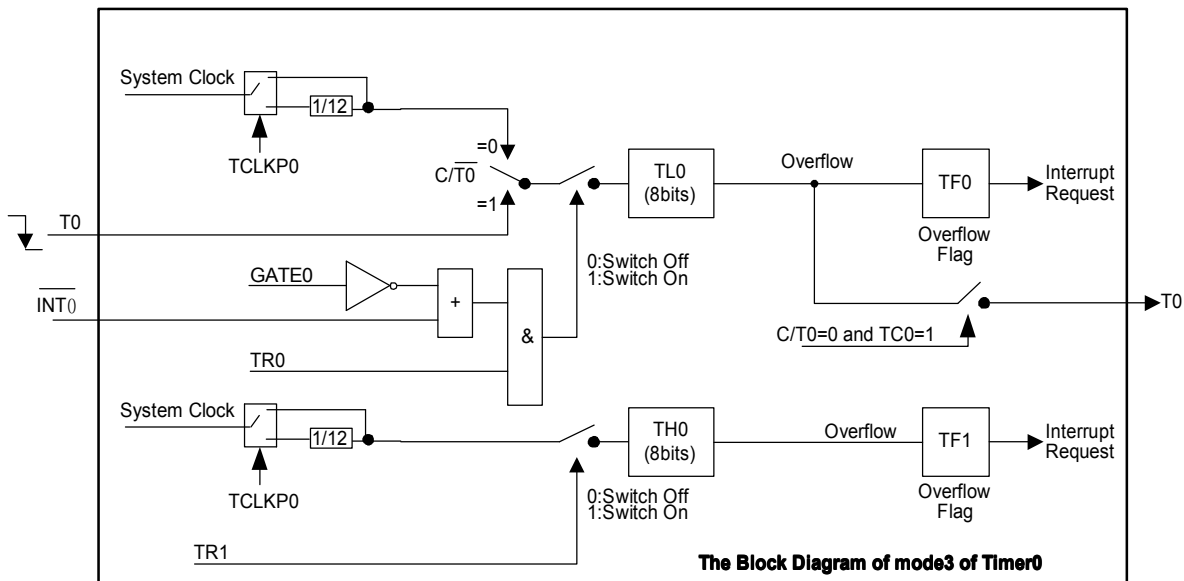
在方式3中，定时器0用作两个独立的8位计数器/定时器，分别由TL0和TH0控制。TL0使用定时器0的控制（在TCON中）和状态（在TMOD中）位：TR0，C/T0，GATE0和TF0。TL0能用系统时钟或外部输入信号作为时钟源。

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，可以用来产生串口的波特率。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。

可配置寄存器TCON1中的TCLKP0位选择系统时钟或系统时钟的1/12作为定时器0的时钟源。

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



**注意：**当定时器1作为波特率发生器时，写入TH1/TL1会影响波特率的准确性，因此也会引起通信出错。



MCU定时器0/1寄存器

Table 8.16 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 8.17 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 8.18** 定时器/计数器x数据寄存器 (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
<b>TL0 (8AH)</b>	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
<b>TH0 (8CH)</b>	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
<b>TL1 (8BH)</b>	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
<b>TH1 (8DH)</b>	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	<b>TLx.y, THx.y</b> <b>x=0-1, y=0-7</b>	定时器x低及高字节计数器

**Table 8.19** 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
<b>TCON1</b>	-	-	-	-	TCLKP1	TCLKP0	TC1	TC0
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
3-2	<b>TCLKPx</b> <b>x = 0,1</b>	定时器x时钟源预分频控制位 0: 选择系统时钟作为定时器x的时钟源 1: 选择系统时钟的1/12作为定时器x的时钟源
1-0	<b>TCx</b> <b>x = 0,1</b>	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



### 8.6.3 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）

定时器2的工作模式与定时器0和定时器1相似。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的工作方式

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，波特率发生器和可编程时钟输出。RCLK, TCLK和CP/RL2的组合能选择这些方式。

#### 定时器2方式选择

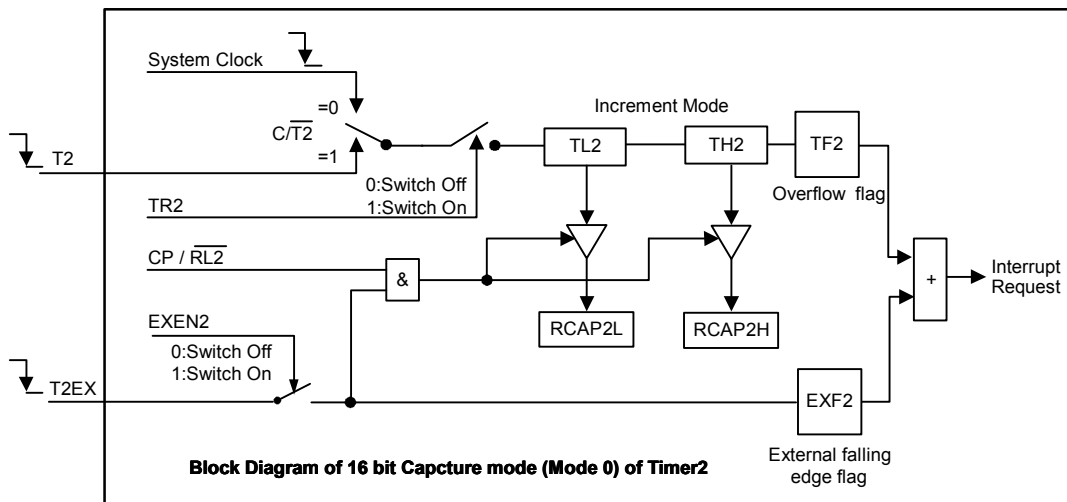
C/T2	T2OE	DCEN	TR2	CP/RL2	RCLK	TCLK	方式		
X	0	X	1	1	0	0	0	16位捕获	
X	0	0	1	0	0	0	1	16位自动重载定时器	
X	0	1	1	0	0	0			
X	0	X	1	X	1	X	2	波特率发生器	
					X	1			
0	1	X	1	X	0	0	3	只用于可编程时钟	
					1	X	3	带波特率发生器的可编程时钟输出	
					X	1			
1	1	X	1	X	X	X		不推荐使用	
X	X	X	0	X	X	X	X		定时器2停止，T2EX通路仍旧允许

#### 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IET2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IET2被允许，EXF2位也像TF2一样也产生一个中断。





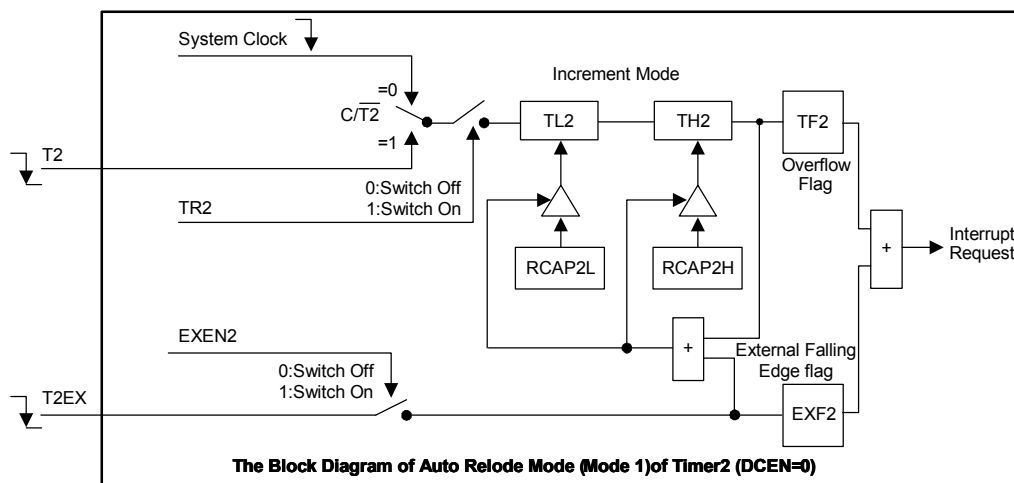
### 方式1：16位自动重载定时器

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。当设置DCEN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IET2被使能，TF2和EXF2位都能产生一个中断。

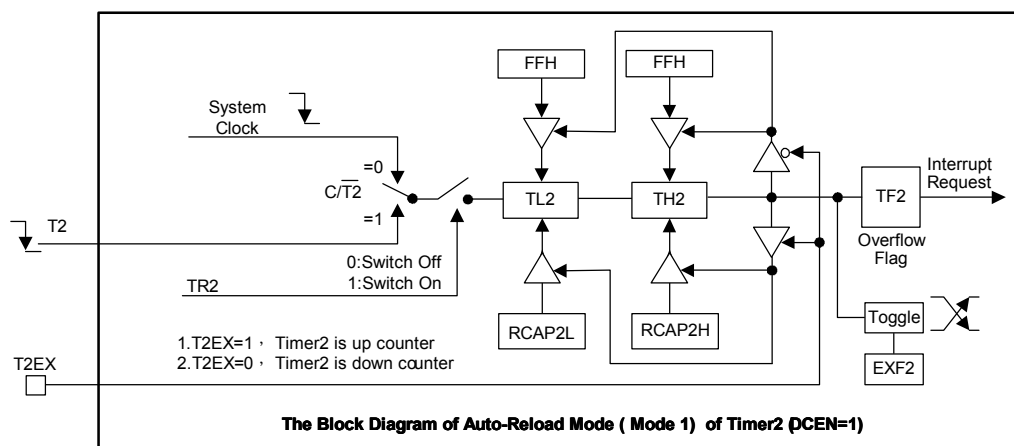


设置DCEN位允许定时器2递增计数或递减计数。当DCEN = 1时，T2EX引脚控制计数的方向，而EXEN2控制无效。

T2EX置1可使定时器2递增计数。定时器递增到0FFFFH，在溢出后设置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的值小于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，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：波特率发生器**

通过设置T2CON寄存器中的TCLK和/或RCLK选择定时器2作为波特率发生器。接收器和发送器的波特率可以不同，如果定时器2作为接收器或发送器则定时器1相应的作为另一种的波特率发生器。

设置RCLK和/或TCLK使定时器2进入波特率发生器方式，该方式与自动重载方式相似。

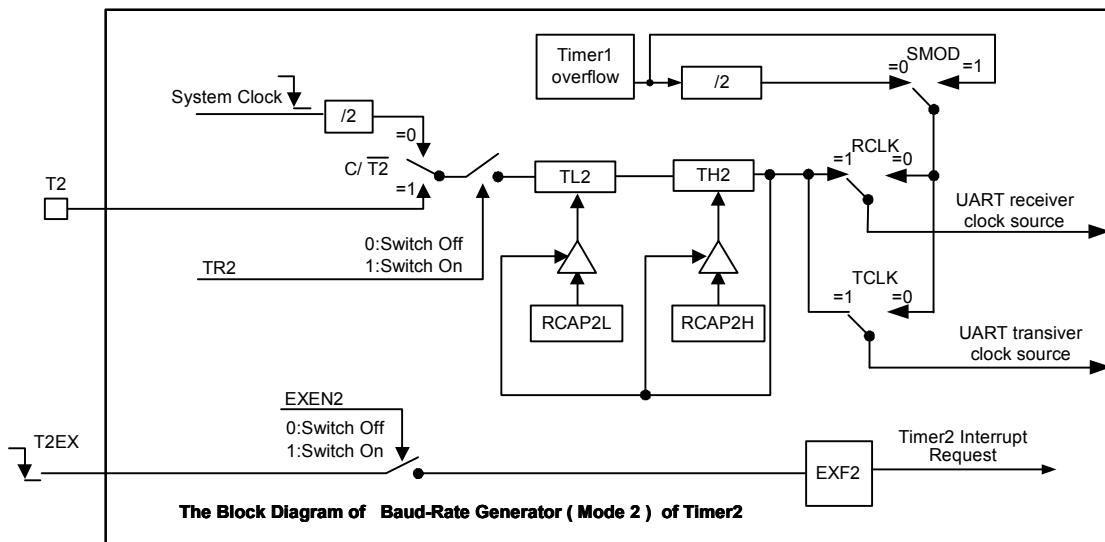
定时器2的溢出会使RCAP2H和RCAP2L寄存器中的值重载入定时器2计数器，但不会产生中断。

如果EXEN2被置1，在T2EX脚上的下降沿会置起EXF2，但不会引起重载。因此当定时器2作为波特率发生器时，T2EX可作为一个额外的外部中断。

在EUART方式1和3中的波特率由定时器2的溢出率根据下列方程式决定。

$$BaudRate = \frac{1}{2 \times 16} \times \frac{f_{SYS}}{65536 - [RCAP2H, RCAP2L]} ; \quad C/\overline{T2} = 0$$

$$BaudRate = \frac{1}{16} \times \frac{f_{T2}}{65536 - [RCAP2H, RCAP2L]} ; \quad C/\overline{T2} = 1$$



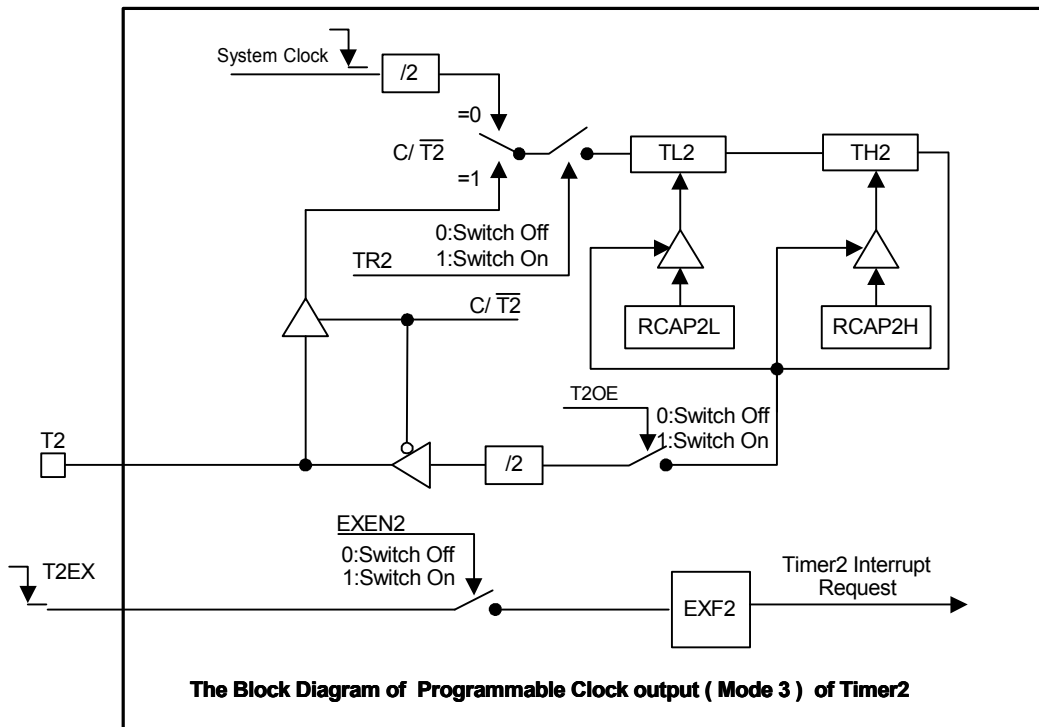


方式3: 可编程时钟输出

在这种方式中, T2输出占空比为50%的时钟:

$$\text{Clock Out Frequency} = \frac{1}{2 \times 2} \times \frac{f_{\text{sys}}}{65536 - [\text{RCAP2H}, \text{RCAP2L}]}$$

定时器2溢出不产生中断, 所以定时器2可以同时以相同频率用作波特率发生器和时钟输出。



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。
- (4) 当定时器 2 作为波特率发生器时, 写入 TH2/TL2, 写入 RCAPH2/RCAPL2 会影响波特率的准确性, 因此也会引起通信出错。



MCU定时器2寄存器

Table 8.20 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	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
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	TF2	定时器2溢出标志位 0: 无溢出 (必须由软件清0) 1: 溢出 (如果RCLK = 0和TCLK = 0, 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	RCLK	EUART0接收时钟控制位 0: 定时器1产生接收波特率 1: 定时器2产生接收波特率
4	TCLK	EUART0发送时钟控制位 0: 定时器1产生发送波特率 1: 定时器2产生发送波特率
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: 当定时器2不做为EUART时钟 (T2EX始终包括上拉电阻) 时, 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 8.21 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	-	-	-	-	-	-	T2OE	DCEN
读/写	-	-	-	-	-	-	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0

位编号	位符号	说明
1	T2OE	定时器2输出允许位 0: 设置P0.2/T2作为时钟输入或I/O端口 1: 设置P0.2/T2作为时钟输出（波特率发生器方式）
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

Table 8.22 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 (CAH)	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H (CBH)	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2 (CCH)	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2 (CDH)	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 RCAP2H.x	定时器2重载/捕获数据，x = 0 - 7
7-0	TL2.x TH2.x	定时器2高位低位计数器，x = 0 - 7



## 8.7 MCU中断

### 8.7.1 特性

- 15个中断源
- 4层中断优先级
- 程序超范围中断

SH99F100中的MCU模块有15个中断源：1个OVL NMI中断，3个外部中断（外部中断0/1/2），3个定时器中断（定时器0/1/2），2个EUART中断，ADC中断，SPI中断，APWM中断，SCM中断，通讯命令中断和通讯标志中断。

### 8.7.2 程序超范围中断（OVL）

MCU有一个不可屏蔽中断（NMI）源——程序超范围中断（OVL），其向量定位在007BH中，不可屏蔽中断用以防止CPU超出有效程序范围。为应用这个特性，用户应该用常量0xA5填满未使用的Flash ROM，如果PC超过了用户的有效程序范围，则运算代码为不存在在8051指令集中的0xA5，CPU因此获知PC已经超出了有效的程序范围，同时OVL中断发生。

不可屏蔽中断OVL享有最高优先级（除复位外），不会被其它中断源中断。同样不可屏蔽中断OVL能自身嵌套，但堆栈不会因此增加。当OVL中断发生后，其它中断仍旧被允许，如果满足设定的条件，其它中断的标志将置1。

由于OVL中断是不可屏蔽中断并且具有最高中断优先级，当产生OVL中断时，其它任何中断都被屏蔽掉，不能响应，所以用户必须处理OVL中断以保护系统免受不必要的影响。用户可以用OVL中断服务程序末端的RETI指令来修改压入栈顶的地址（因为进入OVL中断时，压入堆栈顶端的地址是无用的），这样跳出中断服务程序后，程序可以跳转到用户指定的代码，诸如复位入口或保护程序入口。

```
OVL_NMI_SERVICE:
.....
MOV     SP, #Initial_value
MOV     DPTR, #Start_or_Initial_address
PUSH    DPL
PUSH    DPH
RETI
```

#### 特别提示：

由于**OVL**中断是不可屏蔽中断并且具有最高中断优先级，当产生**OVL**中断时，其它任何中断都被屏蔽掉，不能响应，所以用户必须处理**OVL**中断以保护系统免受不必要的影响。



### 8.7.3 中断允许

任何一个中断源均可通过对寄存器IEN0和IEN1中相应的位置1或清0，实现单独允许或禁止。IEN0寄存器中还包含了一个全局允许位EA，它是所有中断的总开关。一般在复位后，所有中断允许位设置为0，所有中断被禁止。

**Table 8.23** 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	ES0	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	ES0	EUART0中断允许位 0: 禁止EUART0中断 1: 允许EUART0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 8.24 次级中断允许寄存器

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	EDFLAG	EAPWM	EDCMD	-	ES1	EX2	ESPI
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	ESCM	<b>SCM中断允许位</b> 0: 禁止SCM中断 1: 允许SCM中断
6	EDFLAG	<b>DSP至MCU通讯标志中断允许位</b> 0: 禁止通讯标志中断 1: 允许通讯标志中断
5	EAPWM	<b>APWMx周期中断允许位</b> 0: 禁止APWM中断 1: 允许APWM中断
4	EDCMD	<b>DSP至MCU通讯命令中断允许位</b> 0: 禁止通讯命令中断 1: 允许通讯命令中断
2	ES1	<b>EUART1中断允许位</b> 0: 禁止EUART1中断 1: 允许EUART1中断
1	EX2	<b>外部中断2允许位</b> 0: 禁止外部中断2 1: 允许外部中断2
0	ESPI	<b>SPI中断允许位</b> 0: 禁止SPI中断 1: 允许SPI中断



### 8.7.4 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，在中断汇总表中列出各中断标志位。

外部中断源产生外部中断INT<sub>x</sub> (x = 0/1/2) 时，如果中断为边沿触发，CPU在响应中断后，各中断标志位 (TCON寄存器的IE0/1位，EXF0寄存器的IE2位) 被硬件清0；如果中断是低电平触发，外部中断源引脚电平直接控制中断标志，而不是由片上硬件控制。

定时器0/1的计数器溢出时，TCON寄存器的TF<sub>x</sub> (x = 0, 1) 中断标志位置1，产生**定时器0/1**中断，CPU在响应中断后，标志被硬件自动清0。

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，标志必须由软件清0。

SCON寄存器的标志RI<sub>x</sub>或TI<sub>x</sub>置1时，产生**EUARTx (x = 0, 1)**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0。

ADCON寄存器的ADCIF标志位置1时，产生**ADC**中断。如果中断产生，ADCDH/ADC DL中的转换结果是有效的。如果ADC模块的连续比较功能打开，在每次转换中，如果转换结果小于比较值时，ADCIF标志位为0；如果转换结果大于比较值时，ADCIF标志位置1，ADCIF中断标志必须由软件清除。

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。

CLKCON寄存器的SCMIF标志位置1时，产生**SCM**中断，标志只能由硬件清0，软件无法清除。

APWMCON<sub>x</sub>寄存器的APWMxIF (x = 0-2) 标志位置1时，产生**APWM**中断，标志必须由软件清0。

DSP向DTMC寄存器 (这是个16bits的寄存器，DSP可读/写，MCU只读) 写入一个数据时，产生MCU的**通讯命令中断**，DIFLAG寄存器 (这是个16bits的寄存器，DSP可读/写，MCU只能清零) 的DCMDIF置1，MCU向DCMDIF位写0以清除标志。

DSP置位DIFLAG寄存器中的位DFGxIF (x = 14-0) 时，将触发MCU的**通讯标志中断**，MCU向对应的DFGxIF写0以清除标志。

**Table 8.25** 定时器x/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
<b>TCON</b>	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
<b>7, 5</b>	<b>TF<sub>x</sub></b> (x = 0, 1)	<b>定时器x溢出标志</b> 0: 定时器x无溢出 1: 定时器x溢出
<b>6, 4</b>	<b>TR<sub>x</sub></b> (x = 0, 1)	<b>定时器x启动, 停止控制</b> 0: 停止定时器x 1: 启动定时器x
<b>3, 1</b>	<b>IE<sub>x</sub></b> (x = 0, 1)	<b>外部中断x请求标志</b> 0: 无中断挂起 1: 中断挂起
<b>2, 0</b>	<b>IT<sub>x</sub></b> (x = 0, 1)	<b>外部中断x触发方式</b> 0: 低电平触发 1: 下降沿边触发





Table 8.26 外部中断标志寄存器0

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
EXF0	-	-	-	-	IT2.1	IT2.0	-	IE2
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
3-2	IT2[1:0]	外部中断2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
0	IE2	外部中断2请求标志位 0: 无中断挂起 1: 中断挂起



### 8.7.5 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汇总表中详细列出。

### 8.7.6 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。

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 8.27 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 (B8H)	-	PADCL	PT2L	PS0L	PT1L	PX1L	PT0L	PX0L
IPH0 (B4H)	-	PADCH	PT2H	PS0H	PT1H	PX1H	PT0H	PX0H
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
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 (B9H)	PSCML	PDFLAGL	PAPWML	PDCMDL	-	PS1L	PX2L	PSPIL
IPH1 (B5H)	PSCMH	PDFLAGH	PAPWMH	PDCMDH	-	PS1H	PX2H	PSPIH
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
7-0	PxxxL/H	相应中断源xxx优先级选择



### 8.7.7 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：

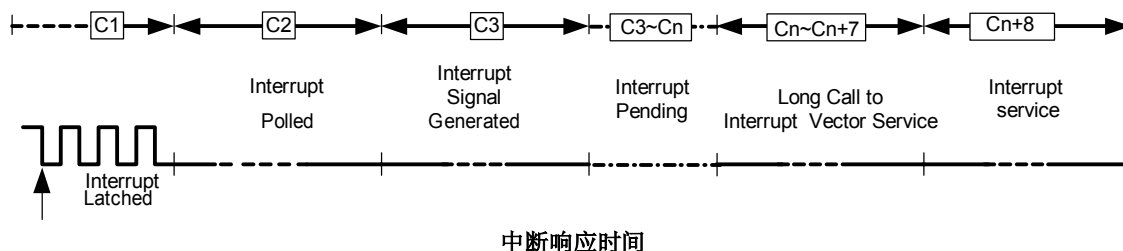
同级或更高级的优先级中断在运行中。

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。

### 8.7.8 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3个机器周期产生中断。如果响应有效且条件允许，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。



### 8.7.9 外部中断输入

MCU有3个外部中断输入。外部中断0-2分别有一个独立的中断源。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ITx = 0 (x = 0, 1) 时, 外部中断INTx (x = 0, 1) 引脚为低电平触发; 当ITx (x = 0, 1) = 1, 外部中断INTx (x = 0, 1) 为沿触发, 在这个模式中, 一个周期内INTx (x = 0, 1) 引脚上连续采样为高电平而下个周期为低电平, 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。

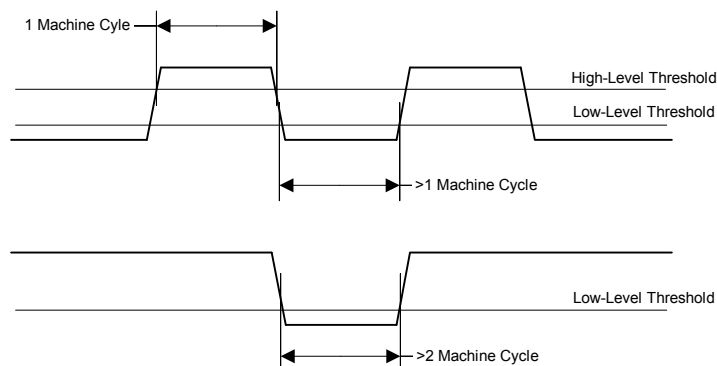
如果外部中断为下降沿触发, 外部中断源应当将中断脚至少保持1个机器周期高电平, 然后至少保持1个机器周期低电平。这样就确保了边沿能够被检测到以使IEx置1。当调用中断服务程序后, CPU自动将IEx清0。

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IEx (x = 0, 1, 2), 因为中断只与输入口电平有关。

外部中断2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MCU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0-2的中断标志位在执行中断服务程序时被自动硬件清0。



外部中断检测

### 8.7.10 中断汇总

中断源	向量地址	允许位	标记位	轮询优先级	中断号 (C语言)
Reset	0000H			0 (最高级)	
INT0	0003H	EX0	IE0	2	0
Timer0	000BH	ET0	TF0	3	1
INT1	0013H	EX1	IE1	4	2
Timer1	001BH	ET1	TF1	5	3
EUART0	0023H	ES0	RI0+TI0	6	4
Timer2	002BH	ET2	TF2+EXF2	7	5
ADC	0033H	EADC	ADCIF	8	6
SPI	003BH	ESPI	SPIF	9	7
INT2	0043H	EX2	IE2	10	8
EUART1	004BH	ES1	RI1+TI1	11	9
D2M命令中断	005BH	EDCMD+DCMDIE	DCMDIF	12	11
APWM0/1/2	0063H	EAPWM+APWMIE0/1/2	APWM0/1/2IF	13	12
D2M标志中断	006BH	EDFALG+DFLAG0EN ~DFLAG14EN	DFLAG0~DFLAG14	14	13
SCM	0073H	ESCM	SCMIF	15 (最低级)	14
OVL NMI	007BH	-	-	1	15



### 8.8 辅助脉宽调制模块APWM (Auxiliary PWM)

#### 8.8.1 特性

- 12 位精度 PWM 模块
- 提供每个 PWM 周期溢出中断
- 输出极性可选择

MCU内建三个12位PWM模块。PWM模块可以产生3路周期和占空比分别可以调整的脉宽调制波形。寄存器APWMEN用于使能3路PWM模块。APWMCONx (x = 0-2) 控制PWM模块的时钟源、输出极性、周期中断等。寄存器APWMPLx/APWMPHx (x = 0-2) 用于设置PWM模块的周期，寄存器APWMDLx/APWMDHx (x = 0-2) 用于设置PWM模块的占空比。

#### 8.8.2 寄存器

**Table 8.28** APWMx控制寄存器APWMCONx (x = 0-2) (APWM CONTROL REGISTER)

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>APWMCON0 (9DH)</b>	APWMEN0	APWMS0	APWM0CK1	APWM0CK0	-	APWMIE0	APWMIF0	APWMCH0
<b>APWMCON1 (CBH) (Bank1)</b>	APWMEN1	APWMS1	APWM1CK1	APWM1CK0	-	APWMIE1	APWMIF1	APWMCH1
<b>APWMCON2 (C3H) (Bank1)</b>	APWMEN2	APWMS2	APWM2CK1	APWM2CK0	-	APWMIE2	APWMIF2	APWMCH2
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	APWMENx	<b>APWMx使能位</b> 0: 禁止APWMx模块 1: 允许APWMx模块
6	APWMSx	<b>APWMx输出模式</b> 0: APWMx占空比期间输出高电平，占空比溢出后输出低电平 1: APWMx占空比期间输出低电平，占空比溢出后输出高电平
5-4	APWMxCK1-0	<b>APWMx时钟选择位</b> 00: MCU系统时钟/1 01: MCU系统时钟/2 10: MCU系统时钟/4 11: MCU系统时钟/8
2	APWMIEx	<b>APWMx周期中断使能位</b> 0: 禁止APWMx周期中断 1: 使能APWMx周期中断
1	APWMIFx	<b>APWMx中断标志位</b> 0: APWMx周期计数器没有溢出 1: APWMx周期计数器溢出，由硬件置1
0	APWMCHx	<b>APWMx引脚输出控制位</b> 0: APWMx输出禁止，用作I/O功能 1: APWMx输出允许

**Table 8.29** APWM0周期寄存器APWMPH/L0 (High/Low byte of APWM0 PERIOD REGISTER)

9FH, 9EH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>APWMPH0 (9FH)</b>	-	-	-	-	APWMP0.11	APWMP0.10	APWMP0.9	APWMP0.8
<b>APWMPL0 (9EH)</b>	APWMP0.7	APWMP0.6	APWMP0.5	APWMP0.4	APWMP0.3	APWMP0.2	APWMP0.1	APWMP0.0
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
<b>11-0</b>	<b>APWMP0[11:0]</b>	<b>APWM0数据寄存器</b>

APWM0的计数器计满至APWMPH/L0中的值后归零,若PWMPH/L0为0时,如果APWMS0为0,则APWM0引脚输出低电平;如果APWMS0为1,则APWM0引脚输出高电平。

**注意:** 修改寄存器APWMPH0将使得APWM0的输出在下一个周期生效。用户需先修改APWMPL0,再修改APWMPH0以修改APWM0周期。

**Table 8.30** APWM1周期寄存器APWMPH/L1 (High/Low byte of APWM1 PERIOD REGISTER)

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>APWMPH1 (CDH) (Bank1)</b>	-	-	-	-	APWMP1.11	APWMP1.10	APWMP1.9	APWMP1.8
<b>APWMPL1 (CCH) (Bank1)</b>	APWMP1.7	APWMP1.6	APWMP1.5	APWMP1.4	APWMP1.3	APWMP1.2	APWMP1.1	APWMP1.0
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
<b>11-0</b>	<b>APWMP1[11:0]</b>	<b>APWM1数据寄存器</b>

APWM1的计数器计满至APWMPH/L1中的值后归零,若PWMPH/L1为0时,如果APWMS1为0,则APWM1引脚输出低电平;如果APWMS1为1,则APWM1引脚输出高电平。

**注意:** 修改寄存器APWMPH1将使得APWM1的输出在下一个周期生效。用户需先修改APWMPL1,再修改APWMPH1以修改APWM1周期。

**Table 8.31** APWM2周期寄存器APWMPH/L2 (High/Low byte of APWM2 PERIOD REGISTER)

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>APWMPH2 (C5H) (Bank1)</b>	-	-	-	-	APWMP2.11	APWMP2.10	APWMP2.9	APWMP2.8
<b>APWMPL2 (C4H) (Bank1)</b>	APWMP2.7	APWMP2.6	APWMP2.5	APWMP2.4	APWMP2.3	APWMP2.2	APWMP2.1	APWMP2.0
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
<b>11-0</b>	<b>APWMP2[11:0]</b>	<b>APWM2数据寄存器</b>

APWM2的计数器计满至APWMPH/L2中的值后归零,若PWMPH/L2为0时,如果APWMS2为0,则APWM2引脚输出低电平;如果APWMS2为1,则APWM2引脚输出高电平。

**注意:** 修改寄存器APWMPH2将使得APWM2的输出在下一个周期生效。用户需先修改APWMPL2,再修改APWMPH2以修改APWM2周期。



**Table 8.32** APWM0占空比寄存器APWMDH/L0 (High/Low byte of APWM0 DUTY REGISTER)

AFH, AEH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
APWMDH0 (AFH)	-	-	-	-	APWMD0.11	APWMD0.10	APWMD0.9	APWMD0.8
APWMDL0 (AEH)	APWMD0.7	APWMD0.6	APWMD0.5	APWMD0.4	APWMD0.3	APWMD0.2	APWMD0.1	APWMD0.0
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
11-0	APWMD0[11:0]	<p><b>APWM0占空比控制，控制APWM0波形占空比的输出时间</b></p> <p>1. 当APWMP0 ≤ APWMD0时            如果APWMS0 = 0，则APWM0引脚输出高电平            如果APWMS0 = 1，则APWM0引脚输出低电平</p> <p>2. 当APWMD0 = 00H时            如果APWMS0 = 0，则APWM0引脚输出低电平            如果APWMS0 = 1，则APWM0引脚输出高电平</p>

**注意：**修改寄存器APWMDH0将使得APWM0的输出在下一个周期生效。用户需先修改APWMDL0，再修改APWMDH0以修改APWM0占空比。

**Table 8.33** APWM1占空比寄存器APWMDH/L1 (High/Low byte of APWM1 DUTY REGISTER)

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
APWMDH1 (CFH) (Bank1)	-	-	-	-	APWMD1.11	APWMD1.10	APWMD1.9	APWMD1.8
APWMDL1 (CEH) (Bank1)	APWMD1.7	APWMD1.6	APWMD1.5	APWMD1.4	APWMD1.3	APWMD1.2	APWMD1.1	APWMD1.0
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
11-0	APWMD1[11:0]	<p><b>APWM1占空比控制，控制APWM1波形占空比的输出时间</b></p> <p>1. 当APWMP1 ≤ APWMD1时            如果APWMS1 = 0，则APWM1引脚输出高电平            如果APWMS1 = 1，则APWM1引脚输出低电平</p> <p>2. 当APWMD1 = 00H时            如果APWMS1 = 0，则APWM1引脚输出低电平            如果APWMS1 = 1，则APWM1引脚输出高电平</p>

**注意：**修改寄存器APWMDH1将使得APWM1的输出在下一个周期生效。用户需先修改APWMDL1，再修改APWMDH1以修改APWM1占空比。



**Table 8.34** APWM2占空比寄存器APWMDH/L2 (High/Low byte of APWM2 DUTY REGISTER)

	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>APWMDH2 (C7H) (Bank1)</b>	-	-	-	-	APWMD2.11	APWMD2.10	APWMD2.9	APWMD2.8
<b>APWMDL2 (C6H) (Bank1)</b>	APWMD2.7	APWMD2.6	APWMD2.5	APWMD2.4	APWMD2.3	APWMD2.2	APWMD2.1	APWMD2.0
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
11-0	APWMD2[11:0]	<p><b>APWM2占空比控制，控制APWM2波形占空比的输出时间</b></p> <ol style="list-style-type: none"> <li>当APWMP2 ≤ APWMD2时 如果APWMS2 = 0，则APWM2引脚输出高电平 如果APWMS2 = 1，则APWM2引脚输出低电平</li> <li>当APWMD2 = 00H时 如果APWMS2 = 0，则APWM2引脚输出低电平 如果APWMS2 = 1，则APWM2引脚输出高电平</li> </ol>

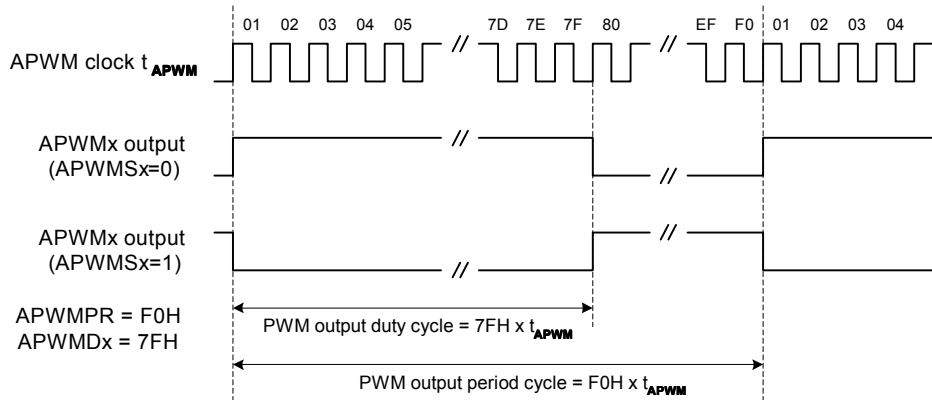
**注意：**修改寄存器APWMDH2将使得APWM2的输出在下一个周期生效。用户需先修改APWMDL2，再修改APWMDH2以修改APWM2占空比。



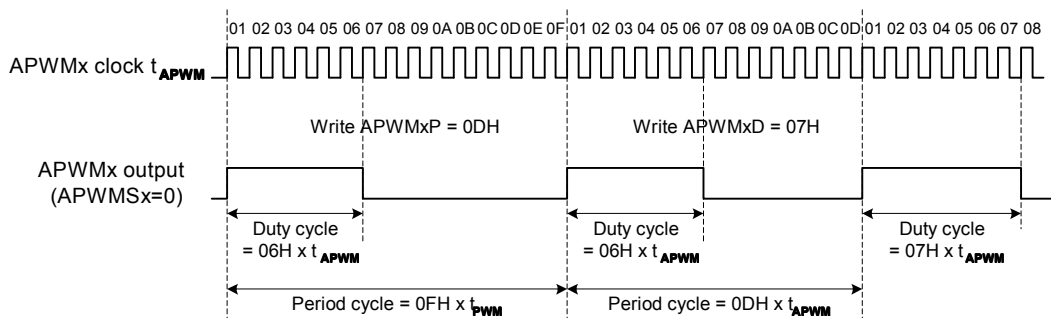


**注意事项:**

- 1. APWMENx位控制APWMx模块打开。
- 2. APWMCHx (x = 0-2) 位能选择 P1.4/P1.5/P1.6端口是作为I/O端口还是PWM输出端口。
- 3. 在IEN1寄存器中的EPWM位和APWMCONx寄存器中的APWMIEx位允许/禁止APWM中断。
- 4. 如果APWMENx置1, PWM模块打开, 但APWMCHx (x = 0-2) = 0, APWMx输出关闭, 此时APWMx模块可以用作一个 12bit timer, 此时如果中断控制寄存器IEN1的EPWM位置1且APWMIEx = 1, 则APWMx中断照样发生。



**APWM输出范例**



**APWM输出周期或者占空比改变范例**



### 8.9 增强型通用异步收发器 (EUART)

#### 8.9.1 特性

- MCU模块带有2个EUART，兼容传统8051
- 波特率可选择为系统时钟分频或定时器1/2的溢出率，EUART1自身带有一个波特率发生器，无需占用定时器
- 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

#### 8.9.2 EUART工作模式

EUART有4种工作模式。在通信之前用户必须先初始化SCON。选择模式和波特率。如果使用模式1或模式3应先初始化定时器1或定时器2。

在所有四种模式中，任何将SBUF作为目的寄存器的指令都会启动发送，在模式0中由设置RI = 0和REN = 1启动接收。这会在TxD引脚上产生一个时钟，然后在RxD引脚上移出8位数据。在其它模式中，如果REN = 1则通过接收起始位来启动。通过发送开始位，外部发送器开始通信。

#### EUART模式列表

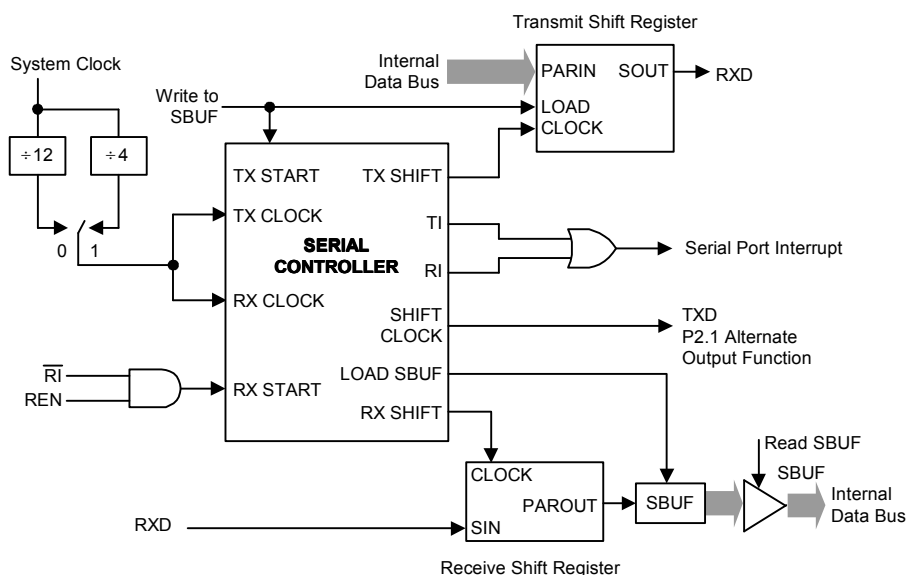
SM0	SM1	模式	类型	波特时钟	帧长度	开始位	中止位	第9位
0	0	0	同步	$f_{SYS}/(4或12)$	8位	无	无	无
0	1	1	异步	定时器1或2的溢出率/(16或32)	10位	1	1	无
1	0	2	异步	$f_{SYS}/(32或64)$	11位	1	1	0, 1
1	1	3	异步	定时器1或2的溢出率/(16或32)	11位	1	1	0, 1

#### 模式0: 同步, 半双工通讯

模式0提供带内部装置的同步通信。在这个模式下，串行数据的收发都在RxD线上。TxD用作发送移位时钟。无论作为接收器还是发送器，TxD时钟都可由MCU提供。因此这个模式是一种半双工模式的串行通信。每次发送或接收都为8位，LSB（最低位）在前。

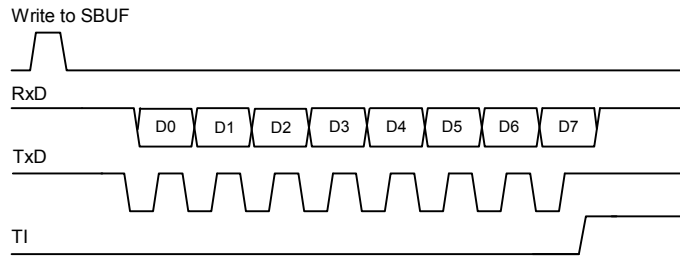
模式0波特率固定为系统时钟的1/12或1/4。波特率由SM2位（SCON.5）决定。当设置SM2位为0时，串行通讯波特率为时钟1/12。当设置为1时，波特率为时钟的1/4。

模块功能框图如下图所示。数据通过RxD线移进和移出串行端口，TxD线用来输出移位时钟，移位时钟用来控制数据移进和移出MCU。



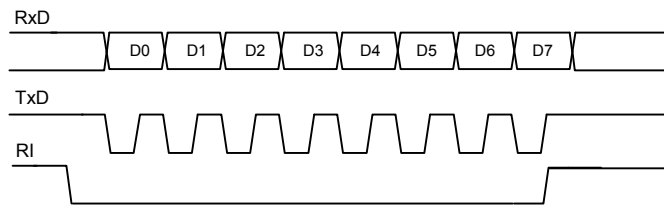


任何将SBUF作为目标寄存器的写操作都会启动发送。下一个系统时钟Tx控制块开始发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置0。当移位寄存器中的所有8位都发送后，Tx控制模块停止发送操作，然后在下一个系统时钟的上升沿将TI置1（SCON.1）。



Send Timing of Mode 0

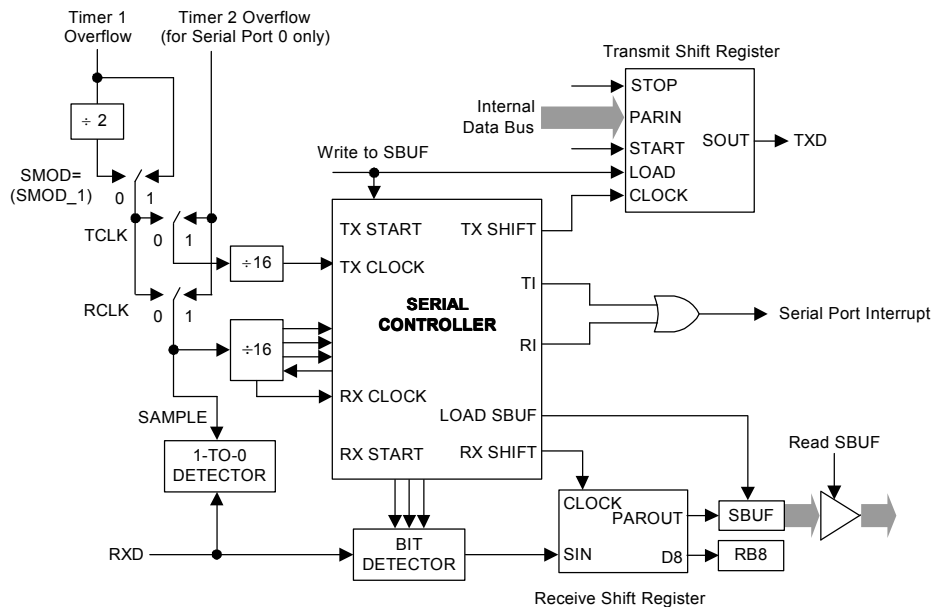
REN（SCON.4）置1和RI（SCON.0）清0初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有8位都接收到接收移位寄存器中后，Rx控制块停止接收，然后在下一个系统时钟的上升沿上RI置1，直到被软件清0才允许接收。



Receive Timing of Mode 0

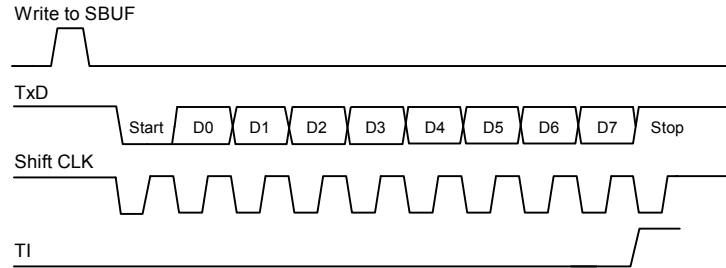
**模式1：8位EUART，可变波特率，异步全双工**

模式1使用全双工异步模式。发送（通过TxD）和接收（通过RxD）都是10位。10位由1个开始位（逻辑0），8个数据位（低位在先）和1个中止位（逻辑1）组成。接收时，这8个数据位存储在SBUF中而中止位储存在RB8（SCON.2）中。模式1中的波特率是可变的。串行收发波特率可编程为定时器1溢出频率的1/16或1/32，或是定时器2溢出频率的1/16。（请见波特率章节）模块功能框图如下图所示。





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1。

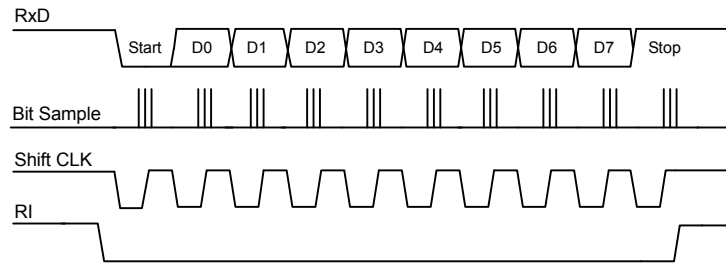


Send Timing of Mode 1

只有REN位置1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。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1，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1。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。用户必须用软件清除RI，然后才能再次接收。

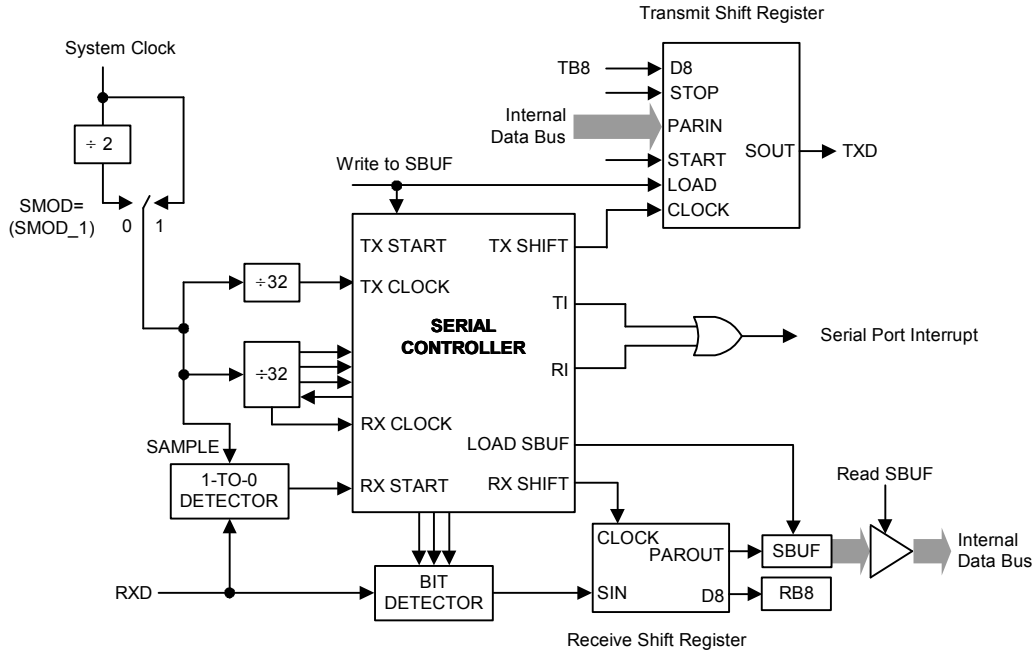


Receive Timing of Mode 1

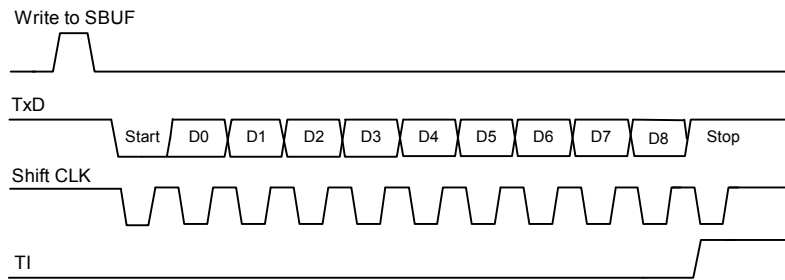


模式2: 9位EUART, 固定波特率, 异步全双工

模式2使用全双工异步模式。发送(通过TxD)和接收(通过RxD)都是11位。模块功能框图如下所示。11位由1个开始位(逻辑0), 8个数据位(低位在先), 1个可编程的第9数据位和1个中止位(逻辑1)组成。模式2支持多处理器通信和硬件地址识别(请见多处理器通信)。当数据被发送时, 第9数据位(SCON中的TB8)可以指派0或1, 例如, 在PSW中奇偶标志位P或用作多处理器通信中的数据/地址标志。当接收到数据时, 第9数据位进入RB8而中止位不保留。PCON寄存器中的SMOD位决定了波特率为系统工作频率的1/32或1/64。



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, 然后是第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1。



Send Timing of Mode 2

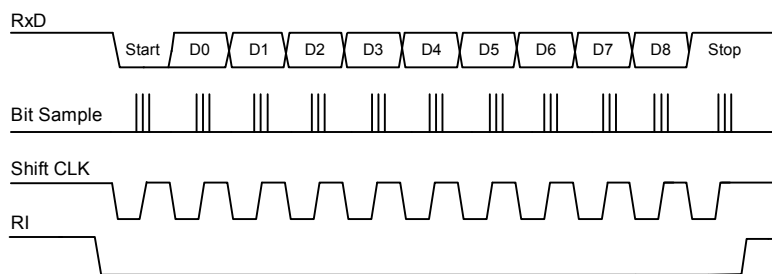


只有REN位置1时才允许接收。当RxD引脚检测到下降沿时串口开始接收串行数据。为此，CPU对RxD不断采样，采样速率为波特率的16倍。当检测下降沿时，16分频计数器立即复位。这有助于16分频计数器与RxD引脚上的串行数据位同步。16分频计数器把每一位的时间分为16个状态，在第7、8、9状态时，位检测器对RxD端的电平进行采样。为抑制噪声，在这3个状态采样中至少有2次采样值一致数据才被接收。如果所接收的第一位不是0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待RxD引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9个数据位和1个停止位移入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

- 1. RI = 0
- 2. SM2 = 0或者接收的第9位 = 1，且接收的字节符合实际从机地址。

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1。否则接收的数据帧会丢失。

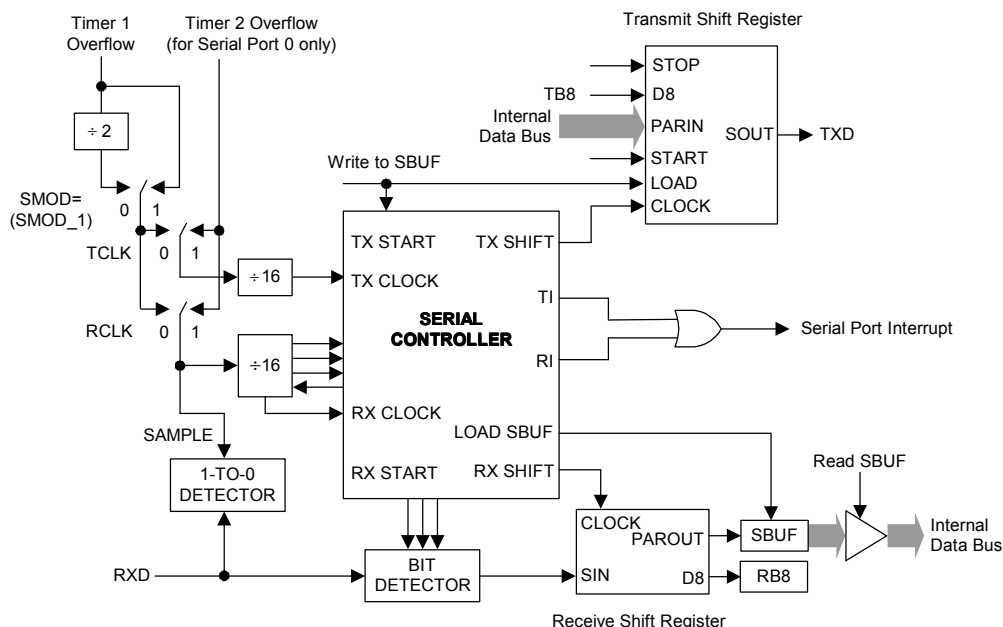
在停止位的当中，接收器回到寻找RxD引脚上的另一个下降沿。用户必须用软件清除RI，然后才能再次接收。



Receive Timing of Mode 2

### 模式3: 9位EUART, 可变波特率, 异步全双工

模式3使用模式2的传输协议以及模式1的波特率产生方式。发送（通过TxD）和接收（通过RxD）都是11位。模块功能框图如下所示。11位由1个开始位（逻辑0），8个数据位（低位在先），1个可编程的第9数据位和1个中止位（逻辑1）。波特率由定时器1或定时器2溢出产生，支持多处理器通讯和硬件地址识别。





### 8.9.3 波特率发生

在模式0中，波特率固定为系统频率的1/12或1/4，由SM2位决定。当设置为0时，串行端口在振荡器的1/12下运行。当设置为1时，串行端口在振荡器的1/4下运行。

在模式1和模式3下，波特率取决于定时器1/2的溢出速率，如同下面的公式所示。EUART的模式2即能使用定时器1工作（8位自动重载入计数器/定时器），也能使用定时器2工作（波特率发生器模式）来产生波特率（**注意必须分别选择Tx和Rx时钟**）。每次定时器从最大值溢出时（定时器1最大值为FFH，定时器2最大值为FFFFH），一个时钟发送到波特率逻辑器。同时定时器1模式0和模式1都能用作波特率发生器，但使用起来颇为不便，因此不推荐使用。

分别设置TCLK（T2CON.4）和RCLK（T2CON.5）位来选择定时器2作为TX和RX的波特时钟源（请见定时器）。无论设置TCLK或RCLK为逻辑1，定时器2都不得成为波特率发生器模式。如果TCLK或RCLK为逻辑0，定时器1为Tx或Rx电路的波特率时钟源。

模式1和模式3波特率公式如下所示，其中TH1是定时器1的8位重载入寄存器，SMOD的EUART波特率加倍器（PCON.7）和[RCAP2H, RCAP2L]是定时器2的16位重载入寄存器，T1CLK是定时器1的时钟源，而T2CLK是定时器2的时钟源。

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{T1CLK}{256 - TH1}, \text{ 波特率使用定时器1, 模式2}$$

$$BaudRate = \frac{1}{2 \times 16} \times \frac{T2CLK}{65536 - [RCAP2H, RCAP2L]}, \text{ 波特率使用定时器2, 系统时钟}$$

$$BaudRate = \frac{1}{16} \times \frac{T2CLK}{65536 - [RCAP2H, RCAP2L]}, \text{ 波特率使用定时器2, T2时钟输入}$$

在模式2中，波特率固定为系统时钟的1/32或1/64，由SM2位（PCON.7）中决定。当SM2位为0时，串行端口在系统时钟的1/64下运行。当SM2位置1时，串行端口在系统时钟的1/32下运行。

$$BaudRate = 2^{SMOD} \times \left( \frac{SYSCLK}{64} \right)$$



### 8.9.4 多机通讯

#### 软件地址识别

模式2和模式3有一个专门的应用领域，即多处理器通讯。在这两个模式下，接收的是9位数据，第9位进到RB8中，然后再来一位中止位。串行口可以这样来编程：当接收到停止位时，只有在RB8 = 1的条件下，串行口中断才会有效（请求标志RI置1）。要使串行口具有这一特点，可以将SCON寄存器的SM2位置1。

在多处理机系统中，以如下所述来利用这一特点。当主处理器想发送一数据块给几个处理机中的一个时，它先送出一地址字节，以辨认目标从机。地址字节与数据字节可用第9数据位来区别，前者的第9位为1，后者的第9位为0。

SM2 = 1，从机不会被一个数据字节中断。然而，地址字节可以中断所有从机，这样，使每一台从机都检查一下所接收到的字节，以判别自己是不是目标从机。被寻址的从机清除它的SM2位，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1。没有被寻址的从机，则维持它们的SM2 = 1，这些从机将不理睬进入到串行口的数据字节，继续做它们自己的事，直到一个新的地址字节的来临。

**注意：**模式0中，SM2用来选择波特率。模式1中，SM2用来检测停止位是否有效，如果SM2 = 1，接收中断不会响应除非接收到一个有效的停止位。

#### 自动（硬件）地址识别

在模式2和模式3中，SM2置1将使EUART在如下状态下运行：当1个停止位被接收，如果载入RB8的第9数据位为1（地址字节）并且接收到的数据字节符合EUART的从机地址，EUART产生一个中断。紧接着接收中断后，从机将SM2清零，使能接收后续数据字节的中断。

在9位模式下要求第9个信息位为1以表明该信息内容是地址而非数据。当主机要发送系列数据给一个从机时，它必须首先发送目标从机的地址。所有从机在等待接收地址字节时，为了确保在接收地址字节时产生中断，SM2位必须置1。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，地址比较通过硬件完成而不是软件。

中断产生后，地址相匹配的从机清零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。一旦全部信息接收完毕，地址匹配的从机SM2置1，忽视所有传送，直到接收到下一个地址字节。

使用自动地址识别特性时，主机通过调用给定的从机地址选择与一个（或多个）从机通信。使用广播地址可以联系所有的从机。

有两个特殊功能寄存器用来定义从机地址SADDR和从机地址掩码SADEN。从机地址是一个8位的值存于SADDR寄存器中。

SADEN用于定义SADDR内哪几位需要使用而哪几位是无关位，如果SADEN中某一位为0，则SADDR中相应位的被忽略，如果SADEN中某一位置1，则SADDR中相应位的将用于得到给定的从机地址。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。使用给定地址可以识别多个从机而排除另外的从机。

记忆码	从机1	从机2
SADDR	10100100	10100111
SADEN (0掩码)	11111010	11111001
给定地址	10100x0x	10100xx1
广播地址（或）	1111111x	11111111

从机1和从机2给定地址的最低位是不同的。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）。类似的，从机1的位1为0，从机2的位1被忽略。因此，只与从机2通讯时，主机必须发送位1为1的地址（10100011）。如果主机希望同时与两从机通讯，则位0为1，位1为0，位3被两从机都忽略，此时有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的逻辑或，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00h，这两个结果设定了给定地址和广播地址为XXXXXXXX（所有位都被忽略）。这有效地去除了多处理机通讯的特性，禁止了自动寻址模式。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识别地址的多机通讯。





### 8.9.5 帧出错检测

当寄存器PCON中的SSTAT位设置为逻辑1时，帧出错检测功能是有有效的。3个错误标志位被置1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才可以访问状态位（FE0, RXOVR0和TXCOL0），SSTAT位为逻辑0时可以访问模式选择位（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（在寄存器SCON中的TXCOL位）显示为“1”。如果发生了冲突，新数据会被忽略，不能被写入发送缓冲器。

#### 接收超限

如果在接收缓冲器中的数据未被读取之前，就将新的数据位锁存入接收缓冲器，那么接收超时位（在SCON寄存器中的RXOVR位）显示为“1”。如果发生了接收超限，接收缓冲器中原来的数据将丢失。

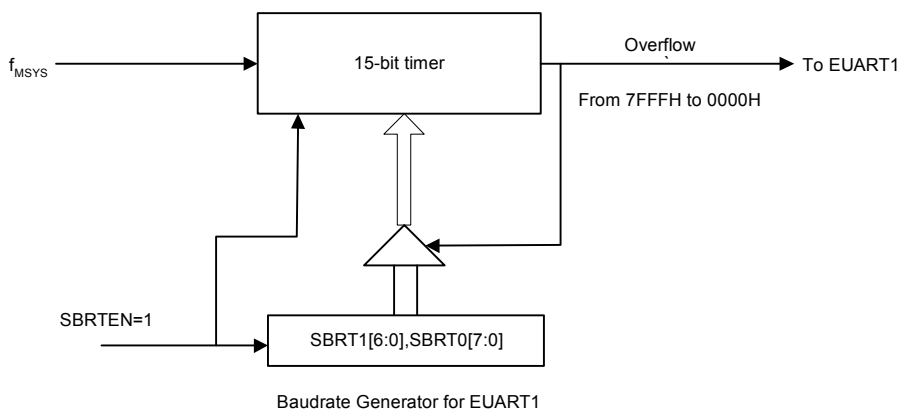
如果检测到一个无效（低）中止位，那么帧出错位（在寄存器SCON中的FE）显示为“1”。

#### 暂停检测

当连续检测到11个位都为低电平时时，则认为检测到一个间隔。由于间隔条件同样满足帧错误条件，因此检测到间隔时也会报告帧错误。一旦检测到间隔条件，UART将进入空闲状态并一直保持，直至接收到有效停止位（RxD线上出现上升沿）。

### 8.9.6 关于EUART1

EUART1自身带有一个波特率发生器，除此之外与EUART0类似。换言之，EUART1不能使用Timer1/2作为波特率发生器。实际上，EUART1的波特率发生器就是一个15位向上计数器。



由图得到，波特率发生器的溢出率为  $SBRT_{overflowrate} = \frac{F_{sys}}{32768 - SBRT}$ ， $SBRT = [SBRT1, SBRT0]$ 。

因此，EUART1在各模式下的波特率计算公式如下。

**模式0：**EUAT1的工作方式与EUAR0相同，波特率与EUART0一致。

**在模式1和模式3中，**波特率固定为自带波特率发生器溢出率的1/16，即  $BaudRate = \frac{SBRT_{overflowrate}}{16}$

**模式2：**工作方式与EUART0相同，波特率计算公式为  $BaudRate = 2^{SMOD} \times \frac{F_{sys}}{64}$



**EUART寄存器**

**EUART0相关SFR**

**Table 8.35** EUART0的控制和状态位在SCON寄存器，说明如下。

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
<b>SCON</b>	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
<b>7-6</b>	<b>SM[0:1]</b>	<b>EUART0串行模式控制位，SSTAT = 0</b> 00: 模式0，同步模式，固定波特率（默认） 01: 模式1，8位异步模式，可变波特率 10: 模式2，9位异步模式，固定波特率 11: 模式3，9位异步模式，可变波特率
<b>7</b>	<b>FE</b>	<b>SSTAT = 1</b> ，该位表示帧错误。该位置 <b>1</b> ，表示帧错误发生，该位只能软件清零
<b>6</b>	<b>RXOV</b>	<b>SSTAT = 1</b> ，该位表示接收超限。该位置 <b>1</b> ，表示接收超限发生，该位只能软件清零
<b>5</b>	<b>SM2/TXCOL</b>	<b>SSTAT = 0</b> ，在模式 <b>2</b> 和 <b>3</b> 中，该位控制多处理器通讯使能 在模式 <b>2</b> 或 <b>3</b> 中，SM2 = 1，如果接收到的第9数据位为0，RI不产生中断 在模式 <b>1</b> 中，SM2 = 1，RI不产生中断，直到收到一个有效停止位 在模式 <b>0</b> 中，SM2 = 0，波特率是系统时钟的1/12；SM2 = 1，波特率是系统时钟的1/4 <b>SSTAT = 1</b> ，该位发送冲突标志，该位置 <b>1</b> ，表示发送冲突发生，只能软件清零
<b>4</b>	<b>REN</b>	<b>EUART接收器允许控制位</b> 0: 接收禁止（默认） 1: 接收允许
<b>3</b>	<b>TB8</b>	发送位 <b>8</b> ，第 <b>9</b> 数据位在模式 <b>2</b> 和 <b>3</b> 中被发送
<b>2</b>	<b>RB8</b>	接收位 <b>8</b> ，第 <b>9</b> 数据位在模式 <b>2</b> 和 <b>3</b> 中被接收 模式 <b>1</b> 中，SM2 = 0，RB8为被接收的停止位 模式 <b>0</b> 中，RB8未用
<b>1</b>	<b>TI</b>	<b>EUART的传送中断标志</b> 0: 由软件清除（默认） 1: 在模式 <b>0</b> 下的第8位结束或在其它模式下的停止位开始时由硬件置1
<b>0</b>	<b>RI</b>	<b>EUART0的接收中断标志</b> 0: 由软件清除（默认） 1: 在模式 <b>0</b> 下的第8位结束，或在其它模式下的停止位期间由硬件置1

**Table 8.36** EUART0数据寄存器

接收到的数据和传送的数据都存储到特殊功能寄存器SBUF中，物理上，它对应着两个寄存器，即发送寄存器和接收寄存器。写SBUF，就是修改发送寄存器，读SBUF，就是读接收寄存器。

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
<b>SBUF</b>	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
<b>7-0</b>	<b>SBUF[7:0]</b>	两个寄存器共享一个SFR地址：传送移位寄存器和接收锁存器 SBUF的写入将发送字节到传送移位寄存器中，然后开始传输 SBUF的读取返回到接收锁存器中的内容



**Table 8.37** EUART0波特率加速器

SMOD位（PCON.7）控制模式1/2/3波特率的加速，如果该位被置位，波特率加倍。

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
<b>PCON</b>	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
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	<b>SMOD</b>	<b>波特率加倍器</b> 若使用定时器1作为波特率发生器，在方式1和3中置1，EUART的波特率会加倍 如果在方式2中置1，EUART的波特率会加倍
6	<b>SSTAT</b>	<b>SCON[7:5]功能选择</b> 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	<b>SSTAT1</b>	<b>SCON1[7:5]功能选择</b> 0: SCON1[7:5]工作方式如同SM10, SM11, SM12 1: SCON1[7:5]工作方式如同FE1, RXOV1, TXCOL1
3-2	<b>GF[1:0]</b>	用于软件的通用标志位
1	<b>PD</b>	掉电模式控制位
0	<b>IDL</b>	空闲模式控制位

**Table 8.38** EUART0从属地址及地址掩码寄存器

自动地址识别功能使用两个寄存器，从机地址SADDR0和掩码地址SADEN0。实际的从机地址由SADEN0寄存器的位对SADDR0相应位的掩码处理获得。如果SADEN0的某一位为0，则SADDR0中相应的位被忽略。

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
<b>SADDR0 (9AH)</b>	SADDR0.7	SADDR0.6	SADDR0.5	SADDR0.4	SADDR0.3	SADDR0.2	SADDR0.1	SADDR0.0
<b>SADEN0 (9BH)</b>	SADEN0.7	SADEN0.6	SADEN0.5	SADEN0.4	SADEN0.3	SADEN0.2	SADEN0.1	SADEN0.0
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	<b>SADDR0[7:0]</b>	<b>UART从机地址</b> SADDR0寄存器用于定义UART从机地址
7-0	<b>SADEN0[7:0]</b>	<b>地址屏蔽</b> SADEN0是一个位掩码决定SADDR中的的哪个位对应接收地址 0: 相应的SADDR寄存器中位被忽略（默认） 1: 相应的SADDR寄存器中位对应接收地址



**EUART1相关SFR**

**Table 8.39** EUART1控制和状态寄存器

9CH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>SCON1</b>	SM10 /FE1	SM11 /RXOV1	SM12 /TXCOL1	REN1	TB18	RB18	TI1	RI1
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
<b>7-6</b>	<b>SM1[0:1]</b>	<b>EUART1串行方式控制位, SSTAT1 = 0</b> 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
<b>7</b>	<b>FE1</b>	<b>EUART1帧出错标志, 当FE1位被读时, SSTAT1位必须被设置为1</b> 0: 无帧出错, 由软件清0 1: 发生帧出错, 由硬件置1
<b>6</b>	<b>RXOV1</b>	<b>EUART1接收完毕标志位, 当RXOV1位被读时, SSTAT1位必须被设置为1</b> 0: 无接收溢出, 由软件清0 1: 接收溢出, 由硬件置1
<b>5</b>	<b>SM12</b>	<b>EUART1多处理机通讯允许位 (第9位“1”校验器), SSTAT1 = 0</b> 0: 在方式0下, 波特率是系统时钟的1/12 在方式1下, 禁止停止位确认检验, 停止位将置RI1为1产生中断 在方式2和3下, 任何字节都会置RI1为1产生中断 1: 在方式0下, 波特率是系统时钟的1/4 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RI1为1产生中断 在方式2和3下, 只有地址字节 (第9位 = 1) 能置RI1为1产生中断
<b>5</b>	<b>TXCOL1</b>	<b>EUART1发送冲突标志位, 当TXCOL1位被读时, SSTAT1位必须被设置为1</b> 0: 无发送冲突, 由软件清0 1: 有发送冲突, 由硬件置1
<b>4</b>	<b>REN1</b>	<b>EUART1接收器允许位</b> 0: 接收禁止 1: 接收允许
<b>3</b>	<b>TB18</b>	在EUART1的方式2和3下发送的第9位, 由软件置1或清除
<b>2</b>	<b>RB18</b>	在EUART1的方式1, 2和3下接收的第9位 在方式0下, 不使用RB18 在方式1下, 如果接收中断发生, RB18是接收到的停止位 在方式2和3下, RB18是接收到的第9位
<b>1</b>	<b>TI1</b>	<b>EUART1的传送中断标志位</b> 0: 由软件清0 1: 由硬件置1, 在方式0下的第8位最后, 或在其它方式下的停止位开始
<b>0</b>	<b>RI1</b>	<b>EUART1的接收中断标志位</b> 0: 由软件清0 1: 由硬件置1, 在方式0下的第8位最后, 或在其它方式下的停止位开始



Table 8.40 EUART1数据缓冲器寄存器

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
<b>SBUF1</b>	SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0
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	<b>SBUF1[7:0]</b>	SFR访问两个寄存器：一个移位寄存器和一个接收锁存寄存器 SBUF1的写入将发送字节到移位寄存器中，然后开始传输 SBUF1的读取返回接收锁存器中的内容

Table 8.41 EUART1从属地址和地址屏蔽寄存器

DAH-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
<b>SADDR1 (DAH)</b>	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0
<b>SADEN1 (DBH)</b>	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0
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	<b>SADDR1[7:0]</b>	<b>SFR SADDR1</b> 定义EUART1的从机地址
7-0	<b>SADEN1[7:0]</b>	<b>SFR SADEN1</b> 是一个位屏蔽寄存器，决定检验 <b>SADDR1</b> 的哪些位对应接收地址 0：在 <b>SADDR1</b> 中的相应位被忽略 1：SADDR1中的相应位被检验是否对应接收地址

Table 8.42 EUART1波特率发生器寄存器

A4H, 9CH	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
<b>SBRT1 (A4H)</b>	SBRTEN	SBRT1.6	SBRT1.5	SBRT1.4	SBRT1.3	SBRT1.2	SBRT1.1	SBRT1.0
<b>SBRT0 (9CH)</b>	SBRT0.7	SBRT0.6	SBRT0.5	SBRT0.4	SBRT0.3	SBRT0.2	SBRT0.1	SBRT0.0
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	<b>SBRTEN</b>	<b>EUART1</b> 波特率发生器使能控制 0：关闭（默认） 1：打开
6-0	<b>SBRT1[6:0]</b>	<b>EUART1</b> 波特率发生器计数器高7位
7-0	<b>SBRT0[7:0]</b>	<b>EUART1</b> 波特率发生器计数器低8位



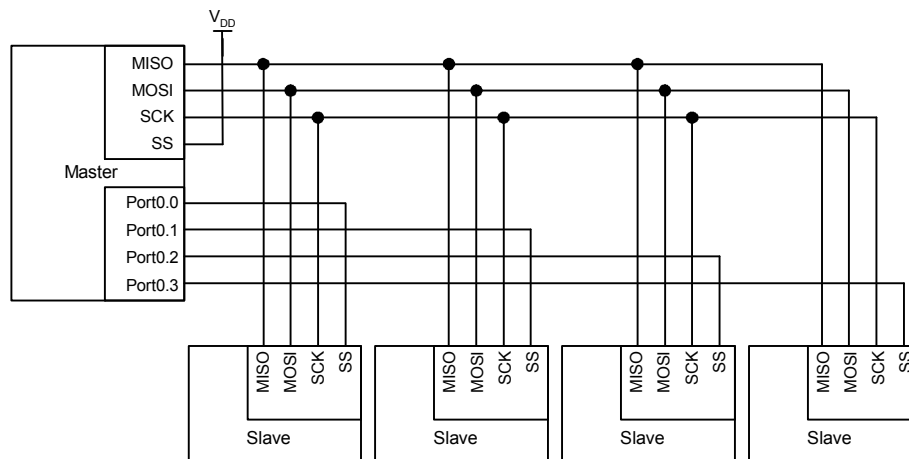
## 8.10 串行外部设备接口 (SPI)

### 8.10.1 特性

- 全双工，三线同步传输
- 主从机操作
- 6 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。



### 8.10.2 信号描述

#### (1) 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。

#### (2) 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 ( $\overline{SS}$ 引脚为高电平)，从设备的MISO引脚处于高阻状态。

#### (3) 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 ( $\overline{SS}$ 引脚为高电平)，SCK信号被此从设备忽略。

#### (4) 从设备选择引脚 ( $\overline{SS}$ )

每个从属外围设备由一个从选择引脚 ( $\overline{SS}$ 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， $\overline{SS}$ 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， $\overline{SS}$ 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 $\overline{SS}$ 引脚选择其作为通讯目标。

**注意：**当CPHA = '0'时， $\overline{SS}$ 引脚产生下降沿表示启动发送。

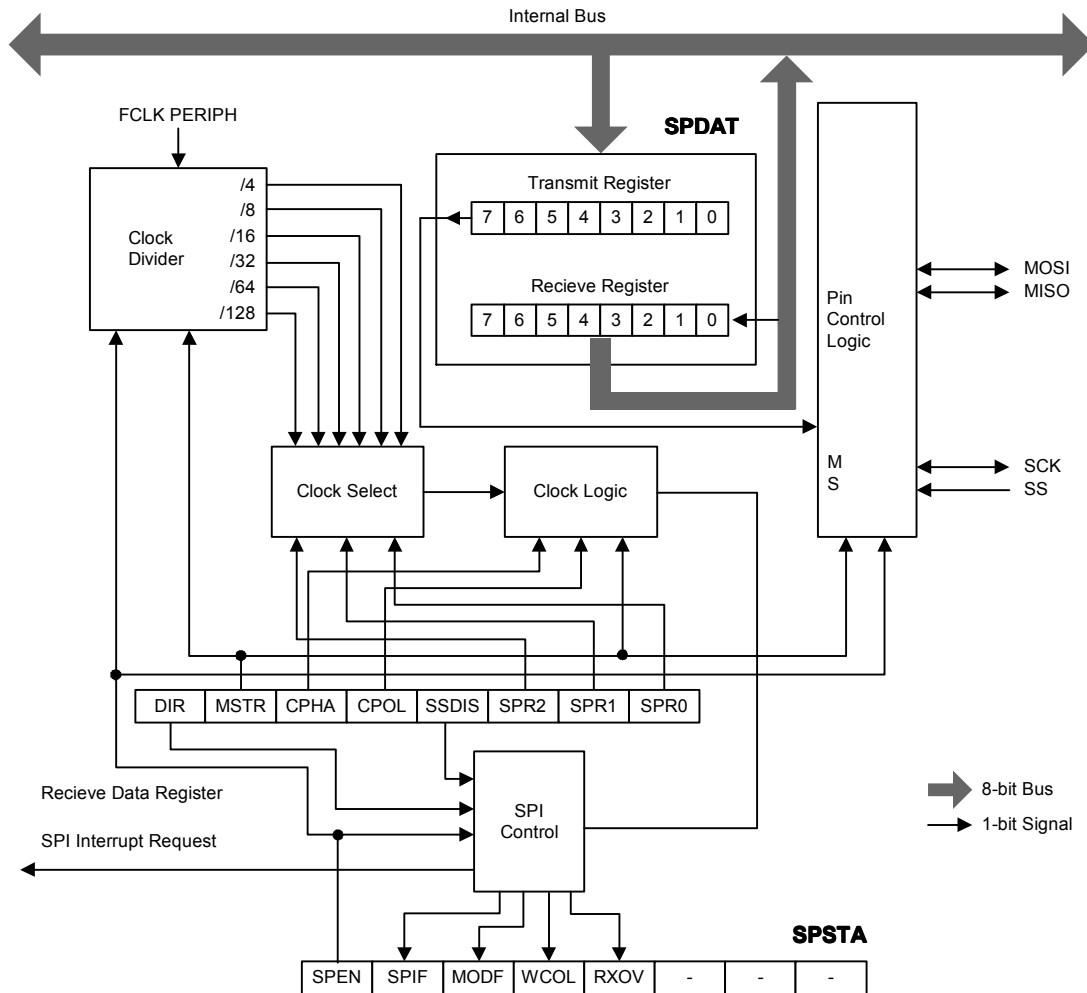


### 8.10.3 波特率

在主模式下，SPI的波特率有六种可选择的频率，分别是内部时钟的4，8，16，32，64或128分频，可以通过设定SPCON寄存器的SPR[2:0]位进行选择。

### 8.10.4 功能描述

下图所示是SPI模块的详细结构。



SPI模块框图

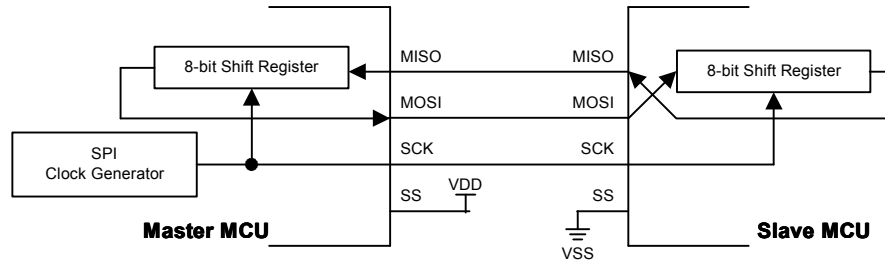


### 8.10.5 工作模式

SPI可配置为主模式或从属模式中的一种。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。

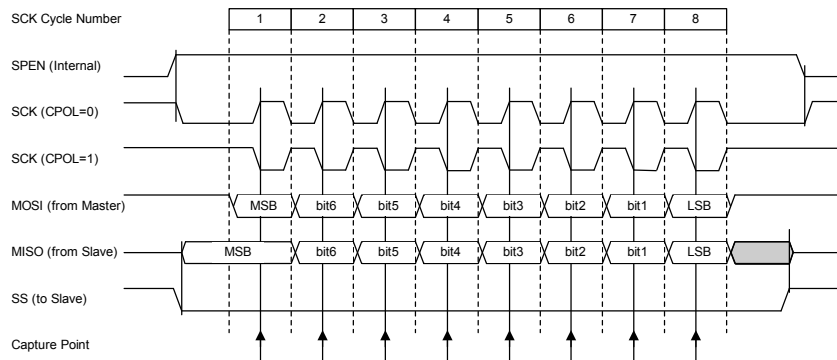
SPI从设备不能启动数据传送，所以SPI从设备必须在主设备开始一次新的数据传送之前将要传送的数据写入发送移位寄存器。如果从设备在第一次开始发送之前未写入数据，从设备将传送“0x00”字节给主设备。如果写SPDAT操作发生在传送过程中，那么SPI从设备的WCOL标志位置1，即如果传送移位寄存器已经含有数据，SPI从设备的WCOL位置1，表示写SPDAT冲突。但是移位寄存器的数据不受影响，传送也不会被中断。





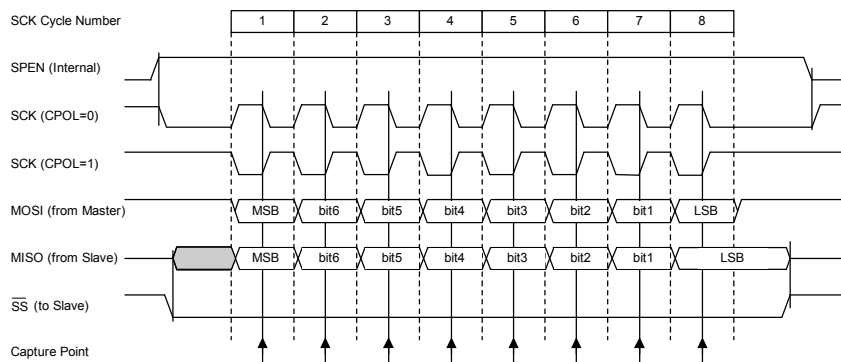
### 8.10.6 传送形式

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。



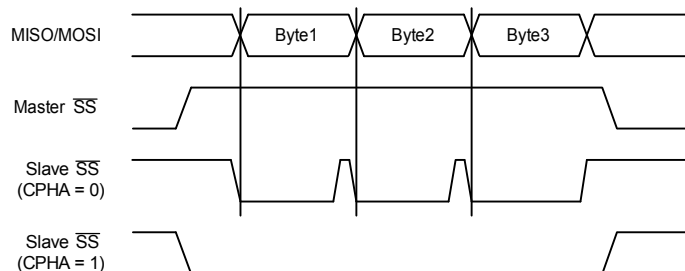
数据传送形式 (CPHA = 0)

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/ $\overline{SS}$ 时序

**注意:** 当SPI用作从设备模式，且SPCON寄存器的CPOL位清0, P4.3/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。



### 8.10.7 出错检测

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)

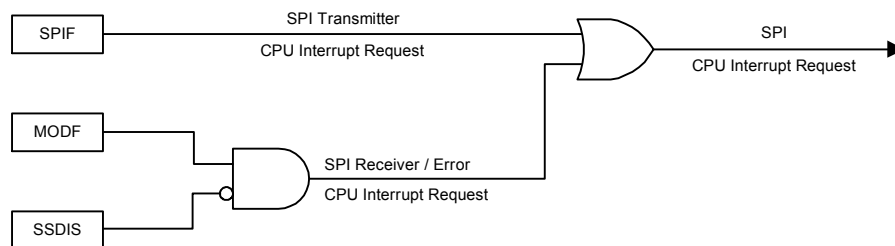
主设备或从设备尚未清除SPIF位,主或从设备又试图发送几个数据字节时,超限情况发生。在这种情况下,接收移位寄存器保持原有数据, SPIF置1,同样SPI设备直到SPIF被清除后才会再接收数据。在SPIF位被清除之前继续调用中断,发送也不会中止。RXOV位置1不会引起中断, RXOV位需由软件清0。

### 8.10.8 SPI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



### 8.10.9 SPI寄存器

Table 8.43 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
<b>SPCON</b>	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	<b>DIR</b>	<b>传送方向选择位</b> 0: MSB优先发送 1: LSB优先发送
6	<b>MSTR</b>	<b>SP设备选择位</b> 0: 配置SPI作为从属设备 1: 配置SPI作为主设备
5	<b>CPHA</b>	<b>时钟相位控制位</b> 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	<b>CPOL</b>	<b>时钟极性控制位</b> 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	<b>SSDIS</b>	<b>SS引脚控制位</b> 0: 在主和从属模式下, 打开SS引脚 1: 在主和从属模式下, 关闭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	<b>SPR[2:0]</b>	<b>串行外部设备时钟速率选择位</b> 000: $f_{SYS}/4$ 001: $f_{SYS}/8$ 010: $f_{SYS}/16$ 011: $f_{SYS}/32$ 100: $f_{SYS}/64$ 其他: $f_{SYS}/128$



Table 8.44 SPI状态寄存器

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
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	<b>SPI控制位</b> 0: 关闭SPI 1: 打开SPI接口
6	SPIF	<b>串行外部设备数据传送标志位</b> 0: 由软件清0 1: 表明已完成数据传输, 由硬件置1
5	MODF	<b>模式故障位</b> 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	<b>写入冲突标志位</b> 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	<b>接收超限位</b> 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 8.45 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。



### 8.11 逐次逼近型ADC

#### 8.11.1 特性

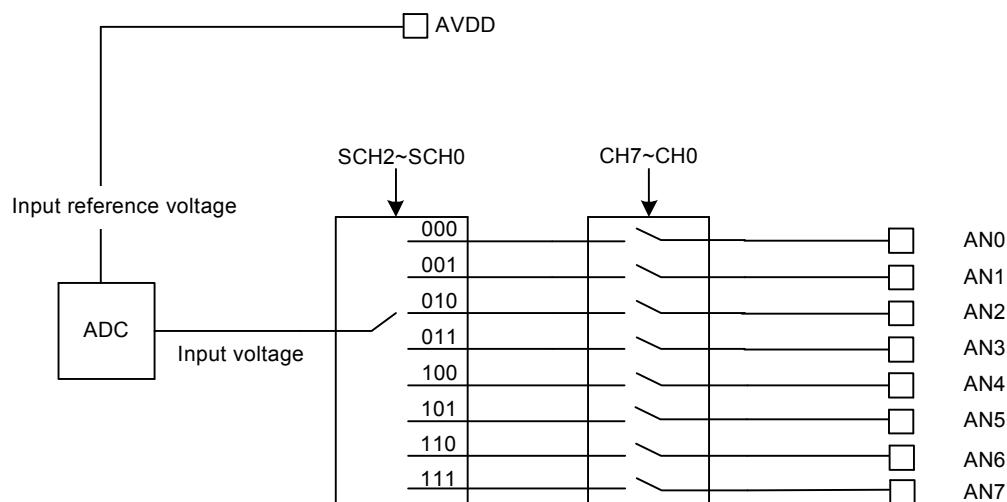
- 10位分辨率
- 内建基准电压
- 可选外接或内建基准电压
- 8模拟通道输入

SH99F100中的MCU包含一个单端型、10位逐次逼近型模数转换器（ADC）。ADC基准电压使用AV<sub>DD</sub>。8个ADC通道都可以独立输入模拟信号，但是每次转换只能使用一个通道。GO/DONE信号控制开始转换，提示转换结束。当转换完成时，更新ADC数据寄存器与此同时，设置ADCON寄存器中的ADCIF位，并且产生一个中断（如果允许ADC中断）。

ADC模块整合数字比较功能可以比较ADC中的模拟输入的值与数字值。如果允许数字比较功能（在ADCON寄存器中的EC位置1），并且ADC模块使能（在ADCON寄存器中的ADON位置1），只有当相应的模拟输入的数字值大于寄存器中的比较值（ADDH/L）时，才会产生ADC中断。当GO/DONE置1时，数字比较功能会持续工作，直到GO/DONE清0。这一点与模数转换工作方式不同。

带数字比较功能的ADC模块能在Idle模式下工作，并且ADC中断能够唤醒Idle模式。但是，在Power-Down模式下，ADC模块被禁止。

#### 8.11.2 ADC模块图



AD转换器模块图



8.11.3 ADC寄存器

Table 8.46 ADC控制寄存器

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
<b>ADCON</b>	ADON	ADCIF	EC	-	SCH2	SCH1	SCH0	GO/DONE
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	<b>ADON</b>	<b>ADC允许位</b> 0: 禁止ADC模块 1: 允许ADC模块
6	<b>ADCIF</b>	<b>ADC中断标志位</b> 0: 无ADC中断 1: 由硬件置1, 表示已完成AD转换或者模拟输入大于ADDH/ADDL (如果允许数字比较模块)
5	<b>EC</b>	<b>比较功能允许</b> 0: 禁止数字比较功能 1: 允许数字比较功能
3-1	<b>SCH[2:0]</b>	<b>ADC通道选择位</b> 000: ADC通道AN0 001: ADC通道AN1 010: ADC通道AN2 011: ADC通道AN3 100: ADC通道AN4 101: ADC通道AN5 110: ADC通道AN6 111: ADC通道AN7
0	<b>GO/DONE</b>	<b>ADC状态标志位</b> 0: 当完成AD转换时, 由硬件自动清0。在转换期间清0这个位会中止AD转换。如果允许数字比较功能, 该位不会由硬件清0只能由软件清0 1: 设置开始AD转换或者启动数字比较功能



Table 8.47 ADC定时控制寄存器

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
ADT	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
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	TADC[2:0]	<b>ADC时钟周期选择位</b> 000: ADC时钟周期 $t_{AD} = 2 t_{SYS}$ 001: ADC时钟周期 $t_{AD} = 4 t_{SYS}$ 010: ADC时钟周期 $t_{AD} = 6 t_{SYS}$ 011: ADC时钟周期 $t_{AD} = 8 t_{SYS}$ 100: ADC时钟周期 $t_{AD} = 12 t_{SYS}$ 101: ADC时钟周期 $t_{AD} = 16 t_{SYS}$ 110: ADC时钟周期 $t_{AD} = 24 t_{SYS}$ 111: ADC时钟周期 $t_{AD} = 32 t_{SYS}$
3-0	TS[3:0]	<b>采样时间选择位</b> $2 t_{AD} \leq \text{采样时间} = (TS[3:0]+1) * t_{AD} \leq 15 t_{AD}$

注意:

- (1) 请确保 $t_{AD} \geq 1\mu s$ ;
- (2) 即使 $TS[3:0] = 0000$ , 最小采样时间为 $2t_{AD}$ ;
- (3) 即使 $TS[3:0] = 1111$ , 最大采样时间为 $15t_{AD}$ ;
- (4) 在设置 $TS[3:0]$ 前, 请估算连接到ADC输入引脚的串联电阻;
- (5) 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小于 $10k\Omega$ ;
- (6) 总共转换时间=  $12t_{AD}$ + 采样时间。

Table 8.48 ADC通道配置寄存器

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
ADCH	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]	<b>通道配置位</b> 0: P3.7-P3.0作为I/O端口 1: P3.7-P3.0作为ADC输入口



Table 8.49 AD转换数据寄存器（比较值寄存器）

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
ADDL	-	-	-	-	-	-	A1	A0
读/写	-	-	-	-	-	-	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	-	-	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
ADDH	A9	A8	A7	A6	A5	A4	A3	A2
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
1-0 7-0	A9-A0	<b>ADC数据寄存器</b> 采样模拟电压的数字值。当完成转换后，这个值会更新。 如果ADC数字比较功能使能（EC = 1），这个值将与模拟输入进行比较。

**启动ADC转换步骤:**

- (1) 选择模拟输入通道以及基准电压
- (2) 使能ADC模块
- (3)  $\overline{GO/DONE}$ 置1开始ADC转换
- (4) 等待 $\overline{GO/DONE} = 0$ 或者 $ADCIF = 1$ ，如果ADC中断使能，则ADC中断将会产生，用户需要软件清0  $ADCIF$
- (5) 从ADDH/ADDL获得转换数据
- (6) 重复步骤3-5开始另一次转换

**启动数字比较功能步骤:**

- (1) 选择模拟输入通道以及基准电压
- (2) 写入ADDH/ADDL，设置比较值
- (3) EC置1使能数字比较功能
- (4) 使能ADC模块
- (5)  $\overline{GO/DONE}$ 置1开始数字比较功能
- (6) 如果模拟输入的值比设置的比较值大，ADIF会被置1。如果ADC中断使能，则ADC中断将会产生，用户需要软件清0  $ADCIF$
- (7) 数字比较功能会持续工作，直到 $\overline{GO/DONE}$ 清0





## 9. DSP部分

### 9.1 DSP核特性

SH99F100中的DSP核是一个16位的定点数字信号处理器，能够在100MIPS的速度下实现多种数据处理，其架构具有高度的并行性，在一个周期内可以完成：

- 产生下一个程序地址
- 读取下一条指令
- 完成一次或两次数据移动
- 更新一个或两个数据地址指针
- 完成一次计算

其内部包括以下功能单元：

- (1) 计算单元：包括3个独立的、功能完备的计算单元：算数/逻辑单元（ALU）、乘加器（MAC）以及桶形移位器（SHIFTER）。计算单元直接处理16位数据并对多精度计算提供硬件支持。
  - 一个单周期内可以完成乘法、带累加乘、任意位的移位及标准算术和逻辑运算。
  - MAC单元支持扩展的乘积和。MAC的累加部分采用了40位的累加器，即提供了8位的溢出保护，从而保证不丢失数据或范围。在数据丢失发生之前要产生256次溢出。有特殊指令用以实现块浮点数。
  - 许多算法，如数字滤波器（PID算法也是FIR的一种）都需要循环的数据缓冲器。DSP的基本框架中包含处理指针卷绕的硬件，简化了片内缓冲器的操作，并且减少了时间开销。
  - 许多算法具有重复性，在逻辑上常表示为循环。DSP的过程控制器支持零开销的循环代码，能将优良的性能和清晰的程序结构结合在一起。同时，条件分支也是零开销的。
- (2) 数据地址产生器和过程控制器：两个专用的数据地址产生器（Data Address Generator, DAG1/2）和一个过程控制器（Program Sequencer, PSQ）提供对DDM、DPM和DCM空间的寻址。PSQ支持单周期的条件分支和无开销循环。双数据地址产生器使DSP能够同时产生两个操作数的地址。数据地址产生器和过程控制器使得DSP保持连续工作状态，使流量达到最大化。
- (3) 增强型哈佛结构：两个数据存储器空间（Program Memory, PM数据空间；Data Memory, DM数据空间）和一个程序存储器空间（Code Memory, CM空间）。3个存储器空间独立编址。处理器可以在一个周期读取两个操作数（一个来自PM空间，一个来之DM空间）和一条指令（来自CM空间）。系统结构图如图9-1-1所示。

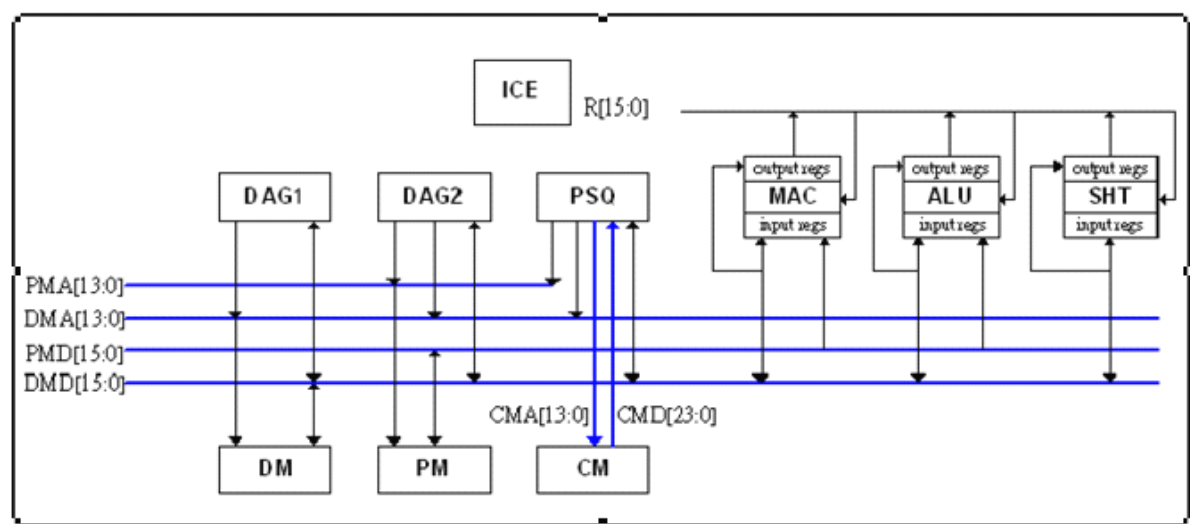


图9-1-1



### 9.1.1 DSP核寄存器

#### 概述

DSP核寄存器是非存储器映射(non-memory mapped)的寄存器,汇编指令中直接用符号表示即可,可直接寻址。如AX0、I0、M4等等。

DSP核寄存器按照硬件功能的不同,分组示意如图9-1-2所示:

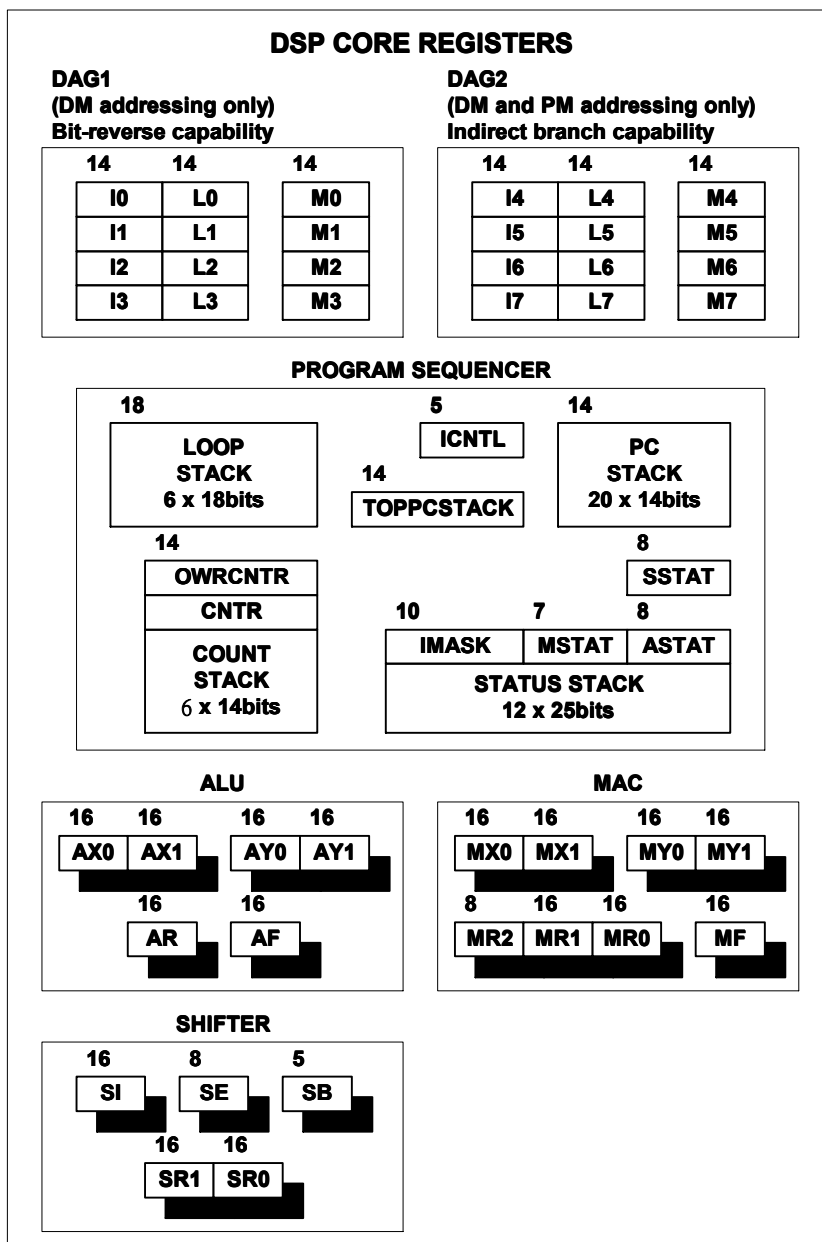


图9-1-2 DSP核寄存器分组示意图



## 数据地址发生器 (DAG1/2) 寄存器

DAG1和DAG2两个数据地址发生器各含有12个14位宽的寄存器。索引 (Index) 寄存器Ix (x = 0-7), 一般用来存放地址指针; 修改 (Modify) 寄存器Mx (x = 0-7), 一般用来更新索引寄存器; 长度 (Length) 寄存器Lx (x = 0-7), 用于循环缓冲操作 (Circular Buffer)。I0 - I3、M0 - M3和L0 - L3属于DAG1, I4 - I7、M4 - M7和L4 - L7属于DAG2。DAG1只能用来寻址DM空间, 并且具有位取反输出功能。DAG2可以寻址DM与PM空间, 并为JUMP和CALL指令提供寻址 (Indirect branch)。

以下的例子是利用I0通过间接寻址读取I0所指向DM空间的存储器单元, 一旦读取完成, I0中的值便更新为M0中的值:

*AX0 = DM (I0, M0)*

以下的例子是利用I4通过间接寻址写I4所指向PM空间的存储器单元, 之后I4中的值被M5更新:

*PM (I4, M5) = MR1*

以下的例子是利用I4实现的跳转:

*JUMP (I4)*

## 过程控制器 (PSQ) 寄存器

PSQ相关的寄存器用来控制子过程 (Subroutines) 的调用、循环 (Loops)、中断 (Interrupts), 同时也用来指示CPU的状态以及控制CPU工作模式。PSQ核寄存器的位定义见9.1.2详细说明。

### (1) 中断控制寄存器ICNTL、IMASK

ICNTL寄存器用于选择中断是否允许嵌套。10位宽的中断屏蔽寄存器IMASK用来使能/禁止10个外设中断源。DSP支持全局使能中断指令 (ENA INTS) 和全局禁止中断指令 (DIS INTS)。执行DIS INTS指令将导致所有用户中断都被屏蔽, 而不必改动IMASK的值。ENA INTS指令将使得IMASK中使能的中断再次有效。

### (2) 循环控制寄存器CNTR、OWRCNTR

寄存器CNTR (Loop Counter Register) 用来存储当前所执行循环的循环次数。写CNTR的动作会导致当前的计数器值被压入计数器堆栈 (Count Stack), 然后CNTR更新为写入的新值。Count Stack深度为6级, 因此在软件不做任何处理的情形下允许循环嵌套6次。OWRCNTR是一个特殊的寄存器, 写这个寄存器可以改变循环计数器CNTR的值而不会导致CNTR压入计数器堆栈Count Stack。

### (3) 状态与模式控制寄存器SSTAT、ASTAT、MSTAT

堆栈状态寄存器SSTAT寄存器含有指示所有堆栈空或满的标志, 算术状态寄存器ASTAT包含所有与计算单元相关的标志, 模式状态寄存器MSTAT包含各种CPU运行选项的控制位。MSTAT中有4个位用来控制是否使用第二套寄存器、是否使能DAG1的位取反功能、ALU溢出时是否锁存溢出标志、ALU是否使用饱和和逻辑。MSTAT中还有1位用来控制MAC运算结果的格式。

可以使用模式控制指令 (ENA或DIS) 来方便地使能/禁止DSP的运行模式。

## 过程控制器 (PSQ) 堆栈

PSQ含有4个硬件堆栈供循环 (Loop)、子程序 (Subroutine) 调用以及中断嵌套 (Interrupts Nesting)。

程序计数器堆栈 (PC Stack) 是一个14位宽、20级深度的硬件堆栈。它存储了子程序的返回地址、中断返回地址以及循环顶部地址 (top-of-loop address)。在发生子程序调用 (call指令) 和中断处理时, PC Stack逻辑会自动执行。此外, 可以通过寄存器TOPPCSTACK来手动地将数据压入PC Stack: TOPPCSTACK = reg, 以及从PC Stack中弹一个数据出来: reg = TOPPCSTACK。

循环堆栈 (Loop Stack) 是一个18位宽、6级深度的硬件堆栈。其中, 14位用来存储循环结束地址 (end-of-loop address), 4位用来存储循环结束条件代码。当执行DO UNTIL指令时, Loop Stack逻辑会自动执行。如果循环嵌套, 那么退出循环时Loop Stack中的内容会自动的弹出至相关寄存器中。可以手动地使用POP LOOP指令来使Loop Stack中的内容出栈。

状态堆栈 (Status Stack) 是一个25位宽、12级深度的硬件堆栈。当DSP处理一个中断, 寄存器IMASK、MSTAT和ASTAT的值会自动压入Status Stack。当执行中断返回指令RTI时, Status Stack中的值会自动弹出至相应的寄存器中。可以通过指令PUSH STS或POP STS来手动将上述三个寄存器的值压入Status Stack 或将Status Stack 中数值弹出至对应的寄存器中。

计数器堆栈 (Count Stack) 是一个14位宽、6级深度的硬件堆栈。写CNTR寄存器的动作会导致当前的计数器值被自动压入计数器堆栈 (Count Stack)。



### 计算单元（ALU、MAC、Shifter）寄存器

DSP核包含3个计算单元：ALU、MAC和Shifter，计算单元寄存器主要用于计数过程的数据存储。

ALU和MAC对于大多数的运算都需要两个输入。AX0、AX1、MX0和MX1寄存器用于存储X输入，而AY0、AY1、MY0和MY1用于存储Y输入。寄存器AR和AF用于存储ALU的结果；AF可以作为ALU的Y输入，而AR可以作为任何运算单元的X输入。同样，MR0、MR1、MR2和MF用来存储MAC的运算结果并且也可以用来作为其它运算单元的输入。16位的MR0和MR1寄存器结合8位的MR2寄存器可以用来存储一个40位的乘/累加结果。

移位器（Shifter）可以接受ALU或MAC的输入寄存器，它以自己的结果寄存器或一个特定的移位器输入寄存器SI（Shifter Input）作为其输入。移位器可以存储一个32位的结果于寄存器SR0/1中。SB寄存器用来存储块浮点格式运算（block floating-point operations）中的指数块（block exponent）。SE寄存器用来存储规格化（normalize）和反规格化（denormalize）操作时移位的值。

计算单元的所有寄存器都有第二套相同功能的备用寄存器（secondary registers），如图9-1-1中的阴影部分所示。在中断不会嵌套的情形下，第二套寄存器可以用来实现单周期上下文切换。用户可以通过MSTAT中的SEC\_REG位来选择使用第二套寄存器，这个位可以利用一下的指令来置位或清除：

```
ENA SEC_REG; /* select secondary registers */
DIS SEC_REG; /* select primary registers */
```

### 9.1.2 PSQ核寄存器的位定义

过程控制器（PSQ）中除了4个硬件堆栈寄存器外，其包含的核寄存器，如下表所示：

类别	缩写符号	功能说明
中断控制	ICNTL	中断控制寄存器
	IMASK	中断屏蔽寄存器
状态指示	ASTAT	算术状态寄存器
	MSTAT	模式状态寄存器
	SSTAT	堆栈状态寄存器
计数器相关	CNTR	循环计数器寄存器
	OWRCNTR	复写计数器寄存器
堆栈相关	TOPPCSTACK	PC栈顶寄存器

**Table 9.1** DSP中断核级控制寄存器ICNTL (Interrupt Control Register)

<b>4</b>	<b>3-0</b>
<i>R/W-0</i>	<i>R-0</i>
<b>INEST</b>	<b>Reserved</b>

位编号	位符号	说明
<b>4</b>	<b>INEST</b>	<b>DSP中断嵌套使能位</b> 0: 禁止中断嵌套 1: 允许中断嵌套
<b>3-0</b>	<b>Reserved</b>	保留位



**Table 9.2** DSP中断核级屏蔽寄存器IMASK (Interrupt Mask Register)

<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>IM_PWM</b>	<b>IM_TMR0</b>	<b>IM_ADC</b>	<b>IM_QEI</b>	<b>IM_DXIOC</b>	<b>IM_TMR1</b>	<b>IM_TMR2</b>	<b>IM_TMR3</b>	<b>IM_CAP</b>	<b>IM_COMM</b>

位编号	位符号	说明
<b>9</b>	<b>IM_PWM</b>	<b>PWM模块中断屏蔽位（核级使能位）</b> 0: 禁止PWM模块中断 1: 允许PWM模块中断
<b>8</b>	<b>IM_TMR0</b>	<b>DTimer0中断屏蔽位（核级使能位）</b> 0: 禁止DTimer0中断 1: 允许DTimer0中断
<b>7</b>	<b>IM_ADC</b>	<b>ADC模块中断屏蔽位（核级使能位）</b> 0: 禁止ADC模块中断 1: 允许ADC模块中断
<b>6</b>	<b>IM_QEI</b>	<b>QEI模块中断屏蔽位（核级使能位）</b> 0: 禁止QEI模块中断 1: 允许QEI模块中断
<b>5</b>	<b>IM_DXIOC</b>	<b>DXINT和输出比较模块中断屏蔽位（核级使能位）</b> 0: 禁止DXINT和输出比较模块中断 1: 允许DXINT和输出比较模块中断
<b>4</b>	<b>IM_TMR1</b>	<b>DTimer1中断屏蔽位（核级使能位）</b> 0: 禁止DTimer1中断 1: 允许DTimer1中断
<b>3</b>	<b>IM_TMR2</b>	<b>DTimer2中断屏蔽位（核级使能位）</b> 0: 禁止DTimer2中断 1: 允许DTimer2中断
<b>2</b>	<b>IM_TMR3</b>	<b>DTimer3中断屏蔽位（核级使能位）</b> 0: 禁止DTimer3中断 1: 允许DTimer3中断
<b>1</b>	<b>IM_CAP</b>	<b>输入捕获模块中断屏蔽位（核级使能位）</b> 0: 禁止输入捕获模块中断 1: 允许输入捕获模块中断
<b>0</b>	<b>IM_COMM</b>	<b>双核通信中断屏蔽位（核级使能位）</b> 0: 禁止双核通信中断 1: 允许双核通信中断



**Table 9.3** 算术状态寄存器ASTAT (Arithmetic Status Register)

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>SS</b>	<b>MV</b>	<b>AQ</b>	<b>AS</b>	<b>AC</b>	<b>AV</b>	<b>AN</b>	<b>AZ</b>

位编号	位符号	说明
<b>7</b>	<b>SS</b>	<b>Shifter</b> 输入符号 ( <b>Shifter input sign</b> )
<b>6</b>	<b>MV</b>	<b>乘法溢出状态位 (MAC overflow)</b> 乘法/累加操作中, 累加器超过32位界限而溢出时, MV置位
<b>5</b>	<b>AQ</b>	<b>ALU商位 (ALU quotient)</b> 由DIVS和DIVQ指令产生的商位
<b>4</b>	<b>AS</b>	<b>ALU输入符号 (ALU X input sign)</b> ALU X输入口的符号位, 只受ABS指令影响
<b>3</b>	<b>AC</b>	<b>ALU进位 (ALU carry)</b> 执行加法时, 最高位上的进位输出
<b>2</b>	<b>AV</b>	<b>ALU溢出 (ALU overflow)</b> 执行加法时, 高两位上的进位输出异或。ALU输出溢出时, AV = 1; 否则AV = 0
<b>1</b>	<b>AN</b>	<b>ALU结果负 (ALU result negative)</b> ALU结果寄存器的符号位。ALU输出最高位为1 (负) 时, AN = 1; 否则AN = 0
<b>0</b>	<b>AZ</b>	<b>ALU结果零 (ALU result zero)</b> ALU结果寄存器中所有位的逻辑或非 (NOR)。ALU输出为零, AZ = 1; 否则AZ = 0

**Table 9.4** 模式状态寄存器MSTAT (Mode Status Register)

<b>6-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>M_MODE</b>	<b>AR_SAT</b>	<b>AV_LATCH</b>	<b>BIT_REV</b>	<b>SEC_REG</b>

位编号	位符号	说明
<b>6-5</b>	<b>Reserved</b>	保留位
<b>4</b>	<b>M_MODE</b>	<b>MAC结果存放模式选择</b> 0: 写0使能小数模式。乘法结果先向左移1位再加至MR中 1: 写1将使能整数模式。乘法结果不做任何处理而直接加至MR中
<b>3</b>	<b>AR_SAT</b>	<b>AR寄存器饱和选择</b> 0: 写0将禁止饱和模式 1: 写1将使能饱和模式, 如果ALU的操作导致AR溢出, 那么AR寄存器会饱和
<b>2</b>	<b>AV_LATCH</b>	<b>ALU溢出状态标志锁存选择</b> 写1将使得ALU在溢出的时候, ASTAT中断AV位会被锁存住
<b>1</b>	<b>BIT_REV</b>	<b>DAG1的位取反模式选择</b> 写1将使得DAG1地址输出按照位取反的模式输出
<b>0</b>	<b>SEC_REG</b>	<b>计算单元寄存器组选择</b> 0: 写0将选择默认的第一套寄存器组 (primary) 1: 写1将选择第二套寄存器组 (secondary)



**Table 9.5** 堆栈状态寄存器SSTAT (Stack Status Register)

7	6	5	4	3	2	1	0
R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1
<b>LPSO</b>	<b>LPSE</b>	<b>SSO</b>	<b>SSE</b>	<b>CSO</b>	<b>CSE</b>	<b>PCSO</b>	<b>PCSE</b>
位编号	位符号	说明					
7	LPSO	循环堆栈溢出标志位 (Loop Stack overflow) 0: 无溢出 1: 溢出					
6	LPSE	循环堆栈空标志位 (Loop Stack empty) 0: 非空 1: 空					
5	SSO	状态堆栈溢出标志位 (Status Stack overflow) 0: 无溢出 1: 溢出					
4	SSE	状态堆栈空标志位 (Status Stack empty) 0: 非空 1: 空					
3	CSO	循环计数器堆栈溢出标志位 (Count Stack overflow) 0: 无溢出 1: 溢出					
2	CSE	循环计数器堆栈空标志位 (Count Stack empty) 0: 非空 1: 空					
1	PCSO	程序计数器堆栈溢出标志位 (PC Stack overflow) 0: 无溢出 1: 溢出					
0	PCSE	程序计数器堆栈空标志位 (PC Stack empty) 0: 非空 1: 空					

**Table 9.6** 循环计数器寄存器CNTR (Counter Register)

13-0		
R/W-0		
CNTR		
位编号	位符号	说明
13-0	CNTR	循环计数器，用于存储循环的次数。每循环一次CNTR的值减1，直至0循环结束。当CNTR中写入一个新值时，CNTR被写入之前的值会压入计数器堆栈 (Count Stack)

**Table 9.7** 复写计数器寄存器OWRCNTR (Overwrite Counter Register)

13-0		
R/W-0		
OWRCNTR		
位编号	位符号	说明
13-0	OWRCNTR	向OWRCNTR写值将写入CNTR中，并且不会导致CNTR的旧值压入堆栈

**Table 9.8** PC栈顶寄存器TOPPCSTACK (Top PC Stack Register)

13-0		
R/W-0		
TOPPCSTACK		
位编号	位符号	说明
13-0	TOPPCSTACK	该寄存器用来读取 (Popping) 或写 (Pushing) PC栈顶的值



### 9.1.3 DSP核的控制寄存器

除了DSP核寄存器之外，还有一些存储器映射（memory mapped）的寄存器用于反映DSP核的运行状态。这些寄存器实际是DM空间地址的存储单元，对它们的读写操作需要采用间接寻址方式，如：

AX0 = 0x023F;

DM (0x3FFE) = AX0;

这些寄存器包括DSP系统控制寄存器DSYSR和反映4个堆栈当前状态的DCCSD、DCLSD、DCSSD、DCPSD寄存器。

**Table 9.9** DSP系统控制寄存器DSYSR

<b>15-2</b>		<b>1</b>	<b>0</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R-0</i>
<b>Reserved</b>		<b>BIASRND</b>	<b>Reserved</b>

位编号	位符号	说明
<b>15-2</b>	<b>Reserved</b>	保留位
<b>1</b>	<b>BIASRND</b>	对MAC运算结果进行Biased Rounding处理的使能控制位 0: 禁止对MAC的40bits运算结果进行Biased Rounding处理 1: 允许对MAC的40bits运算结果进行Biased Rounding处理
<b>0</b>	<b>Reserved</b>	保留位

**Table 9.10** 当前计数器堆栈深度寄存器DCCSD (DSP Current Count Stack Depth Register)

<b>15-3</b>		<b>2-0</b>
<i>R-0</i>		<i>R-0</i>
<b>Reserved</b>		<b>DCCSD</b>

位编号	位符号	说明
<b>15-3</b>	<b>Reserved</b>	保留位
<b>2-0</b>	<b>DCCSD</b>	<b>DSP当前计数器堆栈深度（DCCSD）指示位</b> 指示当前计数器堆栈的深度，当有数据压入COUNT STACK时，DCCSD自增1； 当有数据从COUNT STACK中弹出时，DCCSD自减1

**Table 9.11** 当前循环堆栈深度寄存器DCLSD (DSP Current Loop Stack Depth Register)

<b>15-3</b>		<b>2-0</b>
<i>R-0</i>		<i>R-0</i>
<b>Reserved</b>		<b>DCLSD</b>

位编号	位符号	说明
<b>15-3</b>	<b>Reserved</b>	保留位
<b>2-0</b>	<b>DCLSD</b>	<b>DSP当前循环堆栈深度（DCLSD）指示位</b> 指示当前循环堆栈深度，当有数据压入LOOP STACK时，DCLSD自增1； 当有数据从LOOP STACK中弹出时，DCLSD自减1



**Table 9.12** 当前状态堆栈深度寄存器DCSSD (DSP Current Status Stack Depth Register)

<b>15-4</b>	<b>3-0</b>
<i>R/O</i>	<i>R/O</i>
<b>Reserved</b>	<b>DCSSD</b>

位编号	位符号	说明
<b>15-4</b>	<b>Reserved</b>	保留位
<b>3-0</b>	<b>DCSSD</b>	<b>DSP当前状态堆栈深度 (DCSSD) 指示位</b> 指示当前状态堆栈的深度, 当有数据压入STATUS STACK时, DCSSD自增1; 当有数据从STATUS STACK中弹出时, DCSSD自减1

**Table 9.13** 当前程序计数器堆栈深度寄存器DCPSD (DSP Current PC STACK Depth Register)

<b>15-5</b>	<b>4-0</b>
<i>R/O</i>	<i>R/O</i>
<b>Reserved</b>	<b>DCPSD</b>

位编号	位符号	说明
<b>15-5</b>	<b>Reserved</b>	保留位
<b>5-0</b>	<b>DCSSD</b>	<b>DSP程序计数器堆栈深度 (DCPSD) 指示位</b> 指示当前程序计数器堆栈深度, 当有数据压入PC STACK时, DCPSD自增1; 当有数据从PC STACK中弹出时, DCPSD自减1



## 9.2 DSP中断

### 9.2.1 特性

- 12个中断向量
- 4级软件控制优先级
- 3个外部输入中断源

DSP核包含12个可用中断向量，除复位中断和SCM中断外，其他10个中断向量由各外设模块复用。在DSP核的中断嵌套功能关闭时，用户可以通过设置外设模块中断的软件优先级（共4个级别），调整10个外设模块中断向量的响应顺序。

SH99F100还通过3个外部输入引脚，提供多种电平触发方式的外部输入中断。

### 9.2.2 控制寄存器

DSP中断使用的所有控制寄存器如下表所示：

类别	缩写符号	功能说明
核级控制*	ICNTL	中断控制寄存器
	IMASK	中断屏蔽寄存器
软件控制优先级	INTPR0 INTPR1	软件控制中断优先级设置寄存器
外部中断	DXINTIE	DSP外部中断控制寄存器
	DXINTIF	DSP外部中断标志寄存器

说明：核级控制寄存器ICNTL和IMASK是非存储器映像的寄存器，详细定义参见DSP核寄存器的描述。

**Table 9.14** DSP中断核级控制寄存器ICNTL (Interrupt Control Register)

4		3-0
<i>RW-0</i>		<i>R-0</i>
<b>INEST</b>		<b>Reserved</b>

位编号	位符号	说明
<b>4</b>	<b>INEST</b>	<b>DSP中断嵌套使能位</b> 0: 禁止中断嵌套 1: 允许中断嵌套
<b>3-0</b>	<b>Reserved</b>	保留位



**Table 9.15** DSP中断核级屏蔽寄存器IMASK (Interrupt Mask Register)

<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>IM_PWM</b>	<b>IM_TMR0</b>	<b>IM_ADC</b>	<b>IM_QEI</b>	<b>IM_DXIOC</b>	<b>IM_TMR1</b>	<b>IM_TMR2</b>	<b>IM_TMR3</b>	<b>IM_CAP</b>	<b>IM_COMM</b>

位编号	位符号	说明
<b>9</b>	<b>IM_PWM</b>	<b>PWM</b> 模块中断屏蔽位（核级使能位） 0: 禁止PWM模块中断 1: 允许PWM模块中断
<b>8</b>	<b>IM_TMR0</b>	<b>DTimer0</b> 中断屏蔽位（核级使能位） 0: 禁止DTimer0中断 1: 允许DTimer0中断
<b>7</b>	<b>IM_ADC</b>	<b>ADC</b> 模块中断屏蔽位（核级使能位） 0: 禁止ADC模块中断 1: 允许ADC模块中断
<b>6</b>	<b>IM_QEI</b>	<b>QEI</b> 模块中断屏蔽位（核级使能位） 0: 禁止QEI模块中断 1: 允许QEI模块中断
<b>5</b>	<b>IM_DXIOC</b>	<b>DXINT</b> 和输出比较模块中断屏蔽位（核级使能位） 0: 禁止DXINT和输出比较模块中断 1: 允许DXINT和输出比较模块中断
<b>4</b>	<b>IM_TMR1</b>	<b>DTimer1</b> 中断屏蔽位（核级使能位） 0: 禁止DTimer1中断 1: 允许DTimer1中断
<b>3</b>	<b>IM_TMR2</b>	<b>DTimer2</b> 中断屏蔽位（核级使能位） 0: 禁止DTimer2中断 1: 允许DTimer2中断
<b>2</b>	<b>IM_TMR3</b>	<b>DTimer3</b> 中断屏蔽位（核级使能位） 0: 禁止DTimer3中断 1: 允许DTimer3中断
<b>1</b>	<b>IM_CAP</b>	输入捕获模块中断屏蔽位（核级使能位） 0: 禁止输入捕获模块中断 1: 允许输入捕获模块中断
<b>0</b>	<b>IM_COMM</b>	双核通信中断屏蔽位（核级使能位） 0: 禁止双核通信中断 1: 允许双核通信中断



**Table 9.16** DSP软件控制优先级寄存器0 INTPR0

<b>15-14</b>	<b>13-12</b>	<b>11-10</b>	<b>9-8</b>	<b>7-6</b>	<b>5-4</b>	<b>3-2</b>	<b>1-0</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>IPR_ADC</b>	<b>IPR_QEI</b>	<b>IPR_DXIOC</b>	<b>IPR_TMR1</b>	<b>IPR_TMR2</b>	<b>IPR_TMR3</b>	<b>IPR_CAP</b>	<b>IPR_COMM</b>

位编号	位符号	说明
<b>15-14</b>	<b>IPR_ADC</b>	<b>ADC</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>13-12</b>	<b>IPR_QEI</b>	<b>QEI</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>11-10</b>	<b>IPR_DXIOC</b>	<b>DXINT</b> 和 <b>输出比较</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>9-8</b>	<b>IPR_TMR1</b>	<b>DTimer1</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>7-6</b>	<b>IPR_TMR2</b>	<b>DTimer2</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>5-4</b>	<b>IPR_TMR3</b>	<b>DTimer3</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>3-2</b>	<b>IPR_CAP</b>	<b>输入捕捉</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
<b>1-0</b>	<b>IPR_COMM</b>	<b>双核通信</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)



**Table 9.17** DSP软件控制优先级寄存器1 INTPR1

15-8	7-4	3-2	1-0
R-0	R-0	R/W-0	R/W-0
Reserved	Reserved	IPR_PWM	IPR_TMR0

位编号	位符号	说明
15-4	Reserved	保留位
3-2	IPR_PWM	<b>PWM</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)
1-0	IPR_TMR0	<b>DTimer0</b> 中断的软件控制优先级设置位 00: 软件优先级0 (最高优先级) 01: 软件优先级1 10: 软件优先级2 11: 软件优先级3 (最低优先级)

**Table 9.18** DSP外部中断控制寄存器DXINTCON

15-14	13-12	11-10	9-8	7-3	2	1	0
R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0
Reserved	DXINT2MOD	DXINT1MOD	DXINT0MOD	Reserved	DXINT2IE	DXINT1IE	DXINT0IE

位编号	位符号	说明
15-14	Reserved	保留位
13-12	DXINT2MOD	<b>DXINT2</b> 外部中断触发方式选择位 00: 下降沿触发 01: 上升沿触发 10: 双沿触发 11: 低电平触发
11-10	DXINT1MOD	<b>DXINT1</b> 外部中断触发方式选择位 00: 下降沿触发 01: 上升沿触发 10: 双沿触发 11: 低电平触发
9-8	DXINT0MOD	<b>DXINT0</b> 外部中断触发方式选择位 00: 下降沿触发 01: 上升沿触发 10: 双沿触发 11: 低电平触发
7-3	Reserved	保留位
2	DXINT2IE	<b>DXINT2</b> 外部中断使能控制位 0: 禁止DXINT2外部中断 1: 允许DXINT2外部中断
1	DXINT1IE	<b>DXINT1</b> 外部中断使能控制位 0: 禁止DXINT1外部中断 1: 允许DXINT1外部中断
0	DXINT0IE	<b>DXINT0</b> 外部中断使能控制位 0: 禁止DXINT0外部中断 1: 允许DXINT0外部中断



**Table 9.19** DSP外部中断标志寄存器DXINTIF

<b>15-8</b>		<b>7-3</b>		<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>		<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>Reserved</b>		<b>DXINT2IF</b>	<b>DXINT1F</b>	<b>DXINT0F</b>

位编号	位符号	说明
<b>15-3</b>	<b>Reserved</b>	保留位
<b>2</b>	<b>DXINT2IF</b>	<b>DXINT2外部中断标志位</b> 0: 无DXINT2外部中断发生 1: 有DXINT2外部中断发生 写0清除该标志, 写1无效
<b>1</b>	<b>DXINT1IF</b>	<b>DXINT1外部中断标志位</b> 0: 无DXINT1外部中断发生 1: 有DXINT1外部中断发生 写0清除该标志, 写1无效
<b>0</b>	<b>DXINT0IF</b>	<b>DXINT0外部中断标志位</b> 0: 无DXINT0外部中断发生 1: 有DXINT0外部中断发生 写0清除该标志, 写1无效



### 9.2.3 DSP中断使能的全局控制

DSP包含2条中断全局控制指令：中断全局允许指令（ENA INTS）和中断全局禁止指令（DIS INTS）。

系统复位后，中断允许/禁止的全局控制状态为允许状态。执行DIS INTS指令后，将禁止所有的中断（包括非屏蔽中断SCM）响应，而不管中断屏蔽位和中断使能位的设置内容。再次执行ENA INTS指令后，全局控制状态切换为允许状态，所有未被屏蔽的中断都可以在使能控制位允许的情况下被DSP核响应。

### 9.2.4 DSP核级中断控制

DSP核内有两个寄存器ICNTL和IMASK，分别用于设置中断嵌套方式和中断屏蔽控制。

#### ICNTL与中断嵌套设置

ICNTL寄存器只提供一个控制位INEST，用于控制是否使用DSP核的中断嵌套功能。

INEST = 0，DSP核自动关闭中断嵌套功能。当两个中断同时发生时，DSP核自动按照自然优先级顺序响应高优先级的中断，低优先级中断被挂起。当高优先级中断服务子程序结束后，低优先级中断才被响应。该模式下，用户还可以使用软件优先级灵活调整10个外设中断的优先级顺序。

INEST = 1，DSP核自动使能中断嵌套功能。当DSP核执行一个中断响应的服务子程序时，如果有更高优先级的中断发生，则DSP核会立即响应高优先级中断，在完成高优先级中断的服务子程序后，再继续恢复执行低优先级的服务子程序。使用中断嵌套功能的情况下，10个外设中断的软件优先级设置自动失效，硬件按照自然优先级顺序响应所有的外设中断。

#### IMASK与中断屏蔽设置

IMASK寄存器用于在DSP核内实现各外设中断源的使能控制，系统复位和SCM中断不受IMASK寄存器控制，因此SCM中断也称为不可屏蔽中断。

IMASK寄存器是个10位宽的DSP核寄存器，其每一个位对应控制一个外设中断源，设置值为0表示屏蔽对应的中断源，也即禁止该中断的响应；设置值为1则表示允许该中断源的响应。因此，要使一个外设中断使能，除了要设置该外设的中断使能位有效，还必须同时设置IMASK的对应屏蔽位为1。

在一个中断得到硬件响应，并开始进入中断向量位置的中断服务程序后，硬件会自动把IMASK内容压入状态堆栈STATUS STACK中，以便在中断返回时通过堆栈恢复中断前的IMASK内容。

### 9.2.5 DSP中断的响应

DSP程序运行过程中，当一个中断发生时，下一条指令的PC值会自动压入PC STACK堆栈，IMASK/MSTAT/ASTAT会自动压入STATUS STACK堆栈，然后程序跳转到该中断对应的中断向量入口位置，开始执行中断服务程序。中断服务程序执行完毕后，执行RTI指令可以将PC STACK栈顶的内容弹出到PC中，控制权返回到主程序中，同时STATUS STACK中的内容也会被一一弹出，以恢复到中断响应前处理器的原状态。

如果IMASK压栈时INEST = 0，DSP核通过压栈后硬件修改IMASK=0来实现禁止中断嵌套功能；如果IMASK压栈时INEST = 1，则DSP核会自动载入一个新的IMASK值，屏蔽所有低优先级的中断以实现中断嵌套功能。中断嵌套功能的这种实现方式，为用户提供了一种在关闭硬件中断嵌套功能后，通过软件在中断服务程序中修改IMASK值来实现中断嵌套功能的方法。

系统复位中断具有一定的特殊性，只有系统复位后Autoload过程结束后才会进入向量入口，从0x0000位置开始执行主程序，该中断响应过程不会进行任何堆栈操作。

**9.2.6 DSP中断优先级**

为实现中断优先级的灵活控制，SH99F100采用两级优先级控制来实现：自然优先级和软件优先级。

**自然优先级**

自然优先级，是指DSP核对12个中断源做出的优先级判定，用户不能对核级自然优先级进行修改和调整。12个中断向量对应的核级自然优先级固定如下表所示：（优先级编号越小优先级越高）

中断源	中断向量地址	自然优先级
系统复位	0x0000	0（最高优先级）
PWM & 输入保护	0x0004	2
DTimer0	0x0008	3
ADC	0x000C	4
QEI	0x0010	5
DXINT & 输出比较	0x0014	6
DTimer1	0x0018	7
DTimer2	0x001C	8
DTimer3	0x0020	9
输入捕获	0x0024	10
双核通讯	0x0028	11（最低优先级）
SCM	0x002C	1

**软件优先级**

当INEST = 0，禁止中断嵌套功能时，通过软件优先级寄存器INTPR0/1，用户可以灵活设置10个外设模块的中断源优先级。每个外设模块中断源均有独立的软件优先级控制位，可以分别设置4个软件优先级：0、1、2和3，设置值越小优先级越高。

系统复位和SCM中断没有软件优先级，且其优先级始终高于10个外设模块中断。当软件优先级设置有效时，硬件首先对10个外设中断根据软件优先级设置值进行优先级排序，相同软件优先级的外设中断则再根据自然优先级顺序排序。

**9.2.7 DSP中断外部中断DXINT0-2**

SH99F100提供3个外部中断输入引脚，作为DSP外部中断源。3个外部输入引脚DXINT0、DXINT1和DXINT2分别与I/O口复用，并与输出比较模块共用一个中断向量。

DXINTxIE (x = 0-2) 分别控制每个外部输入引脚的功能。当DXINTxIE = 1，对应外部输入引脚的中断功能打开，当硬件按照DXINTxMOD设置模式检测到有效的外输入中断信号时，自动置位DXINTxIF = 1，同时触发中断。当DXINTxIE = 0，对应外部输入引脚的中断功能关闭，该引脚作为普通I/O使用，但DXINTxIF标志还是会根据DXINTxMOD设置模式检测到的引脚输入状态变化。

通过DXINTxMOD (x = 0-2) 可以设置各输入引脚的触发模式为上升沿触发、下降沿触发、双沿触发或低电平触发。当设置为低电平触发时，DXINTxIF标志始终由外部输入信号电平决定。





### 9.2.8 DSP中断汇总

DSP中断的所有中断源与中断控制/状态信息汇总如下表：

中断源	向量地址	中断屏蔽位	中断使能位	中断标志位
系统复位	0x0000	-	-	-
PWM模块	0x0004	IM_PWM	PZRIE	PZRIF
			PFLIE	PFLIF
			PDP4IE	PDP4IF
			PDP3IE	PDP3IF
			PDP2IE	PDP2IF
			PDP1IE	PDP1IF
			PDP0IE	PDP0IF
DTimer0	0x0008	IM_TMR0	TMR0IE	TMR0IF
ADC模块	0x000C	IM_ADC	ADIE2	ADIF2
			ADIE1	ADIF1
QEI模块	0x0010	IM_QEI	QEIE	QEIF
			QCEIE	QCEIF
			QPEIE	QPEIF
			QTIE	QTIF
			QTCAPIE	QTCAPIF
DXINT & 输出比较	0x0014	IM_DXIOC	DXINT2IE	DXINT2IF
			DXINT1IE	DXINT1IF
			DXINT0IE	DXINT0IF
			OCFIE	OCFIF
			OC2IE	OC2IF
			OC1IE	OC1IF
			OC0IE	OC0IF
DTimer1	0x0018	IM_TMR1	TMR1IE	TMR1IF
DTimer2	0x001C	IM_TMR2	TMR2IE	TMR2IF
DTimer3	0x0020	IM_TMR3	TMR3IE	TMR3IF
输入捕获模块	0x0024	IM_CAP	CAP3IE	CAP3IF
			CAP2IE	CAP2IF
			CAP1IE	CAP1IF
			CAP0IE	CAP0IF
双核通讯	0x0028	IM_COMM	MCMDIE	MCMDIF
			MTDxIE (x = 0-14)	MTDxIF (x = 0-14)
SCM	0x002C	-	DSCMIE	-

对于上述中断源（系统复位中断除外），只有对应的中断屏蔽位和中断使能位都允许触发中断时，对应中断标志被硬件置1的同时才会触发中断，中断响应过程由中断优先级设置和嵌套设置决定。



### 9.3 PWM模块

#### 9.3.1 特性

- 16位PWM计数器
- 6路（3对）PWM输出信号
- 3种时基计数模式：单事件模式、边沿对齐模式和中心对齐模式
- 每对PWM提供2种输出模式：互补输出、独立输出
- 互补输出模式下提供死区控制逻辑
- 最多可在一个PWM时基周期（指寄存器PWMPRD定义的时间周期）内选择三处位置自动触发AD采样
- 5个保护输入引脚，可分别设置滤波时间，用于紧急关闭PWM输出
- 后分频功能

SH99F100内建一个增强型的PWM模块，包含三对PWM输出信号PWM0A/B、PWM1A/B、PWM2A/B，可以产生周期和占空比分别可调的脉宽调制波形，以实现多种电机的调速控制。

PWM模块内含一个16位宽的计数器PWMTCNT，6个占空比寄存器PWMWD0A/B、PWMWD1A/B和PWMWD2A/B分别与PWM计数值比较产生相应的PWM信号。

PWM模块提供3种时基计数模式（单事件计数模式、边沿对齐计数模式、中心对齐计数模式）和2种输出模式（互补输出、独立输出）。中心对齐计数模式下的互补输出还提供2种波形：对称波形和非对称波形。同一对PWM输出信号的所有设置组合如下图所示：

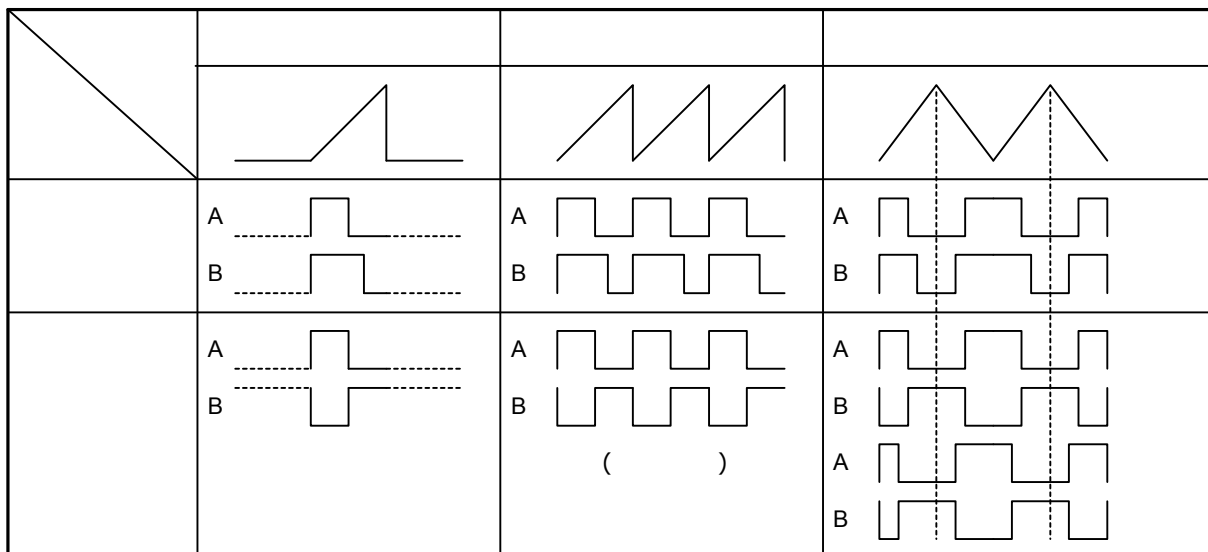


图9-3-1 PWM输出信号对的配置图

上电默认状态下，三对PWM均配置为边沿对齐的互补输出模式，互补模式下每对PWM提供独立的死区控制逻辑。

时基周期寄存器PWMPRD、输出极性控制位PWPLRxA/B（x = 0-2）以及事件触发比较寄存器PWMECMPx（x = 0-2）都存在内部缓存寄存器，用户修改这些寄存器值后，直到当前PWM计数器归零时才更新到对应的缓存寄存器，并产生修改效果。所有的占空比寄存器PWMWDxA/B（x = 0-2）也存在内部缓存寄存器，但用户可以通过控制位来选择寄存器修改值对内部缓存寄存器的更新方式。

6个PWM输出信号的极性可单独控制，互相没有任何制约，也与信号的输出模式（互补或独立输出）无关。用户可以根据应用需要，通过PWMWCON寄存器的PWPLRxA/B（x = 0-2）位灵活设置。



### 9.3.2 逻辑框图

PWM模块的逻辑框图如下图所示，主要包含6个子模块：

- (1) PWM时基子模块 (Time-Base submodule)
- (2) PWM占空比波形生成子模块 (Wave Generator submodule)
- (3) PWM死区延迟与极性控制子模块 (Dead-band Delay & Polarity Control submodule)
- (4) PWM输入保护与输出控制子模块 (Fault Input Detect & Output Control submodule)
- (5) PWM事件触发子模块 (Event Trigger submodule)
- (6) PWM中断子模块 (PWM Interrupt submodule)

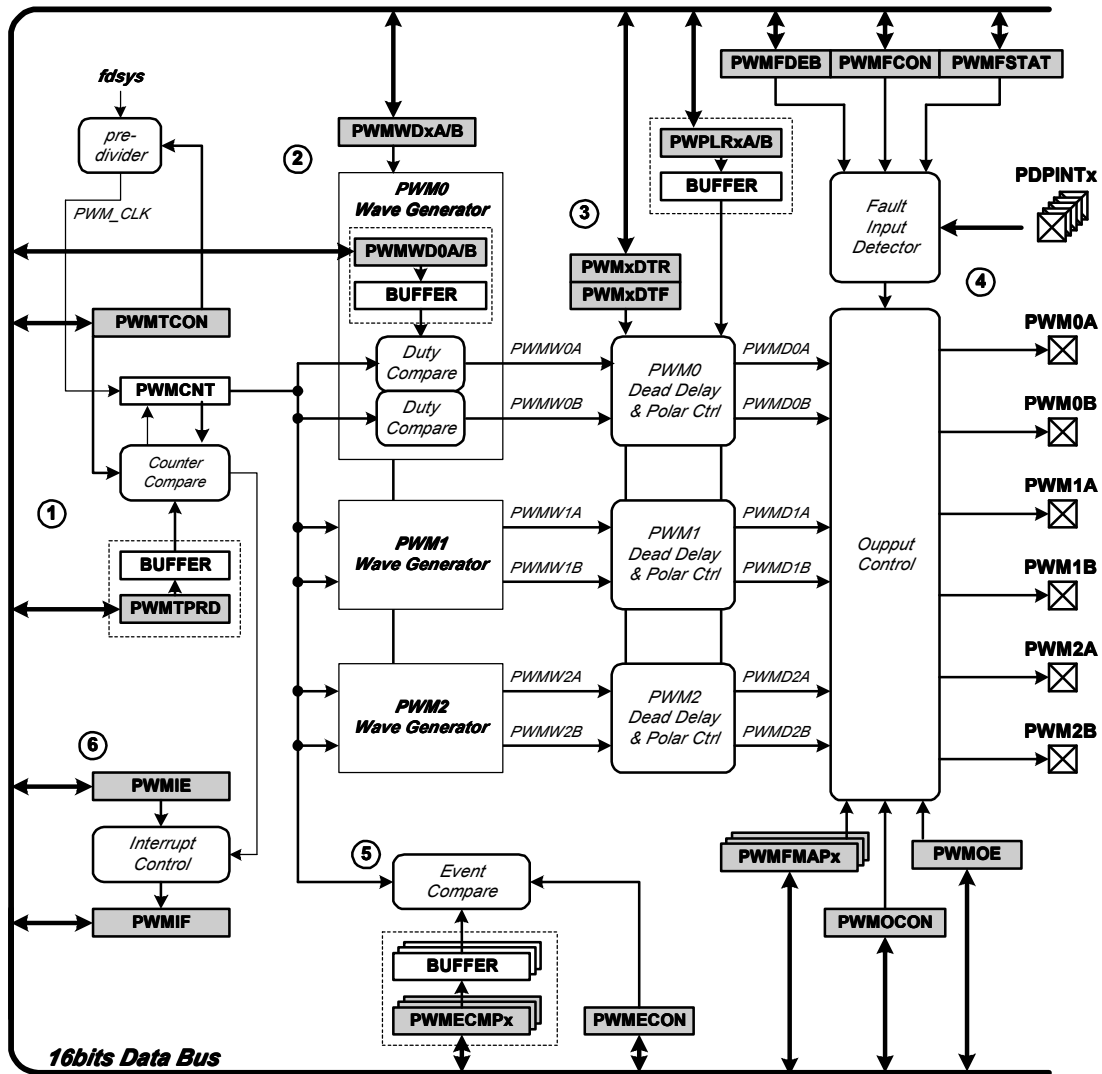


图9-3-2 PWM模块的逻辑框图



### 9.3.3 控制寄存器

PWM模块使用的所有控制寄存器如下表所示:

类别	缩写符号	功能说明	
模块控制	PWMLO	PWM模块锁定控制寄存器	
时基控制	PWMTCON	时基控制寄存器	
	PWMTPRD	时基周期设置寄存器	
	PWMTCNT	时基计数寄存器	
波形发生单元控制	PWMWCON	PWM比较输出波形控制寄存器	
	PWMWD0A PWMWD0B PWMWD1A PWMWD1B PWMWD2A PWMWD2B	6路PWM输出信号占空比设置寄存器	
	PWM0DTR	PWM0A/B上升沿死区时间设置寄存器	
	PWM0DTF	PWM0A/B下降沿死区时间设置寄存器	
	PWM1DTR	PWM1A/B上升沿死区时间设置寄存器	
	PWM1DTF	PWM1A/B下降沿死区时间设置寄存器	
死区控制	PWM2DTR	PWM2A/B上升沿死区时间设置寄存器	
	PWM2DTF	PWM2A/B下降沿死区时间设置寄存器	
	PWMECON	事件触发控制寄存器	
	PWMECMP0 PWMECMP1 PWMECMP2	事件触发比较寄存器	
	输入保护 & 输出控制	PWMFCON	保护输入引脚控制寄存器
		PWMFSTAT	保护输入引脚状态寄存器
PWMFDEB		保护输入引脚滤波设置寄存器	
PWMFMAP0 PWMFMAP1 PWMFMAP2		保护引脚映射控制寄存器	
PWMOCON		PWM输出控制寄存器	
PWMOE		PWM输出使能控制寄存器	
中断控制		PWMIE	PWM中断使能寄存器
	PWMIF	PWM中断标志寄存器	

**Table 9.20** PWM模块锁定寄存器PWMLO (PWM Write-lock Control)

<b>15-0</b>		
<i>R/W-0</i>		
<b>PWMLO</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMLO</b>	= 0x5A5A, 其他PWM寄存器的写操作有效 ≠ 0x5A5A, 其他PWM寄存器的写操作无效 (不改变原值)



**Table 9.21** 时基控制寄存器PWMTCON (PWM Time-Base Control)

15	14-13	12	11-8	7-5	4-3	2-0
<i>R/W-0</i>	<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>PWMEN</b>	<b>PWMEMU</b>	<b>PTDIR</b>	<b>PPOST</b>	<b>Reserved</b>	<b>PTMOD</b>	<b>PTCLKPS</b>
位编号	位符号	说明				
15	PWMEN	<b>PWM时基使能控制位</b> 0: 关闭PWM时基 1: 打开PWM时基 当用户写0关闭时基后, PWM计数器PWMCNT自动清零, 6路PWM口输出高阻态				
14-13	PWMEMU	<b>仿真挂起时PWM模块工作状态控制位</b> 00: 6路PWM口立即输出高阻态 01: 挂起后PWM计数器的状态由PTMOD决定 PTMOD = 00/10/11 (边沿与单次事件模式), PWM计数器将继续计数至与PWMTPRD值匹配后停止运行; PTMOD = 01 (中心对齐模式), PWM计数器将继续计数至0后停止运行。 1x: PWM计数器不受挂起影响, 将继续运行				
12	PTDIR	<b>时基计数方向状态位 (只读)</b> 0: 时基计数器当前为增计数状态 1: 时基计数器当前为减计数状态				
11-8	PPOST	<b>后分频设置位</b> 0000: 1/1后分频 0001: 1/2后分频 ... 1110: 1/15后分频 1111: 1/16后分频				
7-5	Reserved	保留位				
4-3	PTMOD	<b>PWM时基计数模式选择位</b> 00: 边沿对齐计数模式 01: 中心对齐计数模式 10: 单次事件计数模式 11: 单次事件计数模式				
2-0	PTCLKPS	<b>PWM模块工作时钟频率 (<math>f_{PT\_CLK}</math>) 预分频选择控制位</b> 000: = $f_{DSYS}/1$ 001: = $f_{DSYS}/2$ 010: = $f_{DSYS}/4$ 011: = $f_{DSYS}/8$ 100: = $f_{DSYS}/16$ 101: = $f_{DSYS}/32$ 110: = $f_{DSYS}/64$ 111: = $f_{DSYS}/128$				

**Table 9.22** 时基周期寄存器PWMTPRD (PWM Time-Base Period)

15-0		
<i>R/W-0</i>		
<b>PWMTPRD</b>		
位编号	位符号	说明
15-0	PWMTPRD	PWM时基周期设置寄存器 (设置0xFFFF无效) $PWM时基周期T_{TBPWM} = (PWMTPRD+1) \times T_{PT\_CLK}$



**Table 9.23** 时基计数寄存器PWMTCNT (PWM Time-Base Counter)

<b>15-0</b>		
<i>R-0</i>		
<b>PWMTCNT</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMTCNT</b>	PWM时基计数寄存器（只读）

**Table 9.24** 波形控制寄存器PWMWCON (PWM Wave Control)

<b>15-13</b>		<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>PWDL D</b>	<b>PWSYM</b>	<b>PWMOD2</b>	<b>PWMOD1</b>	<b>PWMOD0</b>
<b>7-6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R/W-1</i>	<i>R/W-0</i>	<i>R/W-1</i>	<i>R/W-0</i>	<i>R/W-1</i>	<i>R/W-0</i>
<b>PDMOD</b>	<b>PWPLR2A</b>	<b>PWPLR2B</b>	<b>PWPLR1A</b>	<b>PWPLR1B</b>	<b>PWPLR0A</b>	<b>PWPLR0B</b>
位编号	位符号	说明				
<b>15-13</b>	<b>Reserved</b>	保留位				
<b>12</b>	<b>PWDL D</b>	<b>PWM</b> 占空比寄存器更新方式控制位 0: 周期更新（与下一个PWM时基周期同步） 1: 立即更新				
<b>11</b>	<b>PWSYM</b>	<b>PWM</b> 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				
<b>10</b>	<b>PWMOD2</b>	<b>PWM2A/B</b> 输出模式选择位 0: PWM2A和PWM2B工作于互补输出模式 1: PWM2A和PWM2B工作于独立输出模式				
<b>9</b>	<b>PWMOD1</b>	<b>PWM1A/B</b> 输出模式选择位 0: PWM1A和PWM1B工作于互补输出模式 1: PWM1A和PWM1B工作于独立输出模式				
<b>8</b>	<b>PWMOD0</b>	<b>PWM0A/B</b> 输出模式选择位 0: PWM0A和PWM0B工作于互补输出模式 1: PWM0A和PWM0B工作于独立输出模式				
<b>7-6</b>	<b>PDMOD</b>	死区延迟模式控制位（对互补输出的 <b>PWM</b> 对设置有效） 00: 无死区延迟 01: PWM互补输出的上升沿插入死区延迟，下降沿无死区延迟 10: PWM互补输出的下降沿插入死区延迟，上升沿无死区延迟 11: PWM互补输出的上升沿和下降沿均插入死区延迟				
<b>5</b>	<b>PWPLR2A</b>	<b>PWM2A</b> 波形极性控制位 0: 有效时区为低电平，无效时区为高电平 1: 有效时区为高电平，无效时区为低电平				



续上表

位编号	位符号	说明
4	PWPLR2B	<b>PWM2B波形极性控制位</b> 0: 有效时区为低电平, 无效时区为高电平 1: 有效时区为高电平, 无效时区为低电平
3	PWPLR1A	<b>PWM1A波形极性控制位</b> 0: 有效时区为低电平, 无效时区为高电平 1: 有效时区为高电平, 无效时区为低电平
2	PWPLR1B	<b>PWM1B波形极性控制位</b> 0: 有效时区为低电平, 无效时区为高电平 1: 有效时区为高电平, 无效时区为低电平
1	PWPLR0A	<b>PWM0A波形极性控制位</b> 0: 有效时区为低电平, 无效时区为高电平 1: 有效时区为高电平, 无效时区为低电平
0	PWPLR0B	<b>PWM0B波形极性控制位</b> 0: 有效时区为低电平, 无效时区为高电平 1: 有效时区为高电平, 无效时区为低电平

**Table 9.25** 占空比寄存器组PWMWDxA/B (x = 0, 1, 2) (PWM Wave Duty)

<b>15-0</b>
<i>R/W-0</i>
<b>PWMWDxA/B</b>

位编号	位符号	说明
15-0	<b>PWMWDxA/B (x = 0, 1, 2)</b>	<p><b>PWMWDxA/B (x = 0-2) 是占空比寄存器</b></p> <p><b>独立输出模式:</b> PWMWDxA/B一对一控制PWMxA/B (x = 0-2)</p> <p><b>互补输出模式:</b> PWMWD0A同时控制PWM0A/B输出占空比, PWMWD0B设置值无效 PWMWD1A同时控制PWM1A/B输出占空比, PWMWD1B设置值无效 PWMWD2A同时控制PWM2A/B输出占空比, PWMWD2B设置值无效</p> <p><b>互补输出模式 (非对称波形, 仅在中心对齐计数模式下存在):</b> PWMWD0A同时控制PWM0A/B增计数过程的输出占空比 PWMWD0B同时控制PWM0A/B减计数过程的输出占空比 PWMWD1A同时控制PWM1A/B增计数过程的输出占空比 PWMWD1B同时控制PWM1A/B减计数过程的输出占空比 PWMWD2A同时控制PWM2A/B增计数过程的输出占空比 PWMWD2B同时控制PWM2A/B减计数过程的输出占空比</p>

**Table 9.26** 上升沿死区延迟时间寄存器PWMxDTR (x = 0-2) (PWM Dead-Time Rising Edge Delay register)

<b>15-0</b>
<i>R/W-0</i>
<b>PWMxDTR</b>

位编号	位符号	说明
15-0	<b>PWMxDTR (x = 0-2)</b>	设置PWMx波形信号上升沿死区延迟时间 (详细定义参考死区的描述) 上升沿死区延迟时间 $T_{DTR} = (PWMxDTR+1) \times T_{DSYS}$



**Table 9.27** 下降沿死区延迟时间寄存器PWMxDTF (x = 0-2) (PWM Dead-Time Falling Edge Delay register)

<b>15-0</b>		
<i>R/W-0</i>		
<b>PWMxDTF</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMxDTF (x = 0-2)</b>	设置PWMx波形信号下降沿死区延迟时间（详细定义参考死区的描述） 下降沿死区延迟时间 $T_{DTF} = (PWMxDTF+1) \times T_{DSYS}$

**Table 9.28** 事件触发控制寄存器PWMECON (PWM Event Trigger Control)

<b>15-11</b>	<b>10-8</b>	<b>7-6</b>	<b>5-4</b>	<b>3-2</b>	<b>1-0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PEMOD</b>	<b>PECMP0DIR</b>	<b>PECMP2EN</b>	<b>PECMP1EN</b>	<b>PECMP0EN</b>

位编号	位符号	说明
<b>15-11</b>	<b>Reserved</b>	保留位
<b>10-8</b>	<b>PEMOD</b>	<b>事件触发模式选择位</b> 000: PWM工作过程不触发任何事件 001: 根据PWMECMPxEN (x = 0-2)控制位触发ADC 010: PWM工作过程不触发任何事件 011: PWM工作过程不触发任何事件 100: PWM计数器值归零时, 触发定时器DTimer0运行 101: PWM计数器值归零时, 触发定时器DTimer1运行 110: PWM计数器值归零时, 触发定时器DTimer2运行 111: PWM计数器值归零时, 触发定时器DTimer3运行
<b>7-6</b>	<b>PECMP0DIR</b>	<b>PWM计数器与PWMECMP0比较方式控制位（仅在PTMOD=01且PEMOD=001时有效）</b> 00: 只要PWM计数器值与PWMECMP0相等便触发ADC 01: 只要PWM计数器值与PWMECMP0相等便触发ADC 10: 只有PWM计数器增计数过程与PWMECMP0相等时才触发ADC 11: 只有PWM计数器减计数过程与PWMECMP0相等时才触发ADC
<b>5-4</b>	<b>PECMP2EN</b>	<b>PWMECMP2事件触发方式控制位（仅在PEMOD=001时有效）</b> 00: 禁止PWM计数器值与PWMECMP2比较 01: 禁止PWM计数器值与PWMECMP2比较 10: PWM计数器值与PWMECMP2比较相等时, 置位SOC1 11: PWM计数器值与PWMECMP2比较相等时, 置位SOC2
<b>3-2</b>	<b>PECMP1EN</b>	<b>PWMECMP1事件触发方式控制位（仅在PEMOD=001时有效）</b> 00: 禁止PWM计数器值与PWMECMP1比较 01: 禁止PWM计数器值与PWMECMP1比较 10: PWM计数器值与PWMECMP1比较相等时, 置位SOC1 11: PWM计数器值与PWMECMP1比较相等时, 置位SOC2
<b>1-0</b>	<b>PECMP0EN</b>	<b>PWMECMP0事件触发方式控制位（仅在PEMOD=001时有效）</b> 00: 禁止PWM计数器值与PWMECMP0比较 01: 禁止PWM计数器值与PWMECMP0比较 10: PWM计数器值与PWMECMP0比较相等时, 置位SOC1 11: PWM计数器值与PWMECMP0比较相等时, 置位SOC2





**Table 9.29** 事件触发比较寄存器PWMECMP0 (PWM Event Compare Register0)

<b>15-0</b>		
<i>R/W-0</i>		
<b>PWMECMP0</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMECMP0</b>	事件触发比较寄存器0, PWM计数器值与该设置值相等时可以触发ADC

**Table 9.30** 事件触发比较寄存器PWMECMP1 (PWM Event Compare Register1)

<b>15-0</b>		
<i>R/W-0</i>		
<b>PWMECMP1</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMECMP1</b>	事件触发比较寄存器1, PWM计数器值与该设置值相等时可以触发ADC

**Table 9.31** 事件触发比较寄存器PWMECMP2 (PWM Event Compare Register2)

<b>15-0</b>		
<i>R/W-0</i>		
<b>PWMECMP2</b>		
位编号	位符号	说明
<b>15-0</b>	<b>PWMECMP2</b>	事件触发比较寄存器2, PWM计数器值与该设置值相等时可以触发ADC

**Table 9.32** 保护输入引脚控制寄存器PWMFCON (PWM Fault Input Control)

<b>15-13</b>		<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>PFLT4RM</b>	<b>PFLT3RM</b>	<b>PFLT2RM</b>	<b>PFLT1RM</b>	<b>PFLT0RM</b>
<b>7-5</b>		<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>PFLT4EN</b>	<b>PFLT3EN</b>	<b>PFLT2EN</b>	<b>PFLT1EN</b>	<b>PFLT0EN</b>
位编号	位符号	说明				
<b>15-13</b>	<b>Reserved</b>	保留位				
<b>12-8</b>	<b>PFLTyRM (y = 0-4)</b>	保护输入引脚PDPINTy (y = 0-4) 的故障恢复模式控制位 0: 软件恢复 1: 硬件恢复				
<b>7-5</b>	<b>Reserved</b>	保留位				
<b>4-0</b>	<b>PFLTyEN (y = 0-4)</b>	保护输入引脚PDPINTy (y = 0-4) 的保护功能使能控制位 0: PDPINTy作为I/O口 1: PDPINTy作为保护输入引脚				



**Table 9.33** 保护输入引脚状态寄存器PWMFSTAT (PWM Fault Input Status)

<b>15-13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PFLT4TG</b>	<b>PFLT3TG</b>	<b>PFLT2TG</b>	<b>PFLT1TG</b>	<b>PFLT0TG</b>
<b>7-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R-u</i>	<i>R-u</i>	<i>R-u</i>	<i>R-u</i>	<i>R-u</i>
<b>Reserved</b>	<b>PFLT4ST</b>	<b>PFLT3ST</b>	<b>PFLT2ST</b>	<b>PFLT1ST</b>	<b>PFLT0ST</b>

位编号	位符号	说明
<b>15-13</b>	<b>Reserved</b>	保留位
<b>12-8</b>	<b>PFLTyTG (y = 0-4)</b>	保护输入引脚PDPINTy的触发电平控制位 0: 低电平触发 1: 高电平触发
<b>7-5</b>	<b>Reserved</b>	保留位
<b>4-0</b>	<b>PFLTyST (y = 0-4)</b>	保护输入引脚PDPINTy滤波后电平状态指示位（仅在PFLTxEN=1时有效） 0: PDPINTy信号为低电平 1: PDPINTy信号为高电平

**Table 9.34** 保护输入引脚滤波时间设置寄存器PWMFDEB (PWM Fault Input Debounce)

<b>15</b>	<b>14-12</b>	<b>11-9</b>	<b>8-6</b>	<b>5-3</b>	<b>2-0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PFLT4DEB</b>	<b>PFLT3DEB</b>	<b>PFLT2DEB</b>	<b>PFLT1DEB</b>	<b>PFLT0DEB</b>

位编号	位符号	说明
<b>15</b>	<b>Reserved</b>	保留位
<b>14-12</b>	<b>PFLT4DEB</b>	<b>PDPINT4</b> 的滤波时间控制位 000: 关闭滤波功能 001: 约0.25μs 010: 约0.5μs 011: 约1μs 100: 约2μs 101: 约4μs 110: 约8μs 111: 约16μs
<b>11-9</b>	<b>PFLT3DEB</b>	<b>PDPINT3</b> 的滤波时间控制位（设置值定义同 <b>PFLT4DEB</b> ）
<b>8-6</b>	<b>PFLT2DEB</b>	<b>PDPINT2</b> 的滤波时间控制位（设置值定义同 <b>PFLT4DEB</b> ）
<b>5-3</b>	<b>PFLT1DEB</b>	<b>PDPINT1</b> 的滤波时间控制位（设置值定义同 <b>PFLT4DEB</b> ）
<b>2-0</b>	<b>PFLT0DEB</b>	<b>PDPINT0</b> 的滤波时间控制位（设置值定义同 <b>PFLT4DEB</b> ）



**Table 9.35** 保护输入引脚映射寄存器0 PWMFMAP0 (PWM Fault Input Mapping Control register0)

<b>15-13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP0A</b>	<b>PF3MAP0A</b>	<b>PF2MAP0A</b>	<b>PF1MAP0A</b>	<b>PF0MAP0A</b>
<b>7-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP0B</b>	<b>PF3MAP0B</b>	<b>PF2MAP0B</b>	<b>PF1MAP0B</b>	<b>PF0MAP0B</b>

位编号	位符号	说明
<b>15-13</b>	<b>Reserved</b>	保留位
<b>12-8</b>	<b>PFyMAP0A (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM0A口输出的控制位</b> 0: 使能PDPINTy保护PWM0A口的输出 1: 禁止PDPINTy保护PWM0A口的输出
<b>7-5</b>	<b>Reserved</b>	保留位
<b>4-0</b>	<b>PFyMAP0B (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM0B口输出的控制位</b> 0: 使能PDPINTy保护PWM0B口的输出 1: 禁止PDPINTy保护PWM0B口的输出

**Table 9.36** 保护输入引脚映射寄存器1 PWMFMAP1 (PWM Fault Input Mapping Control register1)

<b>15-13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP1A</b>	<b>PF3MAP1A</b>	<b>PF2MAP1A</b>	<b>PF1MAP1A</b>	<b>PF0MAP1A</b>
<b>7-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP1B</b>	<b>PF3MAP1B</b>	<b>PF2MAP1B</b>	<b>PF1MAP1B</b>	<b>PF0MAP1B</b>

位编号	位符号	说明
<b>15-13</b>	<b>Reserved</b>	保留位
<b>12-8</b>	<b>PFyMAP1A (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM1A口输出的控制位</b> 0: 使能PDPINTy保护PWM1A口的输出 1: 禁止PDPINTy保护PWM1A口的输出
<b>7-5</b>	<b>Reserved</b>	保留位
<b>4-0</b>	<b>PFyMAP1B (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM1B口输出的控制位</b> 0: 使能PDPINTy保护PWM1B口的输出 1: 禁止PDPINTy保护PWM1B口的输出



**Table 9.37** 保护输入引脚映射寄存器2 PWMFMAP2 (PWM Fault Input Mapping Control register2)

<b>15-13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP2A</b>	<b>PF3MAP2A</b>	<b>PF2MAP2A</b>	<b>PF1MAP2A</b>	<b>PF0MAP2A</b>
<b>7-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PF4MAP2B</b>	<b>PF3MAP2B</b>	<b>PF2MAP2B</b>	<b>PF1MAP2B</b>	<b>PF0MAP2B</b>

位编号	位符号	说明
<b>15-13</b>	<b>Reserved</b>	保留位
<b>12-8</b>	<b>PFyMAP2A (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM2A口输出的控制位</b> 0: 使能PDPINTy保护PWM2A口的输出 1: 禁止PDPINTy保护PWM2A口的输出
<b>7-5</b>	<b>Reserved</b>	保留位
<b>4-0</b>	<b>PFyMAP2B (y = 0-4)</b>	<b>5个FLT输入引脚是否保护PWM2B口输出的控制位</b> 0: 使能PDPINTy保护PWM2B口的输出 1: 禁止PDPINTy保护PWM2B口的输出

**Table 9.38** 输出控制寄存器PWMOCON (PWM Output Control)

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>POSYNC</b>	<b>Reserved</b>	<b>POCON2A</b>	<b>POCON2B</b>	<b>POCON1A</b>	<b>POCON1B</b>	<b>POCON0A</b>	<b>POCON0B</b>
<b>7-6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>POUT2A</b>	<b>POUT2B</b>	<b>POUT1A</b>	<b>POUT1B</b>	<b>POUT0A</b>	<b>POUT0B</b>	

位编号	位符号	说明
<b>15</b>	<b>POSYNC</b>	<b>PWM口输出同步位</b> 当PWMMEN = 1时: 0: POCONxA/B位的修改值在下一个Tdsys边沿处产生效果 1: POCONxA/B位的修改值在下一个PWM时基周期开始时产生效果
<b>14</b>	<b>Reserved</b>	保留位
<b>13</b>	<b>POCON2A</b>	<b>PWM2A口输出控制位</b> 0: PWM2A口输出PWM模块的PWM2A波形 1: PWM2A口作为手动输出口, 手动输出值由POUT2A决定
<b>12</b>	<b>POCON2B</b>	<b>PWM2B口输出控制位</b> 0: PWM2B口输出PWM模块的PWM2B波形 1: PWM2B口作为手动输出口, 手动输出值由POUT2B决定
<b>11</b>	<b>POCON1A</b>	<b>PWM1A口输出控制位</b> 0: PWM1A口输出PWM模块的PWM1A波形 1: PWM1A口作为手动输出口, 手动输出值由POUT1A决定
<b>10</b>	<b>POCON1B</b>	<b>PWM1B口输出控制位</b> 0: PWM1B口输出PWM模块的PWM1B波形 1: PWM1B口作为手动输出口, 手动输出值由POUT1B决定
<b>9</b>	<b>POCON0A</b>	<b>PWM0A口输出控制位</b> 0: PWM0A口输出PWM模块的PWM0A波形 1: PWM0A口作为手动输出口, 手动输出值由POUT0A决定



续上表

位编号	位符号	说明
8	POCON0B	<b>PWM0B口输出控制位</b> 0: PWM0B口输出PWM模块的PWM0B波形 1: PWM0B口作为手动输出口, 手动输出值由POUT0B决定
7-6	Reserved	保留位
5	POUT2A	<b>POCON2A = 1时, PWM2A口输出电平控制位</b> 0: 输出0 1: 输出1
4	POUT2B	<b>POCON2B = 1时, PWM2B口输出电平控制位</b> 0: 输出0 1: 输出1
3	POUT1A	<b>POCON1A = 1时, PWM1A口输出电平控制位</b> 0: 输出0 1: 输出1
2	POUT1B	<b>POCON1B = 1时, PWM1B口输出电平控制位</b> 0: 输出0 1: 输出1
1	POUT0A	<b>POCON0A = 1时, PWM0A口输出电平控制位</b> 0: 输出0 1: 输出1
0	POUT0B	<b>POCON0B = 1时, PWM0B口输出电平控制位</b> 0: 输出0 1: 输出1

Table 9.39 输出使能控制寄存器PWMOE (PWM Output Enable)

15-6	5	4	3	2	1	0
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Reserved	PWMOE2A	PWMOE2B	PWMOE1A	PWMOE1B	PWMOE0A	PWMOE0B

位编号	位符号	说明
15-6	Reserved	保留位
5	PWMOE2A	<b>PWM2A输出使能位</b> 1: PWM2A口正常输出PWM2A波形 0: PWM2A口输出高阻态
4	PWMOE2B	<b>PWM2B输出使能位</b> 1: PWM2B口正常输出PWM2B波形 0: PWM2B口输出高阻态
3	PWMOE1A	<b>PWM1A输出使能位</b> 1: PWM1A口正常输出PWM1A波形 0: PWM1A口输出高阻态
2	PWMOE1B	<b>PWM1B输出使能位</b> 1: PWM1B口正常输出PWM1B波形 0: PWM1B口输出高阻态
1	PWMOE0A	<b>PWM0A输出使能位</b> 1: PWM0A口正常输出PWM0A波形 0: PWM0A口输出高阻态
0	PWMOE0B	<b>PWM0B输出使能位</b> 1: PWM0B口正常输出PWM0B波形 0: PWM0B口输出高阻态



**Table 9.40** PWM中断控制寄存器PWMIE (PWM Interrupt Enable)

<b>15-8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PZRIE</b>	<b>PFLIE</b>	<b>Reserved</b>	<b>PDP4IE</b>	<b>PDP3IE</b>	<b>PDP2IE</b>	<b>PDP1IE</b>	<b>PDP0IE</b>

位编号	位符号	说明
<b>15-8</b>	<b>Reserved</b>	保留位
<b>7</b>	<b>PZRIE</b>	计数归零中断使能位 0: 禁止归零中断 1: 使能归零中断
<b>6</b>	<b>PFLIE</b>	计数满值中断使能位 0: 禁止满值中断 1: 使能满值中断
<b>5</b>	<b>Reserved</b>	保留位
<b>4</b>	<b>PDP4IE</b>	<b>PDPINT4</b> 输入引脚触发中断控制位 0: 禁止PDPINT4输入引脚触发中断 1: 允许PDPINT4输入引脚触发中断
<b>3</b>	<b>PDP3IE</b>	<b>PDPINT3</b> 输入引脚触发中断控制位 0: 禁止PDPINT3输入引脚触发中断 1: 允许PDPINT3输入引脚触发中断
<b>2</b>	<b>PDP2IE</b>	<b>PDPINT2</b> 输入引脚触发中断控制位 0: 禁止PDPINT2输入引脚触发中断 1: 允许PDPINT2输入引脚触发中断
<b>1</b>	<b>PDP1IE</b>	<b>PDPINT1</b> 输入引脚触发中断控制位 0: 禁止PDPINT1输入引脚触发中断 1: 允许PDPINT1输入引脚触发中断
<b>0</b>	<b>PDP0IE</b>	<b>PDPINT0</b> 输入引脚触发中断控制位 0: 禁止PDPINT0输入引脚触发中断 1: 允许PDPINT0输入引脚触发中断



**Table 9.41** PWM中断标志寄存器PWMIF (PWM Interrupt Flag)

<b>15-8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>PZRIF</b>	<b>PFLIF</b>	<b>Reserved</b>	<b>PDP4IF</b>	<b>PDP3IF</b>	<b>PDP2IF</b>	<b>PDP1IF</b>	<b>PDP0IF</b>

位编号	位符号	说明
<b>15-8</b>	<b>Reserved</b>	保留位
<b>7</b>	<b>PZRIF</b>	计数归零中断标志位 0: 无归零中断发生 1: 有归零中断发生 写0清除该标志, 写1无效
<b>6</b>	<b>PFLIF</b>	计数满值中断标志位 0: 无满值中断发生 1: 有满值中断发生 写0清除该标志, 写1无效
<b>5</b>	<b>Reserved</b>	保留位
<b>4</b>	<b>PDP4IF</b>	<b>PDPINT4</b> 输入引脚中断标志位 0: PDPINT4无触发中断 1: PDPINT4有触发中断 写0清除该标志, 写1无效
<b>3</b>	<b>PDP3IF</b>	<b>PDPINT3</b> 输入引脚中断标志位 0: PDPINT3无触发中断 1: PDPINT3有触发中断 写0清除该标志, 写1无效
<b>2</b>	<b>PDP2IF</b>	<b>PDPINT2</b> 输入引脚中断标志位 0: PDPINT2无触发中断 1: PDPINT2有触发中断 写0清除该标志, 写1无效
<b>1</b>	<b>PDP1IF</b>	<b>PDPINT1</b> 输入引脚中断标志位 0: PDPINT1无触发中断 1: PDPINT1有触发中断 写0清除该标志, 写1无效
<b>0</b>	<b>PDP0IF</b>	<b>PDPINT0</b> 输入引脚中断标志位 0: PDPINT0无触发中断 1: PDPINT0有触发中断 写0清除该标志, 写1无效



### 9.3.4 PWM时基

#### PWM时基的工作时钟

PWM时基的工作时钟PT\_CLK由DSP系统时钟fdsys分频得到。分频系数由寄存器PWMTCON中的PTCLKPS设置值决定，可以是fdsys的1/1、1/2、1/4、1/8、1/16、1/32、1/64、1/128倍。时基工作时钟PT\_CLK的周期时间由下式决定：

$$T_{PT\_CLK} = 2^{PTCLKPS} \times T_{dsys}$$

当用户修改PWMTCON寄存器的PWMEN、PTCLKPS和PTMOD位为新设置值时，硬件将自动清除PT\_CLK使用的内部分频器值，并开始重新分频计数，以保证PT\_CLK与DSP系统时钟的相位关系。

PWM时基工作时钟周期T<sub>PT\_CLK</sub>是PWM计数器、PWMTPRD、PWMWDxA/B和PWMECMPx寄存器的基本计数单位。

#### PWM时基周期和计数周期

PWM的时基周期T<sub>TBPRD</sub> (Time-Base Period) 是指PWMTPRD寄存器确定的时间周期，由下式决定：

$$T_{TBPRD} = (PWMTPRD + 1) \times T_{PT\_CLK}$$

PWMEN控制位软件置1后，PWM计数器从0开始对PT\_CLK进行增计数，根据计数值PWCNT与PWMTPRD寄存器值相等后不同的计数处理方式，PWM时基分为单事件、边沿对齐和中心对齐三种计数模式。如果PWM时基在正常运行过程中，用户通过PWMEN位软件清零来关闭时基，则PWM计数器在下一个DSP系统时钟边沿自动复位。

PWM的计数周期T<sub>CNTPRD</sub> (Counter Period) 是指PWM计数器从0开始增计数后直至计数值重新归0的时间周期。在单事件和边沿对齐计数模式下，计数周期等同于时基周期；在中心对齐计数模式下，计数周期是时基周期的2倍。

#### 单事件模式 (One-Shot mode)

当PWMTCON寄存器的PTMOD = 10或11时，PWM时基工作于单事件计数模式。

单事件计数模式中，PWM计数器PWCNT在PWMEN置1时从0开始向上增计数。当PWCNT值与PWMTPRD寄存器相等后，PWCNT将在下一个PWM工作时钟边沿复位，并且由硬件将PWMEN位清零，以停止时基。PWCNT归零时若PRZIE = 1，将触发计数归零中断信号。

单事件计数过程中，如果用户对PWMEN位软件清零，PWM计数器在下一个DSP系统时钟边沿自动复位，PWM时基自动结束计数。

单事件计数模式如图9-3-3所示，其中CNT\_MPR信号是PWCNT值与PWMTPRD寄存器值匹配指示信号；ZERO\_INT是计数归零指示信号。

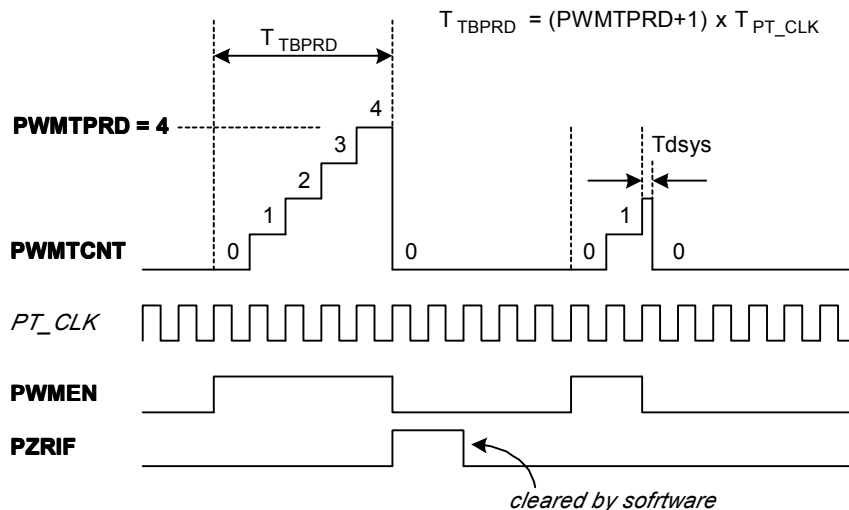


图9-3-3 单事件计数模式





### 边沿对齐模式 (Edge-Aligned mode)

当PWMTCON寄存器的PTMOD = 00时，PWM时基工作于边沿对齐计数模式。

边沿对齐计数模式中，PWM计数器PWMCNT在PWMEN置1时从0开始向上增计数。当PWMCNT值与PWMTPRD寄存器值相等后，PWMCNT将在下一个PWM工作时钟边沿复位为0。如果PWMEN位保持为1，则计数器继续从0向上增计数，并周而复始重复上述过程。PWMCNT归零时若PRZIE=1，将触发计数归零中断信号。

边沿对齐计数过程中，如果用户对PWMEN位软件清零，PWM计数器在下一个DSP系统时钟边沿自动复位，PWM时基自动结束计数。

边沿对齐计数模式如图9-3-4所示，其中**CNT\_MPR**信号是PWMCNT值与PWMTPRD寄存器值匹配指示信号；**ZERO\_INT**是计数归零指示信号。

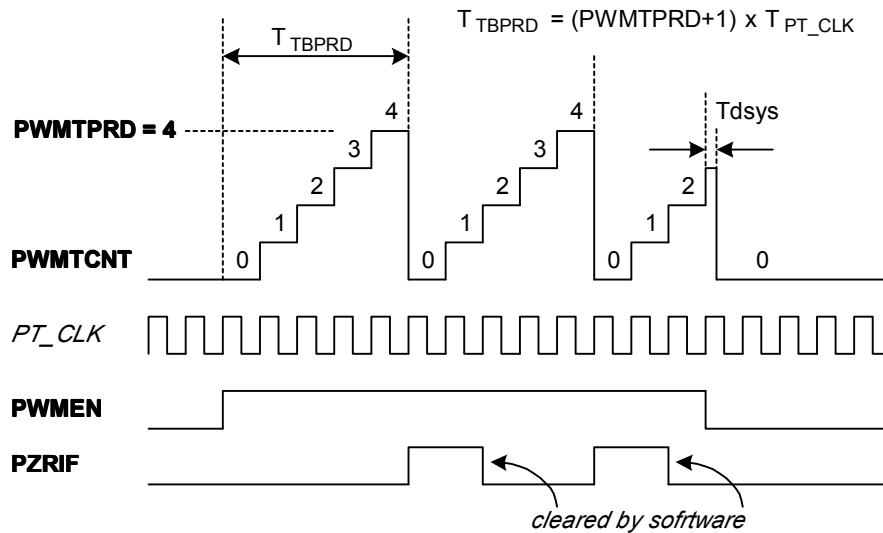


图9-3-4 边沿对齐计数模式

### 中心对齐模式 (Center-Aligned mode)

当PWMTCON寄存器的PTMOD = 01时，PWM时基工作于中心对齐计数模式。

中心对齐计数模式中，PWM计数器PWMCNT在PWMEN置1时从0开始向上增计数。当PWMCNT值与PWMTPRD寄存器值匹配时，PWMCNT将在下一个PWM工作时钟边沿开始向下减计数，当计数值为减为0时，PWMCNT将在下一个PWM工作时钟边沿开始向上增计数。如果PWMEN位保持为1，则计数器从0继续向上计数，并周而复始重复上述过程。PWMCNT归零时若PRZIE=1，将触发计数归零中断信号；PWMCNT与PWMTPRD匹配后若PFLIE=1，将触发计数满值中断信号。

中心对齐计数过程中，如果用户对PWMEN位软件清零，PWM计数器在下一个DSP系统时钟边沿自动复位，PWM时基自动结束计数。

中心对齐计数模式如图9-3-5所示，其中**CNT\_MPR**信号是PWMCNT值与PWMTPRD寄存器值匹配指示信号；**ZERO\_INT**是计数归零指示信号；**FULL\_INT**是计数满值指示信号。

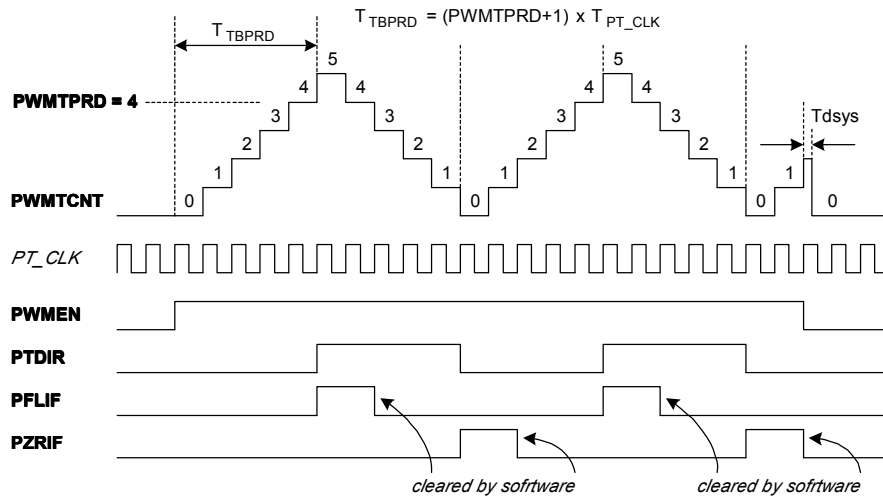


图9-3-5 中心对齐计数模式

### PWMTPRD的缓存寄存器

PWMTPRD寄存器使用内部缓存寄存器以保证PWMTPRD寄存器的修改不会导致PWM信号的毛刺。PWM计数器固定与内部的缓存寄存器进行比较，以控制实现不同的时基模式。PWMTPRD寄存器的内部缓存寄存器不能被用户访问到。

PWMEN = 0时，修改PWMTPRD的设置值，该值将直接更新到内部缓存寄存器。

PWMEN = 1时，修改PWMTPRD的设置值，该值将在当前PWM计数周期结束时（PWM计数器归零时）更新到内部缓存寄存器，新设置值在下一个计数周期有效（在单事件和边沿对齐计数模式下，计数周期等同于时基周期；在中心对齐计数模式下，计数周期是时基周期的2倍）。



### 9.3.5 PWM波形生成器

#### 波形发生单元

PWM模块包含3个独立的波形发生单元，每个PWM波形发生单元使用一对占空比寄存器PWMWDx/A/B (x = 0-2) 与PWM计数值进行比较，产生对应的一对PWM波形输出信号PWMWx/A/B。

以PWM0波形发生单元为例，其基本结构如下图所示：

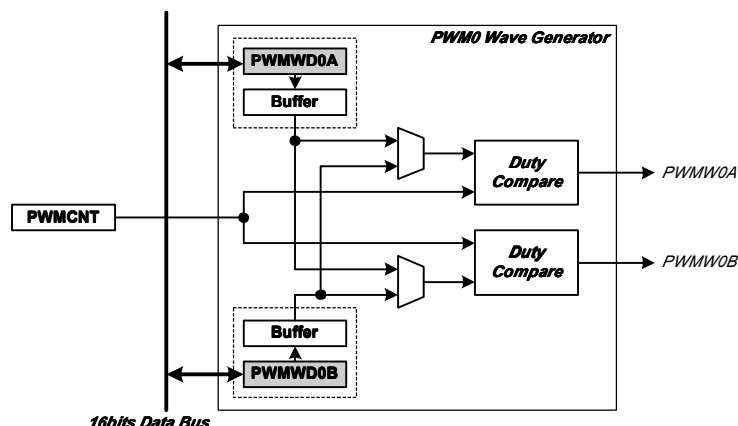


图9-3-6 PWM0波形发生单元结构框图

占空比寄存器PWMWD0A和PWMWD0B均存在内部缓存寄存器，波形发生单元内部的两个比较器直接使用缓存寄存器与PWM计数器进行比较，并产生一对输出比较结果信号PWMW0A/B：

- (1) 当PWM计数值小于占空比缓存寄存器值时，PWMW0A/B输出1，该时间区段称为无效时区（Inactive Time-zone）；
  - (2) 当PWM计数值等于或大于占空比缓存寄存器时，PWMW0A/B输出0，该时间区段称为有效时区（Active Time-zone）；
- 占空比寄存器的设置值对应无效时区的计数宽度，实际输出波形中无效时区的占空比值按照下式计算：

$$Duty = \frac{PWMWDx/A/B}{PWMTPRD + 1} \times 100\%$$

当PWMWDx/A/B = 0时，占空比为0；当PWMWDx/A/B = PWMTPRD+1时，占空比为100%。

两路输出信号PWMW0A/B的比较器使用哪个占空比缓存寄存器作比较，由PWM波形输出模式控制位和对称性控制位决定。

PWM模块的3个波形发生单元使用独立的输出模式控制位（PWMODx, x = 0-2）分别控制各发生单元两路输出波形的输出模式，并共用一个对称性控制位（PWSYM）控制波形的对称性。

#### PWM波形输出模式

3个PWM波形发生单元产生3对PWM波形，分别为PWM0A/B、PWM1A/B和PWM2A/B。每对PWM波形发生单元的输出模式由PWMODx (x = 0-2) 对应位分别控制，可以配置为两种输出模式：独立输出模式和互补输出模式。

##### (1) 独立输出模式

独立输出模式下，同一个PWM波形发生单元的两路PWM波形完全独立，占空比寄存器分别控制相应的比较器输出独立的PWM波形，即PWMWD0A/B分别控制PWM0A/B，PWMWD1A/B分别控制PWM1A/B，PWMWD2A/B分别控制PWM2A/B。

##### (2) 互补输出模式

互补输出模式下，同一个PWM波形发生单元的两路比较器使用相同的占空比寄存器，产生相同的PWMWx/A/B信号。

当PWM时基工作在单事件或边沿对齐计数模式时，互补输出的PWM波形发生单元仅使用其对应的PWMWDxA (x = 0-2) 占空比寄存器，PWMWDxB占空比寄存器不被使用。

当PWM时基工作在中心对齐计数模式时，由波形对称性控制位PWSYM设置输出波形的对称性：

PWSYM = 0时，在PWM计数器增计数和减计数过程中，互补输出的PWM波形发生单元均使用其对应的PWMWDxA (x = 0-2) 占空比寄存器作比较，PWMWDxB (x = 0-2) 占空比寄存器不被使用。

PWSYM = 1时，在PWM计数器增计数过程中，互补输出的PWM波形发生单元使用其对应的PWMWDxA (x = 0-2) 占空比寄存器值作比较，在PWM计数器减计数过程中，则使用其对应的PWMWDxB (x = 0-2) 占空比寄存器值作比较。



单事件模式下的PWM波形

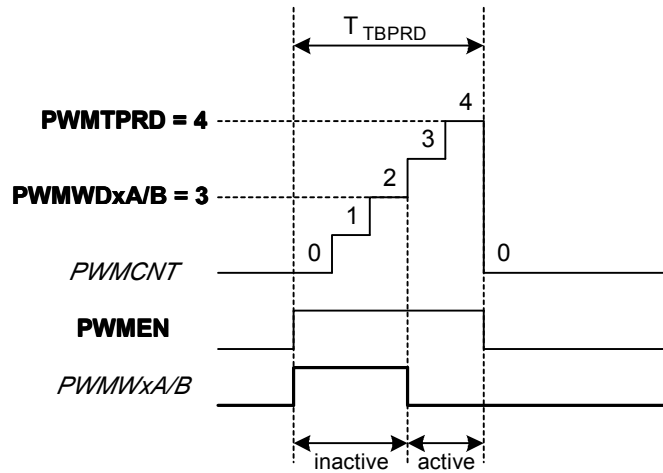


图9-3-7 单事件模式下的PWM波形 (x = 0-2)

单事件模式下的PWM波形如上图所示，PWM计数器值与占空比寄存器设置值比较，计数器值小于或等于占空比设置值的无效时区内，PWMWx/A/B输出为1，计数器值大于占空比设置值的有效时区内输出为0。

边沿对齐模式下的PWM波形

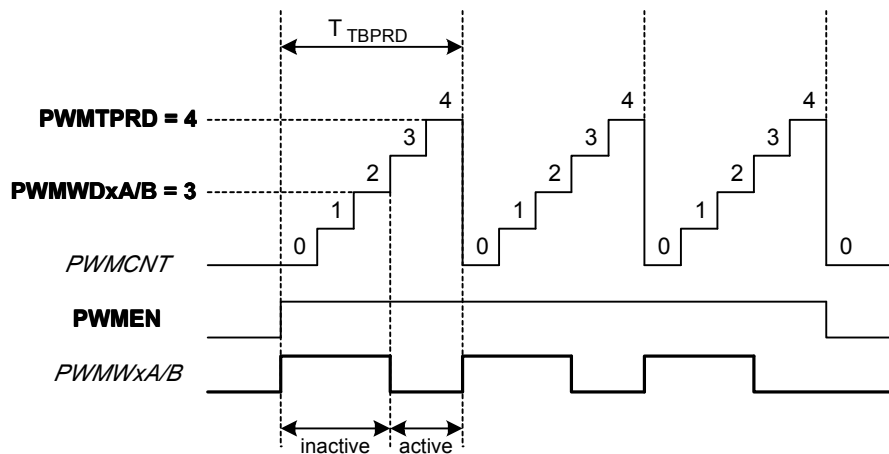


图9-3-8 边沿对齐模式下的PWM波形 (x = 0-2)

边沿对齐模式下的PWM波形如上图所示，PWM计数器值与占空比寄存器设置值比较，计数器值小于或等于占空比设置值的无效时区内，PWMWx/A/B输出为1，计数器值大于占空比设置值的有效时区内输出为0，然后以此周而复始直到PWMEN被清零。



## 中心对齐模式下的对称PWM波形

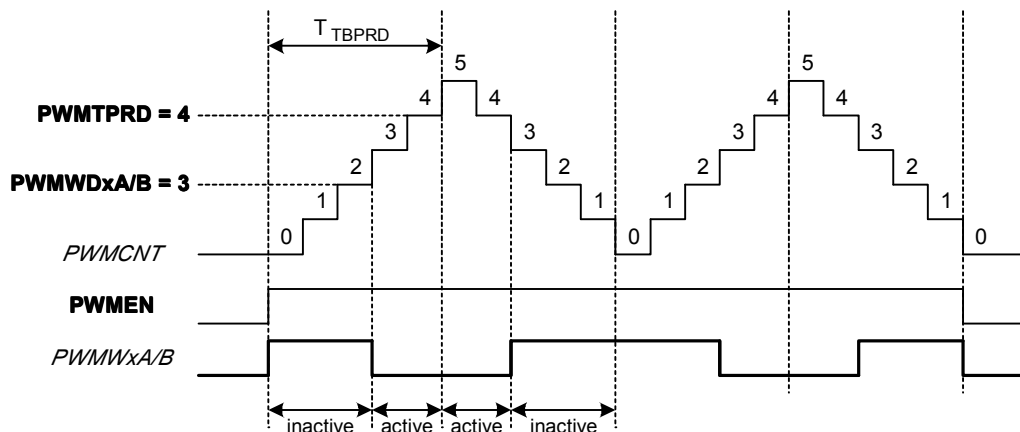


图9-3-9 中心对齐模式下的对称PWM波形 (x = 0-2)

中心对齐模式下的对称PWM波形如上图所示，该模式下PWM计数器在增计数和减计数过程都使用同一个占空比寄存器PWMWDxA/B的设置值进行比较，然后以此周而复始直到PWMEN被清零。

增计数过程中， $PWMCNT < PWMWDxA/B$ （无效时区）， $PWMWxA/B$ 输出为1， $PWMCNT \geq PWMWDxA/B$ （有效时区）输出为0。

减计数过程中， $PWMCNT > PWMWDxA/B$ （有效时区）， $PWMWxA/B$ 输出为0， $PWMCNT \leq PWMWDxA/B$ （无效时区）输出为1。

## 中心对齐模式下的非对称PWM波形（仅在互补输出时才存在）

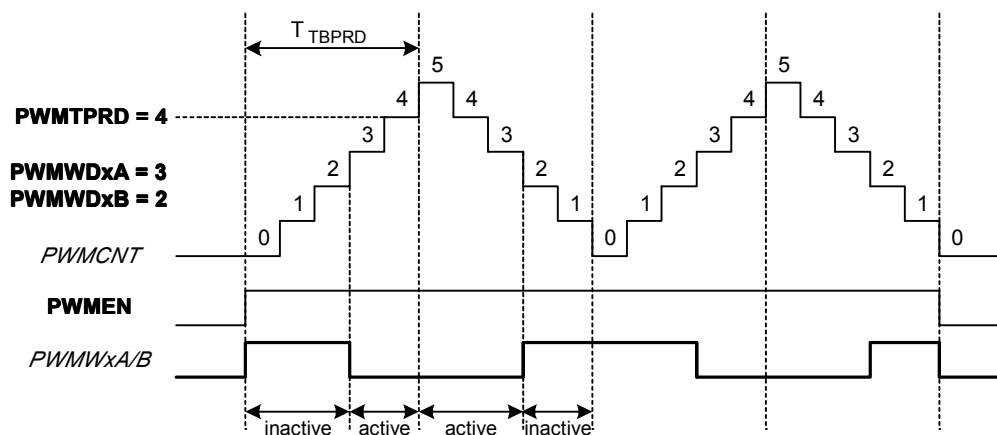


图9-3-10 中心对齐模式下的非对称PWM波形 (x = 0-2)

中心对齐模式下的非对称PWM波形如上图所示，该模式下PWM计数器在增计数过程使用占空比寄存器PWMWDxA的设置值进行比较，在减计数过程使用PWMWDxB的设置值进行比较，然后以此周而复始直到PWMEN被清零。

增计数过程中， $PWMCNT < PWMWDxA$ （无效时区）， $PWMWxA/B$ 均输出为1， $PWMCNT \geq PWMWDxA$ （有效时区）则输出为0。

减计数过程中， $PWMCNT > PWMWDxB$ （有效时区）， $PWMWxA/B$ 均输出为0， $PWMCNT \leq PWMWDxB$ （无效时区）则输出为1。



### PWMWDxA/B的缓存寄存器

6个占空比寄存器PWMWDxA/B都内建缓存寄存器，PWM波形发生单元固定使用缓存寄存器与PWM计数值进行比较。占空比缓存寄存器不能被用户访问到。

PWMEN = 0时，修改PWMWDxA/B的设置值，该值将直接更新到缓存寄存器。

PWMEN = 1时，修改PWMWDxA/B的设置值，该值更新缓存寄存器的方式受PWMWCON寄存器的PVDLD位控制。

(1) PVDLD = 0，所有占空比寄存器PWMWDxA/B在下一个PWM计数周期的开始时刻更新到缓存寄存器，称为周期更新。

(2) PVDLD = 1，所有占空比寄存器PWMWDxA/B在写入的同时立即更新到对应的缓存寄存器，称为立即更新。

立即更新方式下，根据PWMWDxA/B寄存器写入时刻PWM计数值与占空比寄存器新旧设置值的关系，处理结果说明如下：

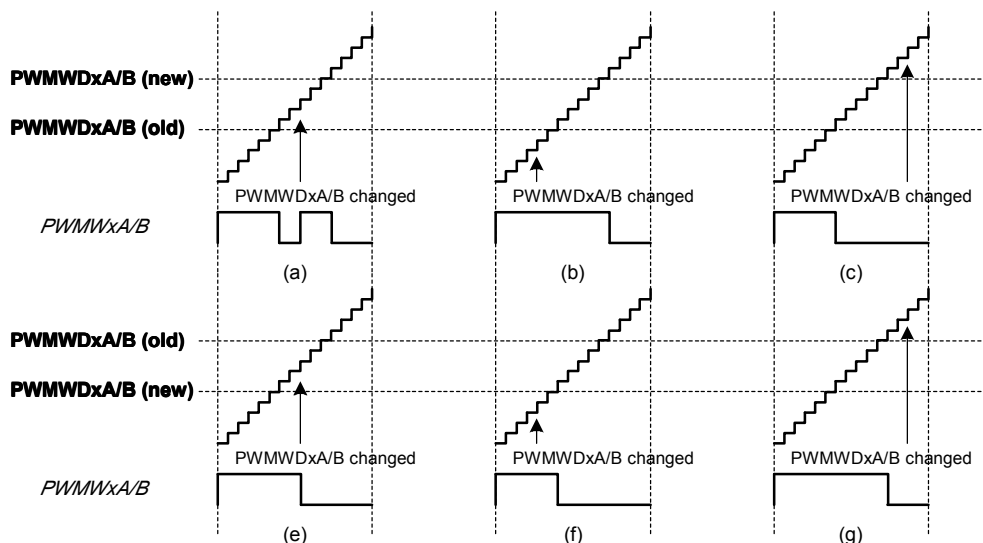


图9-3-11 增计数过程的PWMWDxA/B缓存寄存器更新 (x = 0-2)

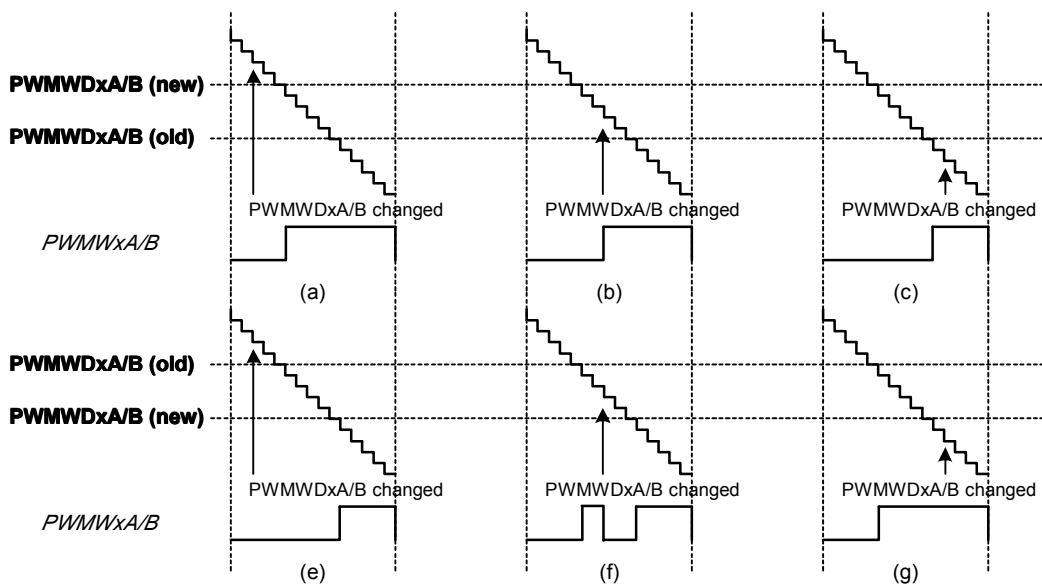


图9-3-12 减计数过程的PWMWDxA/B缓存寄存器更新 (x = 0-2)



### 9.3.6 死区 & 极性控制

#### 死区 & 极性控制单元

当PWM波形发生器工作在互补输出模式时，死区控制逻辑能防止互补输出的两个PWM信号有效时区的互相交叠，以保证实际应用中PWM信号驱动的一对互补功率开关管不会同时导通。

输出极性控制逻辑通过6个独立的寄存器位，分别控制6路PWM波形的输出特性，为用户提供了最大的灵活性。

PWM模块提供了3个死区和极性控制单元，分别控制每对PWM输出波形，每个控制单元的逻辑框图如下图所示：

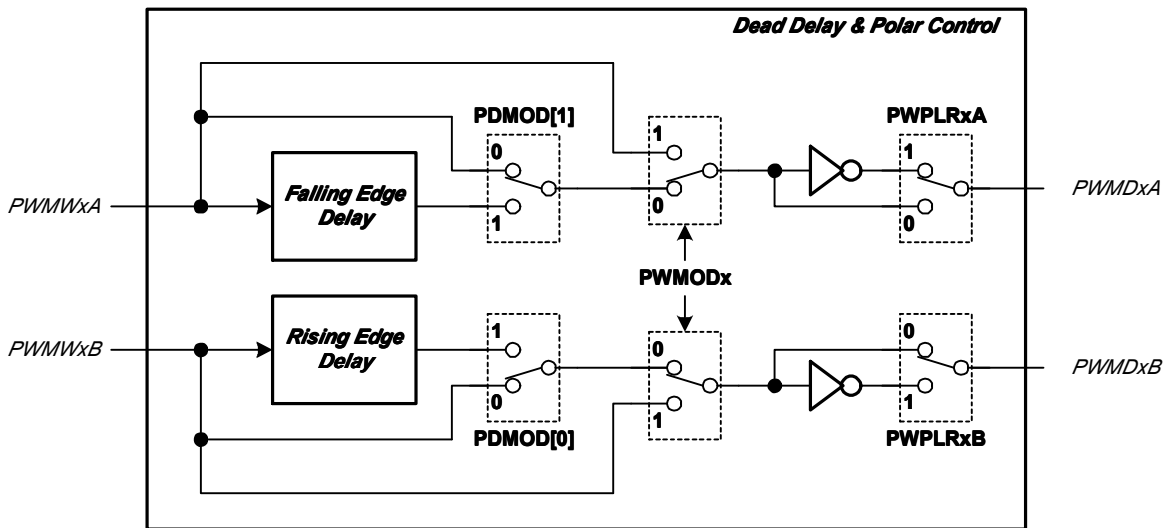


图9-3-13 死区 & 极性控制单元结构 (x = 0-2)

#### 死区控制

只有当PWM模块工作在互补输出模式时，死区控制才有效。从PWM波形发生单元产生的一对PWM波形信号通过死区控制逻辑，可以在上升沿和下降沿分别插入死区延迟时间后再输出。

波形控制寄存器PWMWCON的PDMOD控制位用于选择死区控制方式，2位宽的设置值共提供了4种死区控制方式，该设置值对所有设置为互补输出模式的PWM波形均有效。

每个死区控制电路使用两个独立的16位时间设置寄存器PWMxDTR和PWMxDTF (x = 0-2) 分别控制上升沿和下降沿死区延迟时间，设置值以DSP系统时钟时间为计数单位，实际的上升沿和下降沿延迟时间由下式决定：

$$\begin{aligned} T_{DTRx} &= (PWMxDTR + 1) \times T_{DSYS} \\ T_{DTFx} &= (PWMxDTF + 1) \times T_{DSYS} \end{aligned} \quad (x = 0-2)$$

#### 极性控制

在每一个PWM时基周期，PWM波形发生器产生的PWM波形在时间上分为无效时区和有效时区两个时间区段，6路PWM输出波形在每个时间区段内的输出信号电平由极性控制逻辑实现，以实现无效时区输出高电平、有效时区输出低电平或无效时区输出低电平、有效时区输出高电平的不同设置。

极性控制逻辑使用PWMWCON寄存器的6个设置位PWPLRxA/B (x = 0-2) 独立控制6路PWM输出波形的极性。极性控制与PWM波形的输出模式无关，用户可以根据应用的需要灵活的设置每对PWM波形的输出模式以及每路PWM输出波形的极性。



### 死区 & 极性控制输出波形

死区&极性控制方式与PWM的时基计数模式和波形输出模式均无关。为描述方便，以电机应用中最为常见的中心对齐计数模式下的互补输出对称PWM波形为例，说明死区&极性控制逻辑的输出波形。

如下图所示，PWMW0A/B是PWM波形发生单元产生的信号，当设置PDMOD = 3，同时使能上升沿和下降沿死区延迟时，该图提供了4种不同极性控制值下的PWMD0A/B输出波形对。

其他时基计数方式下的死区&极性控制输出波形与此类似。

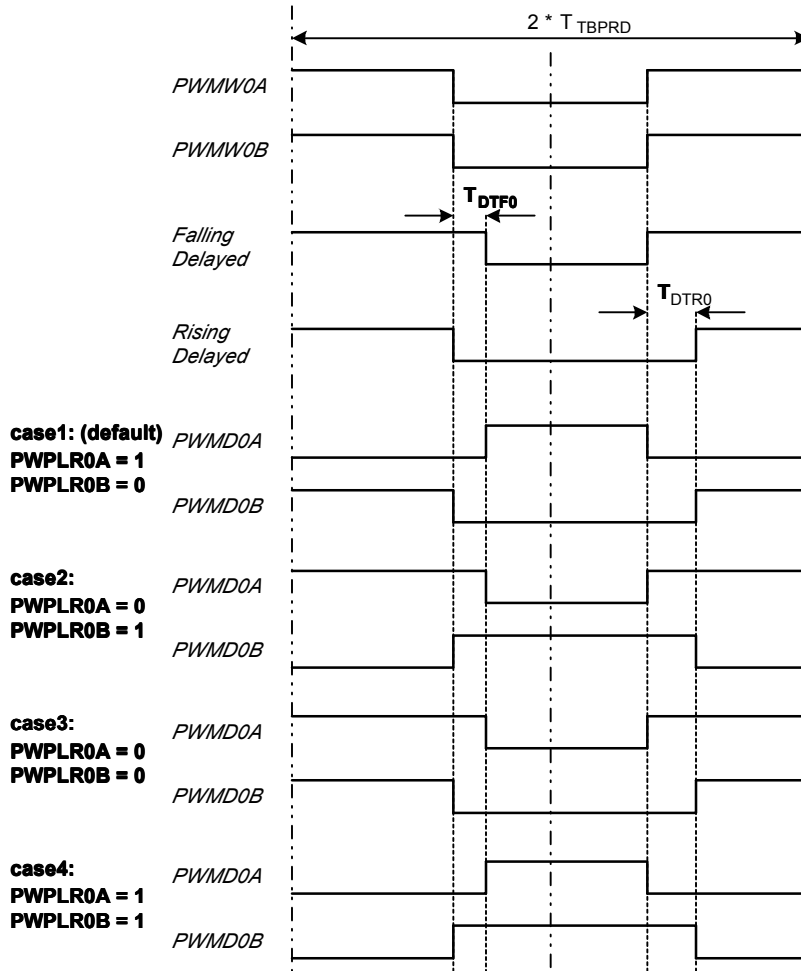


图9-3-14 死区 & 极性控制波形

### PWPLRxA/B的缓存寄存器

PWMWCON寄存器的极性控制位PWPLRxA/B (x = 0-2) 内建缓存寄存器，极性控制电路固定使用PWPLRxA/B的内部缓存寄存器控制输出波形。PWPLRxA/B的内部缓存寄存器不能被用户访问到。

用户通过设置PWMWCON寄存器的PWPLRxA/B控制位来修改输出波形的极性：

PWMEN = 0时，修改PWPLRxA/B控制位的设置值，该值将直接更新到缓存寄存器。

PWMEN = 1时，修改PWPLRxA/B控制位的设置值，该值将在当前PWM计数周期结束时（PWM计数器归零时）更新到内部缓存寄存器，新设置值在下一个计数周期有效（在单事件和边沿对齐计数模式下，计数周期等同于时基周期；在中心对齐计数模式下，计数周期是时基周期的2倍）。





### 9.3.7 保护输入& PWM输出控制

#### 保护输入检测

SH99F100含有五个保护输入引脚PDPINT0-4。每个保护输入引脚都通过独立的输入保护检测电路来检测并产生高电平有效的PWM保护信号FLTy\_SIG (y = 0-4)。PWM保护信号FLTy\_SIG用于控制PWM信号输出口PWMxA/B的输出状态，以实现故障保护功能。

保护输入检测电路的结构框图如下图所示：

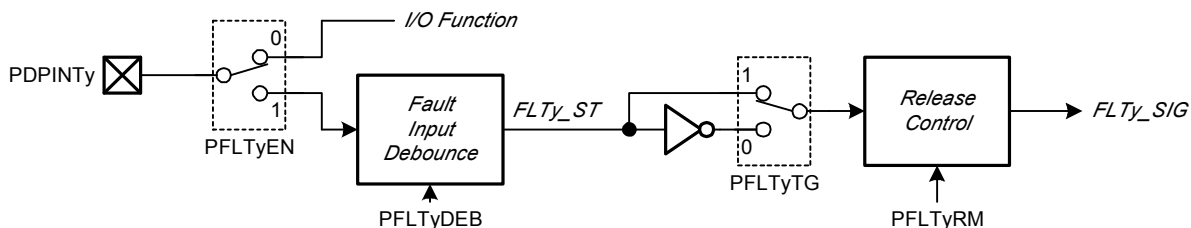


图9-3-15 保护输入检测电路的结构框图 (y = 0-5)

保护输入引脚与普通I/O功能复用，用户通过PWMFCON寄存器的PFLTyEN (y = 0-4) 控制位，可以独立控制每个保护输入引脚用于保护输入功能或用于普通I/O功能。

#### 保护输入引脚的滤波设置

为提高保护输入引脚的抗干扰性，保护输入检测电路内建滤波电路。当保护输入引脚的由低电平变化为高电平后，高电平至少要保持滤波设置的时间长度，才能使滤波输出信号FLTy\_ST变化为高电平。反之，当该电平保护输入引脚的由高电平变化为低电平后，低电平至少要保持滤波设置的时间长度，才能使滤波输出信号FLTy\_ST变化为低电平。每个保护输入引脚滤波时间的设置，由PWMFDEB寄存器的PFLTyDEB (y = 0-4) 控制位分别控制。

以PDPINT0保护输入引脚为例，滤波电路的输入输出信号波形如下图所示：

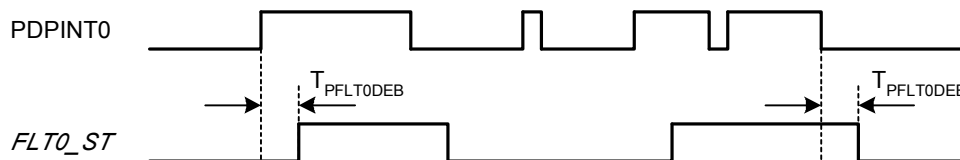


图9-3-16 保护输入滤波器输入输出波形

保护输入引脚内建的滤波电路会延迟PWM输出波形对故障信号的响应，因此用户需要根据实际需要合理设置滤波时间。

#### 保护输入的触发电平和状态指示

PWMFSTAT寄存器的PFLTyTG (y = 0-4) 控制位，用来设置保护输入的触发电平。用户根据外输入故障信号的电平特性，可以设置为低电平触发或高电平触发。当保护输入检测电路检测到满足触发电平要求的保护输入信号后，会产生高电平有效的PWM保护信号FLTy\_SIG (y = 0-4)。

PWMFSTAT寄存器的PFLTyST (y = 0-4) 位是只读位，用来指示保护输入引脚的滤波输出信号FLTy\_ST的电平状态。

#### 保护输入的故障恢复

当保护输入的故障信号被检测到后，PWM保护信号FLTy\_SIG (y = 0-4) 会立即将其对应控制的PWM口输出控制位PWMOExA/B (x = 0-2) 清零，以保证PWM口输出高阻态。当故障信号消失后，其对应控制的PWM口恢复PWM波形输出的方式由PWMFCON寄存器的PFLTyRM (y = 0-4) 位控制。

PFLTyRM = 0，工作在软件恢复模式。故障信号存在时，用户无法对其对应的PWMOExA/B控制位置1。当故障信号消失后，用户可以对PWMOExA/B控制位置1，硬件将在其后的第一次PWM计数器归零时恢复PWM波形输出。

PFLTyRM = 1，工作在硬件恢复模式。故障信号存在时，用户无法对其对应的PWMOExA/B控制位置1。当故障信号消失后，无需用户对PWMOExA/B控制位置1，硬件将自动在其后的第一次PWM计数器归零时恢复PWM波形输出。



## PWM输出控制

5个保护输入引脚产生的PWM保护信号可以分别控制6路PWM口输出信号，PWFMFMAPx (x = 0-2) 寄存器组可以设置各保护信号与PWM输出信号的控制关系。

PWFMFMAP0寄存器的PFyMAP0A (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM0A口输出波形进行控制；  
 PWFMFMAP0寄存器的PFyMAP0B (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM0B口输出波形进行控制；  
 PWFMFMAP1寄存器的PFyMAP1A (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM1A口输出波形进行控制；  
 PWFMFMAP1寄存器的PFyMAP1B (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM1B口输出波形进行控制；  
 PWFMFMAP2寄存器的PFyMAP2A (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM2A口输出波形进行控制；  
 PWFMFMAP2寄存器的PFyMAP2B (y = 0-4) 控制位分别控制5路保护输入信号是否对PWM2B口输出波形进行控制。  
 以PWM0A口为例，其输出控制电路如下图所示：

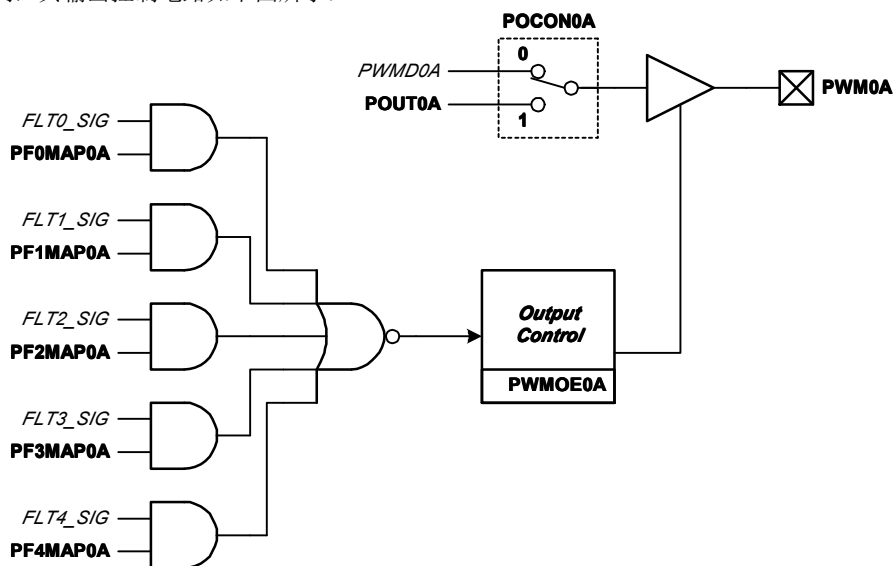


图9-3-17 PWM0A口输出控制电路的结构框图

当输入保护的故障信号发生时，其对应保护PWM口的输出控制位PWMOE<sub>x</sub>/A/B (x = 0-2) 立即清零，以保证该PWM口输出高阻态。当输入保护的故障信号消失后，PWM口根据恢复控制模式的不同，在下一个PWM计数器归零时恢复波形输出。

为提高控制的灵活性，当PWMOE<sub>x</sub>/A/B = 1正常输出PWM波形时，用户可以通过PWMOCON寄存器实现手动控制功能。

PWMOCON寄存器的POCON<sub>x</sub>/A/B控制位用于选择PWM输出波形或手动控制，当设置为手动控制状态时，PWM<sub>x</sub>/A/B口的输出直接由POUT<sub>x</sub>/A/B控制位决定。

### 9.3.8 事件触发控制

#### DTimer事件触发

SH99F100内建四个DTimer<sub>x</sub> (x = 0-3)，PWM模块的事件触发功能能产生对每个DTimer独立的触发信号。DTimer的触发控制由PWMECON寄存器的PEMOD决定：

PEMOD = 100，当PWM计数器值归零时，硬件自动置位TMR0TR，触发定时器DTimer0运行；

PEMOD = 101，当PWM计数器值归零时，硬件自动置位TMR1TR，触发定时器DTimer1运行；

PEMOD = 110，当PWM计数器值归零时，硬件自动置位TMR2TR，触发定时器DTimer2运行；

PEMOD = 111，当PWM计数器值归零时，硬件自动置位TMR3TR，触发定时器DTimer3运行。

#### ADC事件触发

PWMECON寄存器的PEMOD控制位为001时，可以使用PWM模块的ADC事件触发功能。此时三个事件触发比较器的工作方式由PWMECON寄存器的PECMP<sub>x</sub>EN (x = 0-2) 控制位决定。

PECMP<sub>x</sub>EN (x = 0-2) 控制位设置PWMECMP<sub>x</sub>事件比较器的工作模式。事件比较功能有效时，当PWM计数器值等于相应的PWMECMP<sub>x</sub>设置值，硬件将在当前PWM工作时钟边沿立即产生一个AD触发信号，自动触发AD采样（根据PECMP<sub>x</sub>EN选择置位SOC1或SOC2）。



PWMECMP0事件比较器还使用PECMP0DIR位控制在增计数、减计数过程比较，或者在增减计数过程均进行比较。该控制位仅在PTMOD = 01（中心对齐计数模式）时有效。

利用事件触发比较器PWMECMPx可以实现在一个PWM时基周期内选择1至3个点自动触发AD采样（可选择置位SOC1或SOC2）。

当PWMECMPx事件比较器均有效时，三种时基计数模式下，ADC事件触发的示意波形如下图所示：

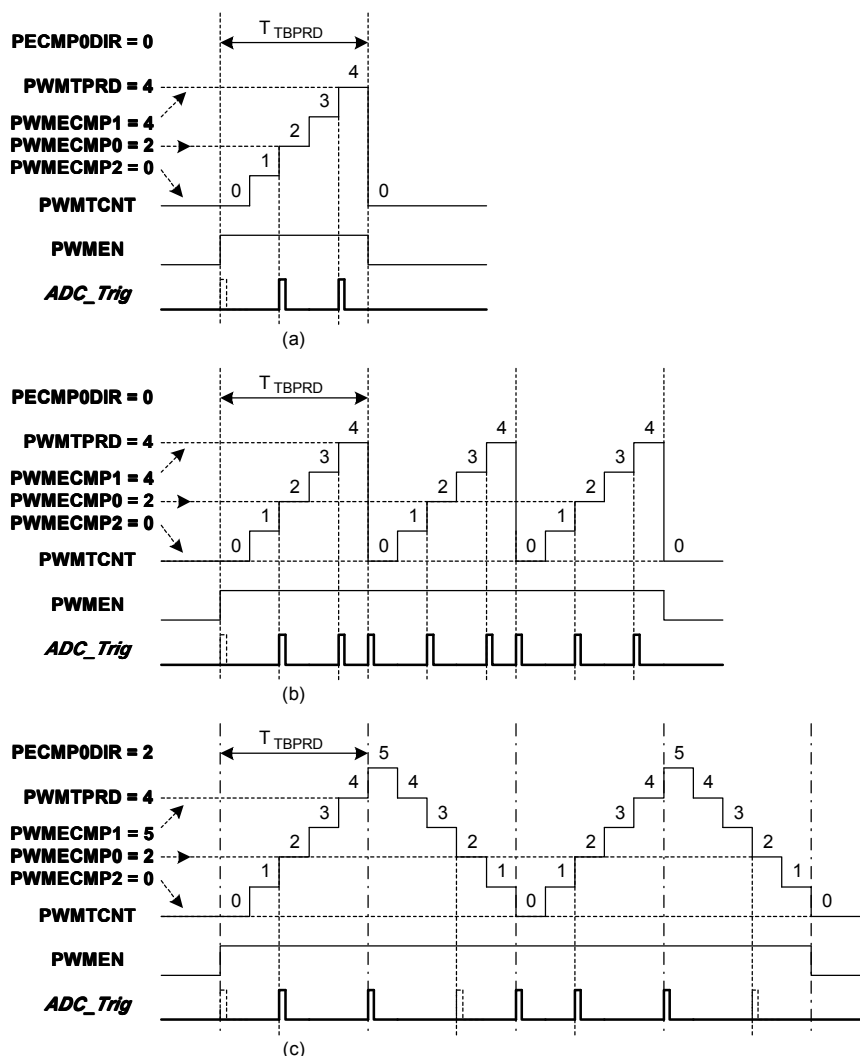


图9-3-18 ADC事件触发波形示意图

### PWMECMPx的缓存寄存器

PWMECMPx (x = 0-2) 寄存器内建缓存寄存器，事件触发比较器固定使用内部缓存寄存器进行比较输出。PWMECMPx的内部缓存寄存器不能被用户访问到。

用户通过设置PWMECMPx寄存器来修改事件触发比较值：

PWMEN = 0时，修改PWMECMPx的设置值，该值将直接更新到缓存寄存器。

PWMEN = 1时，修改PWMECMPx的设置值，该值将在当前PWM计数周期结束时（PWM计数器归零时）更新到内部缓存寄存器，新设置值在下一个计数周期有效（在单事件和边沿对齐计数模式下，计数周期等同于时基周期；在中心对齐计数模式下，计数周期是时基周期的2倍）。

PWMECON寄存器的PECMP0DIR控制位也内建缓存寄存器，其更新方法与PWMECMPx (x = 0-2) 相同，以保证同步。



### 9.3.9 PWM中断

#### PWM时基中断

PWM时基可以产生2个中断源：计数归零中断和计数满值中断。

计数归零中断是PWM计数器从非0值变化为0值时产生的中断。寄存器PWMIE的PZRIE位控制该中断的使能，当PZRIE = 1且中断发生时，硬件自动置位PZRIF标志位，用户向PZRIF位写0可以清除该中断标志。

计数满值中断是PWM计数器PWCNT = (PWMPRD + 1)时产生的中断。寄存器PWMIE的PFLIE位控制该中断的使能，当PFLIE = 1且中断发生时，硬件自动置位PFLIF标志位，用户向PFLIF位写0可以清除该中断标志。

计数归零中断在三种时基计数模式下均可能产生。计数满值中断只有在中心对齐计数模式下才可能产生。

#### PWM保护输入中断

SH99F100的五个保护输入引脚PDPINT0-4经保护输入检测电路可以产生故障保护信号。当保护信号由低电平变为高电平时，硬件自动置位相应的PDPIEy (y = 0-4) 标志位，用户向相应的标志位写0可以清除相应的中断标志。

PWMIE寄存器的PDPIEy (y = 0-4) 位分别控制各保护输入引脚的中断，当相应的中断使能位有效时，硬件置位PDPIEy (y = 0-4) 标志位的同时，会自动产生PWM保护输入中断。

### 9.3.10 后分频系数

边沿对齐模式和中心对齐模式下，有时不需要在每个计数周期内（边沿对齐计数模式下，计数周期等同于时基周期；中心对齐计数模式下，计数周期是时基周期的2倍）均触发硬件动作。利用PWMTCON寄存器中的后分频系数PPOST可以设置PWM时基中断、缓存更新以及ADC和DTimer事件触发产生的间隔。

每次PWMEV位置1后，后分频计数器从0开始，以PWM计数周期为基本计数单位进行增计数，即每个PWM计数周期结束时，后分频计数器自动加1。当分频计数值等于分频系数设置值时，该计数周期内发生的PWM时基中断、缓存更新以及ADC和DTimer事件触发均有效。

#### PWM时基中断的后分频

PWM时基的计数归零中断和计数满值中断都受后分频系数控制。

当分频计数值等于分频系数设置值时，如果中断发生条件满足，相应的中断才真正触发，标志位被置起（边沿对齐模式下，只有计数归零中断；中心对齐模式下，两种中断都可能发生）。

#### 缓存周期更新的后分频

PWMPRD、PWMPLRxA/B和PWMECMPx寄存器，以及PECMP0DIR控制位的内部缓存寄存器，使用周期更新方式，均受后分频系数的控制。只有当分频计数值等于分频系数设置值时，才在PWM计数器归零的同时更新上述寄存器的缓存寄存器。

PWMWDxA/B寄存器的内部缓存寄存器，当设置在周期更新方式时，也受后分频系数的控制。只有当分频计数值等于分频系数设置值时，才在PWM计数器归零的同时更新缓存寄存器。

#### ADC事件触发的后分频

PWMECMPx寄存器比较产生的ADC事件触发信号，也受后分频系数的控制。只有当分频计数值等于0时，PWMECMPx寄存器比较产生的ADC事件触发信号才真正有效。

#### DTimer事件触发的后分频

DTimer事件触发也受后分频系数的控制，但分频方式与其他略有差别。当分频计数值等于0时，满足DTimer事件触发条件的DTimer事件才真正有效。



后分频示意图

在不同时基模式下，上述四类受后分频系数影响的硬件工作结果如下图所示，各关键信号满足如下时序定义：

- (1) PWM时基中断标志PZRIF和PFLIF由硬件置位，软件清0；
- (2) Buffer Update是PWM计数器归零时，内部缓存寄存器的更新事件；
- (3) ADC\_Trig是PWMECMPx与PWM计数值匹配时，ADC被触发的事件；
- (4) TMRx\_Trig (x = 0-3)是PWMECMPx与PWM计数值匹配时，DTimerx被触发的事件；

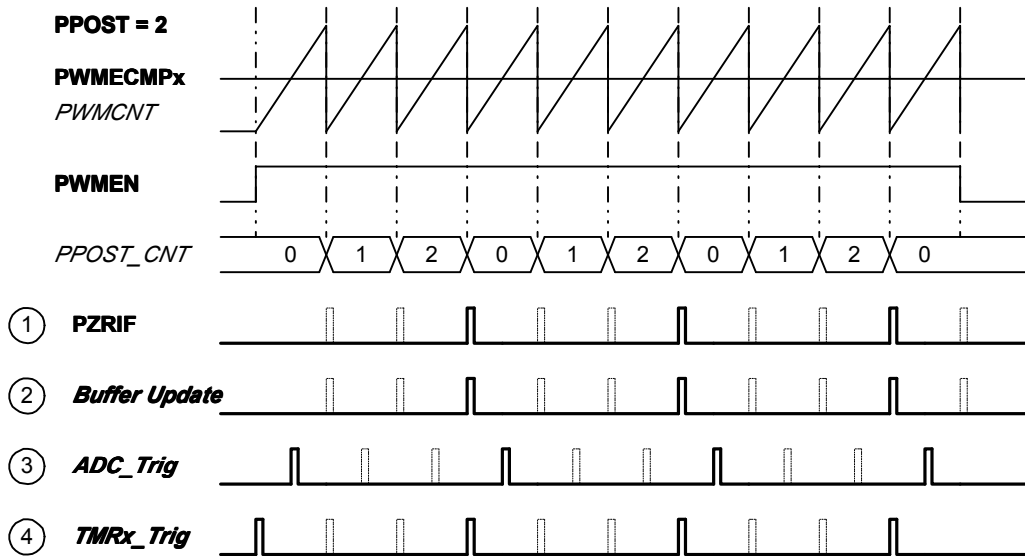


图9-3-19 边沿对齐模式下的后分频示意图

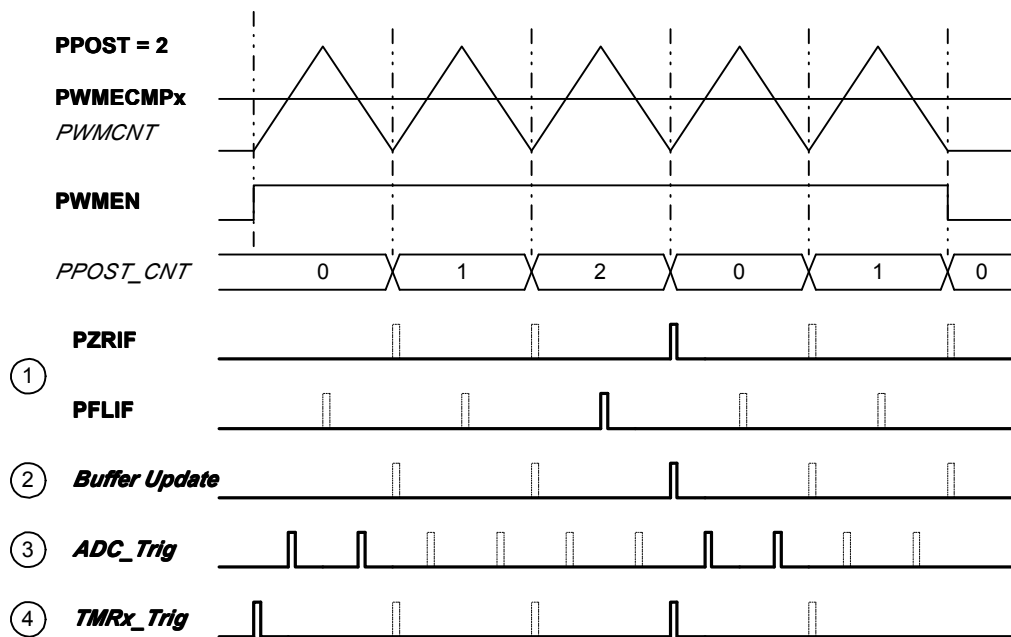


图9-3-20 中心对齐模式下的后分频示意图



### 9.4 正交编码器接口模块QEI

#### 9.4.1 特性

- 3个输入引脚：两个相位信号QEA，QEB和一个索引脉冲信号INDEX
- 16位增/减双向位置计数器
- 2种位置计数模式：x2模式和x4模式
- 输入引脚的数字滤波功能
- 内建16位QTimer定时器
- 提供低转速测量和高转速测量两种应用方案

电机控制系统中常使用增量式旋转编码器来实时监测旋转系统的位置和速度。编码器通常有3个输出信号：两个相位互差90度的方波信号QEA、QEB和一个索引信号INDEX。实际应用中，编码器通常安装在旋转轴，如电机转子上，编码器输出的QEA/QEB信号频率与电机旋转速率成正比，如一个1000线的编码器安装在转速为6000rpm的电机上，编码器将产生100KHz的QEA和QEB信号，因此通过测量QEA或QEB输出即可确定电机的转速。

大多数编码器在顺时针旋转时输出的QEA信号超前QEB信号，逆时针旋转时输出的QEA信号滞后QEB信号，每旋转一周输出一个INDEX脉冲信号指示一个绝对位置。索引信号INDEX的脉冲宽度因编码器不同而不同，大体分为门控索引信号和非门控索引信号两大类。门控索引信号的脉冲上升和下降沿必定与正交信号QEA/QEB的4个边沿中的某2个对齐，脉冲宽度为1/4、1/2或整个QEA/QEB信号的周期，而非门控索引信号的边沿与QEA/QEB的边沿无绝对联系。

编码器输出的正交信号QEA/QEB可以有四种状态，典型的增量式旋转编码器输出信号如下图9-4-1所示：

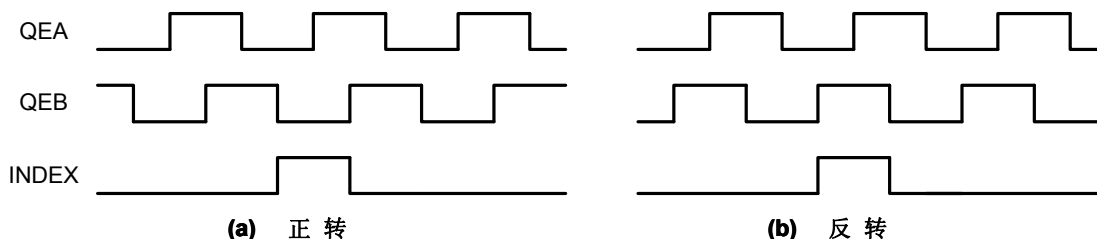


图9-4-1 增量式旋转编码器输出信号

#### 9.4.2 逻辑框图

SH99F100提供了与增量式旋转编码器连接的接口模块QEI，其逻辑框图如下图9-4-2所示：

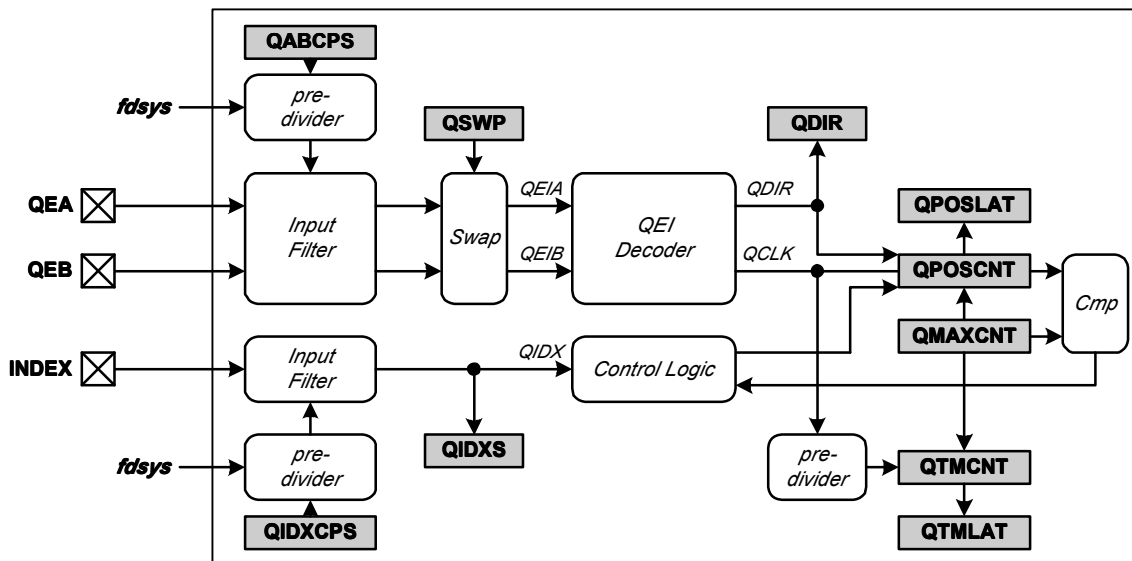


图9-4-2 正交编码器接口模块QEI逻辑框图



### 9.4.3 控制寄存器

QEI模块使用的所有控制寄存器如下表所示：

类别	缩写符号	功能说明
QEI功能控制	QEICON	QEI控制寄存器
	QFLTCN	QEI输入引脚滤波控制寄存器
	QPOSCNT	QEI位置计数器
	QPOSLAT	QEI位置计数值锁存寄存器
QTimer控制	QTMCON	QTimer控制寄存器
	QTMCNT	QTimer计数器
	QTMLAT	QTimer计数值锁存寄存器
计数控制	QCNTMAX	最大计数值设置寄存器
中断控制 & 状态	QEIINT	定时器中断控制寄存器

**Table 9.42** QEI控制寄存器QEICON

15-14	13-10	9	8	7-6	5	4	3	2	1-0
R/W-0	R-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QPEMU	Reserved	QPDIR	QIDXS	Reserved	QSWAP	QIDXEN	QEIEN	QPCAPEN	QPMOD

位编号	位符号	说明
15-14	QPEMU	仿真挂起时位置计数器状态控制位（仅在QEIEN=1时有效） 00: 挂起时，位置计数器QPOSCNT立即停止计数 01: 挂起时，位置计数器将增计数到QCNTMAX或减计数到0后再停止 1x: 挂起后位置计数器继续运行，不受挂起影响(x=0,1)
13-10	Reserved	保留位
9	QPDIR	编码器旋转方向状态位 0: QEA信号相位落后QEB，编码器反向旋转，位置计数器减计数 1: QEA信号相位超前QEB，编码器正向旋转，位置计数器增计数
8	QIDXS	INDEX输入信号（QEI正交解码器输入位置）电平状态指示位 0: INDEX输入低电平 1: INDEX输入高电平
7-6	Reserved	保留位
5	QSWAP	QEA和QEB输入交换选择位 0: QEA相与QEB相输入未交换，直接提供给正交解码器 1: QEA相与QEB相输入作交换后再提供给正交解码器
4	QIDXEN	位置计数器复位使能位（仅在QPMOD=00/10时有效） 0: 索引脉冲INDEX不能复位位置计数器 1: 索引脉冲INDEX可以复位位置计数器
3	QEIEN	QEI功能使能控制位 0: QEI功能关闭，仅QTimer作为独立的16位定时器使用 1: QEI功能打开，具体工作模式受QPMOD控制
2	QPCAPEN	DTimer1作为QEI位置计数值的锁存控制位（适用于高转速测速应用） 0: DTimer1计数溢出后，不会触发硬件锁存QPOSCNT值到QPOSLAT 1: DTimer1计数溢出后，自动触发硬件锁存QPOSCNT值到QPOSLAT
1-0	QPMOD	QEI位置计数器计数选择位（仅在QEI功能打开时有效） 00: x2计数模式，使用索引脉冲INDEX复位位置计数器 01: x2计数模式，位置计数器在0~QCNTMAX间自动循环计数 10: x4计数模式，使用索引脉冲INDEX复位位置计数器 11: x4计数模式，位置计数器在0~QCNTMAX间自动循环计数



**Table 9.43** QEI输入引脚滤波控制寄存器QFLTCON

15-8		7	6-4	3	2-0
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>QIDXFEN</b>	<b>QIDXCPS</b>	<b>QABFEN</b>	<b>QABCPS</b>

位编号	位符号	说明
<b>15-8</b>	<b>Reserved</b>	保留位
<b>7</b>	<b>QIDXFEN</b>	<b>INDEX</b> 输入引脚滤波使能位 0: 禁止INDEX输入引脚的滤波功能 1: 使能INDEX输入引脚的滤波功能
<b>6-4</b>	<b>QIDXCPS</b>	<b>INDEX</b> 引脚滤波时钟分频系数 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$
<b>3</b>	<b>QABFEN</b>	<b>QEA</b> 和 <b>QEB</b> 输入引脚滤波使能位 0: 禁止QEA和QEB输入引脚的滤波功能 1: 使能QEA和QEB输入引脚的滤波功能
<b>2-0</b>	<b>QABCPS</b>	<b>QEA</b> 和 <b>QEB</b> 引脚滤波时钟分频系数 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$

**Table 9.44** QEI位置计数器QPOSCNT

15-0	
<i>R/W-0</i>	
<b>QPOSCNT</b>	

位编号	位符号	说明
<b>15-0</b>	<b>QPOSCNT</b>	16位增/减双向计数器，计数方向由QPDIR位控制

**Table 9.45** QEI位置计数值锁存寄存器QPOSLAT

15-0	
<i>R-0</i>	
<b>QPOSLAT</b>	

位编号	位符号	说明
<b>15-0</b>	<b>QPOSLAT</b>	满足条件时，锁存QPOSCNT计数值





Table 9.46 QTimer控制寄存器 QTMCON

15-14		13-12		11	10	9	8
R/W-0		R-0		R/W-0	R/W-0	R/W-0	R/W-0
QTEMU		Reserved		QTIE	QTCAPIE	QTIF	QTCAPIF
7	6	5-4		3-0			
R/W-0	R-0	R/W-0		R/W-0			
QTEN	Reserved	QTCPS		QTEPS			

位编号	位符号	说明
15-14	QTEMU	仿真挂起时QTimer运行状态控制位（仅在QTEN=1时有效） 00: 挂起时，QTimer计数器立即停止运行 01: 挂起时，QTimer计数器将自增到溢出后停止运行 1x: QTimer不受挂起影响，将继续运行
13-12	Reserved	保留位
11	QTIE	QTimer溢出中断使能位 0: 禁止QTimer计数溢出中断 1: 允许QTimer计数溢出中断
10	QTCAPIE	QTimer捕捉事件中断使能位（仅在QEI功能打开时有效） 0: 禁止捕捉事件发生时触发中断 1: 允许捕捉事件发生时触发中断
9	QTIF	QTimer计数溢出中断标志位 0: 未发生QTimer计数溢出 1: 已发生QTimer计数溢出 写0清除该标志，写1无效
8	QTCAPIF	QTimer捕捉事件中断标志位（仅在QEI功能打开时有效） 0: 无捕捉事件发生 1: 发生捕捉事件，硬件已经锁存QTMCNT值到QTMLAT 写0清除该标志，写1无效
7	QTEN	QTimer启停控制位 0: QTimer计数停止 1: QTimer计数启动 该控制位从0设置为1后，QTimer计数器从0开始增计数
6	Reserved	保留位
5-4	QTCPS	QTimer计数时钟分频设置位 00: 1/1 $f_{DSYS}$ 01: 1/4 $f_{DSYS}$ 10: 1/16 $f_{DSYS}$ 11: 1/64 $f_{DSYS}$



续上表

位编号	位符号	说明
<b>3-0</b>	<b>QTEPS</b>	<b>QTimer</b> 捕捉事件分频设置位（仅在 <b>QEI</b> 功能打开时有效） 0000: QCLK/1 0001: QCLK/2 0010: QCLK/4 0011: QCLK/8 0100: QCLK/16 0101: QCLK/32 0110: QCLK/64 0111: QCLK/128 1000: QCLK/256 1001: QCLK/512 1010: QCLK/1024 1011: QCLK/2048 Others: QCLK/2048

**Table 9.47** QTimer计数器QTCNT

<b>15-0</b>
<i>R-0</i>
<b>QTCNT</b>

位编号	位符号	说明
<b>15-0</b>	<b>QTCNT</b>	QTimer16位增计数寄存器

**Table 9.48** QTimer计数值锁存寄存器QTMLAT

<b>15-0</b>
<i>R-0</i>
<b>QTMLAT</b>

位编号	位符号	说明
<b>15-0</b>	<b>QTMLAT</b>	满足条件时，锁存QTCNT计数值

**Table 9.49** 最大计数值寄存器QCNTMAX

<b>15-0</b>
<i>RW-1</i>
<b>QCNTMAX</b>

位编号	位符号	说明
<b>15-0</b>	<b>QCNTMAX</b>	QEI功能有效（QEIEN=1）时，作为位置计数器的最大计数值使用； QEI功能无效（QEIEN=0）时，作为QTimer的周期设置值使用



Table 9.50 QEI中断和状态寄存器QEIINT

15-11	10	9	8	7-3	2	1	0
R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0
Reserved	QEIE	QCEIE	QPEIE	Reserved	QEIF	QCEIF	QPEIF

位编号	位符号	说明
15-11	Reserved	保留位
10	QEIE	<b>QEI功能中断使能位</b> 0: 禁止QEI中断 1: 使能QEI中断
9	QCEIE	<b>INDEX信号复位位置计数器模式下，错误检测中断使能位</b> 0: 禁止错误检测中断 1: 使能错误检测中断
8	QPEIE	<b>编码器输入相位错误中断使能位，QEA/QEB信号同相或反相时视为相位错误</b> 0: 禁止相位错误中断 1: 使能相位错误中断
7-3	Reserved	保留位
2	QEIF	<b>QEI功能中断标志位，QPMOD=00/10时作为INDEX复位事件标志</b> 0: 未发生INDEX复位事件中断 1: 已发生INDEX复位事件中断 <b>QEI功能中断标志位，QPMOD=01/11时作为位置计数上溢或下溢标志</b> 0: 未发生位置计数器上溢或下溢中断 1: 已发生位置计数器上溢或下溢中断 写0清除该标志，写1无效
1	QCEIF	<b>QEI位置计数器计数错误标志位</b> 0: QEI位置计数器未发生计数错位中断 1: QEI位置计数器已发生计数错位中断 写0清除该标志，写1无效
0	QPEIF	<b>QEI解码器相位错误标志位</b> 0: 未发生输入信号相位检测错误中断 1: 已发生输入信号相位检测错误中断 写0清除该标志，写1无效



### 9.4.4 输入滤波

为提高系统抗干扰能力，QEI模块为3个输入引脚QEA、QEB和INDEX提供了内部数字滤波电路。用户通过QABFEN和QIDXFEN分别控制QEA/QEB和INDEX引脚是否使用滤波功能。

当QABFEN = 1时，引脚QEA和QEB的滤波电路有效，QEA和QEB引脚输入信号的电平必须保持至少3个滤波时钟周期的稳定状态才会使滤波输出信号发生变化。滤波时钟由DSP系统时钟fdsys分频得到，分频值通过QABCPS设置。

当QIDXFEN = 1时，引脚INDEX的滤波电路有效，INDEX引脚输入信号的电平必须保持至少3个滤波时钟周期的稳定状态才会使滤波输出信号发生变化。滤波时钟由DSP系统时钟fdsys分频得到，分频值通过QIDXCPS设置。

当QEA、QEB和INDEX引脚上采样不同的滤波设置状态时，会导致后级电路的三个输入信号的相位发生变化，用户需要合理设置滤波功能，以保证后级电路的工作结果符合设计意图。

为描述方便，后面各节均以关闭滤波电路，且3个输入信号的相位为理想情况进行说明。

### 9.4.5 正交解码器

QEI模块内建的正交解码器，用于对引脚QEA、QEB的输入信号进行解码，以确定增量式旋转编码器的工作状态。

#### 正交解码器的接口信号

从引脚QEA/QEB输入的信号经过滤波电路后首先受QSWAP位控制，再输入给正交解码器进行解码，修改QSWAP控制位可以满足不同的增量式旋转编码器的应用需要。

QSWAP = 0，引脚输入的QEA和QEB信号未作交换直接进行解码，常用于QEA相位超前QEB对应正转的旋转编码器；QSWAP = 1，引脚输入的QEA和QEB信号先作交换后再进行解码，可用于QEA相位超前QEB对应反转的旋转编码器。以下各节均以QSWAP = 0的设置为例进行说明，QSWAP=1的情况可以此类推。

正交解码器输出QCLK信号作为位置计数器QPOSCNT的计数时钟，QCLK的产生受QPMOD[1]控制，不同设置条件下，正交解码器提供了2种QCLK产生模式：x2模式和x4模式。

正交解码器输出QPDIR指示计数方向，用户通过软件读取同名寄存器可判断计数方向。QPDIR = 0，QPOSCNT对QCLK进行减计数；QPDIR = 1，QPOSCNT对QCLK进行增计数。

#### 旋转方向与计数方向

正交解码器根据输入QEA/QEB信号的状态，判断增量式旋转编码器的旋转方向。以[QEA:QEB]为状态字，可能的状态转换过程如下图9-4-3所示，其中各图符的说明如下：

F表示QEA相位超前QEB，编码器正向旋转，[QEA:QEB]状态字按照图示顺时针方向转换，该过程中QPDIR将输出1；

R表示QEB相位超前QEA，编码器反向旋转，[QEA:QEB]状态字按照图示逆时针方向转换，该过程中QPDIR将输出0；

PE表示QEA与QEB同相位或反相位（编码器正常工作时不应该出现图示对角线状态间的转换），此时解码器会判断出现相位错误，置位中断标志QPEIF。

S表示QEA与QEB的相位关系发生变化，编码器开始换方向旋转，QPDIR将从1变为0或从0变为1。

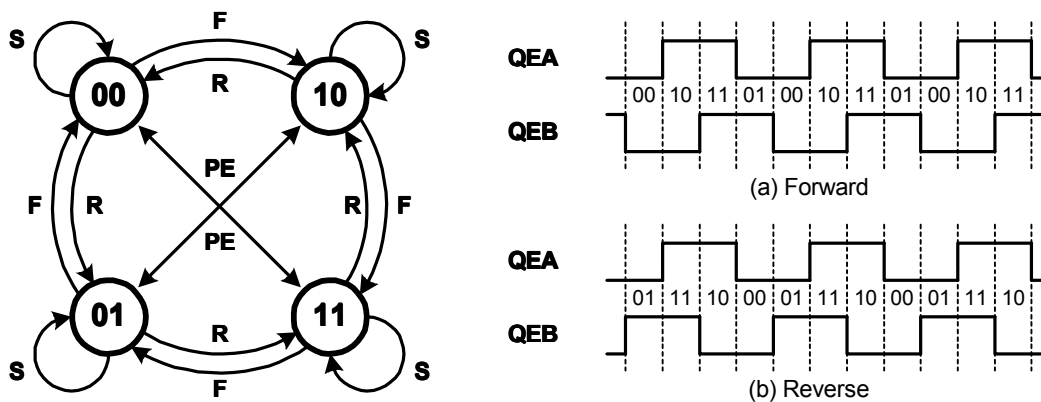


图9-4-3 QEA/QEB状态转换图



### 9.4.6 位置计数器

QEI模块内建16位位置计数器QPOSCNT，对正交解码器输出的QCLK进行增/减双向计数，计数方向由QPDIR控制，计数模式和复位方式受QPMOD控制。

QPMOD[1]控制位提供2种计数模式，也即QCLK产生模式：x2模式和x4模式。

QPMOD[0]控制位提供2种复位方式：自动循环复位和INDEX索引复位。

#### 位置计数器的计数模式

当QPMOD = 00/01时，位置计数器工作于x2计数模式，该模式下QCLK在QEA的上升沿和下降沿产生一个脉冲，因此计数器对QEA的双沿进行计数。x2计数模式下计数器的工作时序如下图9-4-4所示：

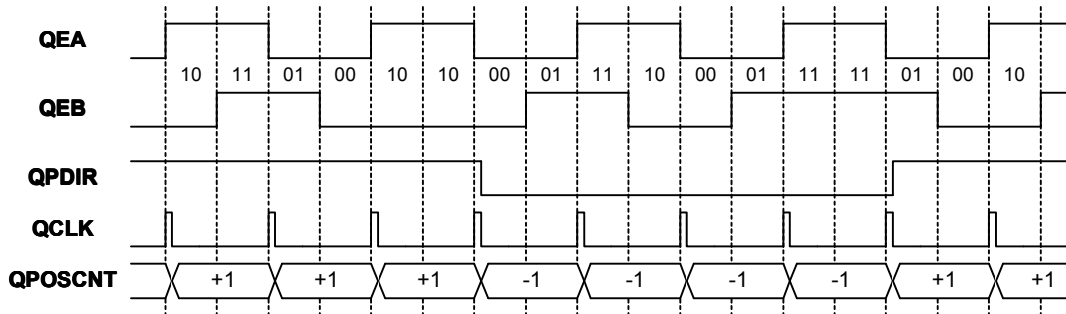


图9-4-4 x2计数模式下QPOSCNT的工作时序

当QPMOD = 10/11时，位置计数器工作于x4计数模式，该模式下QCLK在QEA和QEB的上升沿和下降沿均产生一个脉冲，因此计数器对QEA和QEB的双沿进行计数。x4计数模式下计数器的工作时序如下图9-4-5所示：

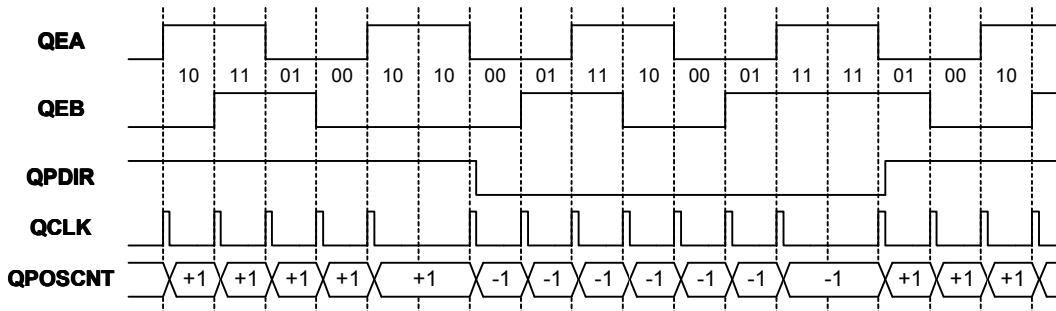


图9-4-5 x4计数模式下QPOSCNT的工作时序

#### 位置计数器的复位方式

当QPMOD = 00/10时，位置计数器使用索引脉冲INDEX复位方式。该复位方式下，硬件会自动检测INDEX引脚的输入信号，每次INDEX由0变1后出现的第一个满足条件的QCLK计数脉冲，会触发一次复位动作。如果复位动作发生时QPOSCNT处于增计数过程，则QPOSCNT自动复位为0；如果此时QPOSCNT处于减计数过程，则QPOSCNT自动重载QCNTMAX设置值。复位动作还受QIDXEN控制，当QIDXEN = 1时，所有复位动作有效，在第一次复位动作后，QPOSCNT将在0-QCNTMAX范围作循环计数。当QIDXEN = 0时，所有复位动作均无效，QPOSCNT增计数到QCNTMAX后自动复位为0，减计数到0后自动复位为QCNTMAX，在0-QCNTMAX范围作循环计数。当发生复位动作时，不管QIDXEN设置状态如何，硬件均会置位QEIIF中断标志。

为保证正反转过程计数值的对称性，每次重新打开QEI功能(QEIEEN由0变1)或每次写QPMOD为新设置值后，硬件会在此后出现的第一次INDEX复位事件时，记录对应的QEA或QEB的变化沿状态。然后在后续每次INDEX由0变1出现时，按照下表（以QEA触发为例，QEB触发与QEA类似）寻找第一个满足条件的复位事件对应的QEA（或QEB）触发沿，并触发复位动作。



重启后第一次复位事件		重启后第n次复位事件 (n ≠ 1)	
增计数	QEA上升沿	增计数	QEA上升沿
		减计数	QEA下降沿
	QEA下降沿	增计数	QEA下降沿
		减计数	QEA上升沿
减计数	QEA上升沿	增计数	QEA下降沿
		减计数	QEA上升沿
	QEA下降沿	增计数	QEA上升沿
		减计数	QEA下降沿

以QPMOD = 10, QCNTMAX = 99为例, 若重新打开QEI功能后第一次复位动作发生在QEB下降沿, 则各关键信号的时序关系如下图9-4-6所示 (灰色框表示复位事件发生位置):

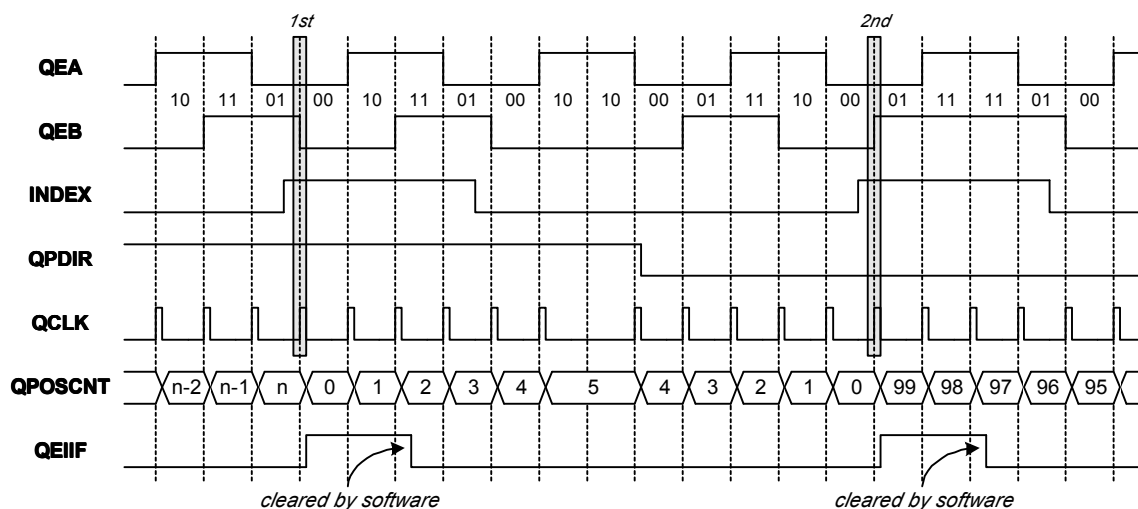


图9-4-6 索引脉冲INDEX复位方式的工作时序 (x4计数模式)

当QPMOD = 01/11时, 位置计数器使用自动循环复位方式。该复位方式下, 当QPOSCNT在增计数过程中等于QCNTMAX设置值后, 在下一个QCLK时QPOSCNT自动复位为0, 然后再继续对QCLK进行增计数; 当QPOSCNT在减计数过程中等于0后, 在下一个QCLK时QPOSCNT重载QCNTMAX设置值, 然后继续减计数。自动循环复位方式不受INDEX信号控制, 自动使位置计数器QPOSCNT在0-QCNTMAX范围作循环计数。当QPOSCNT上溢或下溢复位时, 硬件自动置位QEIIF标志。

以QPMOD = 11, QCNTMAX = 99为例, 各关键信号的时序关系如下图9-4-7所示:

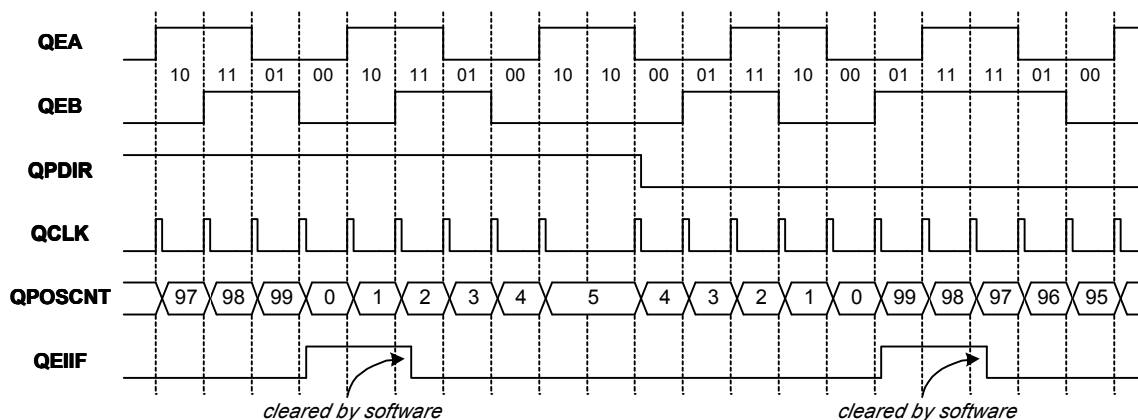


图9-4-7 自动循环复位方式的工作时序 (x4计数模式)



### 9.4.7 QTimer功能

为满足低转速应用中的测速要求，QEI模块内建QTimer以实现高精度测量。使用QTimer进行低转速测速的具体方案参见下节。

QTimer是一个16位计数器，对系统时钟 $fd_{sys}$ 的分频时钟进行增计数，分频系数通过QTCPS进行设置。QTEN位是计数启停控制位，QTEN = 0使QTimer计数停止后，硬件会在下一个系统时钟自动把QTimer计数器和分频计数器复位为0。当下一次QTEN = 1重新启动后，计数器自动从0开始增计数。当QTimer计数溢出时，硬件自动置位QTIF标志，如果此时QTIE = 1，则触发相应的中断。中断标志位QTIF由用户软件写0清除。

**注意：**QTimer正常工作过程（QTEN = 1）中修改QTCPS为新设置值，会立即改变计数时钟的频率。建议用户在QTEN = 0状态下修改QTCPS设置值，以保证计数时钟频率的稳定。

### 9.4.8 测速方案

常规的测速方法是固定一个时间周期T去查询计数器QPOSCNT的值，通过QPOSCNT的计数增量与固定的时间T计算转速。这种方案通常适用于转速较高的情形，当转速较慢时，由于时间周期T内QPOSCNT的变化太小，从而导致转速计数的分辨率降低。因此，在低转速应用中，通过固定QPOSCNT的变化值去检测对应时间的增量，可以大大改善转速测量的精度。

#### 高转速测速方案

在高转速测速应用方案中，QEI模块使用DTimer1作为产生固定时基的定时器。

通过设置QPCAPEN = 1可以使硬件自动将DTimer1溢出标志TMR1IF与QEI模块关联，当TMR1IF由硬件置1的同时，硬件自动将QPOSCNT的值锁存到QPOSLAT寄存器。（DTimer1的具体工作情况请参考DSP定时器描述）。用户可以通过DTimer1的溢出中断，在中断服务程序里计算转速。

#### 低转速测速方案

在低转速测速应用方案中，QEI模块使用QTimer作为时间增量的计数器。

当QEI功能打开时，正交解码器输出的每个QCLK可以触发捕捉事件，捕捉事件的发生频率由QTEPS设置。当有效的捕捉事件发生时，硬件自动把QTMCNT当前计数值锁存到QTMLAT寄存器，同时置位QTCAPIF中断标志（QTEN = 1启动QTimer计数后的第一次捕捉事件不会置位该标志），然后清除QTimer计数器为0，并重新开始增计数。用户在检测到QTCAPIF标志后，可以根据触发事件频率设置（固定的QPOSCNT变化值）和QTMLAT锁存值计算出转速，QTCAPIF标志由用户软件写0清除。

当QTimer计数到0xFFFF后，硬件自动复位计数器为0并重新开始增计数，同时置位QTIF标志。用户需要准确设置QTEPS，以防止捕捉事件发生前QTimer溢出，通过查询QTIF标志可以判断捕捉发生前QTimer是否已经溢出。当QTEN = 1，QTimer启动状态下修改QTEPS为新设置值，新设置值将在下一次捕捉事件发生后才有效。

低转速测速应用方案中，各关键信号的时序关系如下图9-4-8所示（QTCAP信号表示发生了捕捉事件，QTCAPIF和QTIF标志位由用户软件写0清除）：

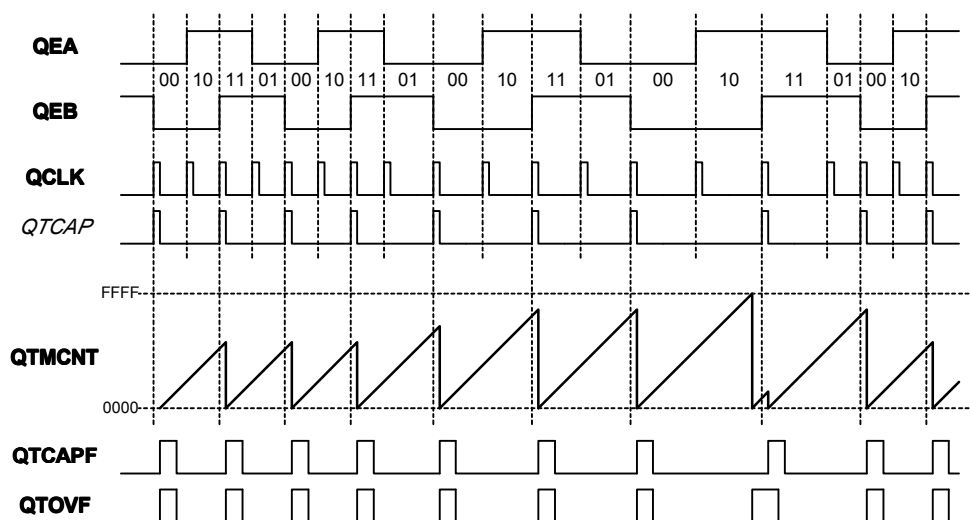


图9-4-8 低转速测速应用方案中关键信号示意图（QTEPS = 0001）



### 9.4.9 QTimer作为独立的定时器

当QEIEN = 0, QEI功能关闭时, QTimer可以作为独立的16位增计数定时器使用。该模式下, QCNTMAX作为QTimer的周期设置寄存器, 当QTimer计数值等于QCNTMAX设置值后, 下一个计数时钟硬件自动复位QTMCNT为0, 同时置位QTIF中断标志。如果此时QTIE = 1, 则触发相应的中断。中断标志位QTIF由用户软件写0清除。

在QEI功能关闭后, QTEPS设置值不变。

### 9.4.10 QEI模块的中断

#### QEI功能的中断

QEI模块使用QEIIE和QEIIF控制和指示QEI功能的中断, 该中断只在QEI功能打开(QEIE = 1)时有效。

QPMOD = 00/10时, 位置计数器使用索引脉冲INDEX复位方式。当发生INDEX复位事件时, 硬件自动置位QEIIF标志, 如果此时QEIIE = 1, 则触发相应的中断。中断标志位QEIIF由用户软件写0清除。

QPMOD = 01/11时, 位置计数器使用自动循环复位方式。该复位方式下, 位置计数器QPOSCNT自动在0-QCNTMAX范围作循环计数。当计数上溢或下溢时, 硬件自动置位QEIIF标志, 如果此时QEIIE = 1, 则触发相应的中断。中断标志位QEIIF由用户软件写0清除。

#### 计数错误中断

当QEI功能打开(QEIE = 1), 且QPMOD = 00/10使用索引脉冲INDEX复位方式时, 硬件使用QCEIE和QCEIF控制和指示计数错误中断。该中断只在发生第一次复位事件后才有效。

当发生第一次复位事件后, QPOSCNT的计数值范围将为0-QCNTMAX。当此后增计数过程计数值从QCNTMAX变为QCNTMAX+1, 或减计数过程计数值从0变为0xFFFF的同时, 硬件自动置位QCEIF标志, 表示发生计数错误。如果此时QCEIE = 1, 则触发相应的中断。中断标志位QCEIF由用户软件写0清除。

#### 相位错误中断

当QEI功能打开(QEIE = 1), 正交解码器会自动检查[QEA:QEB]的状态转换过程, 当检测到QEA与QEB同相位或反相位的同时, 解码器会判断出现相位检测错误, 置位中断标志QPEIF, 如果此时QPEIE = 1, 则触发相应的中断。中断标志位QPEIF由用户软件写0清除。

#### QTimer捕捉事件中中断

当QEI功能打开(QEIE = 1), QTimer作为QEI功能的辅助定时器使用, 硬件使用QTCAPIE和QTCAPIF控制和指示捕捉事件中中断。

当有效的捕捉事件发生时, 硬件自动把QTMCNT当前计数值锁存到QTMLAT寄存器, 同时置位QTCAPIF中断标志(QTEN = 1启动QTimer计数后的第一次捕捉事件不会置位该标志), 如果此时QTCAPIE = 1, 则触发相应的中断。中断标志位QTCAPIF由用户软件写0清除。

#### QTimer溢出中断

当QEI功能打开(QEIE = 1), QTimer作为QEI功能的辅助定时器使用, QTimer计数到0xFFFF后发生溢出, 计数器自动复位为0并重新开始增计数, 同时置位QTIF标志。如果此时QTIE = 1, 则触发相应的中断。中断标志位QTIF由用户软件写0清除。

当QEI功能关闭(QEIE = 0), QTimer作为独立的16位定时器使用, QTimer计数到QCNTMAX设置值后发生溢出, 计数器自动复位为0并重新开始增计数, 同时置位QTIF标志。如果此时QTIE = 1, 则触发相应的中断。中断标志位QTIF由用户软件写0清除。





## 9.5 流水线型ADC模块

### 9.5.1 特性

- 14位流水线型AD转换器，最高采样率为10MSPS（对应AD时钟频率10MHz）
- 16个模拟输入通道，模拟输入电压0V - 3V
- 内建双采样保持器，提供3种采样模式：单序列顺序采样模式、两路同步并行采样模式、两路独立顺序采样模式
- 2种转换方式：连续转换方式和非连续转换方式
- 3种触发启动方式：软件触发启动、PWM比较事件触发启动、DTimer触发启动

### 9.5.2 逻辑框图

SH99F100的ADC模块内建一个14位AD转换器，通过两个采样保持器SHA和SHB分别对两组模拟输入通道AINA0-AINA7和AINB0-AINB7信号的采样后，再选择输出给AD转换器完成数据转换，转换结果存储到结果寄存器ADRRx（x = 0-15）。

ADC模块的逻辑框图如下图9-5-1所示：

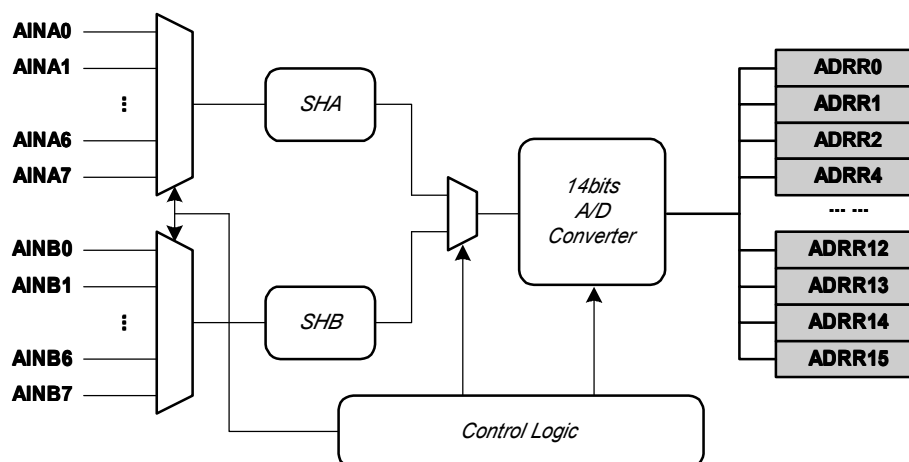


图9-5-1 ADC模块的逻辑框图

### 9.5.3 控制寄存器

ADC模块使用的所有控制寄存器如下表所示：

类别	缩写符号	功能说明
模块控制	ADCON1 ADCON2	ADC控制寄存器
	ADPTR	ADC存储指针寄存器
	ADPCH	ADC通道指针寄存器
	ADCHSEL0 ADCHSEL1 ADCHSEL2 ADCHSEL3	ADC通道选择寄存器
转换结果	ADRR0, ADRR1 ADRR2, ADRR3 ADRR4, ADRR5 ADRR6, ADRR7 ADRR8, ADRR9 ADRR10, ADRR11 ADRR12, ADRR13 ADRR14, ADRR15	AD转换结果寄存器
状态与中断控制	ADINT	ADC中断和状态寄存器



Table 9.51 ADC控制寄存器1 ADCON1

<b>15-14</b>	<b>13-12</b>	<b>11-8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3-0</b>
R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
<b>ADEMU</b>	<b>ADMOD</b>	<b>Reserved</b>	<b>ADEN</b>	<b>ADREN</b>	<b>ADVRS</b>	<b>Reserved</b>	<b>ADCPS</b>

位编号	位符号	说明
<b>15-14</b>	<b>ADEMU</b>	<b>仿真挂起时ADC模块运行状态控制位</b> 00: 不受挂起影响 01: 挂起时, 等待当前通道序列全部转换完成并存储结果后停止 10: 挂起时, 等待当前通道模拟输入的AD转换完成并存储结果后停止 11: 挂起时, 立即停止
<b>13-12</b>	<b>ADMOD</b>	<b>ADC模块采样模式选择控制位</b> 00: 单序列顺序采样模式 01: 两路同步并行采样模式 1x: 两路独立顺序采样模式 (x = 0,1)
<b>11-8</b>	<b>Reserved</b>	保留位
<b>7</b>	<b>ADEN</b>	<b>AD转换器电路电源控制位</b> 0: 关闭AD转换器电路电源和AD运行时钟 1: 打开AD转换器电路电源和AD运行时钟
<b>6</b>	<b>ADREN</b>	<b>ADC模块辅助电路 (Bandgap, V<sub>REF</sub>, Bias等) 电源控制位</b> 0: 关闭辅助电路的电源 1: 打开辅助电路的电源
<b>5</b>	<b>ADVRS</b>	<b>AD转换器参考电平及V<sub>REF</sub>引脚功能选择位</b> 0: AD转换器使用内部参考电平, V <sub>REF</sub> 引脚作为内部参考电平的输出引脚 1: AD转换器使用外部参考电平, V <sub>REF</sub> 引脚作为外部参考电平的输入引脚
<b>4</b>	<b>Reserved</b>	保留位
<b>3-0</b>	<b>ADCPS</b>	<b>AD转换器工作时钟预分频设置位</b> 0: f <sub>DSYS</sub> /1 1: f <sub>DSYS</sub> /2 2: f <sub>DSYS</sub> /4 3: f <sub>DSYS</sub> /6 4: f <sub>DSYS</sub> /8 5: f <sub>DSYS</sub> /10 6: f <sub>DSYS</sub> /12 7: f <sub>DSYS</sub> /14 8: f <sub>DSYS</sub> /16 9: f <sub>DSYS</sub> /18 10: f <sub>DSYS</sub> /20 11: f <sub>DSYS</sub> /22 12: f <sub>DSYS</sub> /24 13: f <sub>DSYS</sub> /26 14: f <sub>DSYS</sub> /28 15: f <sub>DSYS</sub> /30



Table 9.52 ADC控制寄存器2 ADCON2

15	14-12	11	10	9	8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Reserved	ADMAXCH2	ADEV TEN2	ADCTU2	ADRST2	ADSOC2
7-4		3	2	1	0
R/W-0		R/W-0	R/W-0	R/W-0	R/W-0
ADMAXCH1		ADEV TEN1	ADCTU1	ADRST1	ADSOC1

位编号	位符号	说明
15	Reserved	保留位
14-12	ADMAXCH2	SHB通道序列的总长度设置位（仅在两路独立顺序采样模式下有效，设置值0-7） 一次AD转换过程的SHB模拟通道总数为ADMAXCH2+1
11	ADEV TEN2	事件触发SHB通道序列的AD转换使能位（仅在两路独立顺序采样模式下有效） 0: 禁止事件触发SHB通道序列的AD转换 1: 使能事件触发SHB通道序列的AD转换
10	ADCTU2	SHB通道序列的转换方式控制位（仅在两路独立顺序采样模式下有效） 0: 非连续转换方式 1: 连续转换方式 修改该控制位，会在下一次启动AD转换过程时才产生效果
9	ADRST2	AD转换过程复位控制位（仅在两路独立顺序采样模式下有效） 0: 写0无效，硬件完成复位后自动清0 1: 向此位写1将立即复位ADSOC2, ADPSY2, ADPRR2和ADPCH2寄存器为0 若在一次转换过程中将ADRST2置1，硬件会强制结束这次序列转换
8	ADSOC2	SHB通道序列启动AD转换请求位（仅在两路独立顺序采样模式下有效） 0: 写0可取消还未启动的AD转换请求，硬件开始启动一次SHB通道序列的AD转换 过程后，自动清除为0 1: 写1设置SHB通道序列的AD转换请求
7-4	ADMAXCH1	单序列顺序采样模式下，通道序列总长度设置位（设置值0-15） 一次AD转换过程的模拟通道总数为ADMAXCH1+1 两路同步并行采样模式下，通道序列总长度设置位（设置值0-7） 一次AD转换过程的模拟通道总数为2*(ADMAXCH1+1) 两路独立顺序采样模式下，SHA通道序列的总长度设置位（设置值0-7） 一次AD转换过程的SHA模拟通道总数为ADMAXCH1+1
3	ADEV TEN1	单序列顺序和两路同步并行采样模式下，事件触发AD转换使能位 0: 禁止事件触发AD转换 1: 使能事件触发AD转换 两路独立顺序采样模式下，事件触发SHA通道序列的AD转换使能位 0: 禁止事件触发SHA通道序列的AD转换 1: 使能事件触发SHA通道序列的AD转换
2	ADCTU1	单序列顺序和两路同步并行采样模式下，ADC模块转换方式控制位 0: 非连续转换方式 1: 连续转换方式 两路独立顺序采样模式下，SHA通道序列的转换方式控制位 0: 非连续转换方式 1: 连续转换方式 修改该控制位，会在下一次启动AD转换过程时才产生效果



续上表

位编号	位符号	说明
1	ADRST1	<b>AD转换过程复位控制位</b> 0: 写0无效, 硬件完成复位后自动清0 1: 向此位写1将立即复位ADSOC1, ADBSY1, ADPRR1和ADPCH1寄存器为0 若在一次转换过程中将ADRST1置1, 硬件会强制结束这次序列转换
0	ADSOC1	<b>单序列顺序和两路同步并行采样模式下, 启动AD转换请求位</b> 0: 写0可取消还未启动的AD转换请求, 硬件开始启动一次AD转换过程后, 自动清除为0 1: 写1设置AD转换请求 <b>两路独立顺序采样模式下, SHA通道序列启动AD转换请求位</b> 0: 写0可取消还未启动的AD转换请求, 硬件开始启动一次SHA通道序列的AD转换过程后, 自动清除为0 1: 写1设置SHA通道序列的AD转换请求

Table 9.53 ADC存储指针寄存器ADPTR

15	14-12	11	10-8	7-4	3-0
R-0	R-0	R-0	R-0	R-0	R-0
Reserved	ADCNT2	Reserved	ADPRR2	ADCNT1	ADPRR1

位编号	位符号	说明
15	Reserved	保留位
14-12	ADCNT2	AD转换通道计数器 (仅在两路独立顺序采样模式下有效), 每次启动AD转换后, 自动载入ADMAXCH2的值, 然后每完成一次SHB通道选择寄存器指定的模拟通道信号采样后自动减1, 直到减为0为止
11	Reserved	保留位
10-8	ADPRR2	AD转换结果存储寄存器的指针寄存器 (仅在两路独立顺序采样模式下有效), 读取值为x (x = 0-7), 表示当前转换结果将存储到结果寄存器ADRRy (y = x+8) 中
7-4	ADCNT1	<b>AD转换通道计数器</b> 单序列顺序采样模式下, 每次启动AD转换后, 自动载入ADMAXCH1的值, 然后每完成一次通道选择寄存器指定的模拟通道信号采样后自动减1, 直到减为0为止; 两路同步并行采样模式下, 每次启动AD转换后, 自动载入ADMAXCH1的值, 然后每完成一次通道选择寄存器指定的两个模拟通道信号采样后自动减1, 直到减为0为止; 两路独立顺序采样模式下, 每次启动AD转换后, 自动载入ADMAXCH1的值, 然后每完成一次SHA通道选择寄存器指定的模拟通道信号采样后自动减1, 直到减为0为止。
3-0	ADPRR1	<b>AD转换结果存储寄存器的指针寄存器</b> 单序列顺序和两路同步并行采样模式下, 读取值为x (x = 0-15), 表示当前AD转换结果将存储到结果寄存器ADRRx中; 两路独立顺序采样模式下, 读取值为x (x = 0-7), 表示当前转换结果将存储到结果寄存器ADRRx中。



Table 9.54 ADC通道指针寄存器ADPCH

15-11	10-8	7-4	3-0
R-0	R-0	R-0	R-0
Reserved	ADPCH2	Reserved	ADPCH1

位编号	位符号	说明
15-11	Reserved	保留位
10-8	ADPCH2	AD转换通道指针寄存器（仅在两路独立顺序采样模式下有效），读取值为x（x = 0-7），表示当前AD转换的模拟信号为ADCHS <sub>y</sub> （y = x+8）寄存器指定的模拟输入通道采样信号
7-4	Reserved	保留位
3-0	ADPCH1	<b>AD转换通道指针寄存器</b> 单序列顺序采样模式下，读取值为x（x = 0-15），表示当前AD转换的模拟信号为ADCHS <sub>x</sub> 寄存器指定的模拟输入通道采样信号； 两路同步并行采样模式下，读取值为x（x = 0-7），表示当前AD转换的模拟信号为ADCHS <sub>x</sub> 寄存器指定的模拟输入通道采样信号； 两路独立顺序采样模式下，读取值为x（x = 0-7），表示当前AD转换的模拟信号为ADCHS <sub>x</sub> 寄存器指定的模拟输入通道采样信号。

Table 9.55 ADC中断和状态寄存器ADINT

15	14-10	9	8	7	6-2	1	0
R/W-0	R-0	R-0	R/W-0	R/W-0	R-0	R-0	R/W-0
ADIE2	Reserved	ADBSY2	ADIF2	ADIE1	Reserved	ADBSY1	ADIF1

位编号	位符号	说明
15	ADIE2	<b>SHB通道序列转换结束中断使能位</b> （仅在两路独立顺序采样模式下有效） 0: 禁止SHB通道序列转换结束中断 1: 使能SHB通道序列转换结束中断
14-10	Reserved	保留位
9	ADBSY2	<b>ADC模块工作状态指示位</b> （仅在两路独立顺序采样模式下有效） 0: ADC模块没有处于SHB通道序列的转换过程中 1: ADC模块正工作于SHB通道序列的转换过程中
8	ADIF2	<b>SHB通道序列转换结束中断标志位</b> （仅在两路独立顺序采样模式下有效） 0: 无SHB通道序列转换结束中断发生 1: 有SHB通道序列转换结束中断发生 写0清除该标志，写1无效
7	ADIE1	单序列顺序和两路同步并行采样模式下， <b>AD转换过程结束中断使能位</b> 0: 禁止AD转换过程结束中断 1: 使能AD转换过程结束中断 两路独立顺序采样模式下， <b>SHA通道序列转换结束中断使能位</b> 0: 禁止SHA通道序列转换结束中断 1: 使能SHA通道序列转换结束中断
6-2	Reserved	保留位



续上表

位编号	位符号	说明
1	ADBSY1	单序列顺序和两路同步并行采样模式下，ADC模块工作状态指示位 0: ADC模块处于空闲状态，等待启动下一次AD转换过程 1: ADC模块正工作于通道序列的AD转换过程中 两路独立顺序采样模式下，ADC模块工作状态指示位 0: ADC模块没有处于SHA通道序列的转换过程中 1: ADC模块正工作于SHA通道序列的转换过程中
0	ADIF1	单序列顺序和两路同步并行采样模式下，AD转换过程结束中断标志位 0: 无AD转换转换结束中断发生 1: 有AD转换转换结束中断发生 两路独立顺序采样模式下，SHA通道序列转换结束中断标志位 0: 无SHA通道序列转换结束中断发生 1: 有SHA通道序列转换结束中断发生 写0清除该标志，写1无效

Table 9.56 ADC通道选择寄存器ADCHSEL0

15-12	11-8	7-4	3-0
R/W-0	R/W-0	R/W-0	R/W-0
ADCHS3	ADCHS2	ADCHS1	ADCHS0

Table 9.57 ADC通道选择寄存器ADCHSEL1

15-12	11-8	7-4	3-0
R/W-0	R/W-0	R/W-0	R/W-0
ADCHS7	ADCHS6	ADCHS5	ADCHS4

Table 9.58 ADC通道选择寄存器ADCHSEL2

15-12	11-8	7-4	3-0
R/W-0	R/W-0	R/W-0	R/W-0
ADCHS11	ADCHS10	ADCHS9	ADCHS8

Table 9.59 ADC通道选择寄存器ADCHSEL3

15-12	11-8	7-4	3-0
R/W-0	R/W-0	R/W-0	R/W-0
ADCHS15	ADCHS14	ADCHS13	ADCHS12

位编号	位符号	说明
15-12 或11-8 或7-4 或3-0	ADCHSx (x = 0-15)	模拟通道选择寄存器组，分别指定对应的模拟输入通道 单序列顺序采样模式下，ADCHS0-ADCHS15均有效，每个通道选择寄存器指定一个模拟输入通道，设置值与模拟通道的对应关系如下： 0000 - AINA0    1000 - AINB0 0001 - AINA1    1001 - AINB1 0010 - AINA2    1010 - AINB2 0011 - AINA3    1011 - AINB3 0100 - AINA4    1100 - AINB4 0101 - AINA5    1101 - AINB5 0110 - AINA6    1110 - AINB6 0111 - AINA7    1111 - AINB7



续上表

位编号	位符号	说明
15-12 或11-8 或7-4 或3-0	ADCHSx (x = 0-15)	<p>两路同步并行采样模式下, ADCHS0-ADCHS7有效, ADCHS8-ADCHS15无效, 每个通道选择寄存器指定两个关联的模拟输入通道, 设置值与模拟通道的对应关系如下:</p> <p>x000 - AINA0和AINB0 (x表示与最高位无关, 0,1均可)</p> <p>x001 - AINA1和AINB1</p> <p>x010 - AINA2和AINB2</p> <p>x011 - AINA3和AINB3</p> <p>x100 - AINA4和AINB4</p> <p>x101 - AINA5和AINB5</p> <p>x110 - AINA6和AINB6</p> <p>x111 - AINA7和AINB7</p> <p>两路独立顺序采样模式下, 每个通道选择寄存器指定一个模拟输入通道, ADCHS0-ADCHS7指定SHA通道序列的模拟输入通道, 设置值与模拟通道的对应关系如下:</p> <p>0000 - AINA0</p> <p>0001 - AINA1</p> <p>0010 - AINA2</p> <p>0011 - AINA3</p> <p>0100 - AINA4</p> <p>0101 - AINA5</p> <p>0110 - AINA6</p> <p>0111 - AINA7</p> <p>ADCHS8-ADCHS15指定SHB通道序列的模拟输入通道, 设置值与模拟通道的对应关系如下:</p> <p>1000 - AINB0</p> <p>1001 - AINB1</p> <p>1010 - AINB2</p> <p>1011 - AINB3</p> <p>1100 - AINB4</p> <p>1101 - AINB5</p> <p>1110 - AINB6</p> <p>1111 - AINB7</p>

Table 9.60 ADC结果寄存器组ADRRx (x = 0-15)

15-12	11-0
R-0	R-0
Reserved	ADRRx

位编号	位符号	说明
15-12	Reserved	保留位
11-0	ADRRx (x = 0-15)	<p><b>AD转换结果寄存器:</b></p> <p>单序列顺序采样模式下, ADRRx (x = 0-15) 与ADCHSx (x = 0-15) 寄存器一一对应, ADCHSx所指定通道的模拟输入信号, 经采样转换后存储到对应的ADRRx结果寄存器;</p> <p>两路同步并行采样模式下, ADCHS0所指定的两个模拟输入信号经采样转换后分别存储到ADRR0和ADRR1结果寄存器, ADCHS1所指定的两个模拟输入信号经采样转换后分别存储到ADRR2和ADRR3结果寄存器, ...以此类推, ADCHS7所指定的两个模拟输入信号经采样转换后分别存储到ADRR14和ADRR15结果寄存器;</p> <p>两路独立顺序采样模式下, SHA通道序列结果寄存器ADRRx (x = 0-7) 与SHA通道序列选择寄存器ADCHSx (x = 0-7) 寄存器一一对应, SHB通道序列结果寄存器ADRRx (x = 8-15) 与SHB通道序列选择寄存器ADCHSx (x = 8-15) 寄存器一一对应, ADCHSx所指定通道的模拟输入信号, 经采样转换后存储到对应的ADRRx结果寄存器。</p>



#### 9.5.4 AD转换器工作时钟

AD转换器的工作时钟AD\_CLK由DSP系统时钟 $f_{\text{DSYS}}$ 分频得到。分频系数由寄存器ADCON1中的ADCPS设置值决定。

每次修改ADCPS为新的设置值，内部分频计数器自动复位，然后按照新设置值进行时钟分频，分频输出的AD\_CLK时钟不受ADEN等其他控制位影响。

**注意：**当 $ADBSY1 = 1$ 或 $ADBSY2 = 1$ 时修改ADCPS为新的设置值，采样频率会因为AD\_CLK时钟频率的变化而变化，可能对软件算法造成影响。

#### 9.5.5 通道序列

ADC模块包含16个4比特位宽的通道选择寄存器ADCHS $_x$  ( $x = 0-15$ )，每个通道选择寄存器用于选择16路模拟输入通道AINA0-AINA7和AINB0-AINB7中的一个或两个。

通道序列是指由ADCHS $_m$  ( $m = 0-15$ )开始，编号在一定范围内循环递增的若干个通道选择寄存器所指定的模拟输入通道的序列。通道序列对应的有效的通道选择寄存器数量由ADMACH1/2寄存器控制，起始编号由工作方式决定。以单序列顺序采样模式的连续转换方式为例，若ADMACH1 = 2，则ADCHS0、ADCHS1和ADCHS2为有效的通道选择寄存器，若这三个寄存器指定的模拟输入通道依次为AINA5、AINB3和AINB0，则该配置下的通道序列即为AINA5、AINB3、AINB0。

在不同的工作模式下，通道序列的定义略有差别。详见后面分节描述。

**注意：**由于设置ADCHS $_x$  ( $x = 0-15$ )寄存器会立即产生效果，因此，在 $ADBSY1 = 1$ 或 $ADBSY2 = 1$ 时修改ADCHS $_x$  ( $x = 0-15$ )寄存器为新的设置值，可能使当前AD转换过程实际使用的通道序列不可预测，建议用户不要在ADBSY1或ADBSY2为1时修改通道选择寄存器。

#### 9.5.6 AD转换过程

ADC模块的工作过程以一个通道序列为基本单位，当一个AD转换请求被响应后，硬件将自动按照设定的通道序列依次切换各模拟通道输入信号，经采样后输出给AD转换器，当通道序列的所有模拟信号都转换完成后，此次AD转换过程才结束。

ADC模块内建流水线型AD转换器，一个模拟通道信号的采样，其转换结果会在8个AD工作时钟后才存储到对应的结果寄存器中。因此，通道序列的采样完成时间点比AD转换过程的结束时间点要早8个AD工作时钟。

ADSOC1和ADSOC2是AD转换请求控制位。以ADSOC1为例，当ADSOC1由0变为1时，将产生一个AD转换请求。如果此时ADBSY1 = 1，表示前次AD转换过程的通道序列采样还未结束，ADSOC = 1的状态将持续存在，直到软件清0，或硬件完成当前AD转换过程所有通道序列的采样并允许启动新一次的AD转换过程（ADBSY1恢复为0后再过一个AD工作时钟开始）。

如果ADSOC1由0变1时ADBSY1 = 0，ADC模块将在下一个AD\_CLK的开始位置，置ADBSY1为1，清除ADSOC1为0，并开始按照通道序列的设置开始AD转换过程。8个AD工作时钟之后，每个AD工作时钟完成一个模拟通道的转换，并存储转换结果到相应的结果寄存器。当通道序列的所有模拟通道信号都采样并切换给AD转换器后，硬件自动清除ADBSY1为0，允许启动一次新的AD转换过程。当所有模拟通道信号的转换结果均存储到结果寄存器后，硬件置位中断标志ADIF1，完成此次AD转换过程。

ADSOC2的控制过程与ADSOC1相似，但只在两路独立顺序采样模式下有效。

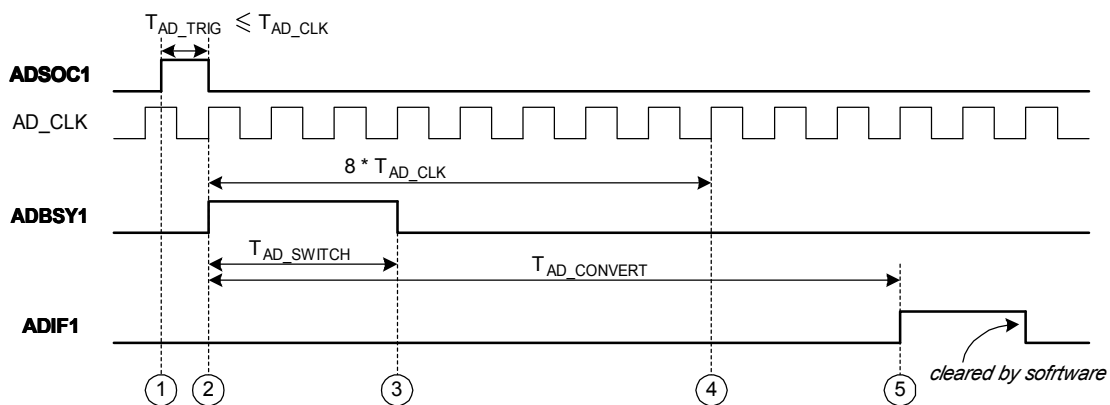


图9-5-2 AD转换过程的时序关系图





AD转换过程中各关键信号的时序关系如图9-5-2所示，各关键时间点的说明如下：

① 由软件或硬件触发，启动AD转换请求；

② 硬件接受请求，并启动AD转换过程，ADBSY1在该位置由硬件置1。从位置①到位置②的 $T_{AD\_TRIG}$ 是硬件响应请求的时间，若ADSOC1置1时ADBSY1 = 0，则 $T_{AD\_TRIG}$ 小于或等于一个AD工作时钟。

③ 硬件完成通道序列所有模拟通道信号采样，并切换给AD转换器。 $T_{AD\_SWITCH}$ 表示所有模拟通道完成一次采样和切换的时间，上图以4个模拟通道的切换为例。

④ 开始输出AD转换结果，在该时间点，第一个模拟通道的AD转换结果被存储到对应的结果寄存器中。此后，每个AD工作时钟将有一个模拟通道信号转换结果存储到对应的结果寄存器中。

⑤ AD转换过程结束，硬件自动置位ADIF1标志。从位置②到位置⑤确定的AD转换过程总时间 $T_{AD\_CONVERT} = T_{AD\_SWITCH} + 8 * T_{AD\_CLK}$ 。ADIF1标志由软件写0清除。

### 9.5.7 AD转换过程的触发方式

SH99F100提供三种AD触发方式。

软件触发：用户可以使用软件直接置位ADSOC1/2来启动AD转换过程，也可以写0来取消还未被响应的AD转换请求。

PWM比较事件触发：当PWM的事件比较功能启用时，如果ADEV TEN1或ADEV TEN2设置为1，则当有效的比较成功事件发生后，硬件自动置位ADSOC1或ADSOC2，启动相应的AD转换过程。PWM比较事件触发的详细内容参考PWM模块的描述。

DTimer事件触发：当DTimer触发AD功能启用时，如果ADEV TEN1或ADEV TEN2设置为1，则当DTimer满足触发条件时，硬件自动置位ADSOC1或ADSOC2，启动相应的AD转换过程。DTimer事件触发AD的详细内容参考DTimer模块的描述。

### 9.5.8 指针和计数寄存器

ADC模块包含2类指针寄存器和1类计数寄存器，所有指针和计数寄存器均为只读特性。

ADPCH1/2通道指针寄存器，用于指示当前AD转换器使用的通道选择寄存器编号。ADPCH1读取值 $x$  ( $x = 0-15$ )，表示当前AD转换的模拟输入信号来自于ADCHS $x$ 寄存器所指定的模拟通道；ADPCH2读取值 $x$  ( $x = 0-7$ )，表示当前AD转换的模拟输入信号来自于ADCHS $y$  ( $y = x+8$ ) 寄存器所指定的模拟通道。

ADPRR1/2存储指针寄存器，用于指示当前AD转换值将存储到的结果寄存器编号。ADPRR1读取值 $x$  ( $x = 0-15$ )，表示当前转换结果将存储到结果寄存器ADRR $x$ 中；ADPRR2读取值 $x$  ( $x = 0-7$ )，表示当前转换结果将存储到结果寄存器ADRR $y$  ( $y = x+8$ ) 中。

ADCNT1/2通道计数寄存器，用于对本次AD转换过程启动后的有效通道选择寄存器数进行减计数。每次AD转换启动后，硬件自动把ADMAXCH1/2设置值载入到ADCNT1/2寄存器中，当一个有效通道选择寄存器指定的一个（或两个）模拟通道信号采样并切换给AD转换器后，ADCNT1/2计数值自动减1，当计数值为0，表示此次启动AD转换后的采样过程将在当前通道选择寄存器指定的模拟通道信号采样完成后结束。

### 9.5.9 ADC模块的转换方式

ADC模块提供了2种转换方式：连续转换方式和非连续转换方式。寄存器ADCTU1/2用于控制转换方式的选择。

连续转换方式下，硬件每次响应AD转换请求并启动AD转换过程后，自动复位ADPCH1/2和ADPRR1/2寄存器为0。因此，每次置位ADSOC1或ADSOC2时，不必关心ADPCH1/2和ADPRR1/2的当前状态，也即可以连续的启动AD转换过程。

非连续转换方式下，硬件每次响应AD转换请求并启动AD转换过程后，并不复位ADPCH1/2和ADPRR1/2寄存器，寄存器会保持原值并在下一次AD转换过程中直接使用。因此，每次置位ADSOC1或ADSOC2时，用户都需要特别注意ADPCH1/2和ADPRR1/2的当前状态，必要时需要通过ADRST1或ADRST2来软件清除ADPCH1/2和ADPRR1/2寄存器为0。

**注意：**当ADCTU1/2设置值发生改变，硬件不会自动清除ADPCH1/2和ADPRR1/2寄存器为0，而在下一次AD转换过程启动后，根据最新的转换方式设置来决定是否清除ADPCH1/2和ADPRR1/2寄存器。

### 9.5.10 单序列顺序采样模式

设置ADMOD = 00时，ADC模块工作在单序列顺序采样模式，SHA和SHB两个采样保持电路组合使用，共同实现一个单序列采样转换过程。该模式下，当ADSOC1 = 1启动AD转换请求并被硬件响应后，硬件会按照设定的通道序列依次对各模拟通道输入信号进行采样，同时将采样结果切换给AD转换器，转换结果依次存储到对应的结果寄存器。通道序列所有模拟通道信号全部转换结束后置位中断标志ADIF1，如果此时ADIE1 = 1，则触发相应的中断。中断标志位ADIF1由用户软件写0清除。

单序列顺序采样模式下，通道序列是指由ADCHS $m$  ( $m = 0-15$ ) 开始，编号以0-15循环递增的若干个通道选择寄存器所指定的模拟输入通道的序列。每个通道选择寄存器指定一个模拟输入通道，起始编号 $m$ 由转换方式决定，通道序列的总长度由ADMAXCH1设定，设置值为0-15，表示ADCHS $m$  ( $m = 0-15$ ) 开始到ADCHS $n$  ( $n = [m+ADMAXCH1]\%16$ , %表示取模) 为有效的通道选择寄存器。

单序列顺序采样模式下，结果存储寄存器与通道选择寄存器一一对应，ADCHS $x$  ( $x = 0-15$ ) 通道选择器指定模拟通道的转换结果，存储到对应的ADRR $x$  ( $x = 0-15$ ) 结果寄存器中。每完成一个模拟通道的切换，ADPCH1自动以0-15循环加1，指向下一个通道选择寄存器；每存储一个转换值，ADPRR1自动以0-15循环加1，指向下一个用于存储转换结果的结果寄存器。

单序列顺序采样模式下，采样时间点与通道切换的时序关系图参见电气特性的图11-1。



### 连续转换方式

ADCTU1 = 1, ADC工作于连续转换方式。在连续转换方式下,当ADSOC1 = 1启动AD转换请求并被硬件响应后,ADPCH1和ADPRR1寄存器自动复位为0。因此,该方式下有效的通道选择寄存器是从ADCHS0开始到ADCHSn (n = ADMAXCH1) 的总计ADMAXCH1+1个通道选择寄存器。

例如, ADCHS0 = 2, ADCHS1 = 13, ADCHS2 = 8, ADCHS3 = 4, ADCHS4 = 7, ..., ADCHS15 = 0, 当ADMAXCH1 = 3 时,在单序列顺序采样模式的连续转换方式下,ADCHS0-ADCHS3为有效的通道选择寄存器,由此组成的通道序列为AINA2, AINB5, AINB0, AINA4。该示例中,各关键信号的时序关系如图9-5-3所示:

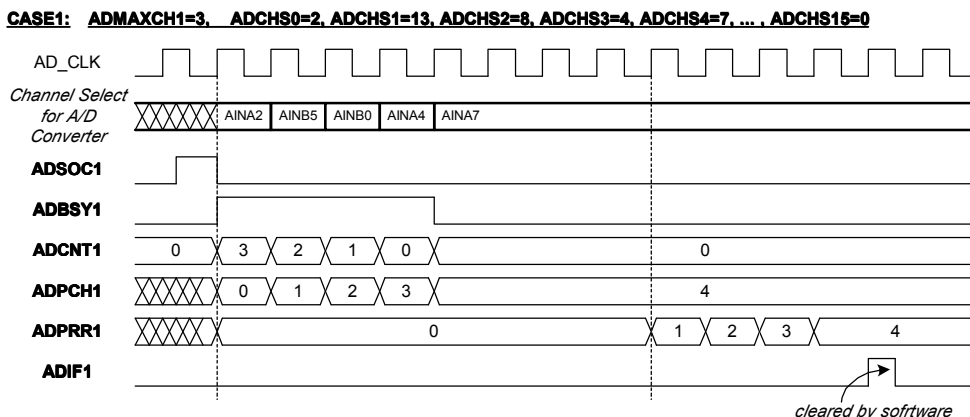


图9-5-3 单序列顺序采样模式,连续转换方式的波形示例图

### 非连续转换方式

ADCTU1 = 0, ADC工作于非连续转换方式。在非连续转换方式下,当ADSOC1 = 1启动AD转换请求并被硬件响应后,ADPCH1和ADPRR1寄存器并不复位,保持原值。因此,该方式下有效的通道选择寄存器是从ADCHSm (m = 0-15) 开始到ADCHSn (n = [m+ADMAXCH1]%16, %表示取模) 的总计ADMAXCH1+1个通道选择寄存器。其中,m是启动本次转换时ADPCH1的值。

例如, ADCHS0 = 2, ADCHS1 = 13, ADCHS2 = 8, ADCHS3 = 4, ..., ADCHS14 = 15, ADCHS15 = 0, 当ADMAXCH1 = 3 时,若前次转换结束时ADPCH1 = 14,在单序列顺序采样模式的非连续转换方式下,ADCHS14, ADCHS15, ADCHS0和ADCHS1 为有效的通道选择寄存器,由此组成的通道序列为AINB7, AINA0, AINA2, AINB5。该示例中,各关键信号的时序关系如图9-5-4 所示:

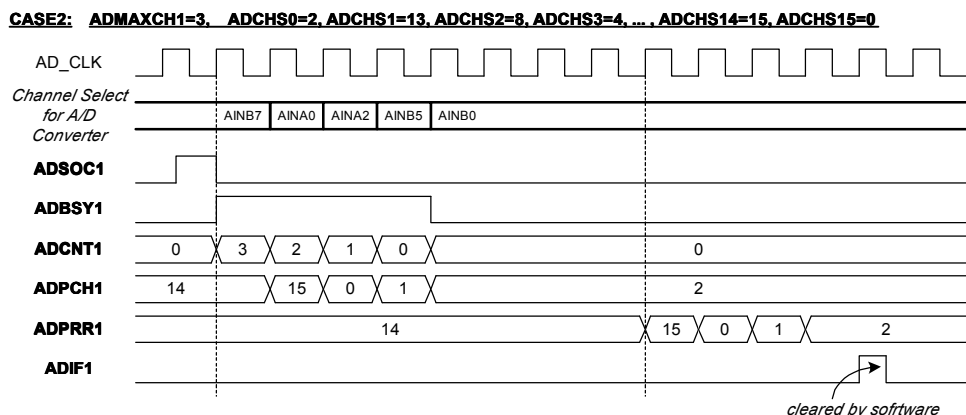


图9-5-4 单序列顺序采样模式,非连续转换方式的波形示例图

**9.5.11 两路同步并行采样模式**

设置ADMOD = 01时，ADC模块工作在两路同步并行采样模式，SHA和SHB两个采样保持电路同时对关联的两个模拟通道进行采样，然后依次切换给AD转换器完成转换。该模式下，当ADSOC1 = 1启动AD转换请求并被硬件响应后，硬件会按照设定的通道序列切换各模拟通道采样信号给AD转换器，转换结果依次存储到对应的结果寄存器。通道序列所有模拟通道的采样信号全部转换结束后置位中断标志ADIF1，如果此时ADIE1 = 1，则触发相应的中断。中断标志位ADIF1由用户软件写0清除。

两路同步并行采样模式下，通道序列是指由ADCHSm (m = 0-7) 开始，编号以0-7循环递增的若干个通道选择寄存器所指定的模拟输入通道的序列，起始编号m由转换方式决定。与单序列顺序采样模式不同的是，每个通道选择寄存器选择SHA和SHB两个关联的模拟输入通道，SHA模拟通道在前，SHB模拟通道在后，两个通道同时采样，顺序转换。通道序列的总长度仍然由ADMAXCH1设定，但设置值为0-7，表示由ADCHSm (m = 0-7) 开始到ADCHSn (n=[m+ADMAXCH1]%8, %表示取模) 为有效的通道选择寄存器。由于每个通道选择寄存器选择两个模拟输入通道，因此该模式下通道序列总长度为2\*(ADMAXCH1+1)。

两路同步并行采样模式下，ADC模块首先同时采样第一个有效的通道选择寄存器指定的SHA和SHB模拟通道信号，然后依次把SHA通道和SHB通道的采样信号切换给AD转换器，转换结果分别存储到对应的结果寄存器中，然后同时采样第二个有效的通道选择寄存器指定的SHA和SHB模拟通道信号，以此类推，直到所有通道选择寄存器的指定通道都完成采样和转换。每完成一个通道对（包含SHA和SHB两个模拟通道）的AD转换，ADPCH1寄存器值自动以0-7循环加1，指向下一个通道选择寄存器；每存储一个AD转换值，ADPRR1寄存器值自动以0-15循环加1，指向下一个用于存储转换结果的结果寄存器。通道选择寄存器与结果寄存器的对应存储关系如下表所示：

通道选择寄存器	对应模拟通道	结果寄存器
ADCHS0	SHA对应的模拟通道	ADRR0
	SHB对应的模拟通道	ADRR1
ADCHS1	SHA对应的模拟通道	ADRR2
	SHB对应的模拟通道	ADRR3
ADCHS2	SHA对应的模拟通道	ADRR4
	SHB对应的模拟通道	ADRR5
ADCHS3	SHA对应的模拟通道	ADRR6
	SHB对应的模拟通道	ADRR7
ADCHS4	SHA对应的模拟通道	ADRR8
	SHB对应的模拟通道	ADRR9
ADCHS5	SHA对应的模拟通道	ADRR10
	SHB对应的模拟通道	ADRR11
ADCHS6	SHA对应的模拟通道	ADRR12
	SHB对应的模拟通道	ADRR13
ADCHS7	SHA对应的模拟通道	ADRR14
	SHB对应的模拟通道	ADRR15

两路同步并行采样模式下，采样时间点与通道切换的时序关系图参见电气特性的图11-2。



### 连续转换方式

ADCTU1 = 1, ADC工作于连续转换方式。在连续转换方式下, 当ADSOC1 = 1启动AD转换请求并被硬件响应后, ADPCH1和ADPRR1寄存器自动复位为0。因此, 该方式下有效的通道选择寄存器是从ADCHS0开始到ADCHSn (n = ADMAXCH1) 的总计ADMAXCH1+1个通道选择寄存器。

例如, ADCHS0 = 2, ADCHS1 = 5, ADCHS2 = 0, ..., ADCHS7 = 4, 当ADMAXCH1 = 1时, 在两路同步并行采样模式的连续转换方式下, ADCHS0和ADCHS1为有效的通道选择寄存器, 由此组成的通道序列为AINA2, AINB2, AINA5, AINB5。该示例中, 各关键信号的时序关系如图9-5-5所示:

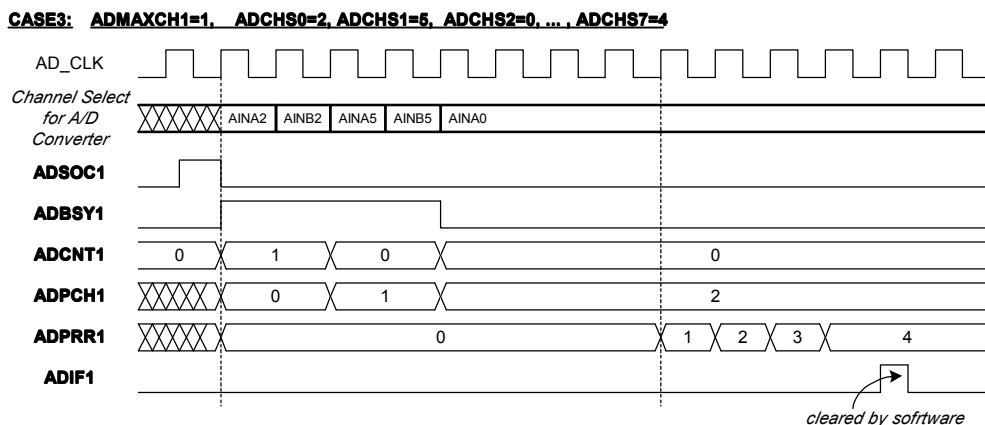


图9-5-5 两路同步并行采样模式, 连续转换方式的波形示例图

### 非连续转换方式

ADCTU1 = 0, ADC工作于非连续转换方式。在非连续转换方式下, 当ADSOC1 = 1启动AD转换请求并被硬件响应后, ADPCH1和ADPRR1寄存器并不复位, 保持原值。因此, 该方式下有效的通道选择寄存器是从ADCHSm (m = 0-7) 开始到ADCHSn (n = [m+ADMAXCH1]%8, %表示取模) 的总计ADMAXCH1+1个通道选择寄存器。其中, m是启动本次转换时ADPCH1的值。

例如, ADCHS0 = 2, ADCHS1 = 5, ADCHS2 = 0, ..., ADCHS7 = 4, 当ADMAXCH1 = 1时, 若前次转换结束时ADPCH1 = 7, 在两路同步并行采样模式的非连续转换方式下, ADCHS7和ADCHS0为有效的通道选择寄存器, 由此组成的通道序列为, AINA4, AINB4, AINA2, AINB2。该示例中, 各关键信号的时序关系如图9-5-6所示:

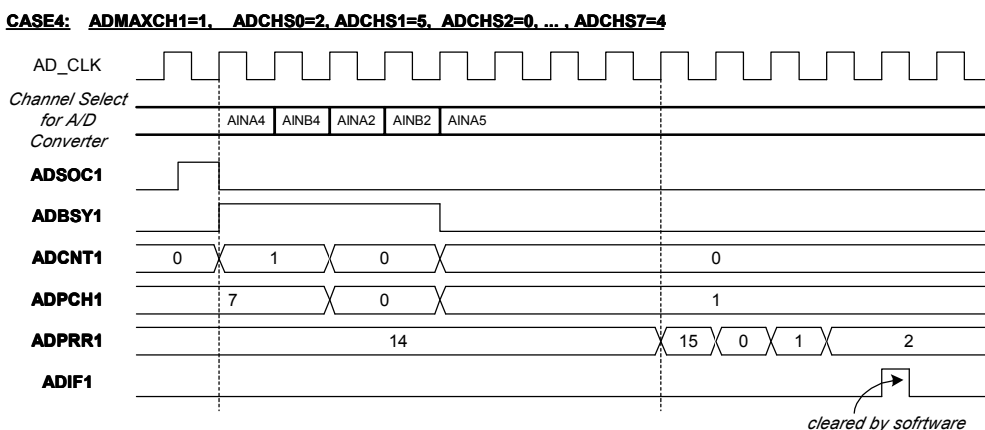


图9-5-6 两路同步并行采样模式, 非连续转换方式的波形示例图



### 9.5.12 两路独立顺序采样模式

设置ADMOD = 10/11时，ADC模块工作在两路独立顺序采样模式，SHA和SHB两个采样保持电路独立使用，组成两个完全独立的通道序列。每个通道序列有独立的操作控制寄存器、状态指示和中断控制寄存器。设置ADSOC1 = 1独立启动SHA通道序列的AD转换请求，设置ADSOC2 = 1独立启动SHB通道序列的AD转换请求，如果两者同时被置位，硬件会先按照SHA通道序列自动切换各模拟通道输入信号给AD转换器，转换结果依次存储到对应的结果寄存器，然后再按照SHB通道序列自动切换各模拟通道输入信号给AD转换器，转换结果依次存储到对应的结果寄存器。SHA通道序列的所有模拟通道信号全部转换结束后置位中断标志ADIF1，如果此时ADIE1 = 1，则触发相应的中断。SHB通道序列的所有模拟通道信号全部转换结束后置位中断标志ADIF2，如果此时ADIE2 = 1，则触发相应的中断。中断标志位ADIF1和ADIF2由用户软件写0清除。

SHA对应的通道序列是指由ADCHSm (m = 0-7) 开始，编号以0-7循环递增的若干个通道选择寄存器所指定的模拟输入通道的序列。每个通道选择寄存器指定一个模拟输入通道，起始编号m由转换方式决定，通道序列的总长度由ADMAXCH1设定，设置值为0-7，表示由ADCHSm (m = 0-7) 开始到ADCHSn (n = [m+ADMAXCH1]%8, %表示取模) 为有效的通道选择寄存器。

SHB对应的通道序列是指由ADCHSm (m = 8-15) 开始，编号以8~15循环递增的若干个通道选择寄存器所指定的模拟输入通道的序列。每个通道选择寄存器指定一个模拟输入通道，起始编号m由转换方式决定，通道序列的总长度由ADMAXCH2设定，设置值为0-7，表示由ADCHSm (m = 8-15) 开始到ADCHSn (n = [m+ADMAXCH2]%8+8, %表示取模) 为有效的通道选择寄存器。

两路独立顺序采样模式下，ADPCH1和ADPCH2独立指示两个通道序列的通道选择寄存器，ADPRR1和ADPRR2则独立指示两个通道序列的结果寄存器位置。每完成一个SHA通道序列中模拟通道的AD转换，ADPCH1寄存器值自动以0-7循环加1；每存储一个SHA通道序列中模拟通道的AD转换值，ADPRR1寄存器值自动以0-7循环加1。每完成一个SHB通道序列中模拟通道的AD转换，ADPCH2寄存器值自动以0-7循环加1；每存储一个SHB通道序列中模拟通道的AD转换值，ADPRR2寄存器值自动以0-7循环加1。通道选择寄存器与结果寄存器的对应存储关系如下表所示：

通道序列	通道选择寄存器	结果寄存器
SHA通道序列	ADCHS0	ADRR0
	ADCHS1	ADRR1
	ADCHS2	ADRR2
	ADCHS3	ADRR3
	ADCHS4	ADRR4
	ADCHS5	ADRR5
	ADCHS6	ADRR6
	ADCHS7	ADRR7
SHB通道序列	ADCHS8	ADRR8
	ADCHS9	ADRR9
	ADCHS10	ADRR10
	ADCHS11	ADRR11
	ADCHS12	ADRR12
	ADCHS13	ADRR13
	ADCHS14	ADRR14
	ADCHS15	ADRR15

两路独立顺序采样模式下，采样时间点与通道切换的时序关系与单序列顺序采样模式相似，参见电气特性的图11-1。



连续转换方式

ADCTU1 = 1, ADC工作于连续转换方式。在连续转换方式下,当ADSOC1 = 1启动AD转换请求并被硬件响应后,ADPCH1/2和ADPRR1/2寄存器自动复位为0。因此,该方式下SHA通道序列的有效的通道选择寄存器是从ADCHS0开始到ADCHSn (n = ADMAXCH1)的总计ADMAXCH1+1个通道选择寄存器,SHB通道序列的有效的通道选择寄存器是从ADCHS8开始到ADCHSn (n = ADMAXCH2+8)的总计ADMAXCH2+1个通道选择寄存器。

例如,ADCHS0 = 2, ADCHS1 = 5, ADCHS2 = 0, ..., ADCHS7 = 4, ADCHS8 = 3, ADCHS9 = 6, ..., ADCHS15 = 1, 当ADMAXCH1 = 2, ADMAXCH2 = 1时,在两路独立顺序采样模式的连续转换方式下,ADCHS0-ADCHS2为SHA通道序列有效的通道选择寄存器,由此组成的通道序列为AINA2, AINA5, AINA0, 而ADCHS8和ADCHS9为SHB通道序列有效的通道选择寄存器,由此组成的通道序列为AINB3, AINB6。该示例中,各关键信号的时序关系如图9-5-7所示:

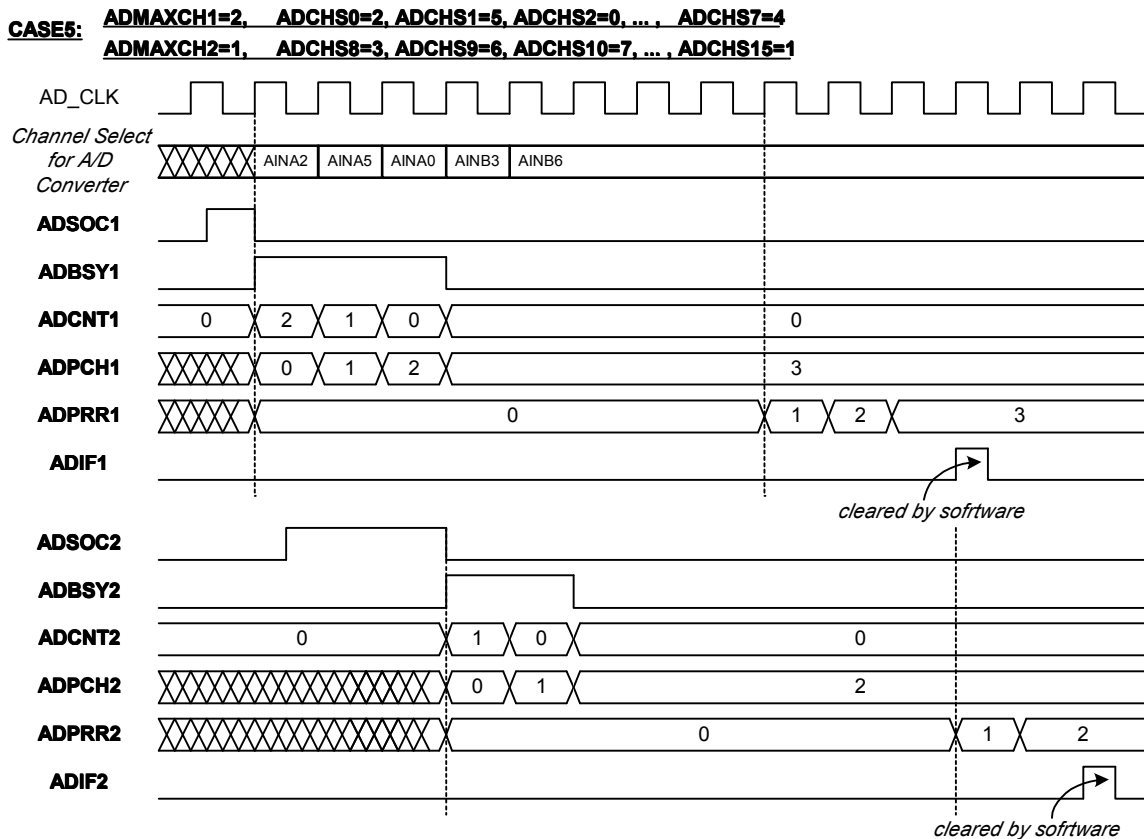


图9-5-7 两路独立顺序采样模式,连续转换方式的波形示例图



### 非连续转换方式

ADCTU1 = 0, ADC工作于非连续转换方式。在非连续转换方式下, 当ADSOC1 = 1启动AD转换请求并被硬件响应后, ADPCH1/2和ADPRR1/2寄存器并不复位, 保持原值。因此, 该方式下SHA通道序列有效的通道选择寄存器是从ADCHSm ( $m = 0-7$ ) 开始到ADCHSn ( $n = [m + \text{ADMAXCH1}] \% 8$ , %表示取模) 的总计ADMAXCH1+1个通道选择寄存器。其中, m是本次转换时ADPCH1的值。SHB通道序列的有效的通道选择寄存器是ADCHSm ( $m = 8-15$ ) 开始到ADCHSn ( $n = [m - 8 + \text{ADMAXCH2}] \% 8 + 8$ , %表示取模) 的总计ADMAXCH2+1个通道选择寄存器。其中, m是本次转换时ADPCH2的值。

例如, ADCHS0 = 2, ADCHS1 = 5, ADCHS2 = 0, ..., ADCHS7 = 4, ADCHS8 = 3, ADCHS9 = 6, ..., ADCHS15 = 1, 当ADMAXCH1 = 2, ADMAXCH2 = 1时, 若前次转换结束时ADPCH1 = 7, ADPCH2 = 7, 在两路独立顺序采样模式的非连续转换方式下, ADCHS7, ADCHS0和ADCHS1为SHA通道序列有效的通道选择寄存器, 由此组成的通道序列为AINA4, AINA2, AINA5, 而ADCHS15和ADCHS0为SHB通道序列有效的通道选择寄存器, 由此组成的通道序列为AINB1, AINB3。该示例中, 各关键信号的时序关系如图9-5-8所示:

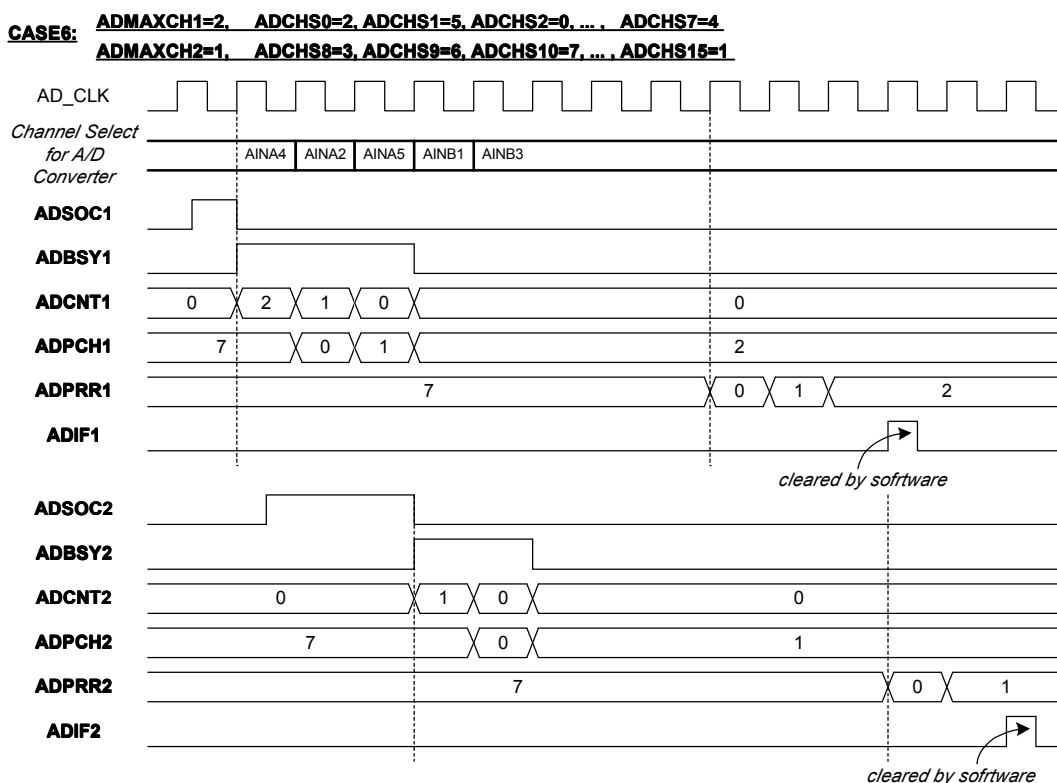


图9-5-8 两路独立顺序采样模式, 非连续转换方式的波形示例图

### 9.5.13 ADC模块的电源控制

ADC模块提供3个控制位ADEN, ADREN和ADVRS用于控制ADC模块的电源和参考电平, 用户需按照如下启动顺序, 以保证ADC模块的可靠工作:

1. 如果使用外部参考电平, 用户需要在开启ADC模块电源前前置位ADVRS, 以避免内部参考电路上电后输出驱动外部电路。
2. 通过置位ADREN, 打开ADC模块辅助电路 (Bandgap,  $V_{REF}$ , Bias等) 的电源, 辅助电路的建立时间至少需要10ms, 用户需要通过软件控制来保证。
3. 当辅助电路正常工作后 (上电10ms), 置位ADEN, 打开AD转换器工作电源, AD转换器电路需要20us才能正常工作, 用户软件必须在20us后再启动AD转换过程。

如果需要关闭ADC模块以降低系统功耗, 用户可以直接对ADREN和ADEN清零来实现, 但下次重新打开ADC模块电源时, 必须重复上述2, 3步骤以保证ADC模块的可靠工作。为提高ADC模块关闭后重新启动的速度, 用户可以对ADEN控制位清零, 以关闭ADC模块主要的工作电路。



## 9.6 DSP定时器

### 9.6.1 特性

- 4个16位定时器DTimerx (x = 0-3)
- 每个定时器提供2种计数模式：单次计数和连续计数

SH99F100内部含有4个16位定时器DTimerx (x = 0-3)。其中，DTimer1在QEI模块中可以作为测速点单位时基发生器；DTimer2和DTimer3可以用来作为输入捕捉以及输出比较模块的时基。

定时器的计数时钟使用分频后的系统时钟（通过8位的预分频寄存器设置），定时时间的最小分辨率为10ns（定时器时钟为100MHz）。使能定时器后，16位的定时计数器TMRxCNT (x = 0-3)从0开始增计数，当与周期寄存器TMRxPRD (x = 0-3)的内部缓存寄存器值相等时，计数器在下一个计数时钟复位为0，并置起相应的定时器中断标志。

每个定时器都提供两种计数模式：单次计数模式和连续计数模式，可以独立的设置和运行。

### 9.6.2 逻辑框图

4个DSP定时器DTimerx (x = 0-3)具有完全相同的实现结构，其逻辑框图如下图所示：

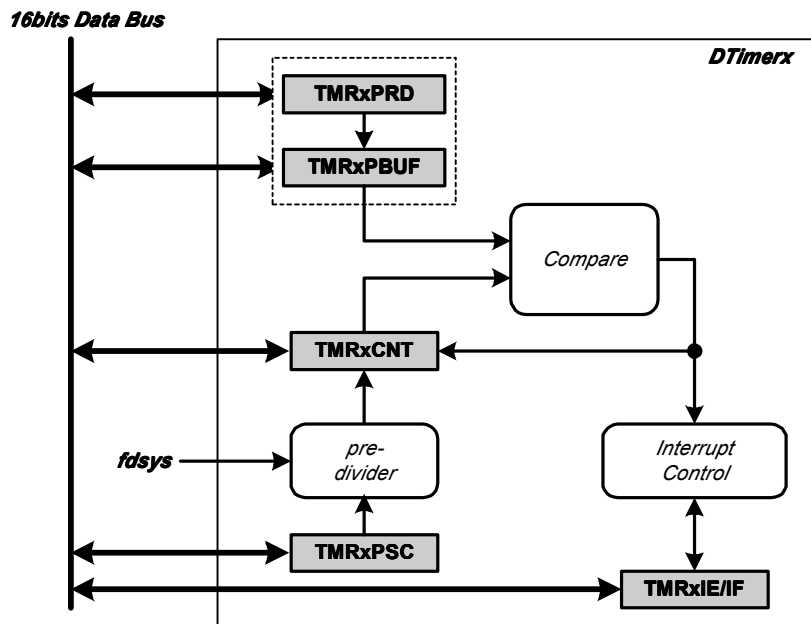


图9-6-1 DTimerx (x = 0-3) 定时器的逻辑框图





### 9.6.3 控制寄存器

DTimerx (x = 0-3) 使用的所有控制寄存器如下表所示:

类别	缩写符号	功能说明
定时器控制	TMR0PRD TMR1PRD TMR2PRD TMR3PRD	周期寄存器
	TMR0PBUF TMR1PBUF TMR2PBUF TMR3PBUF	周期缓存寄存器
	TMRMOD	定时器模式设置寄存器
	TMRCON	定时器控制寄存器
	TMRPSC0 TMRPSC1	定时器预分频寄存器
定时器计数值	TMR0CNT TMR1CNT TMR2CNT TMR3CNT	定时器计数值寄存器
中断控制	TMRINT	定时器中断控制寄存器

**Table 9.61** 周期寄存器TMRxPRD (x = 0-3) (DTimer Period)

15-0		
R/W-0		
TMRxPRD		
位编号	位符号	说明
15-0	<b>TMRxPRD</b> (x = 0-3)	周期寄存器, 与周期缓存寄存器TMRxPBUF配对使用, 每次定时计数器归零时, 硬件会自动把TMRxPRD设置值更新到TMRxPBUF中

**Table 9.62** 周期缓存寄存器TMRxPBUF (x = 0-3) (DTimer Period Buffer)

15-0		
R/W-0		
TMRxPBUF		
位编号	位符号	说明
15-0	<b>TMRxPBUF</b> (x = 0-3)	周期缓存寄存器, 内部比较器固定使用TMRxPBUF的设置值与定时计数器TMRxCNT进行比较。当定时器关闭时, 周期缓存寄存器可读可写; 当定时器工作时, 只能读取不能写入。

**Table 9.63** 定时器计数器TMRxCNT (x = 0-3) (DTimer Counter)

15-0		
R-0		
TMRxCNT		
位编号	位符号	说明
15-0	<b>TMRxCNT</b> (x = 0-3)	DTimerx的16位计数器, 分别以TMRPSC0/1设置的分频后时钟作为计数时钟。每次启动定时器后, TMRxCNT自动从0开始, 在计数时钟下进行增计数, 当与周期缓存寄存器设置值相等时, 在下一个计数时钟自动复位为0



Table 9.64 定时器模式寄存器TMRMOD (DTimer Mode)

<b>15-14</b>	<b>13-12</b>	<b>11-10</b>	<b>9-8</b>		
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>		
<b>TMR3ADT</b>	<b>TMR2ADT</b>	<b>TMR1ADT</b>	<b>TMR0ADT</b>		
<b>7-4</b>		<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>TMR3MOD</b>	<b>TMR2MOD</b>	<b>TMR1MOD</b>	<b>TMR0MOD</b>

位编号	位符号	说明
<b>15-14</b>	<b>TMR3ADT</b>	<b>AD触发方式控制位</b> 00: 不触发AD采样 01: 不触发AD采样 10: DTimer3的计数器值与周期设置值相等时置位SOC1 11: DTimer3的计数器值与周期设置值相等时置位SOC2
<b>13-12</b>	<b>TMR2ADT</b>	<b>AD触发方式控制位</b> 00: 不触发AD采样 01: 不触发AD采样 10: DTimer2的计数器值与周期设置值相等时置位SOC1 11: DTimer2的计数器值与周期设置值相等时置位SOC2
<b>11-10</b>	<b>TMR1ADT</b>	<b>AD触发方式控制位</b> 00: 不触发AD采样 01: 不触发AD采样 10: DTimer1的计数器值与周期设置值相等时置位SOC1 11: DTimer1的计数器值与周期设置值相等时置位SOC2
<b>9-8</b>	<b>TMR0ADT</b>	<b>AD触发方式控制位</b> 00: 不触发AD采样 01: 不触发AD采样 10: DTimer0的计数器值与周期设置值相等时置位SOC1 11: DTimer0的计数器值与周期设置值相等时置位SOC2
<b>7-4</b>	<b>Reserved</b>	保留位
<b>3</b>	<b>TMR3MOD</b>	<b>DTimer3计数模式控制位</b> 0: 连续计数模式 1: 单次计数模式
<b>2</b>	<b>TMR2MOD</b>	<b>DTimer2计数模式控制位</b> 0: 连续计数模式 1: 单次计数模式
<b>1</b>	<b>TMR1MOD</b>	<b>DTimer1计数模式控制位</b> 0: 连续计数模式 1: 单次计数模式
<b>0</b>	<b>TMR0MOD</b>	<b>DTimer0计数模式控制位</b> 0: 连续计数模式 1: 单次计数模式



**Table 9.65** 定时器控制寄存器TMRCON (DTimer Control)

15-14		13-12		11-10		9-8	
<i>R/W-0</i>		<i>R/W-0</i>		<i>R/W-0</i>		<i>R/W-0</i>	
<b>TMR3EMU</b>		<b>TMR2EMU</b>		<b>TMR1EMU</b>		<b>TMR0EMU</b>	
7-4				3	2	1	0
<i>R-0</i>				<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>				<b>TMR3TR</b>	<b>TMR2TR</b>	<b>TMR1TR</b>	<b>TMR0TR</b>
位编号	位符号	说明					
15-14	TMR3EMU	<b>仿真挂起时DTimer3运行状态控制位</b> 00: 挂起时, DTimer3计数器在下一个计数时钟自增1之后停止计数 01: 挂起时, DTimer3计数器将增至与周期设置值相等后复位为0, 然后停止计数 1x: 不受挂起影响, 继续运行					
13-12	TMR2EMU	<b>仿真挂起时DTimer2运行状态控制位</b> 00: 挂起时, DTimer2计数器在下一个计数时钟自增1之后停止计数 01: 挂起时, DTimer2计数器将增至与周期设置值相等后复位为0, 然后停止计数 1x: 不受挂起影响, 继续运行					
11-10	TMR1EMU	<b>仿真挂起时DTimer1运行状态控制位</b> 00: 挂起时, DTimer1计数器在下一个计数时钟自增1之后停止计数 01: 挂起时, DTimer1计数器将增至与周期设置值相等后复位为0, 然后停止计数 1x: 不受挂起影响, 继续运行					
9-8	TMR0EMU	<b>仿真挂起时DTimer0运行状态控制位</b> 00: 挂起时, DTimer0计数器在下一个计数时钟自增1之后停止计数 01: 挂起时, DTimer0计数器将增至与周期设置值相等后复位为0, 然后停止计数 1x: 不受挂起影响, 继续运行					
7-4	Reserved	保留位					
3	TMR3TR	<b>DTimer3启停控制位</b> 0: DTimer3停止计数 1: DTimer3计数使能 该控制位从0设置为1后, DTimer3定时计数器从0开始增计数					
2	TMR2TR	<b>DTimer2启停控制位</b> 0: DTimer2停止计数 1: DTimer2计数使能 该控制位从0设置为1后, DTimer2定时计数器从0开始增计数					
1	TMR1TR	<b>DTimer1启停控制位</b> 0: DTimer1停止计数 1: DTimer1计数使能 该控制位从0设置为1后, DTimer1定时计数器从0开始增计数					
0	TMR0TR	<b>DTimer0启停控制位</b> 0: DTimer0停止计数 1: DTimer0计数使能 该控制位从0设置为1后, DTimer0定时计数器从0开始增计数					

**Table 9.66** 定时器预分频寄存器0 TMRPSC0 (DTimer Prescaler0)

15-8		7-0	
<i>R/W-0</i>		<i>R/W-0</i>	
<b>TMR1PSC</b>		<b>TMR0PSC</b>	
位编号	位符号	说明	
15-8	TMR1PSC	DTimer1 预分频寄存器	
7-0	TMR0PSC	DTimer0 预分频寄存器	



**Table 9.67** 定时器预分频寄存器1 TMRPSC1 (DTimer Prescaler1)

<b>15-8</b>		<b>7-0</b>	
<i>R/W-0</i>		<i>R/W-0</i>	
<b>TMR3PSC</b>		<b>TMR2PSC</b>	
位编号	位符号	说明	
<b>15-8</b>	<b>TMR3PSC</b>	DTimer3预分频寄存器	
<b>7-0</b>	<b>TMR2PSC</b>	DTimer2预分频寄存器	

**Table 9.68** 定时器中断控制寄存器TMRINT (DTimer Interrupt)

<b>15-12</b>		<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>TMR3IE</b>	<b>TMR2IE</b>	<b>TMR1IE</b>	<b>TMR0IE</b>
<b>7-4</b>		<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>TMR3IF</b>	<b>TMR2IF</b>	<b>TMR1IF</b>	<b>TMR0IF</b>
位编号	位符号	说明			
<b>15-12</b>	<b>Reserved</b>	保留位			
<b>11</b>	<b>TMR3IE</b>	<b>DTimer3</b> 中断使能控制位 0: 禁止DTimer3中断 1: 使能DTimer3中断			
<b>10</b>	<b>TMR2IE</b>	<b>DTimer2</b> 中断使能控制位 0: 禁止DTimer2中断 1: 使能DTimer2中断			
<b>9</b>	<b>TMR1IE</b>	<b>DTimer1</b> 中断使能控制位 0: 禁止DTimer1中断 1: 使能DTimer1中断			
<b>8</b>	<b>TMR0IE</b>	<b>DTimer0</b> 中断使能控制位 0: 禁止DTimer0中断 1: 使能DTimer0中断			
<b>7-4</b>	<b>Reserved</b>	保留位			
<b>3</b>	<b>TMR3IF</b>	<b>DTimer3</b> 中断标志位 0: 无DTimer3中断 1: 有DTimer3中断 写0清除该标志, 写1无效			
<b>2</b>	<b>TMR2IF</b>	<b>DTimer2</b> 中断标志位 0: 无DTimer2中断 1: 有DTimer2中断 写0清除该标志, 写1无效			
<b>1</b>	<b>TMR1IF</b>	<b>DTimer1</b> 中断标志位 0: 无DTimer1中断 1: 有DTimer1中断 写0清除该标志, 写1无效			
<b>0</b>	<b>TMR0IF</b>	<b>DTimer0</b> 中断标志位 0: 无DTimer0中断 1: 有DTimer0中断 写0清除该标志, 写1无效			



### 9.6.4 计数时钟与预分频器

4个定时器的计数时钟TMRxCLK都通过8位的预分频计数器对DSP系统时钟 $f_{DSYS}$ 分频而得到。分频系数由寄存器TMRPSC0/1中的TMRxPSC ( $x = 0-3$ ) 设置值决定。计数时钟TMRxCLK的周期时间由下式决定:

$$T_{TMRxCLK} = (TMRxPSC+1) \times T_{DSYS} \quad (x = 0-3)$$

启动定时器DTimer $x$  ( $x = 0-3$ ) 后, 对应的分频计数器自动在系统时钟下从0开始增计数, 当计数值与TMRxPSC设置值相等时, 产生一个有效的定时器计数时钟, 然后再清0重新开始计数, 以此周而复始。

当关闭定时器, 或对TMRxPSC ( $x = 0-3$ ) 写入与原值不同的设置值时, 硬件都会自动把对应定时器的预分频计数器清0。为保证计数时钟频率的稳定, 建议用户在定时器关闭状态下修改TMRxPSC ( $x = 0-3$ ) 设置值。

### 9.6.5 周期寄存器与定时时间

用户使用定时器的周期寄存器TMRxPRD ( $x = 0-3$ ) 来设置每个定时器的定时时间。定时器的定时时间由下式决定:

$$T_{TMRx} = (TMRxPRD+1) \times T_{DTMxCLK} \quad (x = 0-3)$$

周期寄存器TMRxPRD与周期缓存寄存器TMRxPBUF一一对应, 定时器的内部比较器固定使用周期缓存寄存器的设置值与定时计数值比较, 当两者相等时, 计数器在下一个计数时钟复位为0, 并把周期寄存器的设置值更新到周期缓存寄存器中。

当TMRxTR = 0, 即定时器关闭时, 用户可以直接对周期缓存寄存器写入设置值, 以使该设置值在下次定时器工作时立即有效。当TMRxTR = 1定时器正常工作时, 所有对周期缓存寄存器的写入动作均无效, 用户必须通过周期寄存器来设置定时器的定时时间, 以保证定时器工作的正确性。

### 9.6.6 工作模式

每个定时器都有一个独立的控制位TMRxMOD ( $x = 0-3$ ) 用于设置定时器的计数模式。

TMRxMOD = 0, 定时器设置为连续计数模式。该模式下启动定时器后, 定时计数器从0开始增计数, 当计数值等于周期缓存寄存器设置值后, 计数器在下一个计数时钟复位为0, 然后再从0开始增计数, 以此周而复始, 直到TMRxTR控制位软件清0。

TMRxMOD = 1, 定时器设置为单次计数模式。该模式下启动定时器后, 定时计数器从0开始增计数, 当计数值等于周期缓存寄存器设置值后, 计数器在下一个计数时钟复位为0, 硬件自动把TMRxTR控制位清0, 然后停止计数。

### 9.6.7 定时器的启停控制

4个定时器都使用独立的启停控制位TMRxTR ( $x = 0-3$ ) 来控制定时器的运行或停止。

用户可以使用软件置位TMRxTR ( $x = 0-3$ ) 的方式来触发定时器运行, 也可以通过PWM计数归零时硬件自动置位TMRxTR的方法来启动定时器 (PWM计数归零触发DTimer的设置, 见PWM章节的描述)。

当定时器设置在单次计数模式时, 定时计数满后硬件会自动对TMRxTR ( $x = 0-3$ ) 位清0, 并结束计数。用户也可以在定时器运行的任意时刻, 通过软件方法对TMRxTR ( $x = 0-3$ ) 位清0以停止定时器的工作, 硬件在下一个系统时钟自动把定时计数器和分频计数器复位为0。

### 9.6.8 ADC触发

每个DSP定时器都可以用来触发ADC, 使用TMRMOD寄存器的TMRxADT ( $x = 0-3$ ) 控制位可以独立控制每个定时器对ADC的触发模式。

TMRxADT = 00或01, DTimer $x$  ( $x = 0-3$ ) 定时器不会触发ADC。

TMRxADT = 10, 当DTimer $x$  ( $x = 0-3$ ) 计数值等于周期设置值后, 会自动置位SOC1触发ADC。

TMRxADT = 11, 当DTimer $x$  ( $x = 0-3$ ) 计数值等于周期设置值后, 会自动置位SOC2触发ADC。

关于ADC触发的具体情况参见ADC章节。

### 9.6.9 定时器中断

定时器启动后, 当定时计数值等于周期缓存寄存器设置值后, 计数器在下一个计数时钟复位为0, 同时置位中断标志位TMRxIF ( $x = 0-3$ ), 如果此时定时器中断使能位TMRxIE = 1 ( $x = 0-3$ ), 则触发相应的中断。

定时器中断标志只能软件清除, 通过在相应的中断标志位TMRxIF ( $x = 0-3$ ) 写0可以清除中断标志。



### 9.6.10 定时器工作波形示例

以DTimer0为例，设置TMR0PRD = 3，TMR0PSC = 2时的关键信号波形如下图9-6-2所示：

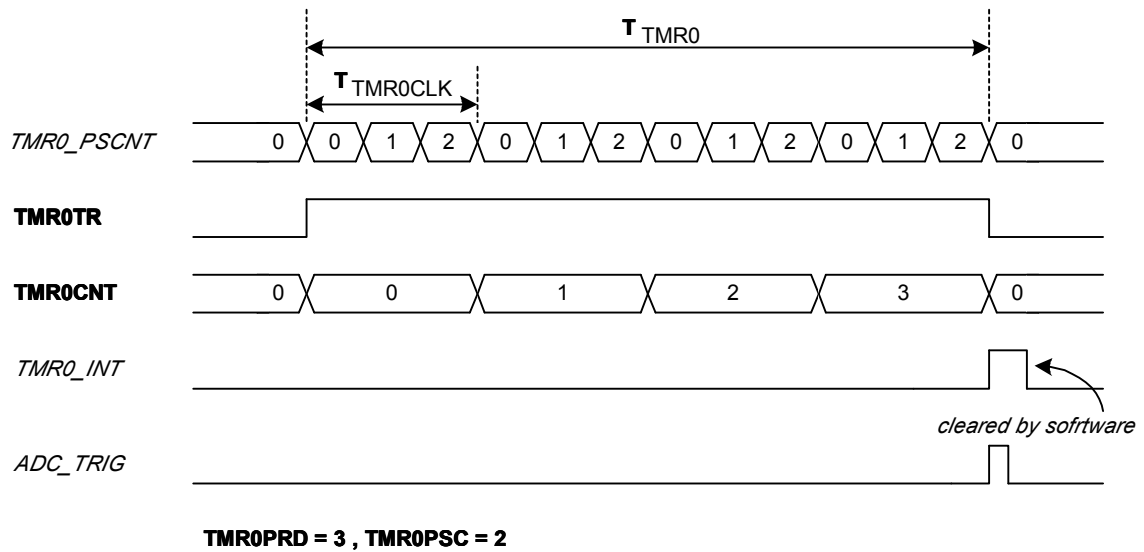


图9-6-2 DTimer0工作波形



## 9.7 输入捕捉模块

### 9.7.1 特性

- 4路施密特输入捕捉引脚
- 每个输入引脚有独立的滤波时钟设置
- 捕捉输入信号预分频系数设定（1-62分频）
- 4个16位2级捕捉存储栈
- 捕捉事件可设置为上升沿、下降沿、双沿
- 使用DTimer2或DTimer3作为捕捉时基

SH99F100内部含有4个输入捕捉单元，每个单元对应一个输入捕捉引脚CAPx(x=0~3)，当该引脚设置为输入捕获功能时，对应的输入捕捉单元工作有效。

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，硬件会根据捕捉时基的设置信息，自动抓取DTimer2或DTimer3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。

### 9.7.2 逻辑框图

每个输入捕捉单元的逻辑框图如下图所示：

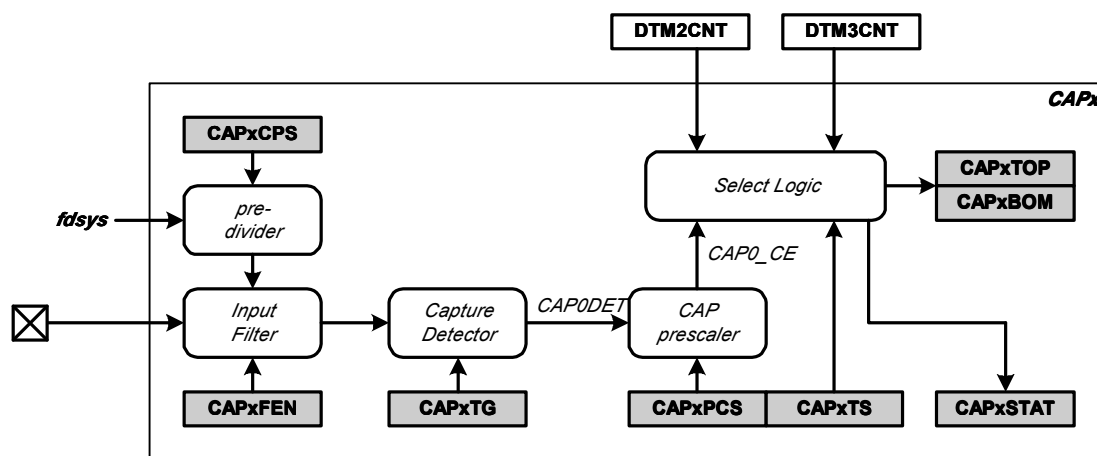


图9-7-1 输入捕捉单元CAPx (x = 0-3) 的逻辑框图

### 9.7.3 控制寄存器

输入捕捉模块使用的所有控制寄存器如下表所示：

类别	缩写符号	功能说明
捕捉控制	CAP0CON CAP1CON CAP2CON CAP3CON	输入捕捉控制寄存器
存储栈	CAP0TOP CAP0BOM	输入捕捉0存储栈栈顶/栈底寄存器
	CAP1TOP CAP1BOM	输入捕捉1存储栈栈顶/栈底寄存器
	CAP2TOP CAP2BOM	输入捕捉2存储栈栈顶/栈底寄存器
	CAP3TOP CAP3BOM	输入捕捉3存储栈栈顶/栈底寄存器
中断控制	CAPINT	输入捕捉中断控制寄存器



Table 9.69 输入捕捉控制寄存器CAP0CON

15	14-13	12-8	7-6	5-4	3	2-0
R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
CAP0TS	CAP0TG	CAP0PSC	Reserved	CAP0STAT	CAP0FEN	CAP0CPS
位编号	位符号	说明				
15	CAP0TS	<b>CAP0捕捉时基选择位</b> 0: 使用DTimer2作为捕捉时基 1: 使用DTimer3作为捕捉时基				
14-13	CAP0TG	<b>CAP0引脚边沿检测控制位</b> 00: 无检测, CAP0引脚作为普通IO口使用 01: 检测上升沿 10: 检测下降沿 11: 检测双沿				
12-8	CAP0PSC	<b>CAP0输入信号预分频系数设置位, X为输入</b> 00000:X/1      00001:X/2 00010:X/4      00011:X/6 00100:X/8      00101:X/10 00110:X/12     00111:X/14 ..... 11010:X/52     11011:X/54 11100:X/56     11101:X/58 11110:X/60     11111:X/62				
7-6	Reserved	保留位				
5-4	CAP0STAT	<b>CAP0存储栈状态标志位</b> 00: 存储栈为空 01: 已有一个值入栈 (CAP0TOP有捕捉值, CAP0BOM无捕捉值) 10: 已有两个值入栈 (CAP0TOP和CAP0BOM都有捕捉值) 11: 存储栈中已有两个值且又压入一个新值, 导致最先压入的值出栈 (即丢失)				
3	CAP0FEN	<b>CAP0引脚滤波使能位</b> 0: 禁止CAP0引脚上的滤波功能 1: 使能CAP0引脚上的滤波功能				
2-0	CAP0CPS	<b>CAP0引脚滤波时钟分频系数</b> 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$				





Table 9.70 输入捕捉控制寄存器CAP1CON

15	14-13	12-8	7-6	5-4	3	2-0
R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
<b>CAP1TS</b>	<b>CAP1TG</b>	<b>CAP1PSC</b>	<b>Reserved</b>	<b>CAP1STAT</b>	<b>CAP1FEN</b>	<b>CAP1CPS</b>
位编号	位符号	说明				
15	CAP1TS	<b>CAP1捕捉时基选择位</b> 0: 使用DTimer2作为捕捉时基 1: 使用DTimer3作为捕捉时基				
14-13	CAP1TG	<b>CAP1引脚边沿检测控制位</b> 00: 无检测, CAP1引脚作为普通IO口使用 01: 检测上升沿 10: 检测下降沿 11: 检测双沿				
12-8	CAP1PSC	<b>CAP1输入信号预分频系数设置位, X为输入</b> 00000:X/1      00001:X/2 00010:X/4      00011:X/6 00100:X/8      00101:X/10 00110:X/12     00111:X/14 ..... 11010:X/52     11011:X/54 11100:X/56     11101:X/58 11110:X/60     11111:X/62				
7-6	Reserved	保留位				
5-4	CAP1STAT	<b>CAP1存储栈状态标志位</b> 00: 存储栈为空 01: 已有一个值入栈 (CAP1TOP有捕捉值, CAP1BOM无捕捉值) 10: 已有两个值入栈 (CAP1TOP和CAP1BOM都有捕捉值) 11: 存储栈中已有两个值且又压入一个新值, 导致最先压入的值出栈 (即丢失)				
3	CAP1FEN	<b>CAP1引脚滤波使能位</b> 0: 禁止CAP1引脚上的滤波功能 1: 使能CAP1引脚上的滤波功能				
2-0	CAP1CPS	<b>CAP1引脚滤波时钟分频系数</b> 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$				



Table 9.71 输入捕捉控制寄存器CAP2CON

15	14-13	12-8	7-6	5-4	3	2-0
R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
<b>CAP2TS</b>	<b>CAP2TG</b>	<b>CAP2PSC</b>	<b>Reserved</b>	<b>CAP2STAT</b>	<b>CAP2FEN</b>	<b>CAP2CPS</b>
位编号	位符号	说明				
15	CAP2TS	<b>CAP2捕捉时基选择位</b> 0: 使用DTimer2作为捕捉时基 1: 使用DTimer3作为捕捉时基				
14-13	CAP2TG	<b>CAP2引脚边沿检测控制位</b> 00: 无检测, CAP2引脚作为普通IO口使用 01: 检测上升沿 10: 检测下降沿 11: 检测双沿				
12-8	CAP2PSC	<b>CAP2输入信号预分频系数设置位, X为输入</b> 00000:X/1      00001:X/2 00010:X/4      00011:X/6 00100:X/8      00101:X/10 00110:X/12     00111:X/14 ..... 11010:X/52     11011:X/54 11100:X/56     11101:X/58 11110:X/60     11111:X/62				
7-6	Reserved	保留位				
5-4	CAP2STAT	<b>CAP2存储栈状态标志位</b> 00: 存储栈为空 01: 已有一个值入栈 (CAP2TOP有捕捉值, CAP2BOM无捕捉值) 10: 已有两个值入栈 (CAP2TOP和CAP2BOM都有捕捉值) 11: 存储栈中已有两个值且又压入一个新值, 导致最先压入的值出栈 (即丢失)				
3	CAP2FEN	<b>CAP2引脚滤波使能位</b> 0: 禁止CAP2引脚上的滤波功能 1: 使能CAP2引脚上的滤波功能				
2-0	CAP2CPS	<b>CAP2引脚滤波时钟分频系数</b> 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$				



Table 9.72 输入捕捉控制寄存器CAP3CON

15	14-13	12-8	7-6	5-4	3	2-0
R/W-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
CAP3TS	CAP3TG	CAP3PSC	Reserved	CAP3STAT	CAP3FEN	CAP3CPS
位编号	位符号	说明				
15	CAP3TS	<b>CAP3捕捉时基选择位</b> 0: 使用DTimer2作为捕捉时基 1: 使用DTimer3作为捕捉时基				
14-13	CAP3TG	<b>CAP3引脚边沿检测控制位</b> 00: 无检测, CAP3引脚作为普通IO口使用 01: 检测上升沿 10: 检测下降沿 11: 检测双沿				
12-8	CAP3PSC	<b>CAP3输入信号预分频系数设置位, X为输入</b> 00000:X/1      00001:X/2 00010:X/4      00011:X/6 00100:X/8      00101:X/10 00110:X/12     00111:X/14 ..... 11010:X/52     11011:X/54 11100:X/56     11101:X/58 11110:X/60     11111:X/62				
7-6	Reserved	保留位				
5-4	CAP3STAT	<b>CAP3存储栈状态标志位</b> 00: 存储栈为空 01: 已有一个值入栈 (CAP3TOP有捕捉值, CAP3BOM无捕捉值) 10: 已有两个值入栈 (CAP3TOP和CAP3BOM都有捕捉值) 11: 存储栈中已有两个值且又压入一个新值, 导致最先压入的值出栈 (即丢失)				
3	CAP3FEN	<b>CAP3引脚滤波使能位</b> 0: 禁止CAP3引脚上的滤波功能 1: 使能CAP3引脚上的滤波功能				
2-0	CAP3CPS	<b>CAP3引脚滤波时钟分频系数</b> 000: 1/1 $f_{DSYS}$ 001: 1/2 $f_{DSYS}$ 010: 1/4 $f_{DSYS}$ 011: 1/16 $f_{DSYS}$ 100: 1/32 $f_{DSYS}$ 101: 1/64 $f_{DSYS}$ 110: 1/128 $f_{DSYS}$ 111: 1/256 $f_{DSYS}$				



**Table 9.73** 存储栈栈顶寄存器CAPxTOP (x = 0-3)

<b>15-0</b>		
<i>R-0</i>		
<b>CAPxTOP</b>		
位编号	位符号	说明
<b>15-0</b>	<b>CAPxTOP (x = 0-3)</b>	<b>CAPx的存储栈栈顶寄存器</b> 只读寄存器, 每次读CAPxTOP后, 栈顶的值出栈, 栈底寄存器的值压入栈顶寄存器

**Table 9.74** 存储栈栈底寄存器CAPxBOM (x = 0-3)

<b>15-0</b>		
<i>R-0</i>		
<b>CAPxBOM</b>		
位编号	位符号	说明
<b>15-0</b>	<b>CAPxBOM (x = 0-3)</b>	<b>CAPx的存储栈栈底寄存器</b> 只读寄存器, 每次读CAPxBOM之后, 栈底寄存器的值清0

**Table 9.75** 输入捕捉中断控制寄存器CAPINT

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>CAP3IM</b>	<b>CAP2IM</b>	<b>CAP1IM</b>	<b>CAP0IM</b>	<b>CAP3IE</b>	<b>CAP2IE</b>	<b>CAP1IE</b>	<b>CAP0IE</b>
<b>7-4</b>				<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R-0</i>				<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>				<b>CAP3IF</b>	<b>CAP2IF</b>	<b>CAP1IF</b>	<b>CAP0IF</b>
位编号	位符号	说明					
<b>15-12</b>	<b>CAPxIM (x = 0-3)</b>	<b>CAPxIE = 1</b> 时, 输入捕捉 <b>CAPx (x = 0-3)</b> 中断模式选择位 0: 每两次捕捉产生一次中断 1: 每次捕捉都产生一次中断					
<b>11-8</b>	<b>CAPxIE (x = 0-3)</b>	输入捕捉 <b>CAPx (x = 0-3)</b> 中断使能控制位 0: 禁止输入捕捉中断 1: 使能输入捕捉中断					
<b>7-4</b>	<b>Reserved</b>	保留位					
<b>3-0</b>	<b>CAPxIF (x = 0-3)</b>	输入捕捉 <b>CAPx (x = 0-3)</b> 中断标志位 0: 无输入捕捉中断发生 1: 有输入捕捉中断发生 写0清除标志, 写1无效					



### 9.7.4 捕捉过程与存储栈操作

每个输入捕捉单元CAPx (x = 0-3) 都有一个专用的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 (CAPxTOP 和 CAPxBOM 都没有捕获值), 则最先进栈的数据存放于栈顶寄存器 CAPxTOP 中; 若存储栈不为空, 则下一次数据进栈时, CAPxBOM 值被压入到 CAPxTOP, 新数据被存放于栈底寄存器 CAPxBOM。

(2) 每次读取栈顶寄存器 CAPxTOP 后, 都会使 CAPxBOM 值被压入到 CAPxTOP, 同时 CAPxBOM 被清0; 每次读取 CAPxBOM 寄存器后, 都会使 CAPxBOM 被清0, 但 CAPxTOP 寄存器值保持不变。

每个输入捕捉单元 CAPx (x = 0-3) 使用 CAPxSTAT[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 CAPxTOP 和 CAPxBOM 为灰底时表示该寄存器中有捕捉数据)。

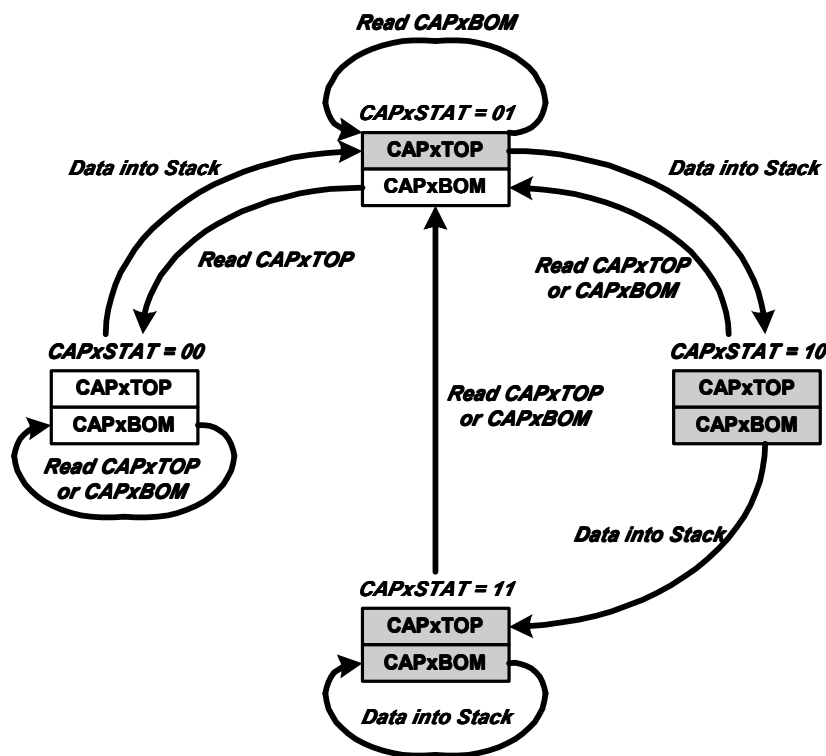


图9-7-2 输入捕捉过程状态机转换与存储栈操作图

第一次捕捉: 此时存储栈为空, CAPxSTAT[1:0] = 00。当 CAPx 引脚上发生了指定的跳变后, DTimer2 或 DTimer3 的计数值被压入栈顶寄存器 CAPxTOP 中, CAPxSTAT[1:0] 变为 01。若在下一捕捉发生之前对栈顶寄存器 CAPxTOP 进行了读取操作, 则读取后 CAPxTOP 寄存器被清0, 且 CAPxSTAT[1:0] 变为 00; 若在下一捕捉发生之前对栈底寄存器 CAPxBOM 进行了读取操作, 由于 CAPxBOM 中没有捕捉值, 因此固定读到0, 且不会改变 CAPxSTAT 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 CAPxBOM 中, 同时 CAPxSTAT[1:0] 变为 10。若在下一捕捉发生之前对栈顶寄存器 CAPxTOP 进行了读取操作, 则 CAPxTOP 的值被读取之后, 栈底寄存器 CAPxBOM 值将压入栈顶寄存器 CAPxTOP 中, 且 CAPxSTAT[1:0] 变为 01; 若在下一捕捉发生之前对栈底寄存器 CAPxBOM 进行了读取操作, 则读取后 CAPxBOM 寄存器被清0, 且 CAPxSTAT[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 CAPxTOP 中数据将被弹出并丢弃, 而栈底寄存器 CAPxBOM 值将压入栈顶寄存器 CAPxTOP 中, 新捕捉到的值被压入栈底寄存器 CAPxBOM 中, 并且 CAPxSTAT[1:0] 变为 11。若在下一捕捉发生之前对栈顶寄存器 CAPxTOP 进行了读取操作, 则 CAPxTOP 的值被读取之后, 栈底寄存器 CAPxBOM 的值将压入栈顶寄存器 CAPxTOP 中, 且 CAPxSTAT[1:0] 变为 01; 若在下一捕捉发生之前对栈底寄存器 CAPxBOM 进行了读取操作, 则读取后 CAPxBOM 寄存器被清0, 且 CAPxSTAT[1:0] 变为 01。

CAPxSTAT[1:0] = 11, 表明存储栈发生了溢出, 有数据已经丢失。



### 9.7.5 输入捕捉引脚滤波设置

4个输入捕捉引脚CAPx (x = 0-3) 的输入信号都可以通过内部滤波电路保证检测的可靠性。

当CAPxFEN = 1使能CAPx引脚的滤波功能后, CAPx引脚上的信号必须保持至少3个滤波时钟周期的稳定电平才会使滤波器输出信号发生变化。CAPx引脚的滤波时钟通过CAPxCON寄存器中CAPxCPS (x = 0-3) 控制位来设置分频系数。

### 9.7.6 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CAPxPSC (x = 0-3) 控制位可以设置每个捕捉单元的预分频参数。

每次对CAPxPSC (x = 0-3) 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。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CAP0PSC[4:0] = 00010时, 实际触发捕捉的信号波形如下图所示:

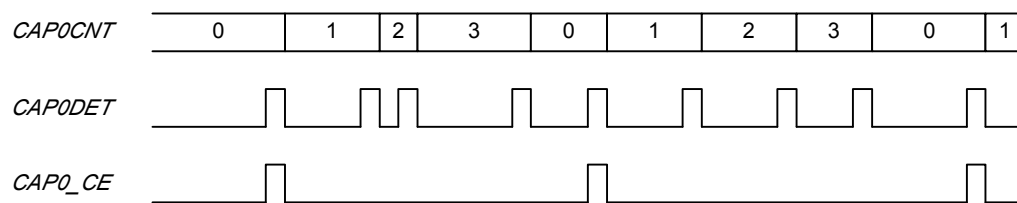


图9-7-3 预分频设置的工作波形

### 9.7.7 输入捕捉中断

每个输入捕捉单元使用CAPINT寄存器的CAPxIM (x = 0-3) 控制位来设置在每次或每两次捕捉之后产生中断。

CAPxIM = 0 (x = 0-3) 时, 每两次捕捉后会置位CAPxIF标志, 如果此时CAPxIE = 1则产生中断。由于在任何复位条件产生之后存储栈都为空, CAPxSTAT[1:0] = 00, 因此中断到来时2级存储栈CAPxTOP和CAPxBOM都是有数据的, 中断服务程序中必须把CAPxTOP和CAPxBOM中的数据都读取, 才能保证下次中断到来时, 不会因为存储栈满而丢掉前面的数据。

CAPxIM = 1 (x = 0-3) 时, 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。



## 9.8 输出比较模块

### 9.8.1 特性

- 3路比较输出通道OCx (x = 0-2)
- 每个比较输出单元提供3种工作模式：单比较模式、双比较模式和边沿对齐的简单PWM模式

输出比较模块包含3个输出比较单元，分别产生3个通道的输出比较信号OCx (x = 0-2)。每个输出比较单元可以独立选择DTimer2或DTimer3作为比较时基。当比较时基定时器的计数值与一个或两个比较寄存器设置值（取决于所选的工作模式）匹配时，会使比较输出信号OCx (x = 0-2) 的电平发生翻转，具体的电平翻转特性由工作模式决定。

当比较输出单元工作在带OCFLT引脚保护的PWM模式时，3个比较输出单元共用一个输入保护引脚OCFLT和滤波电路。

为保证输出比较的结果完全可控，建议用户在操作输出比较模块的控制寄存器时关闭DTimer2或DTimer3比较时基。

### 9.8.2 逻辑框图

每个输出比较单元都有相同的电路结构，其逻辑框图如下图所示：

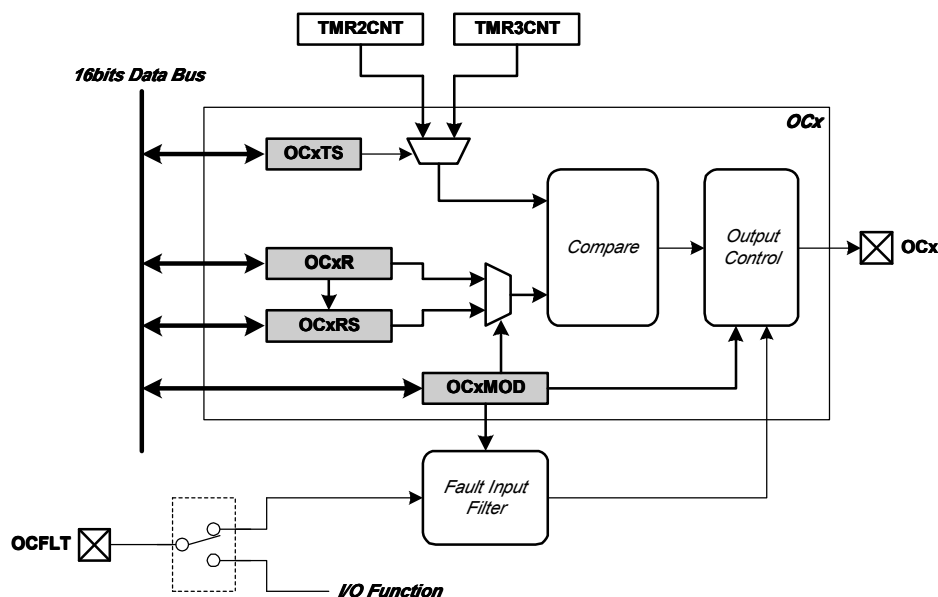


图9-8-1 输出比较单元OCx (x = 0-2) 的逻辑框图

### 9.8.3 控制寄存器

输出比较模块使用的所有控制寄存器如下表所示：

类别	缩写符号	功能说明
通道控制	OC0CON	输出比较0控制寄存器
	OC1CON	输出比较1控制寄存器
	OC2CON	输出比较2控制寄存器
比较值设置	OC0R OC0RS	通道0比较寄存器 通道0比较辅助寄存器
	OC1R OC1RS	通道1比较寄存器 通道1比较辅助寄存器
	OC2R OC2RS	通道2比较寄存器 通道2比较辅助寄存器
	OCINT	输出比较中断控制寄存器



**Table 9.76** 输出比较控制寄存器OCxCON (x = 0-2)

<b>15-4</b>		<b>3</b>	<b>2-0</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>OCxTS</b>	<b>OCxMOD</b>

位编号	位符号	说明
<b>15-4</b>	<b>Reserved</b>	保留位
<b>3</b>	<b>OCxTS (x = 0-2)</b>	<b>OCx (x = 0-2) 比较时基选择位</b> 0: 使用DTimer3作为比较时基 1: 使用DTimer2作为比较时基
<b>2-0</b>	<b>OCxMOD (x = 0-2)</b>	<b>OCx (x = 0-2) 输出比较模式选择控制位</b> 000: 输出比较通道禁止, OCx引脚作为普通IO口使用 001: 初始化OCx引脚为低电平, 比较匹配事件强制OCx引脚为高电平 010: 初始化OCx引脚为高电平, 比较匹配事件强制OCx引脚为低电平 011: 比较匹配事件使OCx引脚的电平交替翻转 100: 初始化OCx引脚为低电平, 在OCx引脚产生单个输出脉冲 101: 初始化OCx引脚为低电平, 在OCx引脚产生连续的输出脉冲 110: 简单PWM输出模式, OCFLT引脚保护禁止 111: 简单PWM输出模式, OCFLT引脚保护使能

**Table 9.77** 通道比较寄存器OCxR (x = 0-2)

<b>15-0</b>	
<i>R/W-0</i>	
<b>OCxR</b>	

位编号	位符号	说明
<b>15-0</b>	<b>OCxR (x = 0-2)</b>	<b>通道x (x = 0-2) 的比较值设置寄存器</b> 在单比较和双比较模式下, 当通道x选定的比较时基计数值与设置值相等时, 产生相应的比较匹配事件; 在简单PWM输出模式下, 作为占空比设置寄存器

**Table 9.78** 通道比较辅助寄存器OCxRS (x = 0-2)

<b>15-0</b>	
<i>R/W-0</i>	
<b>OCxRS</b>	

位编号	位符号	说明
<b>15-0</b>	<b>OCxRS (x = 0-2)</b>	<b>通道x (x = 0-2) 的比较值设置辅助寄存器</b> 在单比较模式下, 辅助寄存器可读可写, 读取值即为前次有效写入值; 在双比较模式下, 辅助寄存器与OCxR类似, 用作同一通道的第二个比较寄存器使用, 当通道x选定的比较时基计数值与设置值相等时, 产生相应的比较匹配事件; 在简单PWM输出模式下, 作为占空比缓存寄存器, 只能读取不能写入





Table 9.79 输出比较中断控制寄存器OCINT

<b>15-12</b>		<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<i>R-0</i>		<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>		<b>OCFIE</b>	<b>OC2IE</b>	<b>OC1IE</b>	<b>OC0IE</b>
<b>7-5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<i>R/W-0</i>	<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>OCFCPS</b>	<b>OCFST</b>	<b>OCFIF</b>	<b>OC2IF</b>	<b>OC1IF</b>	<b>OC0IF</b>

位编号	位符号	说明
<b>15-12</b>	<b>Reserved</b>	保留位
<b>11</b>	<b>OCFIE</b>	<b>OCFLT</b> 保护引脚中断使能位：（仅在 <b>OCxMOD = 111</b> 时有效） 0: 禁止OCFLT保护引脚中断 1: 允许OCFLT保护引脚中断
<b>10</b>	<b>OC2IE</b>	输出比较通道 <b>2</b> 中断使能位 0: 禁止通道 <b>2</b> 的比较中断 1: 允许通道 <b>2</b> 的比较中断
<b>9</b>	<b>OC1IE</b>	输出比较通道 <b>1</b> 中断使能位 0: 禁止通道 <b>1</b> 的比较中断 1: 允许通道 <b>1</b> 的比较中断
<b>8</b>	<b>OC0IE</b>	输出比较通道 <b>0</b> 中断使能位 0: 禁止通道 <b>0</b> 的比较中断 1: 允许通道 <b>0</b> 的比较中断
<b>7-5</b>	<b>OCFCPS</b>	<b>OCFLT</b> 引脚滤波时钟分频系数 000: $1/1 f_{DSYS}$ 001: $1/2 f_{DSYS}$ 010: $1/4 f_{DSYS}$ 011: $1/16 f_{DSYS}$ 100: $1/32 f_{DSYS}$ 101: $1/64 f_{DSYS}$ 110: $1/128 f_{DSYS}$ 111: $1/256 f_{DSYS}$
<b>4</b>	<b>OCFST</b>	<b>OCFLT</b> 保护引脚状态指示位：（仅在 <b>OCxMOD = 111</b> 时有效） 0: 无故障信号发生（OCFLT引脚上无低电平） 1: 有故障信号发生（OCFLT引脚上出现低电平）
<b>3</b>	<b>OCFIF</b>	<b>OCFLT</b> 保护引脚中断标志位：（仅在 <b>OCxMOD = 111</b> 时有效） 0: 无保护引脚中断发生 1: 有保护引脚中断发生 写0清除标志，写1无效
<b>2</b>	<b>OC2IF</b>	输出比较通道 <b>2</b> 中断标志位 0: 无通道 <b>2</b> 的比较中断发生 1: 有通道 <b>2</b> 的比较中断发生 写0清除标志，写1无效
<b>1</b>	<b>OC1IF</b>	输出比较通道 <b>1</b> 中断标志位 0: 无通道 <b>1</b> 的比较中断发生 1: 有通道 <b>1</b> 的比较中断发生 写0清除标志，写1无效
<b>0</b>	<b>OC0IF</b>	输出比较通道 <b>0</b> 中断标志位 0: 无通道 <b>0</b> 的比较中断发生 1: 有通道 <b>0</b> 的比较中断发生 写0清除标志，写1无效



### 9.8.4 单比较工作模式

OCxCON (x = 0-2) 寄存器中的控制位OCxMOD (x = 0-2) 设置为001、010或011时, 对应的输出比较通道工作在单比较模式。

#### 单比较输出高电平

当用户设置OCxCON (x = 0-2) 寄存器中的OCxMOD (x = 0-2) 控制位为001后, 硬件自动初始化OCx引脚输出低电平, 并保持低电平直到选定的比较时基定时器 (DTimer2或DTimer3) 计数值与OCxR设置值相等。当比较值相等后, 在下一个计数时钟, OCx引脚输出高电平, OCx引脚输出的高电平一直保持到改变OCxMOD模式。如果再对OCxMOD写入001, 将重新触发输出比较单元的比较, 即在OCx引脚上再次输出低电平, 并保持低电平直到选定的比较时基定时器计数值与OCxR设置值相等后, OCx引脚输出高电平。

OCx引脚输出由低变高的同时, 硬件置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

单比较输出高电平的关键信号时序关系如下图9-8-2所示 (信号的灰色部分表示不确定状态) :

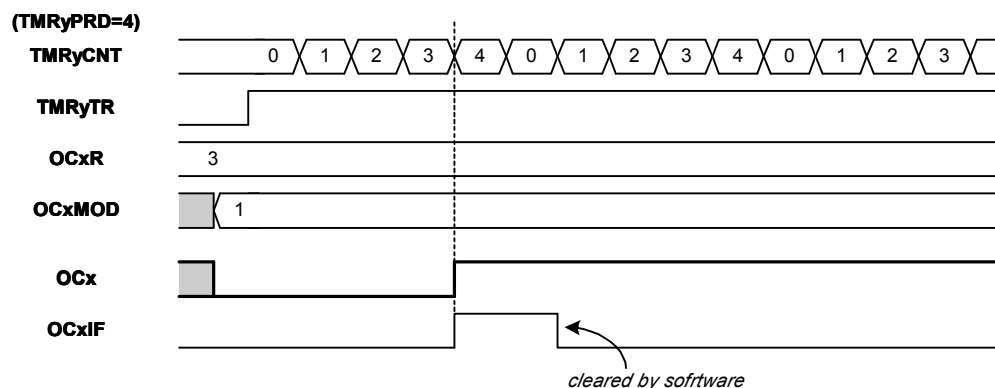


图9-8-2 单比较输出高电平的时序图 (x = 0-2, y = 2,3)

如果OCxR设置值大于选定的比较时基定时器TMRyPRD的设置值, 则比较条件始终不能成立, OCx输出固定为低电平。

#### 单比较输出低电平

当用户设置OCxCON (x = 0-2) 寄存器中的OCxMOD (x = 0-2) 控制位为010后, 硬件自动初始化OCx引脚输出高电平, 并保持高电平直到选定的比较时基定时器 (DTimer2或DTimer3) 计数值与OCxR设置值相等。当比较值相等后, 在下一个计数时钟, OCx引脚输出低电平, OCx引脚输出的低电平一直保持到改变OCxMOD模式。

OCx引脚输出由高变低的同时, 硬件置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

单比较输出低电平的关键信号时序关系如下图9-8-3所示 (信号的灰色部分表示不确定状态) :

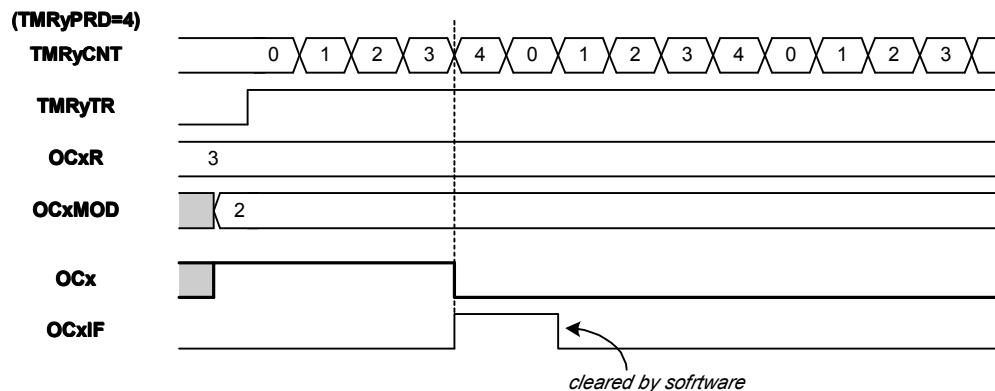


图9-8-3 单比较输出低电平的时序图 (x = 0-2, y = 2,3)

如果OCxR设置值大于选定的比较时基定时器TMRyPRD的设置值, 则比较条件始终不能成立, OCx输出固定为高电平。



### 单比较输出电平交替翻转

当用户设置OCxCON (x = 0-2) 寄存器中的OCxMOD (x = 0-2) 控制位为011后, 硬件自动设置该通道为单比较输出电平交替翻转模式。该模式下, 硬件首先保持OCx引脚输出原电平状态, 并在随后每一次选定的比较时基定时器(DTimer2或DTimer3)计数值与OCxR设置值相等后, 在OCx引脚翻转输出电平, 以交替输出高低电平。OCx引脚输出的电平状态一直保持到发生下一次翻转事件或改变OCxMOD模式。

每次比较匹配后, 硬件自动置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

单比较输出电平交替翻转模式的关键信号线时序关系如下图(以OCx输出原状态为低电平为例)所示:

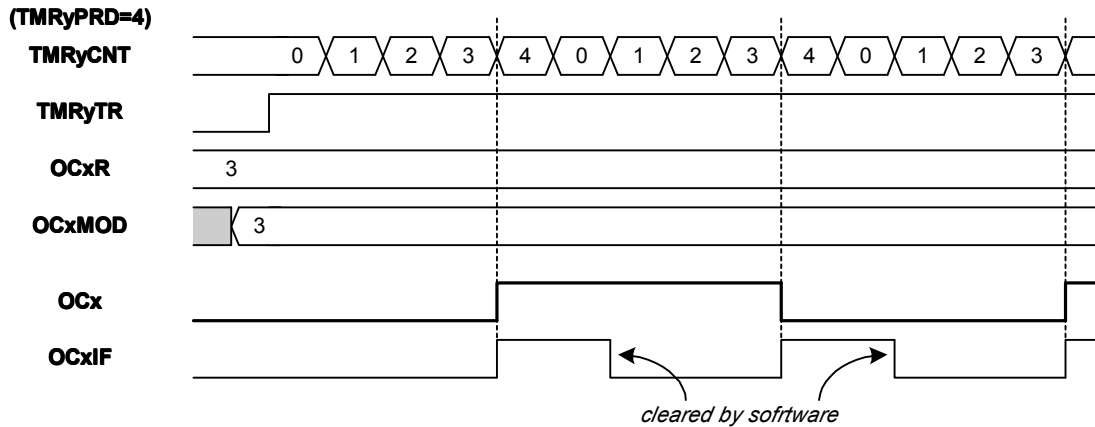


图9-8-4 单比较输出电平交替翻转的时序图 (x = 0-2, y = 2,3)

如果OCxR设置值大于选定的比较时基定时器TMRyPRD的设置值, 则比较条件始终不能成立, OCx输出固定为原电平。用户可以通过设置OCxMOD为001或010值初始化OCx引脚输出电平后, 再设置为011, 以实现初始状态为低电平或高电平的翻转输出波形。

### 9.8.5 双比较工作模式

OCxCON (x = 0-2) 寄存器中的控制位OCxMOD (x = 0-2) 设置为100或101时, 对应的输出比较通道工作在双比较模式。双比较工作模式下可以输出单脉冲或连续脉冲, 但是均满足如下工作要求:

(1) 选定的比较时基定时器计数值TMRyCNT (y = 2,3) 先后与OCxR和OCxRS两个比较寄存器设置值作比较, 当OCxR比较成功后再与OCxRS比较, 如果OCxR比较不成功, 则OCxRS不会被比较(连续脉冲输出时, 前次OCxRS比较不成功, 则下一次OCxR比较也不会发生);

(2) 当比较时基计数值与OCxR设置值相等时, 在下一个计数时钟, OCx引脚开始输出高电平;

(3) 当比较时基计数值与OCxRS设置值相等时, 在下一个计数时钟, OCx引脚开始输出低电平。

#### 单脉冲输出

当用户设置OCxCON (x = 0-2) 寄存器中的OCxMOD (x = 0-2) 控制位为100后, 硬件自动初始化OCx引脚输出低电平。当计数值与OCxR设置值相等后, 在下一个计数时钟, OCx引脚输出由低变高, 产生一个上升沿; 然后当计数值与OCxRS设置值相等后, 在下一个计数时钟, OCx引脚输出由高变低, 产生一个下降沿, 并保持该低电平直到改变OCxMOD模式设置值。

下降沿产生的同时, 硬件会置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

单脉冲输出方式最多只在OCx引脚产生一个上升沿和一个下降沿, 在一个脉冲输出结束后, 如果再对OCxMOD写入100, 将启动输出比较单元在OCx引脚上再次输出一个脉冲。



根据比较时基的周期设置值TMRyPRD (y = 2,3) 与OCxR和OCxRS (x = 0-2) 的特殊关系, 硬件处理结果如下表所示:

设置值的逻辑关系	硬件处理说明
$TMRyPRD < OCxR$	由于OCxR比较不会成功, 因此OCxRS也不会被比较, OCx固定输出低电平
$OCxR \leq TMRyPRD < OCxRS$	当比较时基计数值TMRyCNT = OCxR时, 在下一个计数时钟, OCx引脚输出由低变高, 产生一个上升沿, 然后持续保持高电平
$OCxRS \leq OCxR \leq TMRyPRD$	当比较时基计数值TMRyCNT = OCxR时, 在下一个计数时钟, OCx引脚输出由低变高, 产生一个上升沿, 如果比较时基工作在连续模式, 则时基在计满复位为0后继续增计数, 当计数值TMRyCNT = OCxRS时, OCx引脚输出再由高变低, 产生一个下降沿

下图所示分别是OCxR < OCxRS = TMRyPRD和OCxRS < OCxR < TMRyPRD情况下的示例波形:

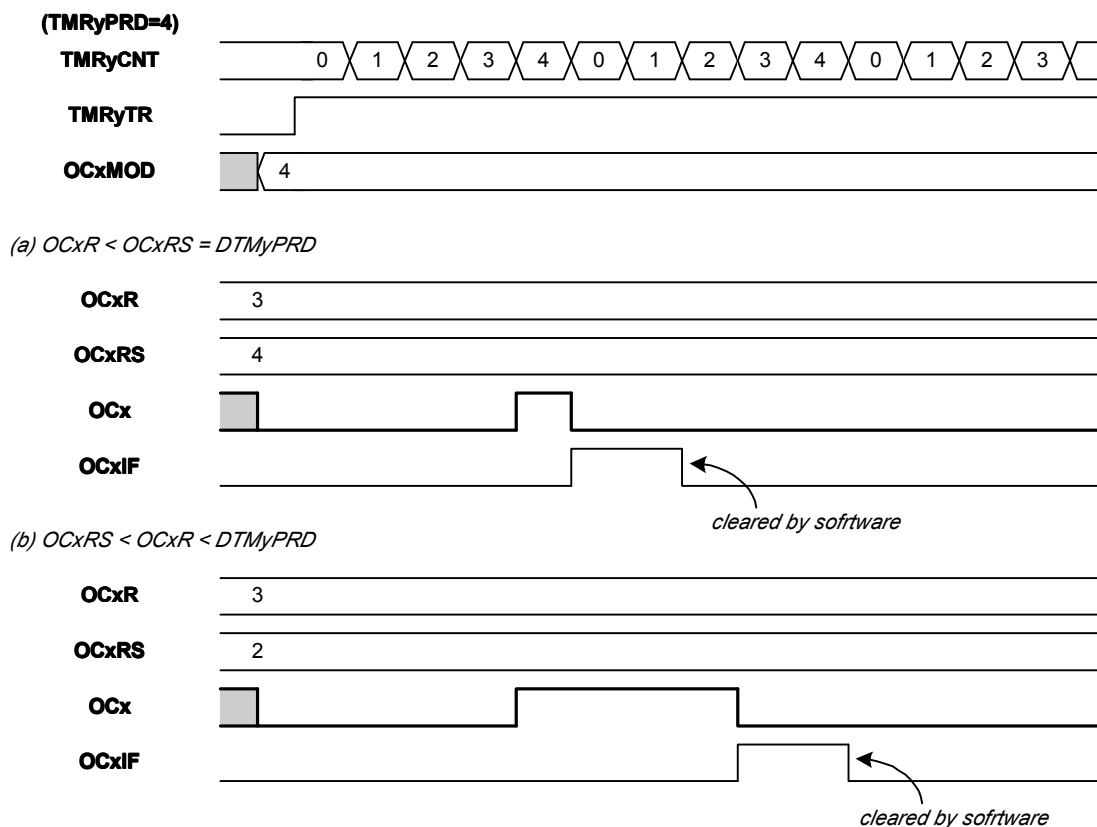


图9-8-5 不同设置条件下, 双比较单脉冲输出的时序图 (x = 0-2, y = 2,3)



### 连续脉冲输出

当用户设置OCxCON (x = 0-2) 寄存器中的OCxMOD (x = 0-2) 控制位为101后, 硬件自动初始化OCx引脚输出低电平。当计数值与OCxR设置值相等后, 在下一个计数时钟, OCx引脚输出由低变高, 产生一个上升沿; 然后当计数值与OCxRS设置值相等后, 在下一个计数时钟, OCx引脚输出由高变低, 产生一个下降沿, 并保持该低电平直到下一次计数值与OCxR相等。此后, 计数值依次与OCxR和OCxRS两个比较寄存器设置值作比较, 并按照这个规则产生上升沿和下降沿。改变OCxMOD模式或再写OCxMOD为101B将停止此功能。

每次下降沿产生的同时, 硬件会置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

根据比较时基的周期设置值TMRyPRD (y = 2,3) 与OCxR和OCxRS (x = 0-2) 的相互关系, 连续脉冲输出的处理方式与单脉冲方式相同, 请参考前表。

下图所示分别是OCxR < OCxRS = TMRyPRD和OCxRS < OCxR < TMRyPRD情况下的示例波形:

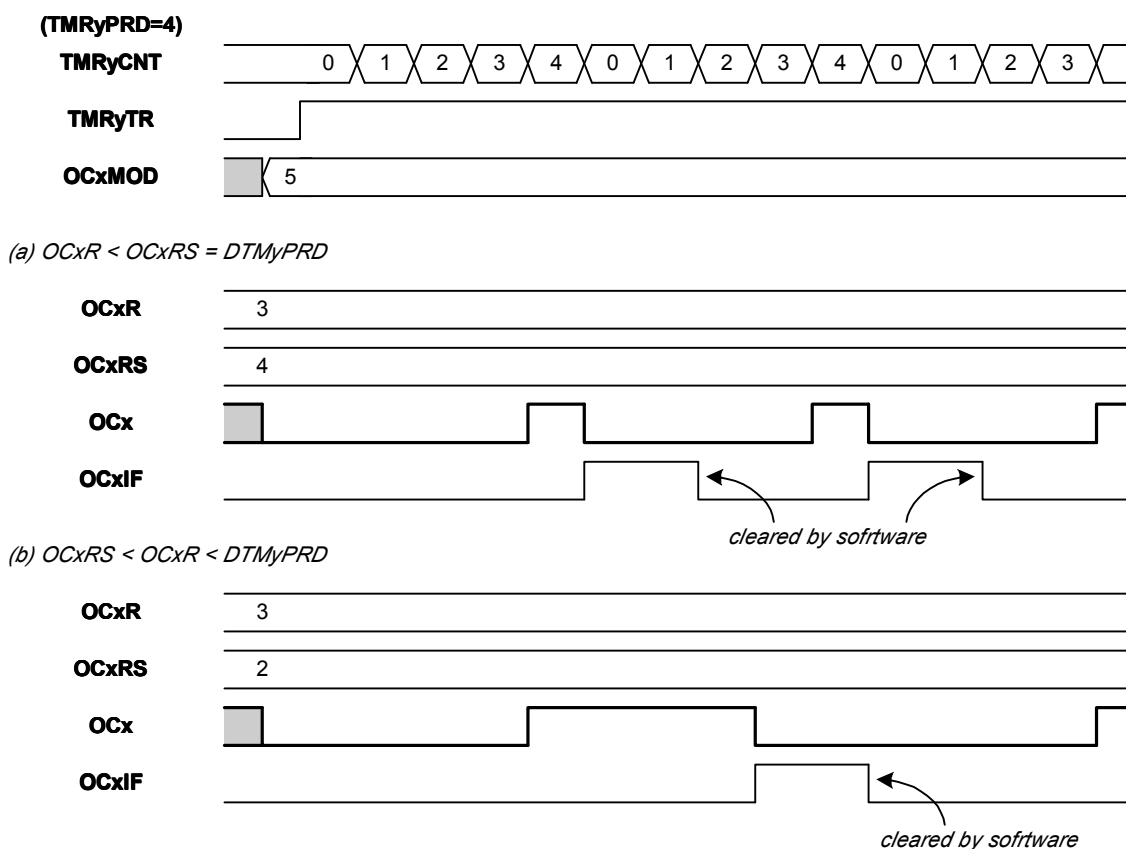


图9-8-6 不同设置条件下, 双比较连续脉冲输出的时序图 (x = 0-2, y = 2,3)



### 9.8.6 简单PWM输出模式

OCxCON (x = 0-2) 寄存器中的控制位OCxMOD (x = 0-2) 设置为110或111时, 对应的输出比较通道工作在简单PWM输出模式, 两种设置值的唯一差别在于是否允许OCFLT引脚的故障保护功能。

#### 简单PWM输出波形

当输出比较单元设置为简单PWM输出模式后, OCx引脚首先输出高阻态, 当比较时基定时器启动后, OCx引脚立即输出高电平, 然后使用OCxRS寄存器设置值与比较时基计数值比较, 当TMRyCNT = (OCxRS-1)后, 在下一个计数时钟, OCx引脚输出由高变低, 直到比较时基计满TMRyPRD的设置值后复位为止。当比较时基定时器复位的同时, OCx引脚输出恢复为高电平, OCxR寄存器设置值被重载到OCxRS寄存器中, 作为下一次比较的设置值, 然后在比较时基增计数过程中, 开始下一次输出比较, 并以此周而复始。

每次OCx引脚输出从高电平变为低电平的同时, 硬件会置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

当输出比较单元设置为简单PWM输出模式时, 软件操作需要遵循以下步骤:

- (1) 设置选定比较时基定时器的周期寄存器TMRyPRD, 设置PWM周期;
- (2) 通过写入OCxR寄存器设置PWM占空比;
- (3) 通过写入OCxRS寄存器设置初始化PWM占空比;
- (4) 设置OCxMOD为简单PWM输出模式;
- (5) 设置选定比较时基定时器的使能位, 开始输出PWM输出波形。

为保证PWM输出波形的可控性, 当OCxMOD设置为110或111后, OCxRS寄存器自动变为只读寄存器, 用户对占空比的修改必须通过设置OCxR寄存器来实现, 每次比较时基定时器复位的同时, OCxR寄存器设置值被重载到OCxRS寄存器中, 作为下一次比较的设置值。

简单PWM输出波形如下图所示。其中, PWM周期由TMRyPRD决定 ( $T_{OCPWM} = (TMRyPRD+1) * T_{TMRyCLK}$ , 详见DSP定时器相关描述); 第一次占空比时间 $T_{OCWD0}$ 由OCxRS寄存器决定, 以后的每次占空比时间 $T_{OCWD1}$ 由OCxR寄存器决定。

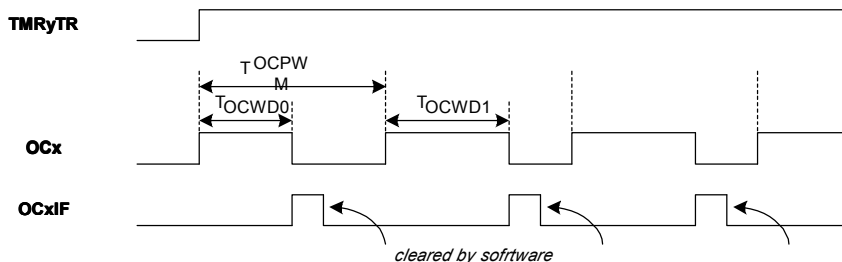


图9-8-6 简单PWM输出波形 (x = 0-2, y = 2,3)

简单PWM输出模式中, OCxR和OCxRS寄存器均表示占空比信息, 取值范围为从0到TMRyPRD+1, 占空比按下式计算:

$$Duty = \frac{OCxR(S)}{TMRyPRD + 1} \times 100\% \quad (x=0\sim 2, y=2,3)$$

(1) OCxR和OCxRS设置值为0, 表示0%占空比, 时基启动或计数复位为0的同时, OCx直接输出低电平。

(2) OCxR和OCxRS设置值大于或等于TMRyPRD+1, 表示100%占空比, OCx持续输出高电平。

每次OCx引脚输出从高电平变为低电平的同时, 硬件会置位相应的中断标志位OCxIF, 如果此时OCxIE = 1, 则触发相应的中断。中断标志位OCxIF由用户软件写0清除。

#### 故障保护输入

当3个输出比较单元中有任意一个工作于带故障保护的PWM输出模式 (OCxMOD = 111), 则OCFLT引脚自动配置为保护输入功能, 同时供所有配置为该模式的输出比较单元使用。当3个输出比较单元均未设置成该模式时, OCFLT引脚自动配置为普通I/O功能。

故障保护引脚OCFLT内建滤波电路以提高抗干扰能力, 用户使用OCFCPS控制位设置滤波时钟的分频系数。OCFLT引脚的低电平必须保持至少3个滤波时钟周期, 才能使滤波输出信号改变状态。

故障保护功能使能时, 当电路检测到OCFLT引脚出现的有效低电平后, 立即控制相应的输出比较单元输出高阻态, 以保护外部电路。此后, 即使故障消失, OCFLT引脚恢复为高电平, PWM输出还是保持高阻态, 用户需要对OCxMOD重新写入111, 才能重新启动PWM模式的正常输出。

故障保护功能使能时, 用户可以通过读取OCFST状态位, 来判断OCFLT保护引脚的输入状态。当电路检测到OCFLT引脚出现的有效低电平后, 如果此时OCFIE = 1, 则触发相应的中断。中断标志位OCFIF由用户软件写0清除。



## 9.9 运算放大器模块

### 9.9.1 特性

- 内建2个高性能运算放大器
- 单电源供电
- 内建频率补偿

### 9.9.2 功能说明

SH99F100的运算放大器模块内建的2个高性能运算放大器，均由单电源 $AV_{DD}$ 供电，可以稳定的提供高增益和中心频率补偿。当不使用运算放大器时，每个放大器都可以通过软件关闭以达到省电的目的。

运算放大器模块的逻辑框图如下图所示：

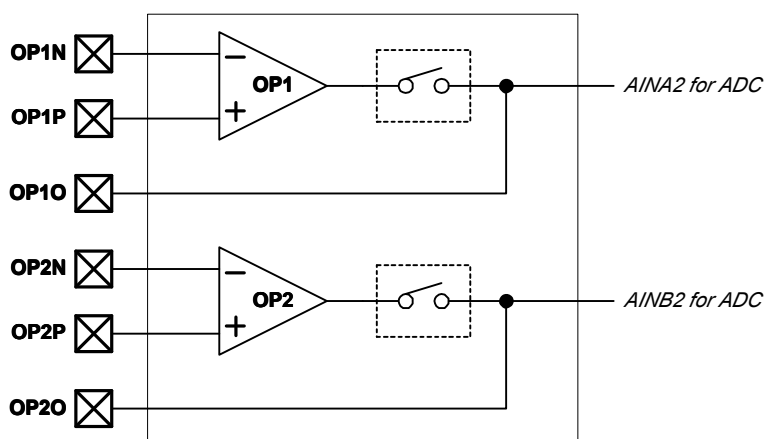


图9-9-1 运算放大器模块的逻辑框图

运算放大器1（OP1）提供三个外接引脚：正输入端口OP1P，负输入端口OP1N，输出端口OP1O。

运算放大器2（OP2）提供三个外接引脚：正输入端口OP2P，负输入端口OP2N，输出端口OP2O。

2个运放的输入、输出引脚均与ADC模块的模拟输入通道复用。OP1EN = 1时，输入引脚AINA0/OP1P、AINA1/OP1N和AINA2/OP1O直接用作OP1的正输入、负输入和输出引脚，OP1的输出在内部自动连接到ADC模块的AINA2模拟输入通道。OP1EN = 0，OP1功能被关闭，输入引脚AINA0/OP1P、AINA1/OP1N和AINA2/OP1O直接作为ADC模块的模拟输入通道。OP2EN = 1，输入引脚AINB0/OP2P、AINB1/OP2N和AINB2/OP2O直接用作OP2的正输入、负输入和输出引脚，OP2的输出在内部自动连接到ADC模块的AINB2模拟输入通道。OP2EN = 0，OP2功能被关闭，输入引脚AINB0/OP2P、AINB1/OP2N和AINB2/OP2O直接作为ADC模块的模拟输入通道。

### 9.9.3 控制寄存器

Table 9.80 运算放大器控制寄存器OPCON

15-2	1	0
<i>R-0</i>	<i>R/W-0</i>	<i>R/W-0</i>
<b>Reserved</b>	<b>OP2EN</b>	<b>OP1EN</b>

位编号	位符号	说明
15-2	<b>Reserved</b>	保留位
1	<b>OP2EN</b>	<b>OP2使能控制位</b> 0: 关闭OP2功能 1: 打开OP2功能
0	<b>OP1EN</b>	<b>OP1使能控制位</b> 0: 关闭OP1功能 1: 打开OP1功能

**10. 代码选项 (Code Option)****OP\_WDT[7]**

- 0: 禁止看门狗复位MCU (默认)
- 1: 允许看门狗复位MCU

**OP\_WDTPD[6]**

- 0: 掉电模式下禁止MCU看门狗工作 (默认)
- 1: 掉电模式下允许MCU看门狗工作

**OP\_DWDT[5]**

- 0: 禁止看门狗复位DSP (默认)
- 1: 允许看门狗复位DSP

**OP\_REG33[2]**

- 0: 禁止内部3.3V稳压源工作 (默认)
- 1: 允许内部3.3V稳压源工作

**OP\_LVREN[7]**

- 0: 禁止低电压复位功能 (默认)
- 1: 允许低电压复位功能

**OP\_LVRLR[6]**

- 0: 低电压复位设定电压为4.3V (默认)
- 1: 低电压复位设定电压为2.8V

**OP\_SCM[3]**

- 0: 在预热期间禁止时钟监测功能 (默认)
- 1: 在预热期间允许时钟监测功能

**OP\_ISP[7]**

- 0: 允许ISP功能 (默认)
- 1: 禁止ISP功能

**OP\_ISPPIN[6]**

- 0: 仅当P2.2和P2.3同时为低时进入ISP模式 (默认)
  - 1: 进入ISP模式时不检测P2.2和P2.3状态
- 注意: 此代码选项仅当OP\_ISP = 0时有效*

**OP\_OVL[5]**

- 0: OVL产生WDT复位 (默认)
- 1: OVL产生OVL中断





## 11. 电气特性

### 极限参数\*

数字供电电压.....-0.3V to +6.0V  
 模拟供电电压.....-0.3V to +3.6V  
 数字输入/输出电压..... GND-0.3V to  $V_{DD}+0.3V$   
 模拟输入电压..... AGND-0.3V to  $AV_{DD}+0.3V$   
 工作环境温度.....-40°C to +85°C  
 存储温度.....-55°C to +125°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 ( $V_{DD} = 2.7 - 5.5V$ ,  $AV_{DD} = 3.0 - 3.6V$ , GND = AGND = 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
数字工作电压	$V_{DD}$	2.7	5.0	5.5	V	$8MHz \leq f_{MSYS} \leq 16MHz$
模拟工作电压	$AV_{DD}$	3.0	3.3	3.6	V	DSP运行时100MHz
工作电流	$I_{OP}$	-	60	90	mA	$f_{MSYS} = 16MHz$ , PLL开; PWM引脚20kHz翻转, 其它输出引脚无负载, 所有输入引脚不浮动; CPU打开(执行NOP指令), DSP 100MHz取指运行, WDT打开, LVR打开, 流水线型ADC运行, 关闭其它所有功能; $V_{DD} = 5.0V$ , $AV_{DD} = 3.3V$
待机电流 (空闲模式: IDLE)	$I_{SB1}$	-	-	8	mA	$f_{MSYS} = 16MHz$ , PLL关; 所有输出引脚无负载, 所有输入引脚不浮动; WDT关闭, LVR打开, DSP复位; 关闭其它所有功能; $V_{DD} = 5.0V$ , $AV_{DD} = 3.3V$
待机电流 (掉电模式: Power-Down)	$I_{SB2}$	-	60	-	$\mu A$	$f_{OSC} = OFF$ , PLL关 所有输出引脚无负载, 所有输入引脚不浮动; WDT关闭, LVR打开, DSP复位; 关闭其它所有功能; $V_{DD} = 5.0V$ , $AV_{DD} = 3.3V$
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, T2EX, INT0, INT1, INT2, CAP0, CAP1, CAP2, CAP3, PDPINT0, PDPINT1, PDPINT2, PDPINT3, PDPINT4, OCFLT, DXINT0, DXINT1, DXINT2 (施密特触发器)
输入高电压2	$V_{IH2}$	$0.8 \times V_{DD}$	-	$V_{DD}$	V	$\overline{RST}$ , T0, T1, T2, T2EX, INT0, INT1, INT2, CAP0, CAP1, CAP2, CAP3, PDPINT0, PDPINT1, PDPINT2, PDPINT3, PDPINT4, OCFLT, DXINT0, DXINT1, DXINT2 (施密特触发器)
输入漏电流	$I_{IL}$	-1	-	1	$\mu A$	输入无上拉, $V_{IN} = DV_{DD}$ or DGND
上拉电阻*	$R_{PH}$	-	30	-	k $\Omega$	$DV_{DD} = 5.0V$ , $V_{IN} = DGND$
输出高电压	$V_{OH}$	$V_{DD} - 0.7$	-	-	V	I/O端口, $I_{OH} = -10mA$ , $DV_{DD} = 5.0V$
输出低电压	$V_{OL}$	-	-	GND+0.6	V	I/O端口, $I_{OL} = 15mA$ , $DV_{DD} = 5.0V$

注意: “\*”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。



逐次逼近型3.3V模/数转换器电气特性 ( $V_{DD} = 3.0 - 5.5V$ ,  $AV_{DD} = 3.0 - 3.6V$ ,  $GND = AGND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值*	最大值	单位	条件
供电电压	$V_{SAD}$	3.0	3.3	3.6	V	
精度	$N_{SR}$	-	10	-	bit	$GND \leq V_{SAIN} \leq AV_{DD}$
A/D输入电压	$V_{SAIN}$	GND	-	$V_{SAD}$	V	
A/D输入电阻	$R_{AIN}$	2	-	-	$M\Omega$	$V_{SAIN} = 3.0V$
A/D转换电流	$I_{SAD}$	-	1	3	mA	ADC模块工作, $AV_{DD} = 3.3V$
A/D输入电流	$I_{SADIN}$	-	-	10	$\mu A$	$AV_{DD} = 3.3V$
模拟电压源推荐阻抗	$Z_{AIN}$	-	-	10	$k\Omega$	
微分非线性误差	$D_{LE}$	-	-	$\pm 1$	LSB	$f_{OSC} = 16MHz$ , $AV_{DD} = 3.3V$
积分非线性误差	$I_{LE}$	-	-	$\pm 2$	LSB	$f_{OSC} = 16MHz$ , $AV_{DD} = 3.3V$
满刻度误差	$E_F$	-	$\pm 1$	$\pm 3$	LSB	$f_{OSC} = 16MHz$ , $AV_{DD} = 3.3V$
偏移误差	$E_Z$	-	$\pm 0.5$	$\pm 3$	LSB	$f_{OSC} = 16MHz$ , $AV_{DD} = 3.3V$
总绝对误差	$E_{AD}$	-	-	$\pm 3$	LSB	$f_{OSC} = 16MHz$ , $AV_{DD} = 3.3V$
总转换时间	$T_{SCON}$	14	-	-	$T_{AD}$	10位精度, $f_{MSYS} = 16MHz$ , $AV_{DD} = 3.3V$

注意: “\*”表示“典型值”下的数据是在3.3V, 25°C下测得的, 除非另有说明。

流水线型模/数转换器电气特性 ( $V_{DD} = 3.0 - 5.5V$ ,  $AV_{DD} = 3.0 - 3.6V$ ,  $GND = AGND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
供电电压	$V_{PAD}$	3.0	3.3	3.6	V	
精度	$N_{PR}$	-	14	-	bit	$GND \leq V_{PAIN} \leq 3V$
工作时钟	$CLK_{ADC}$	0.1	-	10	MHz	
A/D输入电压	$V_{PAIN}$	0	-	3	V	
A/D工作电流	$I_{PAD}$	-	30	40	mA	ADC模块工作, $AV_{DD} = 3.3V$
A/D输入泄漏电流	$I_{PADIN}$	-	-	$\pm 1$	$\mu A$	$AV_{DD} = 3.3V$
积分非线性误差	$I_{NL}$	-	-	$\pm 1.5$	LSB	
微分非线性误差	$D_{NL}$	-	-	$\pm 1$	LSB	
偏移误差	$E_{OFF}$	-	-	0.3	%FSR	
增益误差	$G_{ERR}$	-	-	1.5	%FSR	
转换速率	$T_{PCON}$	-	-	10	MSPS	$f_{DSYS} = 100MHz$ , $AV_{DD} = 3.3V$
单调性						设计保证
信噪比	$S_{NR}$	62	67.5	-	dB	
总谐波失真	$T_{HD}$	-	-72	-68	dB	100kHz
动态范围	$S_{FDR}$	69	75	-	dB	
有效位数	$E_{NOB}$	10.1	10.5	-	Bits	
基准源	$V_{REF}$	1.485	1.5	1.515	V	
基准源温度系数	-	-	50	-	PPM/°C	
基准源负载调整	$VR_{LOAD}$	-	15	20	mV	1.0mA Load Current

交流电气特性 ( $V_{DD} = 2.7V - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ ,  $f_{OSC} = 16MHz$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
PLL输出频率范围	$F_{PLL}$	20	-	100	MHz	$8MHz \leq f_{OSC} \leq 16MHz$
PLL建立时间	$T_{PLL}$	-	2	-	ms	不包括振荡器起振时间
复位脉冲宽度	$t_{RESET}$	10	-	-	$\mu s$	低电平有效
复位引脚上拉电阻	$R_{RPH}$	-	30	-	$k\Omega$	$V_{DD} = 5.0V$ , $V_{IN} = GND$

低电压复位电气特性 ( $V_{DD} = 2.7V - 5.5V$ ,  $DGND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
LVR1	$V_{LVRL1}$	2.7	2.8	2.9	V	LVR1使能
LVR2	$V_{LVRL2}$	4.0	4.1	4.2	V	LVR2使能
LVR低电压复位宽度	$T_{LVR}$	-	30	-	$\mu s$	

晶体谐振器电气特性 ( $V_{DD} = 2.7V - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
频率	$f_{OSC}$	4	-	16	MHz	
占空比	$D_{OSC}$	45	50	55	%	
负载电容	$C_L$	-	12	-	pF	

内部**16M RC**振荡器 ( $V_{DD} = 2.7V - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
频率	$F_{RC}$	15.68	16	16.32	MHz	$V_{DD} = 3.0 - 5.5V$

内部**3.3V**稳压源 ( $V_{DD} = 4.2 - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ , REG33使能,  $AV_{DD}$ 引脚接47 $\mu F$ 电容, 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
供电电压	$V_{IN}$	4.2	5.0	5.5	V	$V_{IN} = V_{DD}$
输出电压	$V_{OUT}$	3.1	3.3	3.5	V	$AV_{DD}$ 不加负载
掉电压	$T_{DROP}$	-	25	100	mV	10mA Load Current



运算放大器电气特性 ( $V_{DD} = 3.0 - 5.5V$ ,  $AV_{DD} = 3.0 - 3.6V$ ,  $GND = AGND = 0V$ ,  $T_A = 25^\circ C$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
输入信号带宽	$S_{BD}$	0	-	10	KHz	
输入失调电压	$V_{IO}$	-	3	8	mV	
输入共模电压范围	$V_{ICM}$	0	-	$AV_{DD} - 1.5$	V	
输出电压	$V_{OPO}$	0.1	-	$V_{OP} - 0.3$	V	
转换速率	$S_R$	0.4	1	-	V/ $\mu s$	
开环差模增益	$G_{OL}$	80	-	-	dB	设计保证
共模抑制比	$C_{MRR}$	-	80	-	dB	设计保证
反馈电阻范围	$R_{FD}$	5	-	20	k $\Omega$	
输入端外接电阻范围	$R_{EIN}$	1.5	-	15	k $\Omega$	
闭环增益	$G_{CL}$	2	-	10		
输出驱动电流	$V_{OPO}$	-	1.5	-	mA	$V_{IN} = 1V, V_{OUT} = 1.5V$
		-	-600	-	$\mu A$	$V_{IN} = -1V, V_{OUT} = 1.5V$

PWM模块时序特性

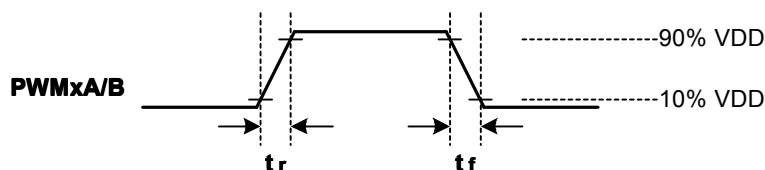


图11-1 PWM模块时序特性

符号	特性	最小值	典型值	最大值	单位	条件
$t_r$	PWMxA/B输出上升沿时间	-	-	10	ns	$V_{DD} = 5.0V$ , IO口负载电容为50pF
$t_f$	PWMxA/B输出下降沿时间	-	-	10	ns	同上

x = 0 - 2



流水线型模/数转换器时序特性  
顺序采样模式（单信道）

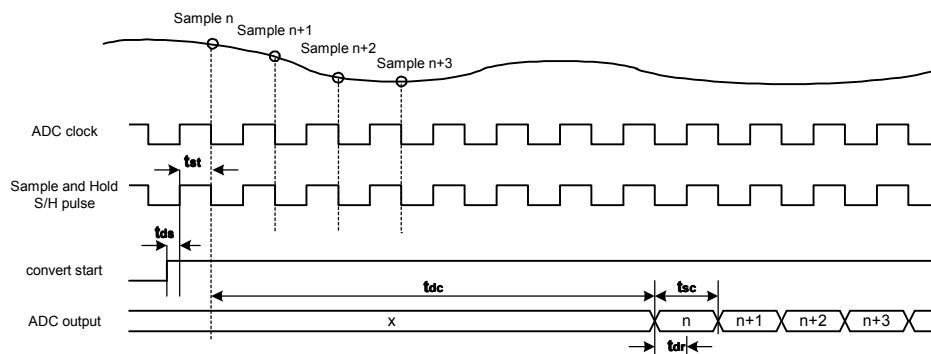


图11-2 流水线型模/数转换器顺序采样时序图

	采样 n	采样 n+1
转换启动延时 $t_{ds}$	$< t_{adcclock}$	
信号建立延时 $t_{st}$	$1/2 t_{adcclock}$	
采样转换延时 $t_{dc}$	$7 t_{adcclock}$	
转换完成到寄存器值有效的延时 $t_{dr}$	$1/2 t_{adcclock}$	
连续转换完成延时 $t_{sc}$		$1 t_{adcclock}$

并行采样模式（双信道）

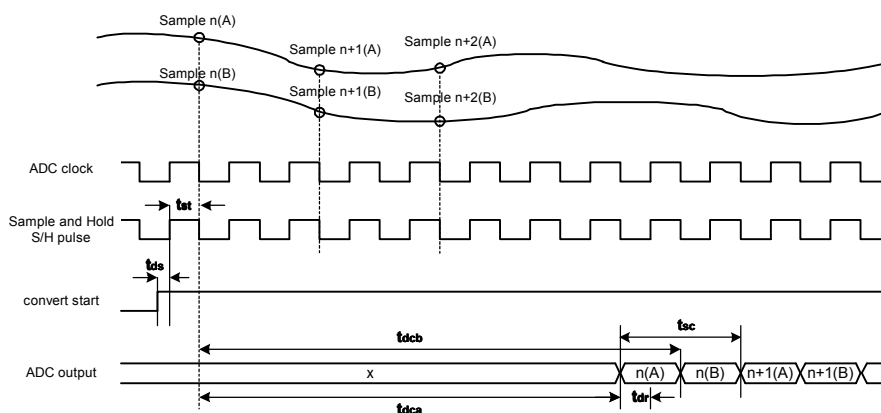


图11-3 流水线型模/数转换器并行采样时序图

	采样n	采样n+1
转换启动延时 $t_{ds}$	$< t_{adcclock}$	
信号建立延时 $t_{st}$	$1/2 t_{adcclock}$	
A通道采样转换延时 $t_{dca}$	$7 t_{adcclock}$	
B通道采样转换延时 $t_{dcb}$	$8 t_{adcclock}$	
转换完成到寄存器值有效的延时 $t_{dr}$	$1/2 t_{adcclock}$	
连续转换完成延时 $t_{sc}$		$2 t_{adcclock}$



**12. 订购信息**

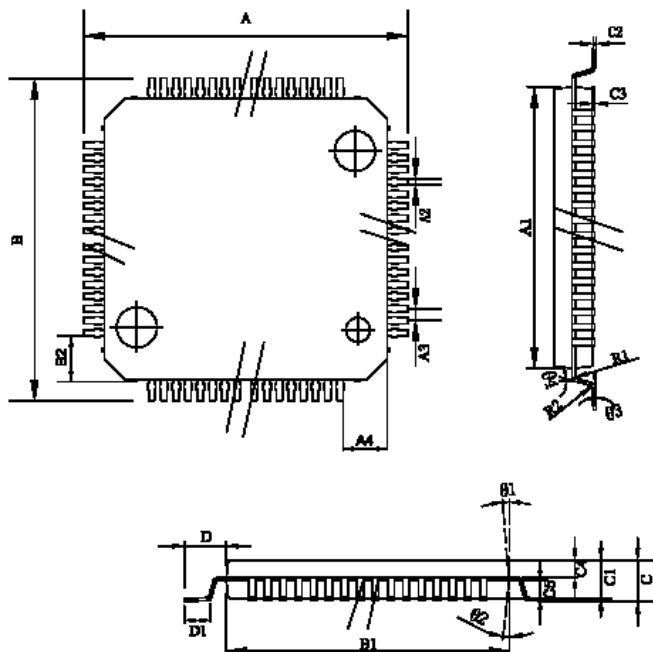
产品编号	封装
SH99F100P/100PR	LQFP100



13. 封装信息

LQFP 100 Outline Dimensions

单位: 英寸/毫米



Symbol	Dimensions in inches		Dimensions in mm	
	Min.	Max.	Min.	Max.
A	0.622	0.638	15.80	16.20
A1	0.547	0.555	13.90	14.10
A2	0.007	0.011	0.17	0.27
A3	0.020 Typ.		0.5 Typ.	
A4	0.035 Typ.		0.9 Typ.	
B	0.622	0.638	15.80	16.20
B1	0.547	0.555	13.90	14.10
B2	0.035 Typ.		0.9 Typ.	
C	0.055	0.063	1.40	1.60
C1	0.053	0.057	1.35	1.45
C2	0.004	0.007	0.09	0.18
C3	0.002	0.006	0.05	0.15
C4	0.025 Typ.		0.6365 Typ.	
C5	0.025 Typ.		0.6365 Typ.	
D	0.035	0.043	0.90	1.10
D1	0.018	0.028	0.45	0.70
R1	0.006 Typ.		0.15 Typ.	
R2	0.006 Typ.		0.15 Typ.	
θ1	12° Typ.		12° Typ.	
θ2	12° Typ.		12° Typ.	
θ3	4° Typ.		4° Typ.	
θ4	4° Typ.		4° Typ.	

**14. 规格更改记录**

版本	记录	日期
2.1	调整待机电流参数值	2012年12月
2.0	1. 增加了中断号(c语言) 2. 更改DSP系统时钟控制寄存器DCLKCON内容	2012年9月
1.0	初始版本	2012年1月



**目录**

1. 特性.....	1
2. 概述.....	2
3. 方框图.....	2
4. 引脚配置.....	3
5. 引脚描述.....	5
6. 寄存器映像.....	7
6.1 MCU寄存器映像.....	7
6.2 DSP寄存器映像.....	9
6.2.1 核寄存器 (non-memory mapped) .....	9
6.2.2 存储器映射寄存器 (memory-mapped) .....	9
7. 系统集成模块SIM (SYSTEM INTEGRATED MODULE) .....	13
7.1 存储器架构.....	13
7.2 复位.....	15
7.2.1 上电复位POR (Power On Reset) .....	16
7.2.2 按键复位PIN (Pin Reset) .....	16
7.2.3 看门狗复位WDT (WatchDogTimer Reset) .....	16
7.2.4 低电压复位LVR (Low Voltage Reset) .....	16
7.2.5 软复位SWR (Software Reset) .....	16
7.2.6 上电Autoload过程.....	18
7.2.7 复位过程综述.....	18
7.3 系统时钟.....	19
7.3.1 DSP系统时钟.....	19
7.3.2 MCU系统时钟.....	21
7.3.3 系统时钟监控模块SCM (System Clock Monitor) .....	21
7.4 看门狗.....	22
7.5 预热计数器.....	23
7.5.1 电源上电预热计数时间.....	23
7.5.2 振荡器上电预热计数时间.....	23
7.6 DSP核与MCU核之间的通讯.....	24
7.6.1 通讯存储器结构.....	24
7.6.2 通讯控制/数据寄存器.....	25
7.6.3 DSP使用的通讯寄存器.....	25
7.6.4 MCU使用的通讯寄存器.....	28
7.6.5 通信中断.....	31
7.7 在系统可编程功能ISP (IN SYSTEM PROGRAM) .....	32
7.7.1 上电复位触发ISP.....	32
7.7.2 按键复位触发ISP.....	33
7.8 电源管理.....	34
7.8.1 空闲模式 (Idle) .....	34
7.8.2 掉电模式 (Power-Down) .....	34
7.8.3 寄存器.....	35
8. MCU部分.....	36
8.1 MCU核特性.....	36
8.1.1 CPU内核特殊功能寄存器.....	36
8.1.2 CPU增强内核特殊功能寄存器.....	37
8.1.3 寄存器.....	37
8.2 低速RAM区 (MIDM/MEDM区) .....	38
8.2.1 特性.....	38
8.2.2 寄存器.....	38
8.3 FLASH程序存储器 (MCM区与DCSM区) .....	39
8.3.1 特性.....	39
8.3.2 ICP模式下的Flash操作.....	39
8.4 FLASH的扇区自编程 (SSP) 功能.....	41
8.4.1 SSP寄存器.....	41



8.4.2 SSP编程注意事项.....	43
8.5 I/O端口.....	44
8.5.1 特性.....	44
8.5.2 寄存器.....	44
8.5.3 端口模块图.....	45
8.5.4 端口共用.....	46
8.6 定时器TIMER0/1/2.....	50
8.6.1 特性.....	50
8.6.2 定时器0和定时器1.....	50
8.6.3 定时器2.....	55
8.7 MCU中断.....	61
8.7.1 特性.....	61
8.7.2 程序超范围中断 (OVL).....	61
8.7.3 中断允许.....	62
8.7.4 中断标志.....	64
8.7.5 中断向量.....	66
8.7.6 中断优先级.....	66
8.7.7 中断处理.....	67
8.7.8 中断响应时间.....	67
8.7.9 外部中断输入.....	68
8.7.10 中断汇总.....	68
8.8 辅助脉宽调制模块APWM (AUXILIARY PWM).....	69
8.8.1 特性.....	69
8.8.2 寄存器.....	69
8.9 增强型通用异步收发器 (EUART).....	74
8.9.1 特性.....	74
8.9.2 EUART工作模式.....	74
8.9.3 波特率发生.....	79
8.9.4 多机通讯.....	80
8.9.5 帧出错检测.....	81
8.9.6 关于EUART1.....	81
8.10 串行外部设备接口 (SPI).....	86
8.10.1 特性.....	86
8.10.2 信号描述.....	86
8.10.3 波特率.....	87
8.10.4 功能描述.....	87
8.10.5 工作模式.....	88
8.10.6 传送形式.....	89
8.10.7 出错检测.....	90
8.10.8 SPI中断.....	90
8.10.9 SPI寄存器.....	91
8.11 逐次逼近型ADC.....	93
8.11.1 特性.....	93
8.11.2 ADC模块图.....	93
8.11.3 ADC寄存器.....	94
9. DSP部分.....	97
9.1 DSP核特性.....	97
9.1.1 DSP核寄存器.....	98
9.1.2 PSQ核寄存器的位定义.....	100
9.1.3 DSP核的控制寄存器.....	104
9.2 DSP中断.....	106
9.2.1 特性.....	106
9.2.2 控制寄存器.....	106
9.2.3 DSP中断使能的全局控制.....	111
9.2.4 DSP核级中断控制.....	111



9.2.5 DSP中断的响应.....	111
9.2.6 DSP中断优先级.....	112
9.2.7 DSP中断外部中断DXINT0-2.....	112
9.2.8 DSP中断汇总.....	113
9.3 PWM模块.....	114
9.3.1 特性.....	114
9.3.2 逻辑框图.....	115
9.3.3 控制寄存器.....	116
9.3.4 PWM时基.....	128
9.3.5 PWM波形生成器.....	131
9.3.6 死区 & 极性控制.....	135
9.3.7 保护输入& PWM输出控制.....	137
9.3.8 事件触发控制.....	138
9.3.9 PWM中断.....	140
9.3.10 后分频系数.....	140
9.4 正交编码器接口模块QEI.....	142
9.4.1 特性.....	142
9.4.2 逻辑框图.....	142
9.4.3 控制寄存器.....	143
9.4.4 输入滤波.....	148
9.4.5 正交解码器.....	148
9.4.6 位置计数器.....	149
9.4.7 QTimer功能.....	151
9.4.8 测速方案.....	151
9.4.9 QTimer作为独立的定时器.....	152
9.4.10 QEI模块的中断.....	152
9.5 流水线型ADC模块.....	153
9.5.1 特性.....	153
9.5.2 逻辑框图.....	153
9.5.3 控制寄存器.....	153
9.5.4 AD转换器工作时钟.....	160
9.5.5 通道序列.....	160
9.5.6 AD转换过程.....	160
9.5.7 AD转换过程的触发方式.....	161
9.5.8 指针和计数寄存器.....	161
9.5.9 ADC模块的转换方式.....	161
9.5.10 单序列顺序采样模式.....	161
9.5.11 两路同步并行采样模式.....	163
9.5.12 两路独立顺序采样模式.....	165
9.5.13 ADC模块的电源控制.....	167
9.6 DSP定时器.....	168
9.6.1 特性.....	168
9.6.2 逻辑框图.....	168
9.6.3 控制寄存器.....	169
9.6.4 计数时钟与预分频器.....	173
9.6.5 周期寄存器与定时时间.....	173
9.6.6 工作模式.....	173
9.6.7 定时器的启停控制.....	173
9.6.8 ADC触发.....	173
9.6.9 定时器中断.....	173
9.6.10 定时器工作波形示例.....	174
9.7 输入捕捉模块.....	175
9.7.1 特性.....	175
9.7.2 逻辑框图.....	175
9.7.3 控制寄存器.....	175



9.7.4 捕捉过程与存储栈操作.....	181
9.7.5 输入捕捉引脚滤波设置.....	182
9.7.6 输入捕捉预分频设置.....	182
9.7.7 输入捕捉中断.....	182
9.8 输出比较模块.....	183
9.8.1 特性.....	183
9.8.2 逻辑框图.....	183
9.8.3 控制寄存器.....	183
9.8.4 单比较工作模式.....	186
9.8.5 双比较工作模式.....	187
9.8.6 简单PWM输出模式.....	190
9.9 运算放大器模块.....	191
9.9.1 特性.....	191
9.9.2 功能说明.....	191
9.9.3 控制寄存器.....	191
10. 代码选项 (CODE OPTION) .....	192
11. 电气特性.....	193
极限参数.....	193
直流电气特性.....	193
逐次逼近型3.3V模/数转换器电气特性.....	194
流水线型模/数转换器电气特性.....	194
交流电气特性.....	195
低电压复位电气特性.....	195
晶体谐振器电气特性.....	195
内部16M RC振荡器.....	195
内部3.3V稳压源.....	195
运算放大器电气特性.....	196
PWM模块时序特性.....	196
流水线型模/数转换器时序特性.....	197
12. 订购信息.....	198
13. 封装信息.....	199
14. 规格更改记录.....	200