芯群

产品规格书

经济型 A/D OTP 单片机

BX66R006

版本：V1.00　　日期：2024-08-12

目录

1. 特性

1.1 CPU 特性

- 工作电压：
  - ♦ f_{SYS}=8MHz：1.8V~5.5V
  - ♦ f_{SYS}=12MHz：2.7V~5.5V
  - ♦ f_{SYS}=16MHz：3.3V~5.5V
- V_{DD}=5V，系统时钟为 16MHz 时，指令周期为 0.25μs
- 暂停和唤醒功能，以降低功耗
- 振荡器类型：
  - ♦ 内部高速 8/12/16MHz RC – HIRC
  - ♦ 内部低速 32kHz RC – LIRC
- 多种工作模式：快速模式、低速模式、空闲模式和休眠模式
- 内部集成的振荡器，无需外接元件
- 所有指令都可在 1~3 个指令周期内完成
- 查表指令
- 115 条功能强大的指令系统
- 8 层硬件堆栈
- 位操作指令

1.2 周边特性

- OTP 程序存储器：4K×16
- 数据存储器：256×8
- OTP ROM 参数烧录功能 – ORPP
- 看门狗定时器功能
- 18 个双向 I/O 口
- 两个与 I/O 口复用的外部中断输入
- 两个定时器模块用于时间测量、捕捉输入、比较匹配输出、PWM 输出及单脉冲输出
- 脉冲宽度调制功能
- 8 个外部通道 12-bit 分辨率的 A/D 转换器，具有内部参考电压 V_{BG}
- 2 个时基功能，用于产生固定时间的中断信号
- 低电压复位功能
- 低电压检测功能
- 软件控制的 4-SCOM 口 1/2 偏压的 LCD 驱动器
- 封装类型：16-pin NSOP，20-pin SOP/NSOP/SSOP，20-pin QFN

2. 概述

该单片机是一款具有 8 位高性能精简指令集的 OTP 单片机，专为经济型产品设计。

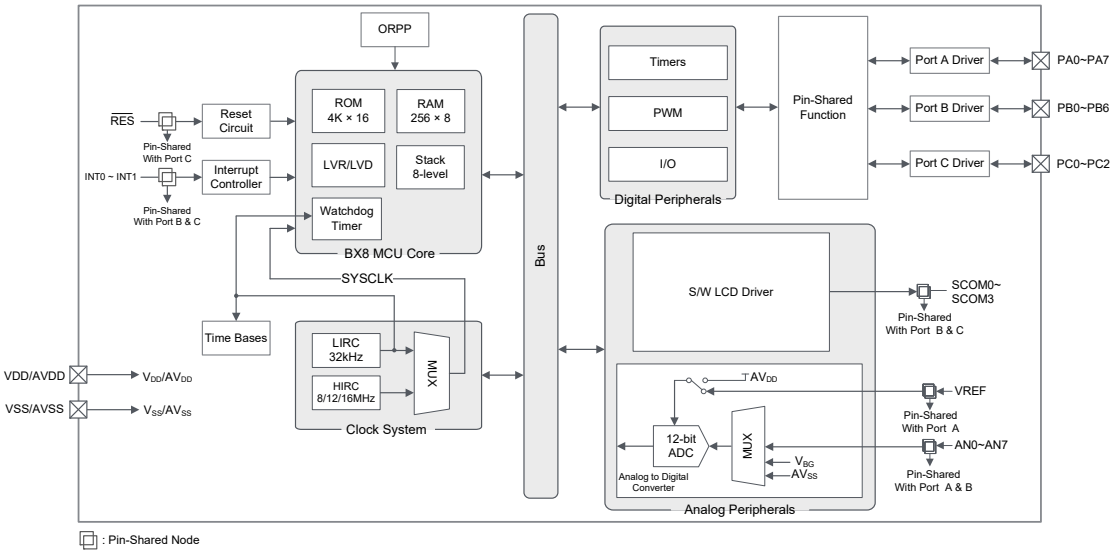
在存储器特性方面，此单片机提供一次可编程的 OTP 存储器，此外还包含了一个 RAM 数据存储器。

在模拟特性方面，该单片机包含一个多通道 12-bit A/D 转换器。在内部定时器方面，该单片机带有多个使用灵活的定时器模块，可提供定时功能、脉冲产生功能及 PWM 产生等功能。内部看门狗定时器、低电压复位和低电压检测等保护特性，外加优秀的抗干扰和 ESD 保护性能，确保单片机在恶劣的电磁干扰环境下可靠地运行。

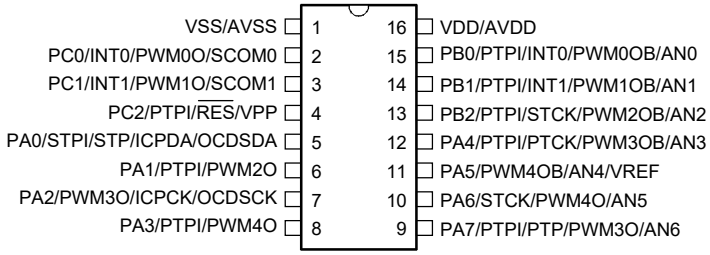
该单片机提供了内部高速和低速振荡器功能选项，这两个内建的系统振荡器无需外围元器件。其在不同工作模式之间动态切换的能力，为用户提供了一个优化单片机操作和减少功耗的手段。

外加 I/O 使用灵活、时基功能等其它特性增强了该单片机的功能和灵活性。该单片机还包含一个 16-bit PWM 控制电路，可产生 PWM 信号和线性调整输出占空比。该单片机特别适合应用于 LED 灯控及各式家电产品，例如：LED 控制器、咖啡机、电热水壶、电茶炉、电饭煲、豆浆机等。

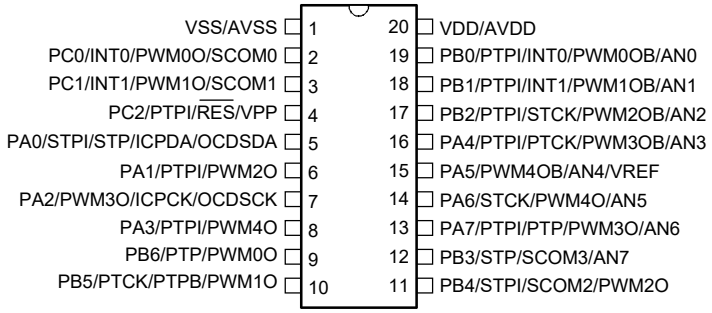
3. 方框图



4. 引脚图



BX66R006/BX66EV006
16 NSOP-A



BX66R006/BX66EV006
20 SOP-A/NSOP-A/SSOP-A

注：1. 若共用脚同时有多种输出，所需引脚共用功能通过引脚共用寄存器中相应的软件控制位决定。
2. OCDSCK 和 OCDSDA 引脚为片上调试功能 (OCDS) 专用引脚，仅存在于 BX66R006 (OTP 型) 的 EV 芯片 BX66EV006 (Flash 型) 中。
3. 在较小封装类型中可能含有未引出的引脚，需合理设置其状态以避免输入浮空造成额外耗电，详见“待机电流注意事项”和“输入 / 输出端口”章节。
4. VPP 引脚为 OTP 烧录高压输入，仅存在于 BX66R006 单片机。

5. 引脚说明

每个引脚的功能如下表所述，而引脚配置的详细内容见规格书其它章节。注意，对于存在不止一种封装的单片机，该表格反映的是较大封装类型的情况。

引脚名称	功能	OPT	I/T	O/T	说明
PA0/STPI/STP/ICPDA/OCDSDA	PA0	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STPI	PAS0 IFS	ST	—	STM 捕捉输入
	STP	PAS0	—	CMOS	STM 输出
	ICPDA	—	ST	CMOS	ICP 数据 / 地址引脚
	OCDSDA	—	ST	CMOS	OCDS 数据 / 地址，仅用于 EV 芯片
PA1/PTPI/PWM2O	PA1	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTPI	PAS0 IFS	ST	—	PTM 捕捉输入
	PWM2O	PAS0	—	CMOS	PWM2 信号输出
PA2/PWM3O/ICPCK/OCDSCK	PA2	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PWM3O	PAS0	—	CMOS	PWM3 信号输出
	ICPCK	—	ST	—	ICP 时钟引脚
	OCDSCK	—	ST	—	OCDS 时钟引脚，仅用于 EV 芯片
PA3/PTPI/PWM4O	PA3	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTPI	PAS0 IFS	ST	—	PTM 捕捉输入
	PWM4O	PAS0	—	CMOS	PWM4 信号输出
PA4/PTPI/PTCK/PWM3OB/AN3	PA4	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTPI	PAS1 IFS	ST	—	PTM 捕捉输入
	PTCK	PAS1 IFS	ST	—	PTM 时钟输入
	PWM3OB	PAS1	—	CMOS	PWM3 信号反向输出
	AN3	PAS1	AN	—	A/D 转换器外部输入通道

引脚名称	功能	OPT	I/T	O/T	说明
PA5/PWM4OB/AN4/VREF	PA5	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PWM4OB	PAS1	—	CMOS	PWM4 信号反向输出
	AN4	PAS1	AN	—	A/D 转换器外部输入通道
	VREF	PAS1	AN	—	A/D 转换器外部参考电压
PA6/STCK/PWM4O/AN5	PA6	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STCK	PAS1 IFS	ST	—	STM 时钟输入
	PWM4O	PAS1	—	CMOS	PWM4 信号输出
	AN5	PAS1	AN	—	A/D 转换器外部输入通道
PA7/PTPI/PTP/PWM3O/AN6	PA7	PAWU PAS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTPI	PAS1 IFS	ST	—	PTM 捕捉输入
	PTP	PAS1	—	CMOS	PTM 输出
	PWM3O	PAS1	—	CMOS	PWM3 信号输出
	AN6	PAS1	AN	—	A/D 转换器外部输入通道
PB0/PTPI/INT0/PWM0OB/AN0	PB0	PBPU PBS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTPI	PBS0 IFS	ST	—	PTM 捕捉输入
	INT0	INTEG INTC1 PBS0 IFS	ST	—	外部中断输入 0
	PWM0OB	PBS0	—	CMOS	PWM0 信号反向输出
	AN0	PBS0	AN	—	A/D 转换器外部输入通道
PB1/PTPI/INT1/PWM1OB/AN1	PB1	PBPU PBS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTPI	PBS0 IFS	ST	—	PTM 捕捉输入
	INT1	INTEG INTC1 PBS0 IFS	ST	—	外部中断输入 1
	PWM1OB	PBS0	—	CMOS	PWM1 信号反向输出
	AN1	PBS0	AN	—	A/D 转换器外部输入通道

引脚名称	功能	OPT	I/T	O/T	说明
PB2/PTPI/STCK/PWM2OB/AN2	PB2	PBPU PBS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTPI	PBS0 IFS	ST	—	PTM 捕捉输入
	STCK	PBS0 IFS	ST	—	STM 时钟输入
	PWM2OB	PBS0	—	CMOS	PWM2 信号反向输出
	AN2	PBS0	AN	—	A/D 转换器外部输入通道
PB3/STP/SCOM3/AN7	PB3	PBPU PBS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	STP	PBS0	—	CMOS	STM 输出
	SCOM3	PBS0	—	AN	软件 LCD COM 输出
	AN7	PBS0	AN	—	A/D 转换器外部输入通道
PB4/STPI/SCOM2/PWM2O	PB4	PBPU PBS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	STPI	PBS1 IFS	ST	—	STM 捕捉输入
	SCOM2	PBS1	—	AN	软件 LCD COM 输出
	PWM2O	PBS1	—	CMOS	PWM2 信号输出
PB5/PTCK/PTPB/PWM1O	PB5	PBPU PBS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTCK	PBS1 IFS	ST	—	PTM 时钟输入
	PTPB	PBS1	—	CMOS	PTM 反向输出
	PWM1O	PBS1	—	CMOS	PWM1 信号输出
PB6/PTP/PWM0O	PB6	PBPU PBS1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP	PBS1	—	CMOS	PTM 输出
	PWM0O	PBS1	—	CMOS	PWM0 信号输出
PC0/INT0/PWM0O/SCOM0	PC0	PCPU PCS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	INT0	INTEG INTC0 PCS0 IFS	ST	—	外部中断输入 0
	PWM0O	PCS0	—	CMOS	PWM0 信号输出
	SCOM0	PCS0	—	AN	软件 LCD COM 输出

引脚名称	功能	OPT	I/T	O/T	说明
PC1/INT1/PWM1O/SCOM1	PC1	PCPU PCS0	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	INT1	INTEG INTC0 PCS0 IFS	ST	—	外部中断输入 1
	PWM1O	PCS0	—	CMOS	PWM1 信号输出
	SCOM1	PCS0	—	AN	软件 LCD COM 输出
PC2/PTPI/$\overline{\text{RES}}$/VPP	PC2	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTPI	IFS	ST	—	PTM 捕捉输入
	$\overline{\text{RES}}$	CO RSTC	ST	—	外部复位输入
	VPP	—	PWR	—	OTP 烧录高压输入，不存在于 EV 芯片
VDD/AVDD	VDD	—	PWR	—	数字正电源
	AVDD	—	PWR	—	A/D 转换器正电源
VSS/AVSS	VSS	—	PWR	—	数字负电源，接地
	AVSS	—	PWR	—	A/D 转换器负电源，接地

注：I/T：输入类型；　　O/T：输出类型；
OPT：通过配置选项 (CO) 或寄存器选项来配置；　CO：配置选项；
PWR：电源；　　ST：施密特触发输入；
CMOS：CMOS 输出；　　AN：模拟信号。

6. 极限参数

电源供应电压 ..... V_{SS}-0.3V~6.0V
端口输入电压 ..... V_{SS}-0.3V~V_{DD}+0.3V
储存温度 ..... -60°C~150°C
工作温度 ..... -40°C~85°C
I_{OH} 总电流 ..... -80mA
I_{OL} 总电流 80mA
总功耗 500mW

注：这里只强调额定功率，超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述标示范围外的工作状态，而且若长期在标示范围外的条件下工作，可能影响芯片的可靠性。

7. 直流电气特性

以下表格中参数测量结果可能受多个因素影响，如振荡器类型、工作电压、工作频率、引脚负载状况、温度和程序指令等等。

7.1 工作电压特性

Ta=-40°C~85°C

符号	参数	测试条件	最小	典型	最大	单位
V_{DD}	工作电压 – HIRC	f_{SYS}=8MHz	1.8	—	5.5	V
		f_{SYS}=12MHz	2.7	—	5.5	V
		f_{SYS}=16MHz	3.3	—	5.5	V
	工作电压 – LIRC	f_{SYS}=32kHz	1.8	—	5.5	V

7.2 待机电流特性

Ta=25°C，除非另有说明。

符号	待机模式	测试条件		最小	典型	最大	最大 @85°C	单位
		V_{DD}	条件					
I_{STB}	休眠模式	1.8V	WDT off	—	0.45	0.80	7.00	μA
		3V		—	0.45	0.90	8.00	
		5V		—	0.5	2.0	10.0	
		1.8V	WDT on	—	1.2	2.4	2.9	μA
		3V		—	1.5	3.0	3.6	
		5V		—	3	5	6	
	空闲模式 0 – LIRC	1.8V	f_{SUB} on	—	2.4	4.0	4.8	μA
		3V		—	3	5	6	
		5V		—	5	10	12	
	空闲模式 1 – HIRC	1.8V	f_{SUB} on，f_{SYS}=8MHz	—	288	400	480	μA
		3V		—	360	500	600	
		5V		—	600	800	960	
		2.7V	f_{SUB} on，f_{SYS}=12MHz	—	432	600	720	μA
		3V		—	540	750	900	
		5V		—	800	1200	1440	
		3.3V	f_{SUB} on，f_{SYS}=16MHz	—	0.80	1.20	1.44	mA
		5V		—	1.4	2.0	2.4	

注：当使用该表格电气特性数据时，以下几点需注意：

1. 任何数字输入都设置为非浮空的状态。
2. 所有测量都在无负载且所有外围功能关闭的条件下进行。
3. 无直流电流路径。
4. 所有待机电流数值都是在 HALT 指令执行后即停止执行所有指令后测得。

7.3 工作电流特性

Ta=-40°C~85°C

符号	工作模式	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
I_{DD}	低速模式 – LIRC	1.8V	f_{SYS}=32kHz	—	8	16	μA
		3V		—	10	20	
		5V		—	30	50	
	快速模式 – HIRC	1.8V	f_{SYS}=8MHz	—	0.6	1.0	mA
		3V		—	0.8	1.2	
		5V		—	1.6	2.4	
		2.7V	f_{SYS}=12MHz	—	1.0	1.4	mA
		3V		—	1.2	1.8	
		5V		—	2.4	3.6	
		3.3V	f_{SYS}=16MHz	—	1.5	3.0	mA
		5V		—	2.5	5.0	

注：当使用该表格电气特性数据时，以下几点需注意：
1. 任何数字输入都设置为非浮空的状态。
2. 所有测量都在无负载且所有外围功能关闭的条件下进行。
3. 无直流电流路径。
4. 所有工作电流数值通过一个连续的 NOP 指令循环程序测得。

8. 交流电气特性

以下表格中参数测量结果可能受多个因素影响，如振荡器类型、工作电压、工作频率和温度等等。

8.1 内部高速振荡器 – HIRC – 频率精准度

程序烧录时，烧录器会依据用户选择的 HIRC 频率和工作电压 (3V 或 5V) 对 HIRC 进行频率精准度调整。

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	温度				
f_{HIRC}	通过烧录器调整后的 8MHz HIRC 频率	3V/5V	25°C	-1%	8	+1%	MHz
			-20°C~60°C	-3%	8	+3%	
			-40°C~85°C	-4.5%	8	+4.5%	
		1.8V~5.5V	25°C	-10%	8	+10%	
			-20°C~60°C	-13.5%	8	+13.5%	
			-40°C~85°C	-15%	8	+15%	
		2.0V~5.5V	25°C	-7%	8	+7%	
			-20°C~60°C	-8.5%	8	+8.5%	
			-40°C~85°C	-10%	8	+10%	
		2.2V~5.5V	25°C	-3.5%	8	+3. 5%	
			-20°C~60°C	-4.5%	8	+4.5%	
			-40°C~85°C	-5%	8	+5%	

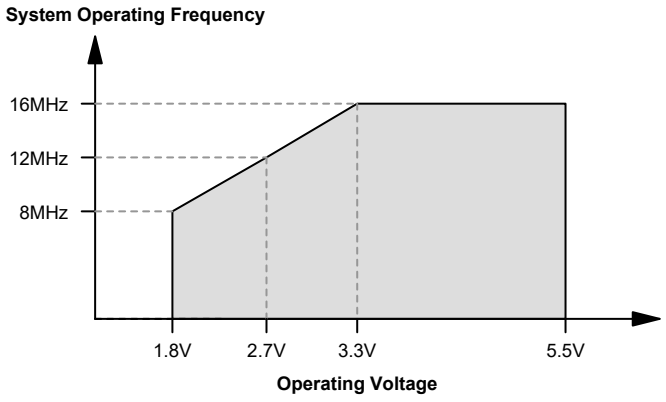
符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	温度				
f_{HIRC}	通过烧录器调整后的 12MHz HIRC 频率	3V/5V	25°C	-1%	12	+1%	MHz
			-20°C~60°C	-1.5%	12	+1.1%	
			-40°C~85°C	-2%	12	+2%	
		2.7V~5.5V	25°C	-2.5%	12	+2.5%	
			-20°C~60°C	-2.8%	12	+2.8%	
			-40°C~85°C	-3%	12	+3%	
	通过烧录器调整后的 16MHz HIRC 频率	5V	25°C	-1%	16	+1%	MHz
			-20°C~60°C	-1.5%	16	+1.1%	
			-40°C~85°C	-2%	16	+2%	
		3.3V~5.5V	25°C	-2.5%	16	+2.5%	
			-20°C~60°C	-2.8%	16	+2.8%	
			-40°C~85°C	-3%	16	+3%	

注：1. 烧录器可在 3V/5V 这两个可选的固定电压下对 HIRC 频率进行调整，在此提供 V_{DD}=3V/5V 时的参数值。
2. 3V/5V 表格列下面提供的是全压条件下的参数值。当应用电压范围是 1.8V~3.6V 时，建议烧录器电压固定在 3V；当应用电压范围是 3.3V~5.5V 时，建议烧录器电压固定在 5V。
3. 表格中提供的最小和最大误差值仅在对应的烧录器调整频率下有效。当烧录器已对所选的频率进行调整，此后再通过程序中振荡器控制位将其频率改为其它值时，频率误差范围将增加到 ±20%。

8.2 内部低速振荡器电气特性 – LIRC – 频率精准度

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	温度				
f_{LIRC}	LIRC 频率	1.8V~5.5V	25°C	-20%	32	+20%	kHz
			-40°C~85°C	-50%	32	+60%	
t_{START}	LIRC 启动时间	—	-40°C~85°C	—	—	500	μs

8.3 工作频率特性曲线



8.4 系统上电时间电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
t_{SST}	系统启动时间 (从 f_{SYS} off 的状态下唤醒)	—	$f_{SYS}=f_H\sim f_H/64$，$f_H=f_{HIRC}$	—	16	—	t_{HIRC}
		—	$f_{SYS}=f_{SUB}=f_{LIRC}$	—	2	—	t_{LIRC}
	系统启动时间 (从 f_{SYS} on 的状态下唤醒)	—	$f_{SYS}=f_H\sim f_H/64$，$f_H=f_{HIRC}$	—	2	—	t_H
		—	$f_{SYS}=f_{SUB}=f_{LIRC}$	—	2	—	t_{SUB}
	系统速度切换时间 (快速模式 → 低速模式或 低速模式 → 快速模式)	—	f_{HIRC} off → on	—	16	—	t_{HIRC}
t_{RSTD}	系统复位延迟时间 (上电复位或 LVR 硬件复位)	—	RR_{POR}=5V/ms	10	16	24	ms
	系统复位延迟时间 (LVRC/WDTC/RSTC 软件复位)	—	—				
	系统复位延迟时间 (WDT 溢出或 $\overline{RES}$ 引脚复位)	—	—				
t_{SRESET}	软件复位最小脉宽	—	—	45	90	120	μs

注：1. 系统启动时间里提到的 f_{SYS} on/off 状态取决于工作模式类型以及所选的系统时钟振荡器。更多相关细节请参考系统工作模式章节。
2. t_{HIRC} 等符号所表示的时间单位，是对应频率值的倒数，相关频率值在前面表格有说明。例如，$t_{HIRC}=1/f_{HIRC}$，$t_{LIRC}=1/f_{LIRC}$ 等等。
3. 若 LIRC 被选择作为系统时钟源且在休眠模式下 LIRC 关闭，则上面表格中对应 t_{SST} 数值还需加上 LIRC 频率表格里提供的 LIRC 启动时间 t_{START}。
4. 系统速度切换时间实际上是指新使能的振荡器的启动时间。

9. 输入 / 输出口电气特性

Ta=-40°C~85°C，除非另有说明。

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{IL}	I/O 口低电平输入电压	5V	—	0	—	1.5	V
		—	—	0	—	$0.2V_{DD}$	
	$\overline{RES}$ 引脚低电平输入电压	—	$V_{DD}\geq 2.7$	0	—	$0.4V_{DD}$	V
		—	$1.8\leq V_{DD}<2.7$	0	—	$0.3V_{DD}$	V
V_{IH}	I/O 口高电平输入电压	5V	—	3.5	—	5	V
		—	—	$0.8V_{DD}$	—	V_{DD}	
	$\overline{RES}$ 引脚高电平输入电压	—	—	$0.9V_{DD}$	—	V_{DD}	V
I_{OL}	I/O 口灌电流	3V	$V_{OL}=0.1V_{DD}$	5	10	—	mA
		5V		10	20	—	

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
I_{OH}	I/O 口源电流	3V	V_{OH}=0.9V_{DD}, SLEDCn[m+1:m]=00B (n=0, 1; m=0, 2, 4, 6)	-0.5	-1.0	—	mA
		5V		-1.0	-1.8	—	
		3V	V_{OH}=0.9V_{DD}, SLEDCn[m+1:m]=01B (n=0, 1; m=0, 2, 4, 6)	-0.9	-1.7	—	
		5V		-1.7	-3.2	—	
		3V	V_{OH}=0.9V_{DD}, SLEDCn[m+1:m]=10B (n=0, 1; m=0, 2, 4, 6)	-1.2	-2.3	—	
		5V		-2.3	-4.6	—	
		3V	V_{OH}=0.9V_{DD}, SLEDCn[m+1:m]=11B (n=0, 1; m=0, 2, 4, 6)	-2.5	-5.0	—	
		5V		-5	-10	—	
R_{PH}	I/O 口上拉电阻(注)	3V	—	20	60	100	kΩ
		5V	—	10	30	50	
I_{LEAK}	I/O 口输入漏电流	5V	V_{IN}=V_{DD} 或 V_{IN}=V_{SS}	—	—	±1	μA
t_{TCK}	xTM xTCK 输入引脚最小脉宽	—	—	0.3	—	—	μs
t_{TPI}	STM STPI 输入引脚最小脉宽	—	—	0.3	—	—	μs
	PTM PTPI 输入引脚最小脉宽	—	—	50	—	—	ns
f_{TMCLK}	xTM 最大时钟源频率	5V	—	—	—	1	f_{SYS}
t_{CPW}	xTM 捕捉输入最小脉宽	—	—	t_{CPW}(2)	—	—	μs
t_{INT}	中断引脚最小输入脉宽	—	—	10	—	—	μs
t_{RES}	外部复位最小低电压保持时间	—	—	10	—	—	μs

注：1. R_{PH} 内部上拉电阻值的计算方法是：将引脚接地并设置为输入且使能上拉电阻功能，然后在特定电源电压下测量该引脚上的电流，最后电压除以测量的电流值从而得到此上拉电阻值。

2. 对于 PTM：

若 PTCAPTS=0，t_{CPW}=max(2×t_{TMCLK}, t_{TPI})

若 PTCAPTS=1，t_{CPW}=max(2×t_{TMCLK}, t_{TCK})

例 1：若 PTCAPTS=0，f_{TMCLK}=16MHz，t_{TPI}=0.05μs，则 t_{CPW}=max(0.125μs, 0.05μs)=0.125μs

例 2：若 PTCAPTS=1，f_{TMCLK}=12MHz，t_{TCK}=0.3μs，则 t_{CPW}=max(0.16μs, 0.3μs)=0.3μs

例 3：若 PTCAPTS=0，f_{TMCLK}=8MHz，t_{TPI}=0.05μs，则 t_{CPW}=max(0.25μs, 0.05μs)=0.25μs

对于 STM：

t_{CPW}=max(2×t_{TMCLK}, t_{TPI})

例 1：若 f_{TMCLK}=16MHz，t_{TPI}=0.3μs，则 t_{CPW}=max(0.125μs, 0.3μs)=0.3μs

例 2：若 f_{TMCLK}=12MHz，t_{TPI}=0.3μs，则 t_{CPW}=max(0.16μs, 0.3μs)=0.3μs

例 3：若 f_{TMCLK}=8MHz，t_{TPI}=0.3μs，则 t_{CPW}=max(0.25μs, 0.3μs)=0.3μs

其中 t_{TMCLK}=1/f_{TMCLK}

10. 存储器电气特性

Ta=-40°C~85°C，除非另有说明。

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
OTP 程序存储器							
V_{DD}	V_{DD} 读工作电压 – ORPP	—	—	1.8	5.0	5.5	V
	V_{DD} 写工作电压 – ORPP	—	—	4.5	5.0	5.5	V
V_{PP}	V_{PP} 写工作电压 – ORPP	—	—	8.25	8.50	8.75	V
t_{WR}	写周期时间 – ORPP	—	—	—	300	450	μs
E_P	存储单元耐受性 – ORPP	—	—	1	—	—	W
t_{RETD}	ROM 数据保存时间	—	Ta=25°C	—	40	—	Year
Flash 程序存储器 – 仅用于 BX66EV006							
t_{WR}	写周期时间	—	—	—	2.2	2.7	ms
t_{ACTV}	ROM 激活时间 – 从暂停模式唤醒	—	—	32	—	64	μs
RAM 数据存储器							
V_{DR}	RAM 数据保存电压	—	—	1.0	—	—	V

注：1. “W” 表示写次数。
2. 在计算从空闲 / 休眠模式唤醒的系统总启动时间时，还需加上 ROM 激活时间 t_{ACTV}。

11. LVR & LVD 电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{LVR}	低电压复位电压	—	LVR 使能，电压选择 1.7V	-5%	1.7	+5%	V
		—	LVR 使能，电压选择 1.9V	-5%	1.9	+5%	
		—	LVR 使能，电压选择 2.55V	-3%	2.55	+3%	
		—	LVR 使能，电压选择 3.15V	-3%	3.15	+3%	
		—	LVR 使能，电压选择 3.8V	-3%	3.8	+3%	
V_{LVD}	低电压检测电压	—	LVD 使能，电压选择 1.8V	-5%	1.8	+5%	V
		—	LVD 使能，电压选择 2.0V	-5%	2.0	+5%	
		—	LVD 使能，电压选择 2.4V	-5%	2.4	+5%	
		—	LVD 使能，电压选择 2.7V	-5%	2.7	+5%	
		—	LVD 使能，电压选择 3.0V	-5%	3.0	+5%	
		—	LVD 使能，电压选择 3.3V	-5%	3.3	+5%	
		—	LVD 使能，电压选择 3.6V	-5%	3.6	+5%	
		—	LVD 使能，电压选择 4.0V	-5%	4.0	+5%	
$I_{LVRLVDBG}$	工作电流	3V	LVD 使能，LVR 使能，V_{LVR}=1.9V，V_{LVD}=2V VBGEN=0	—	—	10	μA
		5V		—	8	15	μA
		3V	LVD 使能，LVR 使能，V_{LVR}=1.9V，V_{LVD}=2V VBGEN=1	—	—	200	μA
		5V		—	210	245	μA

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
t_{LVDS}	LVDO 稳定时间	—	LVR 使能，VBGEN=0，LVD off → on	—	—	18	μs
		—	LVR 除能，VBGEN=0，LVD off → on	—	—	150	μs
t_{LVR}	产生 LVR 复位的低电压最短保持时间	—	TLVR[1:0]=00B	120	240	480	μs
			TLVR[1:0]=01B	0.5	1.0	2.0	ms
			TLVR[1:0]=10B	1	2	4	
			TLVR[1:0]=11B	2	4	8	
t_{LVD}	产生 LVD 中断的低电压最短保持时间	—	TLVD[1:0]=00B / 11B	60	140	220	μs
		—	TLVD[1:0]=01B	90	200	340	μs
		—	TLVD[1:0]=10B	150	320	580	μs
I_{LVR}	LVR 使能的额外电流	5V	LVD 除能，VBGEN=0	—	—	8	μA
I_{LVD}	LVD 使能的额外电流	5V	LVR 除能，VBGEN=0	—	—	8	μA

12. 内部参考电压特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{BG}	Bandgap 参考电压	—	—	-5%	1.2	+5%	V
t_{BGS}	V_{BG} 开启稳定时间	—	无负载	—	—	50	μs
I_{BG}	Bandgap 使能的额外电流	—	LVR 除能，LVD 除能	—	—	230	μA

注：V_{BG} 电压可用作 A/D 转换器内部信号输入。

13. A/D 转换器电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DD}	工作电压	—	—	1.8	—	5.5	V
V_{ADI}	输入电压	—	—	0	—	V_{REF}	V
V_{REF}	参考电压	—	—	1.8	—	V_{DD}	V
N_R	分辨率	—	—	—	—	12	Bit

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
DNL	非线性微分误差	1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=2.0\mu s$	-3	—	3	LSB
		2V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
INL	非线性积分误差	1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=2.0\mu s$	-4	—	4	LSB
		2V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=0.5\mu s$				
		1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，$V_{REF}=V_{DD}$，$t_{ADCK}=10\mu s$				
I_{ADC}	A/D 转换器使能的额外电流	1.8V	无负载，$t_{ADCK}=2.0\mu s$	—	280	400	μA
		3V	无负载，$t_{ADCK}=0.5\mu s$	—	450	600	μA
		5V		—	850	1000	μA

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
t_{ADCK}	时钟周期	—	1.8V≤V_{DD}<2.0V	2.0	—	10.0	μs
			2.0V≤V_{DD}≤5.5V	0.5	—	10.0	μs
t_{ON2ST}	A/D 转换器 On-to-Start 时间	—	—	4	—	—	μs
t_{ADS}	采样时间	—	—	—	4	—	t_{ADCK}
t_{ADC}	转换时间 (包括 A/D 采样和保持时间)	—	—	—	16	—	t_{ADCK}
GERR	A/D 转换器增益误差	1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	LSB
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	
OSRR	A/D 转换器偏置误差	1.8V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	LSB
		3V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	LSB
		5V	SAINS[2:0]=000B，SAVRS[1:0]=01B，V_{REF}=V_{DD}	-4	—	4	LSB

14. LCD 电气特性

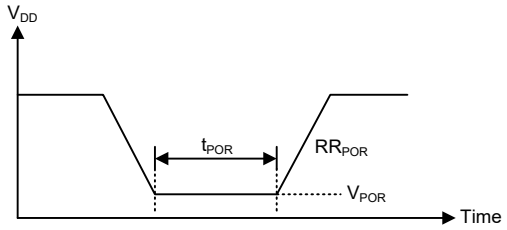
Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
I_{BIAS}	LCD V_{DD}/2 偏压电流	3V	ISEL[2:0]=000B	10.5	15.0	22.5	μA
		5V		17.5	25.0	34.5	μA
		3V	ISEL[2:0]=001B	21	30	39	μA
		5V		35	50	65	μA
		3V	ISEL[2:0]=010B	42	60	78	μA
		5V		70	100	130	μA
		3V	ISEL[2:0]=011B	82.6	118.0	153.4	μA
		5V		140	200	260	μA
		3V	ISEL[2:0]=100B	1.5	3.0	4.5	μA
		5V		2.5	5.0	7.5	μA
		3V	ISEL[2:0]=101B~111B	5.2	7.5	9.8	μA
		5V		8.7	12.5	16.3	μA
V_{SCOM}	LCD COM 端口 V_{DD}/2 电压	2.2V~5.5V	无负载	0.475V_{DD}	0.5V_{DD}	0.525V_{DD}	V

15. 上电复位特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{POR}	上电复位电压	—	—	—	—	100	mV
RR_{POR}	上电复位电压速率	—	—	0.035	—	—	V/ms
t_{POR}	V_{DD} 保持为 V_{POR} 的最小时间	—	—	1	—	—	ms



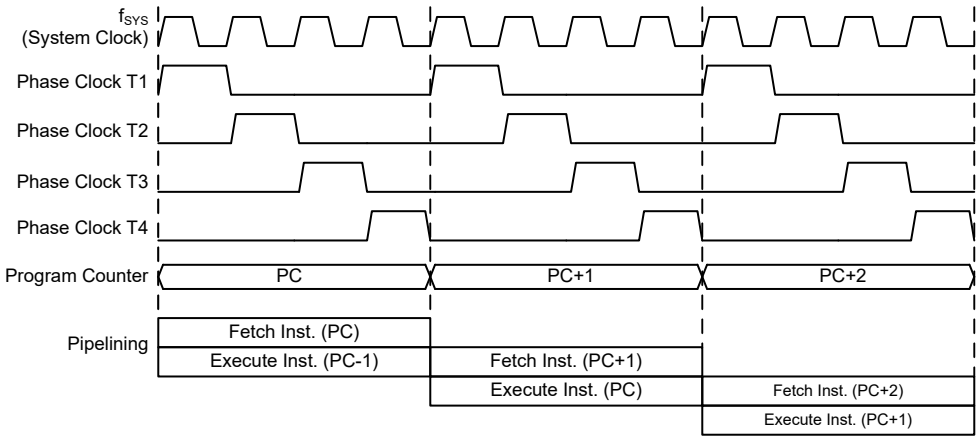
16. 系统结构

内部系统结构是芯群单片机具有良好性能的主要因素。由于采用 RISC 结构，此单片机具有高运算速度和高性能的特点。通过流水线的方式，指令的取得和执行同时进行，此举使得除了跳转和调用指令需要一个以上指令周期外，大部分的标准指令或扩展指令分别能在一个指令周期或两个指令周期内完成。8 位 ALU 参与指令集中所有的运算，它可完成算术运算、逻辑运算、移位、递增、递减和分支等功能，而内部的数据路径则是以通过累加器和 ALU 的方式加以简化。有些寄存器在数据存储器中被实现，且可以直接或间接寻址。简单的寄存器寻址方式和结构特性，确保了在提供具有较大可靠度和灵活性的 I/O 和 A/D 控制系统时，仅需要少数的外部器件。使得这些单片机适用于经济型和批量生产的控制应用。

16.1 时序和流水线结构

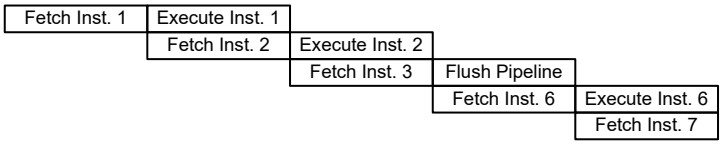
主系统时钟由 HIRC 或 LIRC 振荡器提供，它被细分为 T1~T4 四个内部产生的非重叠时序。在 T1 时间，程序计数器自动加一并抓取一条新的指令。剩下的时间 T2~T4 完成译码和执行功能，因此，一个 T1~T4 时钟周期构成一个指令周期。虽然指令的抓取和执行发生在连续的指令周期，但单片机流水线结构会保证指令在一个指令周期内被有效执行。除非程序计数器的内容被改变，如子程序的调用或跳转，在这种情况下指令将需要多一个指令周期的时间去执行。

如果指令牵涉到分支，例如跳转或调用等指令，则需要两个指令周期才能完成指令执行。需要一个额外周期的原因是程序先用一个周期取出实际要跳转或调用的地址，再用另一个周期去实际执行分支动作，因此用户需要特别考虑额外周期的问题，尤其是在执行时间要求较严格的时候。



系统时序和流水线

```
1          MOV A,[12H]
2          CALL DELAY
3          CPL [12H]
4          :
5          :
6  DELAY:  NOP
```



指令捕捉

16.2 程序计数器

在程序执行期间，程序计数器用来指向下一个要执行的指令地址。除了“JMP”和“CALL”指令需要跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完成以后自动加一。只有较低的 8 位，即所谓的程序计数器低字节寄存器 PCL，可以被用户直接读写。

当执行的指令要求跳转到不连续的地址时，如跳转指令、子程序调用、中断或复位等，单片机通过加载所需要的地址到程序寄存器来控制程序，对于条件跳转指令，一旦条件符合，在当前指令执行时取得的下一条指令将会被舍弃，而由一个空指令周期来取代。

程序计数器	
高字节	低字节 (PCL)
PC11~PC8	PCL7~PCL0

程序计数器

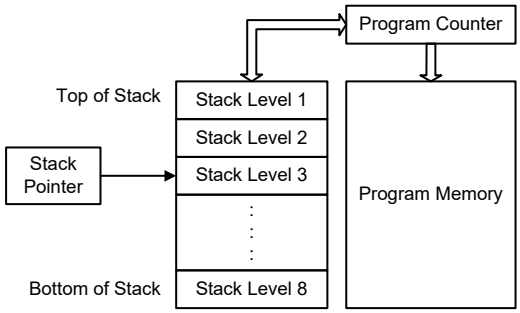
程序计数器的低字节，即程序计数器的低字节寄存器 PCL，可以通过程序控制，且它是可以读取和写入的寄存器。通过直接写入数据到这个寄存器，一个程序短跳转可直接执行，然而只有低字节的操作是有效的，跳转被限制在存储器的当前页中，即 256 个存储器地址范围内，当这样一个程序跳转要执行时，会插入一个空指令周期。程序计数器的低字节可由程序直接进行读取，PCL 的使用可能引起程序跳转，因此需要额外的指令周期。

16.3 堆栈

堆栈是一个特殊的存储空间，用来存储程序计数器中的内容。该单片机有 8 层堆栈，堆栈既不是数据部分也不是程序空间部分，而且它既不是可读取也不是可写入的。当前层由堆栈指针 (SP) 加以指示，同样也是不可读写的。在子程序调用或中断响应服务时，程序计数器的内容被压入到堆栈中。当子程序或中断响应结束时，返回指令 (RET 或 RETI) 使程序计数器从堆栈中重新得到它以前的值。当一个芯片复位后，堆栈指针将指向堆栈顶部。

如果堆栈已满，且有非屏蔽的中断发生，中断请求标志会被置位，但中断响应将被禁止。当堆栈指针减少 (执行 RET 或 RETI)，中断将被响应。这个特性提供程序设计者简单的方法来预防堆栈溢出。然而即使堆栈已满，CALL 指令仍然可以被执行，而造成堆栈溢出。使用时应避免堆栈溢出的情况发生，因为这可能导致不可预期的程序分支指令执行错误。

若堆栈溢出，则首个存入堆栈的程序计数器数据将会丢失。



16.4 算术逻辑单元 – ALU

算术逻辑单元是单片机中很重要的部分，执行指令集中的算术和逻辑运算。ALU 连接到单片机的数据总线，在接收相关的指令码后执行需要的算术与逻辑操作，并将结果存储在指定的寄存器，当 ALU 计算或操作时，可能导致进位、借位或其它状态的改变，而相关的状态寄存器会因此更新内容以显示这些改变，ALU 所提供的功能如下：

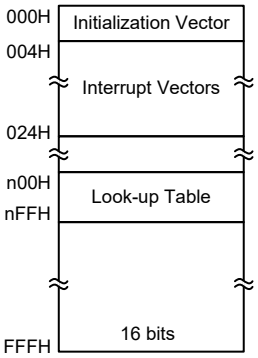
- 算术运算：
  ADD，ADDM，ADC，ADCM，SUB，SUBM，SBC，SBCM，DAA
  LADD，LADDM，LADC，LADCM，LSUB，LSUBM，LSBC，LSBCM，LDAA
- 逻辑运算：
  AND，OR，XOR，ANDM，ORM，XORM，CPL，CPLA
  LAND，LOR，LXOR，LANDM，LORM，LXORM，LCPL，LCPLA
- 移位运算：
  RRA，RR，RRCA，RRC，RLA，RL，RLCA，RLC
  LRRA，LRR，LRRCA，LRRC，LRLA，LRL，LRLCA，LRLC
- 递增和递减：
  INCA，INC，DECA，DEC
  LINCA，LINC，LDECA，LDEC
- 分支判断：
  JMP，SZ，SZA，SNZ，SIZ，SDZ，SIZA，SDZA，CALL，RET，RETI
  LSZ，LSZA，LSNZ，LSIZ，LSDZ，LSIZA，LSDZA

17. OTP 程序存储器

程序存储器用来存放用户代码即储存程序。此单片机提供一次可烧录的存储器(OTP)，用户可将应用程序写入到芯片一次。

17.1 结构

程序存储器的容量为 4K×16 位。程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口。数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。



程序存储器结构

17.2 特殊向量

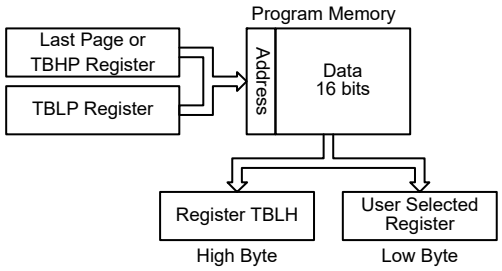
程序存储器内部某些地址保留用做诸如复位和中断入口等特殊用途。地址 000H 是芯片复位后的程序起始地址。在芯片复位之后，程序将跳到这个地址并开始执行。

17.3 查表

程序存储器中的任何地址都可以定义成一个表格，以便储存固定的数据。使用表格时，表格指针必须先行设定，其方式是将表格的地址放在表格指针寄存器 TBLP 和 TBHP 中。这些寄存器定义表格总的地址。

在设定完表格指针后，当数据存储器 [m] 位于 Sector 0，表格数据可以使用如“TABRD [m]”或“TABRDL [m]”等指令分别从程序存储器查表读取。如果存储器 [m] 位于其它 Sector，表格数据可以使用如“LTABRD [m]”或“LTABRDL [m]”等指令分别从程序存储器查表读取。当这些指令执行时，程序存储器中表格数据低字节，将被传送到使用者所指定的数据存储器 [m]，程序存储器中表格数据的高字节，则被传送到 TBLH 特殊寄存器。

下图是查表中寻址 / 数据流程：



17.4 查表范例

以下范例说明表格指针和表格数据如何被定义和执行。这个例子使用的表格 数据用 ORG 伪指令储存在存储器中。ORG 指令的值“0F00H”指向的地址是 4K 程序存储器中

最后一页的起始地址。表格指针低字节寄存器的初始值设为 06H，这可保证从数据表格读取的第一笔数据位于程序存储器地址 1F06H，即最后一页起始地址后的第六个地址。值得注意的是，假如“TABRD [m]”或“LTABRD [m]”指令被使用，则表格指针指向 TBLP 和 TBHP 指定的地址。在这个例子中，表格数据的高字节等于零，而当“TABRD [m]”或“LTABRD [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。TBLH 寄存器为可读 / 可写寄存器，且能重复储存，若主程序和中断服务程序都使用表格读取指令，应该注意它的保护。使用表格读取指令，中断服务程序可能会改变 TBLH 的值，若随后在主程序中再次使用这个值，则会发生错误，因此建议避免同时使用表格读取指令。然而在某些情况下，如果同时使用表格读取指令是不可避免的，则在执行任何主程序的表格读取指令前，中断应该先除能，另外要注意的是所有与表格相关的指令，都需要两个指令周期去完成操作。

表格读取程序范例

```
tempreg1 db ?     ; temporary register #1
tempreg2 db ?     ; temporary register #2
:
:
mov a,06h         ; initialise low table pointer - note that this address
                  ; is referenced
mov tblp,a        ; to the last page or the page that tbhp pointed
mov a,0Fh         ; initialise high table pointer
mov tbhp,a
:
:
tabrd tempreg1    ; transfers value in table referenced by table pointer data at
                  ; program memory address "0F06H" transferred to tempreg1 and
                  ; TBLH
dec tblp          ; reduce value of table pointer by one
tabrd tempreg2    ; transfers value in table referenced by table pointer
                  ; data at program memory address "0F05H" transferred to
                  ; tempreg2 and TBLH in this example the data "1AH" is
                  ; transferred to tempreg1 and data "0FH" to register tempreg2
:
:
org 0F00h         ; sets initial address of program memory
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
:
:
```

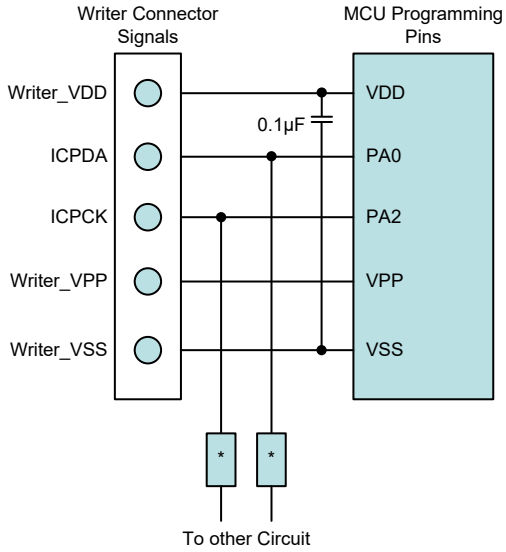
17.5 在线烧录 – ICP

提供 OTP 型程序存储器，用户可将应用程序写入到芯片一次。另外，芯群单片机提供 5 线接口的在线烧录方式。用户可将未经过烧录的单片机芯片连同电路板一起制成，最后阶段进行程序的烧录。

芯群烧录器引脚名称	MCU 在线烧录引脚名称	功能
ICPDA	PA0	串行数据 / 地址烧录
ICPCK	PA2	时钟烧录
VPP	VPP	OTP ROM 可烧录电源 (8.5V)
VDD	VDD	电源；烧录时需在 VDD 和 VSS 之间接一个 0.1μF 的电容
VSS	VSS	地

程序存储器可以通过 5 线的接口在线进行烧录。其中一条线用于数据串行下载或上传、一条线用于串行时钟、剩下三条用于提供电源。芯片在线烧写的详细使用说明超出此文档的描述范围，将由专门的参考文献提供。

烧录过程中，用户必须确保 ICPDA 和 ICPCK 这两个引脚没有连接至其它输出脚。



注：1. ICP 烧录时，需在 VDD 和 VSS 之间接一个 0.1μF 的电容，并且尽可能靠近这两个引脚。
2. * 可能为电阻或电容。若为电阻则其值必须大于 1kΩ，若为电容则其必须小于 1nF。

17.6 片上调试 – OCDS

EV 芯片 BX66EV006 用于单片机 BX66R006 仿真。EV 芯片提供片上调试功能 (OCDS) 用于开发过程中的单片机调试。除了片上调试功能外，单片机和 EV 芯片在功能上几乎是兼容的。用户可将 OCDSDA 和 OCDSCK 引脚连接至芯群 BX-IDE 开发工具，从而实现 EV 芯片对单片机的仿真。OCDSDA 引脚为 OCDS 数据 / 地址输入 / 输出脚，OCDSCK 引脚为 OCDS 时钟输入脚。当用户用 EV 芯片进行调试时，OCDSDA 和 OCDSCK 引脚上的其它共用功能在 EV 芯片无效。关于 OCDS 功能的详细描述，请参考“芯群 e-Link for 8-bit MCU OCDS 用户手册”文件。

芯群 e-Link 引脚名称	EV 芯片引脚名称	功能
OCDSDA	OCDSDA	片上调试串行数据 / 地址输入 / 输出
OCDSCK	OCDSCK	片上调试时钟输入
VDD	VDD	电源
VSS	VSS	地

17.7 OTP ROM 参数烧录功能 – ORPP

该单片机内建 ORPP 功能。单片机提供的 ORPP 功能使用户可以方便地对 OTP 程序存储器进行烧录。注意，写操作只可以对 OTP 程序存储器的最后一页写入数据，数据只能写入一次且无法擦除。

执行写操作之前需将 VPP 引脚接到 8.5V 电源，并且在写入操作完成后将高压电源从 VPP 引脚上移除。若 VPP 功能与 I/O 口共用引脚，则在使用 VPP 功能时，相应的 I/O 口不能设置为输出。

17.7.1 ORPP 寄存器

有三个寄存器可控制内部 ORPP 功能，即数据寄存器 ODL 和 ODH 以及控制寄存器 OCR。

寄存器名称	位							
	7	6	5	4	3	2	1	0
OCR	—	—	—	—	WREN	WR	—	—
ODL	D7	D6	D5	D4	D3	D2	D1	D0
ODH	D15	D14	D13	D12	D11	D10	D9	D8

ORPP 寄存器列表

- **ODL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：ORPP 程序存储器数据 bit 7~bit 0

- **ODH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：ORPP 程序存储器数据 bit 15~bit 8

- **OCR 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	—	—
R/W	—	—	—	—	R/W	R/W	—	—
POR	—	—	—	—	0	0	—	—

Bit 7~4 未定义，读为“0”

Bit 3 **WREN**：ORPP 写使能位
0：除能
1：使能
此位为 ORPP 写使能位，向 ORPP 写操作之前需将此位置高。写周期结束后，硬件自动将此位清零。将此位清零时，则禁止向 ORPP 写操作。

Bit 2 **WR**：ORPP 写控制位
0：写周期结束
1：开始写周期
此位为 ORPP 写控制位，由应用程序将此位置高将激活写周期。写周期结束后，硬件自动将此位清零。当 WREN 未先置高时，此位置高无效。

Bit 1~0 未定义，读为“0”

注：1. 在同一条指令中 WREN 和 WR 不能同时置为“1”。
2. 应注意，当写操作成功启动后，CPU 操作将停止。
3. 需确保 f_{SUB} 时钟运行稳定后才可执行写操作。
4. 需确保写操作已执行完毕后才可执行其它操作。

17.7.2 ORPP 写数据到 OTP 程序存储器

为了实现 ORPP 写操作，ORPP 中写入的数据需存入 ODH 和 ODL 寄存器中，要写入数据的地址需放入 TBLP 寄存器中。OCR 寄存器中的写使能位 WREN 先置高以使能写功能，然后 OCR 寄存器中的 WR 位需立即置高以开始写操作。这两条指令必须在两个指令周期内连续执行才可成功启动一个写操作。进行写操作之前应先将总中断使能位 EMI 清零，在一个有效的写启动步骤完成之后再将其使能。应注意，当写操作成功启动后，CPU 操作将停止。若写周期完成，CPU 将继续执行应用程序，WR 位将自动清除为“0”，通知用户数据已写入 OTP 程序存储器。

17.7.3 ORPP 从 OTP 程序存储器中读取数据

对于 ORPP 读操作，应首先将要读取数据的地址放入 TBLP 寄存器中。然后该数据可以使用“TABRDL [m]”指令从程序存储器中读取。当指令执行时，程序存储器中表格数据低字节，将被传送到使用者所指定的数据存储器 [m]，程序存储器中表格数据的高字节，则被传送到 TBLH 特殊寄存器。

17.7.4 编程注意事项

必须注意的是数据不会无意写入 OTP 程序存储器。在没有写动作时写使能位被正常清零可以增强保护功能。尽管没有必要，写一个简单的读回程序以检查新写入的数据是否正确还是应该考虑的。

WREN 位置位后，OCR 寄存器中的 WR 位需立即置位，以确保写周期正确地执行。写周期开始前总中断位 EMI 应先清零，在一个有效的写启动步骤完成之后再将此位重新使能。注意，单片机不应在 ORPP 写操作完全完成之前进入空闲或休眠模式，否则 ORPP 写操作将失败。

17.7.5 程序举例

ORPP 从 OTP 程序存储器中读取数据

```
Tempreg1 db?              ; temporary register
MOV A, 03H
MOV TBLP, A               ; set read address 03H
TABRDL Tempreg1           ; transfers value in table (last page)
                          ; referenced by table pointer,
                          ; data at program memory address “0F03H”
                          ; transferred to tempreg1 and TBLH
```

ORPP 写数据到 OTP 程序存储器

```
MOV A, ORPP_ADRES         ; user defined address
MOV TBLP, A
MOV A, ORPP_DATA_L        ; user defined data
MOV ODL, A
MOV A, ORPP_DATA_H
MOV ODH, A
MOV A, 00H
MOV OCR, A
CLR EMI
SET WREN                  ; set WREN bit, enable write operation
SET WR                    ; start Write Cycle - set WR bit - executed immediately
                          ; after setting WREN bit
SET EMI
BACK:
SZ  WR                    ; check for write cycle end
JMP BACK
NOP
```

18. 数据存储器

数据存储器是内容可更改的 8 位 RAM 内部存储器，用来储存临时数据。

数据存储器分为两种类型，第一种是特殊功能数据存储器。这些寄存器有固定的地址且与单片机的正确操作密切相关。大多特殊功能寄存器都可在程序控制下直接读取和写入，但有些被加以保护而不对用户开放。第二种数据存储器是做一般用途使用，都可在程序控制下进行读取和写入。

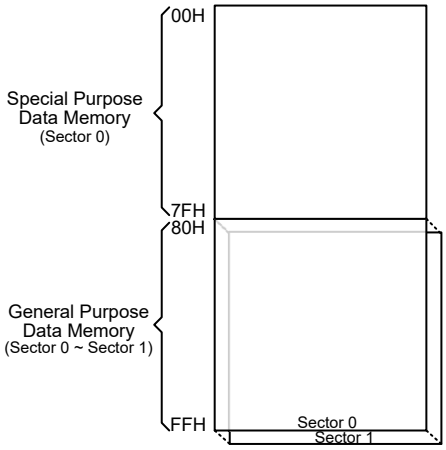
当使用间接寻址时，切换不同的数据存储器 Sector 通过设置正确的存储器指针值实现。

18.1 结构

数据存储器被分为多个 Sector，都可在 8 位的存储器中实现。每个数据存储器 Sector 分为两种类型，即特殊功能数据存储器和通用数据存储器。特殊功能数据存储器的地址范围为 00H~7FH，而通用数据存储器的地址范围为 80H~FFH。

特殊功能数据存储器	通用数据存储器	
所在 Sector	容量	Sector：地址
0	256×8	0：80H~FFH 1：80H~FFH

数据存储器概要



数据存储器结构

18.2 数据存储器寻址

此单片机支持扩展指令架构，因此它没有提供用于数据存储器 Sector 选择的存储区指针。当使用间接寻址方式对数据存储器进行寻址时，通过 MP1H 或 MP2H 寄存器指定所需 Sector，通过 MP1L 或 MP2L 寄存器指定所选 Sector 的具体地址。

直接寻址可用于所有 Sector，通过扩展指令可以寻址所有可用的数据存储器空间。当所访问的地址位于除 Sector 0 外的任何数据存储器 Sector 时，扩展指令可代替间接寻址方式用来访问数据存储器。标准指令和扩展指令的主要区别在于扩展指令中的数据存储器地址“m”可以是 9 位，高字节表示 Sector，低字节表示指定的地址。

18.3 通用数据存储器

所有的单片机程序需要一个读 / 写的存储区，让临时数据可以被储存和再使用，该 RAM 区域就是通用数据存储器。这个数据存储区可让使用者进行读取和写入的操作。使用位操作指令可对个别的位做置位或复位的操作，较大地方便了用户在数据存储器内进行位操作。

18.4 特殊功能数据存储器

这个区域的数据存储器是存放特殊寄存器的，这些寄存器与单片机的正确操作密切相关，大多数的寄存器可进行读取和写入，只有一些是被写保护而只能读取的，相关细节的介绍请参看有关特殊功能寄存器的部分。要注意的是，任何读取指令对存储器中未定义的地址进行读取将返回“00H”。

Sector 0	
00H	IAR0
01H	MP0
02H	IAR1
03H	MP1L
04H	MP1H
05H	ACC
06H	PCL
07H	TBLP
08H	TBLH
09H	TBHP
0AH	STATUS
0BH	
0CH	IAR2
0DH	MP2L
0EH	MP2H
0FH	RSTFC
10H	INTC0
11H	INTC1
12H	INTC2
13H	INTEG
14H	PA
15H	PAC
16H	PAPU
17H	PAWU
18H	PB
19H	PBC
1AH	PBPU
1BH	PC
1CH	PCC
1DH	PCPU
1EH	MFI0
1FH	MFI1
20H	MFI2
21H	TB0C
22H	TB1C
23H	PSC0R
24H	PSC1R
25H	
26H	SCOMC
27H	SADC0
28H	SADC1
29H	
2AH	SADOL
2BH	SADOH
2CH	OCR
2DH	ODL
2EH	ODH
2FH	PWMC
30H	PWMPL
31H	PWMPH
32H	PWMCL
33H	PWMCH
34H	PWM0DL
35H	PWM0DH
36H	PWM1DL
37H	PWM1DH
38H	PWM2DL
39H	PWM2DH
3AH	PWM3DL
3BH	PWM3DH
3CH	PWM4DL
3DH	PWM4DH
3EH	RSTC
3FH	LVDC

Sector 0	
40H	SCC
41H	HIRCC
42H	ORMC
43H	TLVRC
44H	WDTC
45H	LVRC
46H	PTMC0
47H	PTMC1
48H	PTMC2
49H	PTMDL
4AH	PTMDH
4BH	PTMAL
4CH	PTMAH
4DH	PTMBL
4EH	PTMBH
4FH	PTMRPL
50H	PTMRPH
51H	STMC0
52H	STMC1
53H	STMDL
54H	STMDH
55H	STMAL
56H	STMAH
57H	PAS0
58H	PAS1
59H	PBS0
5AH	PBS1
5BH	PCS0
5CH	IFS
5DH	SLEDC0
5EH	SLEDC1
5FH	
7FH	

: Unused, read as 00H

特殊功能数据存储器结构

19. 特殊功能寄存器

大部分特殊功能寄存器的细节将在相关功能章节描述，但有几个寄存器需在此章节单独描述。

19.1 间接寻址寄存器 – IAR0, IAR1, IAR2

间接寻址寄存器 IAR0、IAR1 和 IAR2 的地址虽位于数据存储区域，但不同于普通寄存器，它没有实际的物理地址。与定义实际存储器地址的直接存储器寻址不同，间接寻址是使用间接寻址寄存器和存储器指针来执行存储器数据操作。在间接寻址寄存器 IAR0、IAR1 和 IAR2 上的任何动作，将对存储器指针 MP0、MP1L/MP1H 或 MP2L/MP2H 所指定的存储器地址产生对应的读 / 写操作。它们总是成对出现，IAR0 和 MP0 可以访问 Sector 0，而 IAR1 和 MP1L/MP1H、IAR2 和 MP2L/MP2H 可以访问任何 Sector。因为这些间接寻址寄存器不是实际存在的，直接读取将返回“00H”的结果，而直接写入此寄存器则不做任何操作。

19.2 存储器指针 – MP0, MP1L, MP1H, MP2L, MP2H

该单片机提供五个存储器指针，即 MP0、MP1L、MP1H、MP2L 和 MP2H。由于这些指针在数据存储器中能像普通的寄存器一般被操作，因此提供了一个寻址和数据追踪的有效方法。当对间接寻址寄存器进行任何操作时，单片机指向的实际地址是由存储器指针所指定的地址。MP0、IAR0 用于访问 Sector 0，而 MP1L/MP1H 和 IAR1、MP2L/MP2H 和 IAR2 可根据 MP1H 或 MP2H 寄存器访问所有的 Sector。使用扩展指令可对所有的数据 Sector 进行直接寻址。以下例子说明如何清除一个具有 4 RAM 地址的区块，它们已事先定义成地址 adres1 到 adres4。

间接寻址程序范例 1

```
data .section ´data´
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block  db ?
code .section at 0 ´code´
org 00h
start:
   mov a, 04h             ; set size of block
   mov block, a
   mov a, offset adres1   ; Accumulator loaded with first RAM address
   mov mp0, a             ; setup memory pointer with first RAM address
loop:
   clr IAR0               ; clear the data at address defined by MP0
   inc mp0                ; increase memory pointer
   sdz block              ; check if last memory location has been cleared
   jmp loop
continue:
```

间接寻址程序范例 2

```
data .section ´data´
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block  db ?
code .section at 0 ´code´
```

```
org 00h
start:
   mov a, 04h             ; setup size of block
   mov block, a
   mov a, 01h             ; setup the memory sector
   mov mp1h, a
   mov a, offset adres1   ; Accumulator loaded with first RAM address
   mov mp1l, a            ; setup memory pointer with first RAM address
loop:
   clr IAR1               ; clear the data at address defined by MP1L
   inc mp1l               ; increase memory pointer MP1L
   sdz block              ; check if last memory location has been cleared
   jmp loop
continue:
```

在上面的例子中有一点值得注意，即并没有确定 RAM 地址。

使用扩展指令直接寻址程序范例

```
data .section ´data´
temp db ?
code .section at 0 ´code´
org 00h
start:
   lmov a, [m]            ; move [m] data to acc
   lsub a, [m+1]          ; compare [m] and [m+1] data
   snz  c                 ; [m]>[m+1]?
   jmp  continue          ; no
   lmov a, [m]            ; yes, exchange [m] and [m+1] data
   mov  temp, a
   lmov a, [m+1]
   lmov [m], a
   mov  a, temp
   lmov [m+1], a
continue:
```

注：“m”是位于数据存储器任意 Sector 的某一地址。例如，m=01F0H 表示 Sector 1 中的地址 F0H。

19.3 累加器 – ACC

对任何单片机来说，累加器是相当重要的，且与 ALU 所完成的运算有密切关系，所有 ALU 得到的运算结果都会暂时存在 ACC 累加器里。若没有累加器，ALU 必须在每次进行如加法、减法和移位的运算时，将结果写入到数据存储器，这样会造成程序编写和时间的负担。另外数据传送也常常牵涉到累加器的临时储存功能，例如在使用者定义的一个寄存器和另一个寄存器之间传送数据时，由于两寄存器之间不能直接传送数据，因此必须通过累加器来传送数据。

19.4 程序计数器低字节寄存器 – PCL

为了提供额外的程序控制功能，将程序计数器低字节规划在数据存储器的特殊功能区域内，通过对此寄存器进行操作，便可直接跳转到其它程序地址。直接给 PCL 寄存器赋值将导致程序直接跳转到程序存储器的某一地址，然而由于寄存器只有 8 位长度，因此只允许在本页的程序存储器范围内进行跳转。注意，当执行此操作时，会插入一个空指令周期。

19.5 查表寄存器 – TBLP, TBHP, TBLH

这三个特殊功能寄存器对存储在程序存储器中的表格进行操作。TBLP 和 TBHP 为表格指针，指向表格数据存储的地址。它们的值必须在任何表格读取指令执行前加以设定，由于它们的值可以被如“INC”或“DEC”的指令所改变，这就提供了一种简单的方法对

表格数据进行读取。表格读取数据指令执行之后，表格数据高字节存储在 TBLH 中。其中要注意的是，表格数据低字节会被传送到使用者指定的地址。

19.6 Option 存储器映射寄存器 – ORMC

ORMC 寄存器用于使能 Option 存储器映射功能。Option 存储器的容量为 64 个字。当连续写入特定数据序列 55H 和 AAH 到该寄存器，Option 存储器映射功能将使能，通过使用查表指令即可读到 Option 存储器的内容，Option 存储器的 00H~3FH 地址会一一对应到程序存储器最后一页的 C0H~FFH 地址。

要成功使能 Option 存储器映射功能，该特定的数据序列 55H 和 AAH 必须在两个指令周期内连续写入。建议在写入该特定数据序列前应当先将总中断位 EMI 清零，在数据序列成功写入后，根据用户的需求在适当的时间再将其置高。当数据序列成功写入时会启动内部定时器，$4\times t_{LIRC}$ 时间之后会自动结束映射。因此，用户需及时读出数据，否则需要重新启动 Option 存储器映射功能。每次 ORMC 寄存器被连续写入后，定时器都会重新计数。

当使用查表指令来读取 Option 存储器内容时，“TABRD [m]”和“TABRDL [m]”指令皆可使用。然而，若使用“TABRD [m]”指令来读取，必须配置 TBHP 寄存器将表格指针设定在最后一页。更多查表的描述请参考相关章节。

- **ORMC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	ORMC7	ORMC6	ORMC5	ORMC4	ORMC3	ORMC2	ORMC1	ORMC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **ORMC7~ORMC0**：Option 存储器映射特定数据序列

当将特定数据序列 55H 和 AAH 连续写入该寄存器，会使能 Option 存储器映射功能。需注意，单片机从空闲 / 休眠模式唤醒后，该寄存器的内容将被清除。

19.7 状态寄存器 – STATUS

这 8 位的状态寄存器由 SC 标志位、CZ 标志位、零标志位 (Z)、进位标志位 (C)、辅助进位标志位 (AC)、溢出标志位 (OV)、暂停标志位 (PDF) 和看门狗定时器溢出标志位 (TO) 组成。这些算术 / 逻辑操作和系统运行标志位是用来记录单片机的运行状态。

除了 PDF 和 TO 标志外，状态寄存器中的位像其它大部分寄存器一样可以被改变。任何数据写入到状态寄存器将不会改变 TO 或 PDF 标志位。另外，执行不同的指令后，与状态寄存器有关的运算可能会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出或执行“CLR WDT”或“HALT”指令影响。PDF 标志位只会受执行“HALT”或“CLR WDT”指令或系统上电影响。

SC、CZ、Z、OV、AC 和 C 标志位通常反映最近运算的状态。

- SC：当 OV 与当前指令操作结果 MSB 执行“XOR”所得结果。
- CZ：不同指令不同标志位的操作结果。详细资料请参考寄存器定义部分。
- C：当加法运算的结果产生进位，或减法运算的结果没有产生借位时，则 C 被置位，否则 C 被清零，同时 C 也会被带进位的移位指令所影响。
- AC：当低半字节加法运算的结果产生进位，或低半字节减法运算的结果没有产生借位时，AC 被置位，否则 AC 被清零。
- Z：当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零。
- OV：当运算结果高两位的进位状态异或结果为 1 时，OV 被置位，否则 OV 被清零。

- PDF：系统上电或执行“CLR WDT”指令会清零 PDF，而执行“HALT”指令则会置位 PDF。
- TO：系统上电或执行“CLR WDT”或“HALT”指令会清零 TO，而当 WDT 溢出则会置位 TO。

另外，当进入一个中断程序或执行子程序调用时，状态寄存器不会自动压入到堆栈保存。假如状态寄存器的内容是重要的且子程序可能改变状态寄存器的话，则需谨慎的去做正确的储存。

- **STATUS 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	SC	CZ	TO	PDF	OV	Z	AC	C
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
POR	x	x	0	0	x	x	x	x

“x”：未知

Bit 7　**SC**：当 OV 与当前指令操作结果 MSB 执行“XOR”所得结果

Bit 6　**CZ**：不同指令不同标志位的操作结果
对于 SUB/SUBM/LSUB/LSUBM 指令，CZ 等于 Z 标志位。
对于 SBC/SBCM/LSBC/LSBCM 指令，CZ 等于上一个 CZ 标志位与当前零标志位执行“AND”所得结果。对于其它指令，CZ 标志位无影响。

Bit 5　**TO**：看门狗溢出标志位
0：系统上电或执行“CLR WDT”或“HALT”指令后
1：看门狗溢出发生

Bit 4　**PDF**：暂停标志位
0：系统上电或执行“CLR WDT”指令后
1：执行“HALT”指令

Bit 3　**OV**：溢出标志位
0：无溢出
1：运算结果高两位的进位状态异或结果为 1

Bit 2　**Z**：零标志位
0：算术或逻辑运算结果不为 0
1：算术或逻辑运算结果为 0

Bit 1　**AC**：辅助进位标志位
0：无辅助进位
1：在加法运算中低四位产生了向高四位进位，或减法运算中低四位不发生从高四位借位

Bit 0　**C**：进位标志位
0：无进位
1：如果在加法运算中结果产生了进位，或在减法运算中结果不发生借位
C 标志位也受循环移位指令的影响。

20. 振荡器

不同的振荡器选择可以让使用者在不同的应用需求中实现更大范围的功能。振荡器的灵活性使得在速度和功耗方面可以达到较佳的优化。振荡器的选择和操作是通过配置选项和相关的控制寄存器完成的。

20.1 振荡器概述

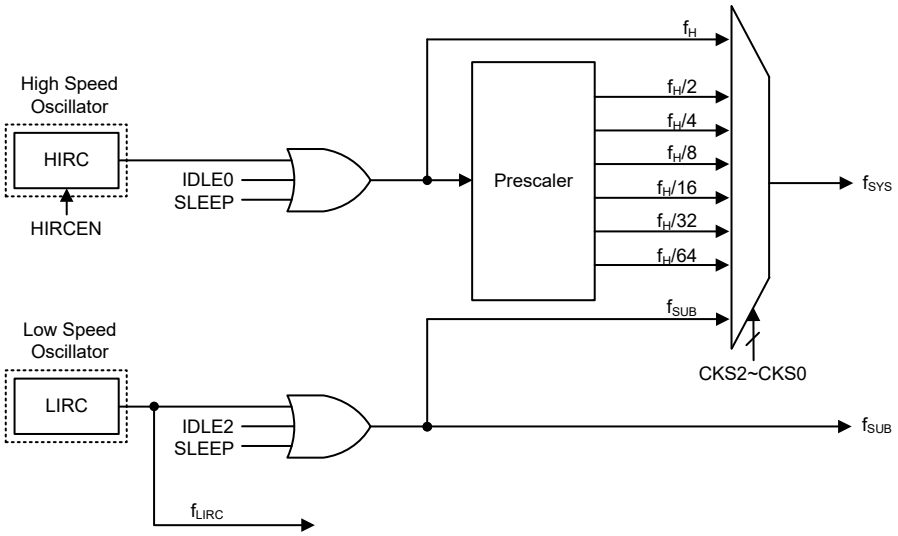
振荡器除了作为系统时钟源，还作为看门狗定时器和时基功能的时钟源。两个完全集成的内部振荡器不需要任何外围器件。它们提供的高速和低速系统振荡器具有较宽的频率范围。较高频率的振荡器提供更高的性能，但要求有更高的功率，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活而优化的性能 / 功耗比，此特性对功耗敏感的应用领域尤为重要。

类型	名称	频率
内部高速 RC	HIRC	8/12/16MHz
内部低速 RC	LIRC	32kHz

振荡器类型

20.2 系统时钟配置

该单片机有两个系统振荡器，包括一个高速振荡器和一个低速振荡器。高速振荡器为内部 8/12/16MHz 高速振荡器 HIRC，低速振荡器为内部 32kHz 低速振荡器 LIRC。使用高速或低速振荡器作为系统时钟的选择是通过设置 SCC 寄存器中的 CKS2~CKS0 位决定的，系统时钟可动态选择。



系统时钟配置选项

20.3 内部 RC 振荡器 – HIRC

内部 RC 振荡器是一个集成的系统振荡器，无需其它外部器件。内部 RC 振荡器具有三种固定的频率：8MHz、12MHz 和 16MHz，可通过 HIRCC 寄存器中的 HIRC1~HIRC0 位进行选择。为了确保能达到交流电气特性里描述的 HIRC 频率精准度，HIRC1~HIRC0 位需要与配置选项中选择的频率吻合。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因 V_{DD}、温度以及芯片制成工艺不同的影响减至较大程度地降低。

20.4 内部 32kHz 振荡器 – LIRC

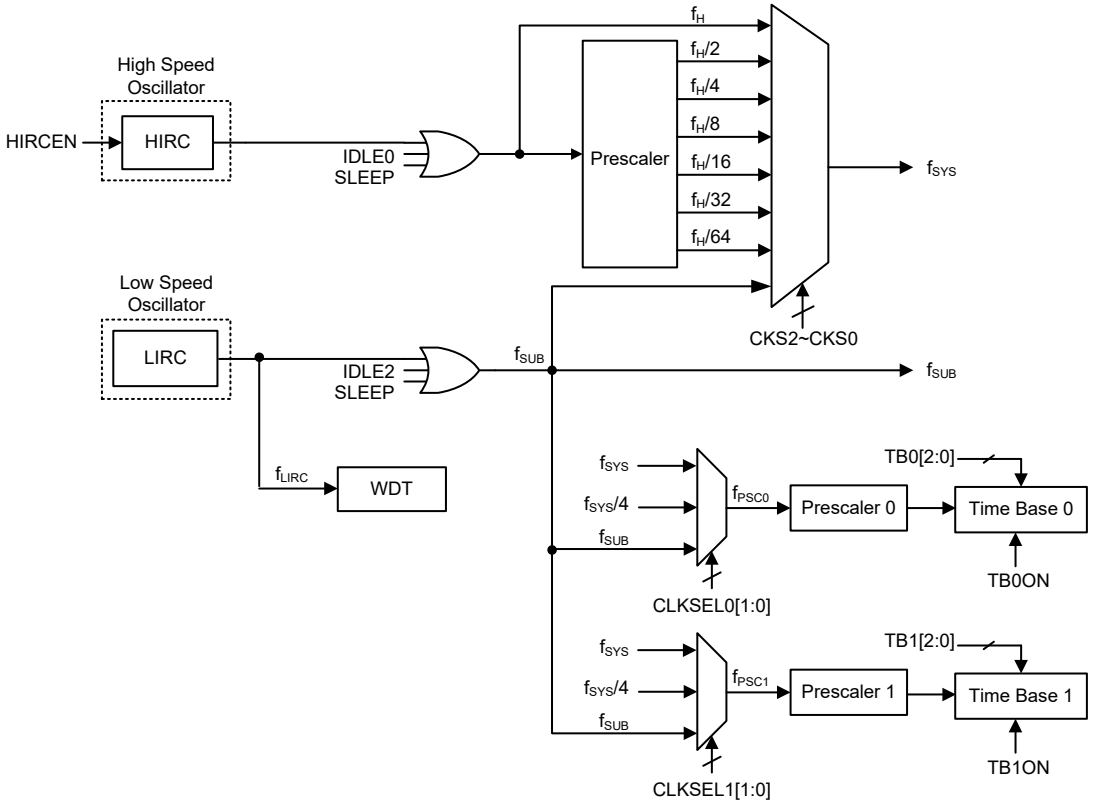
内部 32kHz 系统振荡器是一个完全集成的低频 RC 振荡器，它的典型频率值为 32kHz 且无需外部元件。

21. 工作模式和系统时钟

现今的应用要求单片机具有较高的性能及尽可能低的功耗，这种矛盾的要求在便携式电池供电的应用领域尤为明显。高性能所需要的高速时钟将增加功耗，反之亦然。此单片机提供高、低速两种时钟源，它们之间可以动态切换，用户可通过优化单片机操作来获得较佳性能 / 功耗比。

21.1 系统时钟

单片机为 CPU 和外围功能操作提供了多种不同的时钟源。用户使用寄存器编程可获取多种时钟，进而使系统时钟获取最大的应用性能。主系统时钟可来自高频时钟源 f_H 或低频时钟源 f_{SUB}，通过 SCC 寄存器中的 CKS2~CKS0 位进行选择。高频时钟来自 HIRC 振荡器，低频系统时钟源来自内部时钟 f_{SUB}，若 f_{SUB} 被选择，低频时钟来自 LIRC 振荡器。其它系统时钟还有高速系统振荡器的分频 $f_H/2$~$f_H/64$。



单片机时钟选项

注：当系统时钟源 f_{SYS} 由 f_H 到 f_{SUB} 转换时，可以通过设置相应的高速振荡器使能控制位，选择停止以节省耗电，或者继续振荡，为外围电路提供 f_H~$f_H/64$ 频率的时钟源。

21.2 系统工作模式

单片机有 6 种不同的工作模式，每种有它自身的特性，根据应用中不同的性能和功耗要求可选择不同的工作模式。单片机正常工作有两种模式：快速模式和低速模式。剩余的 4 种工作模式：休眠模式、空闲模式 0、空闲模式 1 和空闲模式 2 用于单片机 CPU 关闭时以节省耗电。

工作模式	CPU	寄存器设置			f_{SYS}	f_H	f_{SUB}	f_{LIRC}
		FHIDEN	FSIDEN	CKS2~CKS0				
快速模式	On	x	x	000~110	f_H~f_H/64	On	On	On
低速模式	On	x	x	111	f_{SUB}	On/Off(1)	On	On
空闲模式 0	Off	0	1	000~110	Off	Off	On	On
				111	On			
空闲模式 1	Off	1	1	xxx	On	On	On	On
空闲模式 2	Off	1	0	000~110	On	On	Off	On
				111	Off			
休眠模式	Off	0	0	xxx	Off	Off	Off	On/Off(2)

“x”：无关

注：1. 在低速模式中，f_H 开启或关闭由相应的振荡器使能位控制。
2. 在休眠模式中，f_{LIRC} 开启或关闭由 WDT 功能使能或除能控制。

21.2.1 快速模式

这是主要的工作模式之一，单片机的所有功能均可在此模式中实现且系统时钟由一个高速振荡器提供。该模式下单片机正常工作的时钟源来自 HIRC 振荡器。高速振荡器频率可被分为 1~64 的不等比率，实际的比率由 SCC 寄存器中的 CKS2~CKS0 位选择。单片机使用高速振荡器分频作为系统时钟可减少工作电流。

21.2.2 低速模式

此模式的系统时钟虽为较低速时钟源，但单片机仍能正常工作。该低速时钟源可来自 f_{SUB}，而 f_{SUB} 来自 LIRC 振荡器。

21.2.3 休眠模式

执行 HALT 指令后且 SCC 寄存器中的 FHIDEN 和 FSIDEN 位都为低时，系统进入休眠模式。在休眠模式中，CPU 停止运行，f_{SUB} 停止为外围功能提供时钟。若看门狗定时器功能由 WDTC 寄存器设定为使能，f_{LIRC} 继续运行。

21.2.4 空闲模式 0

执行 HALT 指令后且 SCC 寄存器中的 FHIDEN 位为低、FSIDEN 位为高时，系统进入空闲模式 0。在空闲模式 0 中，CPU 停止，但低速振荡器会开启以驱动一些外围功能。

21.2.5 空闲模式 1

执行 HALT 指令后且 SCC 寄存器中的 FHIDEN 和 FSIDEN 位都为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但高速和低速振荡器都会开启以确保一些外围功能继续工作。

21.2.6 空闲模式 2

执行 HALT 指令后且 SCC 寄存器中的 FHIDEN 位为高、FSIDEN 位为低时，系统进入空闲模式 2。在空闲模式 2 中，CPU 停止，但高速振荡器会开启以确保一些外围功能继续工作。

21.3 控制寄存器

SCC 和 HIRCC 寄存器用于控制系统时钟和相应的振荡器配置。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SCC	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
HIRCC	—	—	—	—	HIRC1	HIRC0	HIRCF	HIRCEN

系统工作模式控制寄存器列表

- **SCC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
R/W	R/W	R/W	R/W	—	—	—	R/W	R/W
POR	0	0	0	—	—	—	0	0

Bit 7~5 **CKS2~CKS0**：系统时钟选择位
000：f_H
001：$f_H/2$
010：$f_H/4$
011：$f_H/8$
100：$f_H/16$
101：$f_H/32$
110：$f_H/64$
111：f_{SUB}
这三位用于选择系统时钟源。除了 f_H 或 f_{SUB} 提供的系统时钟源外，也可使用高频振荡器的分频作为系统时钟。

Bit 4~2 未定义，读为“0”

Bit 1 **FHIDEN**：CPU 关闭时高频振荡器控制位
0：除能
1：使能
此位用来控制在执行 HALT 指令关闭 CPU 后高速振荡器是开启还是停止。

Bit 0 **FSIDEN**：CPU 关闭时低频振荡器控制位
0：除能
1：使能
此位用来控制在执行 HALT 指令关闭 CPU 后低速振荡器是开启还是停止。

注：使用 CKS2~CKS0 位进行时钟切换设置之后，在相关时钟成功切换至目标时钟源之前需要一定的延时。因此，若接下来执行的操作需要目标时钟源立即响应，则在此之前必须规划适当的延迟时间。
时钟切换延迟时间 = $4\times t_{SYS} + [0 \sim (1.5\times t_{Curr.}+0.5\times t_{Tar.})]$，其中 $t_{Curr.}$ 指代当前的时钟周期，$t_{Tar.}$ 指代目标时钟周期，t_{SYS} 指代当前系统时钟周期。

- **HIRCC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	HIRC1	HIRC0	HIRCF	HIRCEN
R/W	—	—	—	—	R/W	R/W	R	R/W
POR	—	—	—	—	0	0	0	1

Bit 7~4　未定义，读为“0”

Bit 3~2　**HIRC1~HIRC0**：HIRC 频率选择位
00：8MHz
01：12MHz
10：16MHz
11：8MHz
当 HIRC 振荡器使能或通过应用程序改变 HIRC 频率选择位时，在 HIRCF 标志位置高后时钟频率会自动改变。
建议这里选择的频率与配置选项中选定的频率保持一致，以确保能够达到交流电气特性中标示的 HIRC 频率精准度。

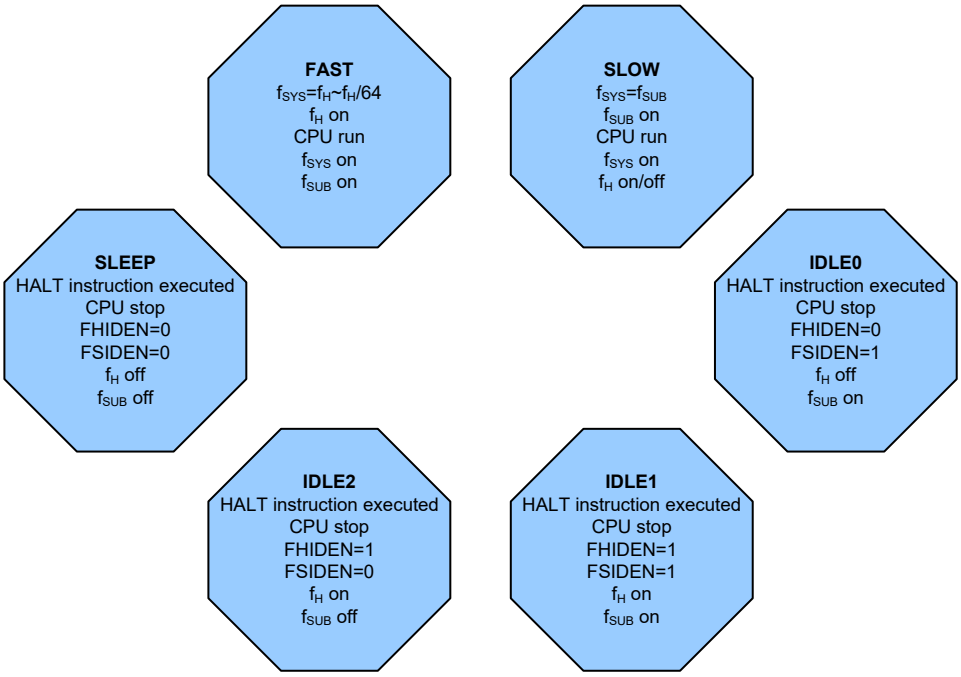
Bit 1　**HIRCF**：HIRC 振荡器稳定标志位
0：HIRC 未稳定
1：HIRC 稳定
此位用于表明 HIRC 振荡器是否稳定。HIRCEN 位置高使能 HIRC 振荡器，或者通过应用程序改变 HIRC 频率选择位时，HIRCF 位会先被清零，待 HIRC 振荡器稳定后会被置高。

Bit 0　**HIRCEN**：HIRC 振荡器使能控制位
0：除能
1：使能

21.4 工作模式切换

此单片机可在各个工作模式间自由切换，使得用户可根据所需选择较佳的性能 / 功耗比。用此方式，对单片机工作的性能要求不高的情况下，可使用较低频时钟以减少工作电流，在便携式应用上延长电池的使用寿命。

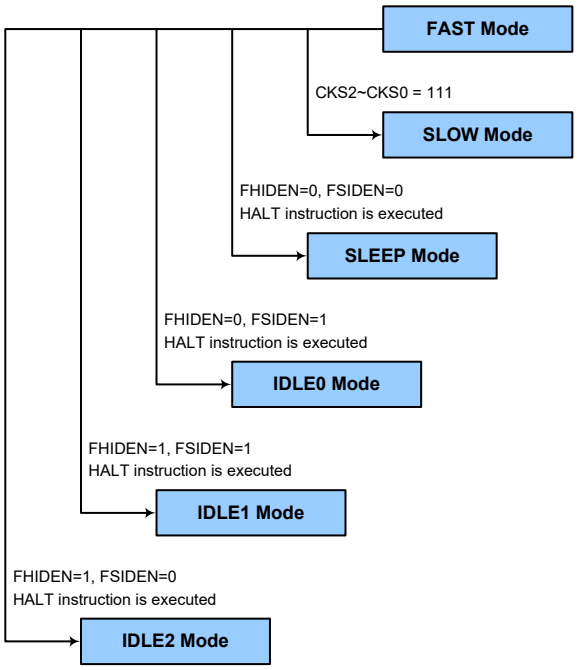
简单来说，快速模式和低速模式间的切换仅需设置 SCC 寄存器中的 CKS2~CKS0 位即可实现，而快速模式 / 低速模式与休眠模式 / 空闲模式间的切换经由 HALT 指令实现。当 HALT 指令执行后，单片机是否进入空闲模式或休眠模式由 SCC 寄存器中的 FHIDEN 和 FSIDEN 位决定的。



21.4.1 快速模式切换到低速模式

系统运行在快速模式时使用高速系统振荡器，因此较为耗电。可通过设置 SCC 寄存器中的 CKS2~CKS0 位为“111”使系统时钟切换至运行在低速模式下。此时将使用低速系统振荡器以节省耗电。用户可在对性能要求不高的操作中使用此方法以减少耗电。

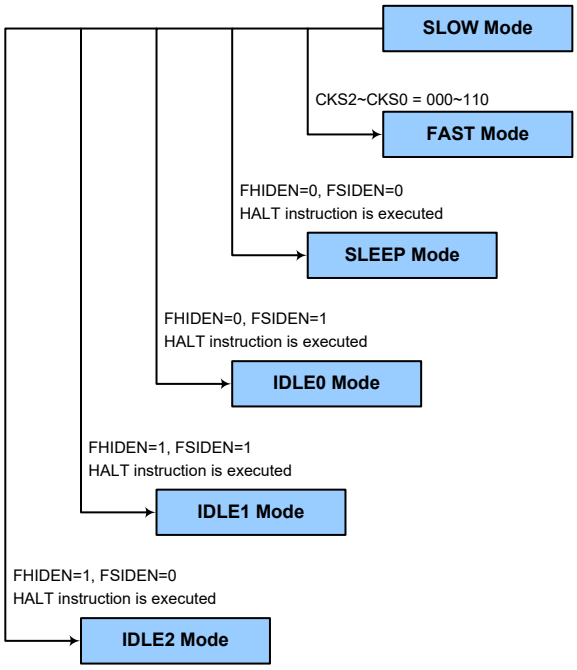
低速模式的系统时钟源来自 LIRC 振荡器，因此要求此振荡器在所有模式切换动作发生前稳定下来。



21.4.2 低速模式切换到快速模式

在低速模式时系统时钟来自 f_{SUB}。切换回快速模式时，需设置 CKS2~CKS0 位为“000”~“110”使系统时钟从 f_{SUB} 切换到 f_H~f_H/64。

然而，如果在低速模式下 f_H 因未使用而关闭，那么从低速模式切换到快速模式时，它需要一定的时间来重新起振和稳定，可通过检测 HIRCC 寄存器中的 HIRCF 位进行判断，所需的高速系统振荡器稳定时间在系统上电时间电气特性中有说明。



21.4.3 进入休眠模式

进入休眠模式的方法仅有一种，即应用程序中执行“HALT”指令前需设置 SCC 寄存器中的 FHIDEN 和 FSIDEN 位都为“0”。在这种模式下，除了 WDT 以外的所有时钟和功能都将关闭。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟停止运行，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清零。
- 如果 WDT 功能使能，WDT 将被清零并重新开始计数。如果 WDT 功能除能，WDT 将被清零并停止计数。

21.4.4 进入空闲模式 0

进入空闲模式 0 的方法仅有一种，即应用程序中执行“HALT”指令前需设置 SCC 寄存器中的 FHIDEN 位为“0”且 FSIDEN 位为“1”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 时钟停止运行，应用程序停止在“HALT”指令处，但 f_{SUB} 时钟将继续运行。
- 数据存储器中的内容和寄存器将保持当前值。
- 输入 / 输出口将保持当前值。

- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清零。
- 如果 WDT 功能使能，WDT 将被清零并重新开始计数。如果 WDT 功能除能，WDT 将被清零并停止计数。

21.4.5 进入空闲模式 1

进入空闲模式 1 的方法仅有一种，即应用程序中执行“HALT”指令前需设置 SCC 寄存器中的 FHIDEN 和 FSIDEN 位都为“1”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 和 f_{SUB} 时钟开启，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清零。
- 如果 WDT 功能使能，WDT 将被清零并重新开始计数。如果 WDT 功能除能，WDT 将被清零并停止计数。

21.4.6 进入空闲模式 2

进入空闲模式 2 的方法仅有一种，即应用程序中执行“HALT”指令前需设置 SCC 寄存器中的 FHIDEN 位为“1”且 FSIDEN 位为“0”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 时钟开启，f_{SUB} 时钟关闭，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清零。
- 如果 WDT 功能使能，WDT 将被清零并重新开始计数。如果 WDT 功能除能，WDT 将被清零并停止计数。

21.5 待机电流的注意事项

由于单片机进入休眠或空闲模式的主要原因是将单片机的电流降低到尽可能低，可能到只有几个微安的级别 (空闲模式 1 和空闲模式 2 除外)，所以如果要将电路的电流进一步降低，电路设计者还应有其它的考虑。应该特别注意的是单片机的输入 / 输出引脚。所有高阻抗输入脚都必须连接到固定的高或低电平，因为引脚浮空会造成内部振荡并导致耗电增加。这也应用于有不同封装的单片机，因为它们可能含有未引出的引脚，这些引脚也必须设为输出或带有上拉电阻的输入。

另外还需注意单片机设为输出的 I/O 引脚上的负载。应将它们设置在有最小拉电流的状态或将它们和其它的 CMOS 输入一样接到没有拉电流的外部电路上。还应注意的是，如果选择 LIRC 振荡器，会导致耗电增加。

在空闲模式 1 和空闲模式 2 中，高速振荡器开启。若外围功能时钟源来自高速振荡器，额外的待机电流也可能会有几百微安。

21.6 唤醒

单片机进入休眠模式或空闲模式后，系统时钟将停止以降低功耗。然而单片机再次唤醒，原来的系统时钟重新起振、稳定且恢复正常工作需要一定的时间。

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- PA 口下降沿
- 外部 $\overline{\text{RES}}$ 引脚复位
- 系统中断
- WDT 溢出

执行 HALT 指令后单片机将进入空闲或休眠模式，PDF 将被置位。系统上电或执行清除看门狗的指令，PDF 将被清零。

如果单片机由外部 $\overline{\text{RES}}$ 引脚唤醒，系统会完全复位；若系统由看门狗计数器溢出唤醒，将会启动看门狗复位并置位 TO 标志，这种复位只会重置程序计数器和堆栈指针，其它标志保持原有状态。

PA 口中的每个引脚都可以通过 PAWU 寄存器使能下降沿唤醒功能。PA 端口唤醒后，程序将在“HALT”指令后继续执行。如果系统是通过中断唤醒，则有两种可能发生。第一种情况是：相关中断除能或是中断使能且堆栈已满，则程序会在“HALT”指令之后继续执行。这种情况下，唤醒系统的中断会等到相关中断使能或有堆栈层可以使用之后才执行。第二种情况是：相关中断使能且堆栈未满，则中断可以马上执行。如果在进入休眠或空闲模式之前中断标志位已经被设置为“1”，则相关中断的唤醒功能将无效。

22. 看门狗定时器

看门狗定时器的功能在于防止如电磁的干扰等外部不可控制事件，所造成的程序不正常动作或跳转到未知的地址。

22.1 看门狗定时器时钟源

WDT 定时器时钟源 f_{LIRC} 由内部低速振荡器 LIRC 提供。内部振荡器 LIRC 的频率大约为 32kHz，这个特殊的内部时钟周期会随 V_{DD}、温度和制成的不同而变化。看门狗定时器的时钟源可分频为 2^8~2^{18} 以提供更大的溢出周期，分频比由 WDTC 寄存器中的 WS2~WS0 位来决定。

22.2 看门狗定时器控制寄存器

WDTC 寄存器用于选择溢出周期、控制 WDT 功能的使能 / 除能和 MCU 复位操作。

- **WDTC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0**：WDT 功能软件控制
10101：除能
01010：使能
其它值：单片机复位
如果由于不利的环境因素使这些位变为其它值，单片机将复位。复位动作发生在 t_{SRESET} 延迟时间后，且 RSTFC 寄存器的 WRF 位将置为“1”。

Bit 2~0 **WS2~WS0**：WDT 溢出周期选择位
000：$2^8/f_{LIRC}$
001：$2^{10}/f_{LIRC}$
010：$2^{12}/f_{LIRC}$
011：$2^{14}/f_{LIRC}$
100：$2^{15}/f_{LIRC}$
101：$2^{16}/f_{LIRC}$
110：$2^{17}/f_{LIRC}$
111：$2^{18}/f_{LIRC}$
这三位控制 WDT 时钟源的分频比，从而实现对 WDT 溢出周期的控制。

- **RSTFC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	RSTF	LVRF	LRF	WRF
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	x	0	0

“x”：未知

Bit 7~4 未定义，读为“0”

Bit 3 **RSTF**：复位控制寄存器软件复位标志位
具体描述见引脚复位章节。

Bit 2 **LVRF**：LVR 复位标志位
具体描述见低电压复位章节。

Bit 1 **LRF**：LVR 控制寄存器软件复位标志位
具体描述见低电压复位章节。

Bit 0 **WRF**：WDTC 寄存器软件复位标志位
0：未发生
1：发生
当 WDTC 寄存器软件复位发生时此位被置为“1”，且该位只能通过应用程序清零。

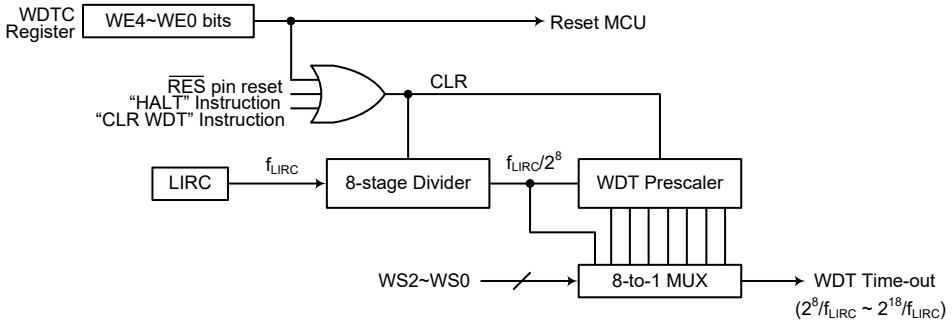
22.3 看门狗定时器操作

WDT 溢出时，它产生一个单片机复位的动作。这也就意味着正常工作期间，用户需在应用程序中看门狗溢出前有策略地清看门狗定时器以防止其产生复位，可使用清除看门狗指令实现。无论什么原因，程序失常跳转到一个未知的地址或进入一个死循环，这个清除指令不能被正确执行，此种情况下，看门狗将溢出以使单片机复位。看门狗定时器控制寄存器 WDTC 中的 WE4~WE0 位可提供 WDT 功能的使能 / 除能控制以及单片机复位操作。当 WE4~WE0 设置为“10101B”时除能 WDT 功能，而当设置为“01010B”时使能 WDT 功能。如果 WE4~WE0 设置为除“01010B”和“10101B”以外的值时，单片机将在 t_{SRESET} 延迟时间后复位。上电后这些位初始化为“01010B”。

WE4~WE0 位	WDT 功能
10101B	除能
01010B	使能
其它值	单片机复位

程序正常运行时，WDT 溢出将导致单片机复位，并置位状态标志位 TO。若系统处于休眠或空闲模式，当 WDT 发生溢出时，状态寄存器中的 TO 和 PDF 位应置位，仅 PC 和堆栈指针复位。有四种方法可以用来清除 WDT 的内容。第一种是 WDTC 软件复位，即将 WE4~WE0 位设置成除了“10101B”和“01010B”外的任意值；第二种是通过 WDT 软件清除指令，而第三种是通过“HALT”指令。最后一种是外部硬件复位 (外部复位引脚低电平)。

该单片机只使用一条软件指令清看门狗。只要执行“CLR WDT”便清除 WDT。当设置分频比为 2^{18} 时，溢出周期最大。当时钟源为 32kHz LIRC 振荡器，分频比为 2^{18} 时最大溢出周期约 8s，分频比为 2^8 时最小溢出周期约 8ms。



看门狗定时器

23. 复位和初始化

复位功能是任何单片机中基本的部分，使得单片机可以设定一些与外部参数无关的先置条件。最重要的复位条件是在单片机首次上电以后，经过短暂的延迟，内部硬件电路使得单片机处于预期的稳定状态并开始执行第一条程序指令。上电复位以后，在程序执行之前，部分重要的内部寄存器将会被设定为预先设定的状态。程序计数器就是其中之一，它会被清为零，使得单片机从最低的程序存储器地址开始执行程序。

除了上电复位外，即使单片机处于正常工作状态，有些情况的发生也会迫使单片机复位。譬如当单片机上电后已经开始执行程序，$\overline{\text{RES}}$ 引脚被强制拉为低电平。这种复位为正常操作复位，单片机中只有一些寄存器受影响，而大部分寄存器不会改变，在复位引脚恢复至高电平后，单片机可以正常运行。

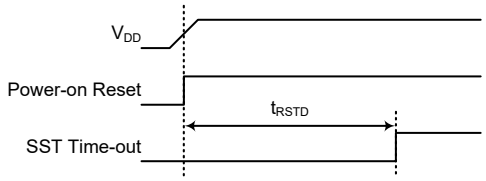
另一种复位为低电压复位即 LVR 复位，在电源供应电压低于 LVR 设置值时，系统会产生 LVR 复位，这种复位与 $\overline{\text{RES}}$ 脚拉低复位方式相似。另一种复位为看门狗溢出单片机复位，不同方式的复位操作会对寄存器产生不同的影响。不同方式的复位操作会对寄存器产生不同的影响。

23.1 复位功能

单片机的几种内部复位和外部复位方式将在此处做具体介绍。

23.1.1 上电复位

这是最基本且不可避免的复位，发生在单片机上电后。除了保证程序存储器从开始地址执行，上电复位也使得其它寄存器被设定在预设条件。所有的输入 / 输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设定为输入状态。



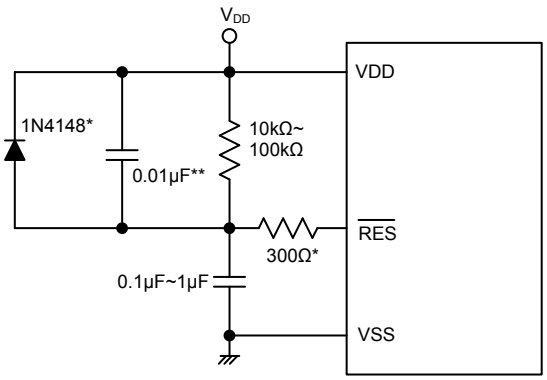
上电复位时序图

23.1.2 $\overline{\text{RES}}$ 引脚复位

由于复位引脚与 I/O 口共用，复位功能必须通过配置选项以及 RSTC 控制寄存器选择。虽然单片机有一个内部 RC 复位功能，如果电源上升缓慢或上电时电源不稳定，内部 RC 振荡可能导致芯片复位不良，所以推荐使用和 $\overline{\text{RES}}$ 引脚连接的外部 RC 电路，由 RC 电路所造成的时间延迟使得 $\overline{\text{RES}}$ 引脚在电源供应稳定前的一段延长周期内保持在低电平。在这段时间内，单片机的正常操作是被禁止的。$\overline{\text{RES}}$ 引脚达到一定电压值后，再经过延迟时间 t_{RSTD} 单片机可以开始进行正常操作。下图中 SST 是系统延迟周期 System Start-up Timer 的缩写。

在许多应用场合，可以在 VDD 和 $\overline{\text{RES}}$ 之间接入一个电阻，在 VSS 与 $\overline{\text{RES}}$ 之间接入一个电容作为外部复位电路。与 $\overline{\text{RES}}$ 脚上所有相连接的线段必须尽量短以减少噪声干扰。

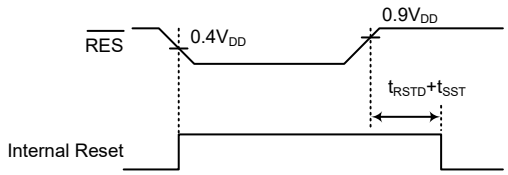
当系统在较强干扰的场合工作时，建议使用增强型的复位电路，如下图所示。



注：“*”表示建议加上此元件以加强静电保护。
“**”表示建议在电源有较强干扰场合加上此元件。

外部 $\overline{\text{RES}}$ 电路

$\overline{\text{RES}}$ 引脚通过外部硬件强迫拉至低电平时，此种复位形式即会发生。这种复位方式和其它的复位方式一样，程序计数器会被清除为零且程序从头开始执行。



$\overline{\text{RES}}$ 复位时序图

内部复位控制寄存器 RSTC 用于选择 $\overline{\text{RES}}$ 引脚功能以及为单片机在受到环境噪声干扰而异常工作时提供复位。如果 RSTC 寄存器的内容被设置为除 01010101B 或 10101010B 以外的任何值，单片机会在 t_{SRESET} 延迟时间后发生复位。上电后寄存器的值为 01010101B。

I/O 或 $\overline{\text{RES}}$ 引脚配置选项	RSTC7~RSTC0 位	功能描述
Always I/O	01010101B 或 10101010B	PC2 引脚或其它引脚共用功能
	其它值	单片机复位
Always $\overline{\text{RES}}$	01010101B 或 10101010B	$\overline{\text{RES}}$ 引脚
	其它值	单片机复位
RSTC 寄存器控制	01010101B	PC2 引脚或其它引脚共用功能
	10101010B	$\overline{\text{RES}}$ 引脚
	其它值	单片机复位

内部复位功能控制

- **RSTC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	RSTC7	RSTC6	RSTC5	RSTC4	RSTC3	RSTC2	RSTC1	RSTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **RSTC7~RSTC0**：复位功能控制位

如果配置选项中设置为“Always I/O”
01010101B 或 10101010B：PC2 引脚或其它引脚共用功能
其它值：单片机复位

如果配置选项中设置为“Always $\overline{\text{RES}}$”
01010101B 或 10101010B：$\overline{\text{RES}}$ 引脚
其它值：单片机复位

如果配置选项中设置为“RSTC 寄存器控制”
01010101B：PC2 引脚或其它引脚共用功能
10101010B：$\overline{\text{RES}}$ 引脚
其它值：单片机复位

通过配置选项的选择决定 RSTC7~RSTC0 位如何设置。

如果由于不利的环境因素使这些位变为除 01010101B 和 10101010B 的其它值，单片机将复位。复位动作发生在 t_{SRESET} 延迟时间后，且 RSTFC 寄存器的 RSTF 位将置为“1”。

除了 WDT 溢出硬件复位外，其它所有复位发生时此寄存器恢复至上电复位值。注意，当通过设置此寄存器为“10101010B”选择 $\overline{\text{RES}}$ 引脚功能时，此设置的优先级高于其引脚共用功能选择设置。

- **RSTFC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	RSTF	LVRF	LRF	WRF
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	x	0	0

“x”：未知

Bit 7~4 未定义，读为“0”

Bit 3 **RSTF**：复位控制寄存器软件复位标志位
0：未发生
1：发生
当 RSTC 控制寄存器软件复位发生时，此位被置为“1”，且只能通过应用程序清零。

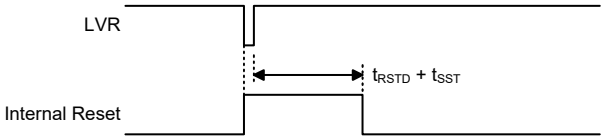
Bit 2 **LVRF**：LVR 复位标志位
详见其它章节

Bit 1 **LRF**：LVR 控制寄存器软件复位标志位
详见其它章节

Bit 0 **WRF**：WDT 控制寄存器软件复位标志位
详见其它章节

23.1.3 低电压复位 – LVR

单片机具有低电压复位电路，用来监测它的电源电压。当电源电压低于某一预定值时，它将复位单片机。LVR 功能可通过 LVRC 寄存器使能或除能。若 LVRC 控制寄存器配置为使能 LVR，低电压复位功能在快速和低速模式下始终使能，并会设定一个电源复位低电压 V_{LVR}。例如在更换电池的情况下，单片机供应的电压可能会在 0.9V~V_{LVR} 之间，这时 LVR 将会自动复位单片机且 RSTFC 寄存器中的 LVRF 标志位置位。LVR 包含以下的规格：有效的 LVR 信号，即在 0.9V~V_{LVR} 的低电压状态的时间，必须超过 LVR/LVD 电气特性中 t_{LVR} 参数的值。如果低电压存在时间不超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。实际的 t_{LVR} 值可通过 TLVRC 寄存器中的 TLVR1~TLVR0 位设置。实际的 V_{LVR} 参数值可通过 LVRC 寄存器中的 LVS7~LVS0 位设置。若由于受到干扰 LVS7~LVS0 变为其它值时，需经过一段 t_{SRESET} 延迟时间才会响应复位。此时 RSTFC 寄存器的 LRF 位被置位。上电复位后 LVRC 的初始值是 01100110B。注意当单片机进入空闲或休眠模式，LVR 功能将自动关闭。



低电压复位时序图

- **LVRC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	0	0	1	1	0

Bit 7~0 **LVS7~LVS0**：LVR 电压选择
01100110：1.7V
01010101：1.9V
00110011：2.55V
10011001：3.15V
10101010：3.8V
11110000：除能
其它值：复位单片机 – 寄存器复位为 POR 值。
若有以上定义的五个低电压复位值的低电压情况发生，且检测到此低电压的保持时间大于 t_{LVR}，则单片机复位发生。实际的 t_{LVR} 值可通过 TLVRC 寄存器中的 TLVR1~TLVR0 位设置。此种复位后的寄存器内容保持不变。
除了以上定义的五个低电压复位值及 11110000B 外，其它值也能导致单片机复位。需要经过一段 t_{SRESET} 延迟时间来响应复位。但此时寄存器内容将复位为 POR 值。

- **TLVRC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	TLVR1	TLVR0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	1

Bit 7~2　未定义，读为“0”

Bit 1~0　**TLVR1~TLVR0**：产生 LVR 复位的低电压最短保持时间 (t_{LVR}) 选择
00：(7~8) × t_{LIRC}
01：(31~32) × t_{LIRC}
10：(63~64) × t_{LIRC}
11：(127~128) × t_{LIRC}

- **RSTFC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	RSTF	LVRF	LRF	WRF
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	x	0	0

“x”：未知

Bit 7~4　未定义，读为“0”

Bit 3　**RSTF**：复位控制寄存器软件复位标志位
具体描述见引脚复位章节。

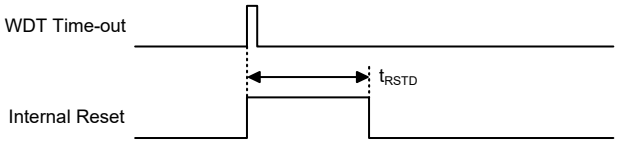
Bit 2　**LVRF**：LVR 复位标志位
0：未发生
1：发生
当特定的低电压复位条件发生时，此位被置为“1”，且该位只能通过应用程序清零。

Bit 1　**LRF**：LVR 控制寄存器软件复位标志位
0：未发生
1：发生
如果 LVRC 寄存器包含任何非定义的 LVR 电压值，此位被置为“1”，这类似于软件复位功能，且只能通过应用程序清零。

Bit 0　**WRF**：WDTC 寄存器软件复位标志位
0：未发生
1：发生
具体描述见看门狗定时器控制寄存器章节。

23.1.4 正常运行时看门狗溢出复位

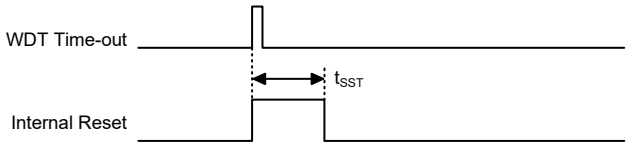
除了看门狗溢出标志位 TO 将被设为“1”之外，在快速或低速模式下正常运行时看门狗溢出复位和 LVR 硬件复位相同。



正常运行时看门狗溢出复位时序图

23.1.5 休眠或空闲时看门狗溢出复位

休眠或空闲时看门狗溢出复位和其它种类的复位有些不同。除了程序计数器与堆栈指针将被清“0”及 TO 与 PDF 位被设为“1”外，绝大部分的条件保持不变。图中 t_{SST} 的详细说明请参考系统上电时间电气特性。



休眠或空闲时看门狗溢出复位时序图

23.2 复位初始状态

不同的复位形式以不同的途径影响复位标志位。这些标志位，即 PDF 和 TO 位存放在状态寄存器中，由休眠或空闲模式功能或看门狗计数器等几种控制器操作控制。复位标志位如下所示：

TO	PDF	复位条件
0	0	上电复位
u	u	快速模式或低速模式时的 $\overline{RES}$ 复位或 LVR 复位
1	u	快速模式或低速模式时的 WDT 溢出复位
1	1	空闲或休眠模式时的 WDT 溢出复位

“u”代表不改变

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

项目	复位后情况
程序计数器	清除为零
中断	所有中断被除能
看门狗定时器，时基	都清为零，且 WDT 重新计数
定时器模块	所有定时器模块停止
输入 / 输出口	I/O 口设为输入模式
堆栈指针	堆栈指针指向堆栈顶端

不同的复位形式对单片机内部寄存器的影响是不同的。为保证复位后程序能正常执行，了解寄存器在特定条件复位后的设置是非常重要的。下表即为不同方式复位后内部寄存器的状况。

寄存器	上电复位	$\overline{RES}$ 复位 (正常运行)	WDT 溢出 (正常操作)	WDT 溢出 (空闲 / 休眠)
IAR0	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP0	0000 0000	0000 0000	0000 0000	uuuu uuuu
IAR1	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1L	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1H	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常操作)	WDT 溢出 (空闲 / 休眠)
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu
STATUS	xx00 xxxx	uuuu uuuu	uu1u uuuu	uu11 uuuu
IAR2	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2L	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2H	0000 0000	0000 0000	0000 0000	uuuu uuuu
RSTFC	---- 0x00	---- uuuu	---- uuuu	---- uuuu
INTC0	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTC2	--00 --00	--00 --00	--00 --00	--uu --uu
INTEG	---- 0000	---- 0000	---- 0000	---- uuuu
PA	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAPU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAWU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PB	-111 1111	-111 1111	-111 1111	-uuu uuuu
PBC	-111 1111	-111 1111	-111 1111	-uuu uuuu
PBPU	-000 0000	-000 0000	-000 0000	-uuu uuuu
PC	---- -111	---- -111	---- -111	---- -uuu
PCC	---- -111	---- -111	---- -111	---- -uuu
PCPU	---- -000	---- -000	---- -000	---- -uuu
MFI0	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI2	--00 --00	--00 --00	--00 --00	--uu --uu
TB0C	0--- -000	0--- -000	0--- -000	u--- -uuu
TB1C	0--- -000	0--- -000	0--- -000	u--- -uuu
PSC0R	---- --00	---- --00	---- --00	---- --uu
PSC1R	---- --00	---- --00	---- --00	---- --uu
SCOMC	0000 ----	0000 ----	0000 ----	uuuu ----
SADC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
SADC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
SADOL	xxxx ----	xxxx ----	xxxx ----	uuuu ---- (ADRFS=0)
	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRFS=1)
SADOH	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRFS=0)
	---- xxxx	---- xxxx	---- xxxx	---- uuuu (ADRFS=1)
OCR	---- 00--	---- 00--	---- 00--	---- uu--
ODL	0000 0000	0000 0000	0000 0000	uuuu uuuu
ODH	0000 0000	0000 0000	0000 0000	uuuu uuuu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常操作)	WDT 溢出 (空闲 / 休眠)
PWMC	0--- -000	0--- -000	0--- -000	u--- -uuu
PWMPL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWMPH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWMCL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWMCH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM0DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM0DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM1DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM1DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM2DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM2DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM3DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM3DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM4DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM4DH	0000 0000	0000 0000	0000 0000	uuuu uuuu
RSTC	0101 0101	0101 0101	0101 0101	uuuu uuuu
LVDC	0000 0000	0000 0000	0000 0000	uuuu uuuu
SCC	000- --00	000- --00	000- --00	uuu- --uu
HIRCC	---- 0001	---- 0001	---- 0001	---- uuuu
ORMC	0000 0000	0000 0000	0000 0000	0000 0000
TLVRC	---- --01	---- --01	---- --01	---- --uu
WDTC	0101 0011	0101 0011	0101 0011	uuuu uuuu
LVRC	0110 0110	0110 0110	0110 0110	uuuu uuuu
PTMC0	0000 0---	0000 0---	0000 0---	uuuu u---
PTMC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMC2	---- -000	---- -000	---- -000	---- -uuu
PTMDL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMDH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMAL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMAH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMBL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMBH	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMRPL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMRPH	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDH	---- --00	---- --00	---- --00	---- --uu
STMAL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAH	---- --00	---- --00	---- --00	---- --uu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常操作)	WDT 溢出 (空闲 / 休眠)
PAS0	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAS1	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBS0	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBS1	--00 0000	--00 0000	--00 0000	--uu uuuu
PCS0	---- 0000	---- 0000	---- 0000	---- uuuu
IFS	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLEDC0	1111 1111	1111 1111	1111 1111	uuuu uuuu
SLEDC1	---- --11	---- --11	---- --11	---- --uu

注：“u”表示不改变；
“x”表示未知；
“-”表示未定义。

24. 输入 / 输出端口

芯群单片机的输入 / 输出口控制具有很大的灵活性。大部分引脚可在用户程序控制下被设定为输入或输出。所有引脚的上拉电阻设置以及指定引脚的唤醒设置也都由软件控制，这些特性也使得此类单片机在广泛应用上都能符合开发的需求。

该单片机提供 PA~PC 双向输入 / 输出口。这些寄存器在数据存储器有特定的地址。所有 I/O 口用于输入输出操作。作为输入操作，输入引脚无锁存功能，也就是说输入数据必须在执行“MOV A, [m]”，T2 的上升沿准备好，m 为端口地址。对于输出操作，所有数据都是被锁存的，且保持不变直到输出锁存被重写。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PB	—	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	—	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBPU	—	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PC	—	—	—	—	—	PC2	PC1	PC0
PCC	—	—	—	—	—	PCC2	PCC1	PCC0
PCPU	—	—	—	—	—	PCPU2	PCPU1	PCPU0

“—”：未定义，读为“0”

I/O 逻辑功能寄存器列表

24.1 上拉电阻

许多产品应用在端口处于输入状态时需要外加一个上拉电阻来实现上拉的功能。为了免去外部上拉电阻，当引脚规划为数字输入时，可由内部连接到一个上拉电阻。这些上拉电阻可通过相关上拉电阻控制寄存器来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。

需要注意的是，当 I/O 引脚设为数字输入或 NMOS 输出时，上拉功能才会受 PxPU 控制开启，其它状态下上拉功能不可用。

- **PxPU 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PxPU7	PxPU6	PxPU5	PxPU4	PxPU3	PxPU2	PxPU1	PxPU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PxPUn：I/O Px 口上拉电阻控制位
0：除能
1：使能

PxPUn 位用于控制对应引脚的上拉电阻功能。这里的 x 可以是端口 A、B 或 C。但是，每个 I/O 端口实际有效位可能不同。

24.2 PA 口唤醒

当使用暂停指令“HALT”迫使单片机进入休眠或空闲模式，单片机的系统时钟将会停止以降低功耗，此功能对于电池及低功耗应用很重要。唤醒单片机有很多种方法，其中之一就是使 PA 口的其中一个引脚从高电平转为低电平。这个功能特别适合于通过外部开关来唤醒的应用。PA 口的每个引脚可以通过设置 PAWU 寄存器来单独选择是否具有唤醒功能。

需要注意的是，只有当引脚功能被设置为通用 I/O 功能输入类型且单片机处于空闲 / 休眠模式时，唤醒功能才会受 PAWU 控制开启，其它状态下此唤醒功能不可用。

- **PAWU 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PAWUn：PA 口引脚唤醒功能控制位
0：除能
1：使能

24.3 输入 / 输出端口控制寄存器

每一个输入 / 输出口都具有各自的控制寄存器，即 PAC~PCC，用来控制输入 / 输出状态。从而每个 I/O 引脚都可以通过软件控制，动态的设置为 CMOS 输出或输入。所有的 I/O 端口的引脚都各自对应于 I/O 端口控制的某一位。若 I/O 引脚要实现输入功能，则对应的控制寄存器的位需要设置为“1”。这时程序指令可以直接读取输入脚的逻辑状态。若控制寄存器相应的位被设定为“0”，则此引脚被设置为 CMOS 输出。当引脚设置为输出状态时，程序指令读取的是输出端口寄存器的内容。注意，如果对输出口做读取动作时，程序读取到的是内部输出数据锁存器中的状态，而不是输出引脚上实际的逻辑状态。

- **PxC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PxC7	PxC6	PxC5	PxC4	PxC3	PxC2	PxC1	PxC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

PxCn：I/O Px 口引脚类型选择位
0：输出
1：输入

PxCn 位用于控制对应引脚的状态类型。这里的 x 可以是端口 A、B 或 C。但是，每个 I/O 端口实际有效位可能不同。

24.4 输入 / 输出端口源电流选择

该单片机的每个引脚都支持不同的源电流驱动能力，通过相应的源电流选择位控制。仅当对应的引脚被设为 CMOS 输出时，其源电流选择位才有效。否则，这些选择位无效。用户可参考输入 / 输出口电气特性章节为不同应用选择所需的源电流。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SLEDC0	SLEDC07	SLEDC06	SLEDC05	SLEDC04	SLEDC03	SLEDC02	SLEDC01	SLEDC00
SLEDC1	—	—	—	—	—	—	SLEDC11	SLEDC10

源电流选择寄存器列表

- **SLEDC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	SLEDC07	SLEDC06	SLEDC05	SLEDC04	SLEDC03	SLEDC02	SLEDC01	SLEDC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **SLEDC07~SLEDC06**：PB6~PB4 源电流选择位
00：源电流 = Level 0 (最小)
01：源电流 = Level 1
10：源电流 = Level 2
11：源电流 = Level 3 (最大)

Bit 5~4 **SLEDC05~SLEDC04**：PB3~PB0 源电流选择位
00：源电流 = Level 0 (最小)
01：源电流 = Level 1
10：源电流 = Level 2
11：源电流 = Level 3 (最大)

Bit 3~2 **SLEDC03~SLEDC02**：PA7~PA4 源电流选择位
00：源电流 = Level 0 (最小)
01：源电流 = Level 1
10：源电流 = Level 2
11：源电流 = Level 3 (最大)

Bit 1~0 **SLEDC01~SLEDC00**：PA3~PA0 源电流选择位
00：源电流 = Level 0 (最小)
01：源电流 = Level 1
10：源电流 = Level 2
11：源电流 = Level 3 (最大)

- **SLEDC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	SLEDC11	SLEDC10
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **SLEDC11~SLEDC10**：PC2~PC0 源电流选择位
00：源电流 = Level 0 (最小)
01：源电流 = Level 1

10：源电流 = Level 2
11：源电流 = Level 3 (最大)

24.5 引脚共用功能

引脚的多功能可以增加单片机应用的灵活性。有限的引脚个数将会限制应用设计，而引脚的多功能将会解决很多此类问题。此外，这些引多功能引脚功能可以通过一系列寄存器进行设定。

24.5.1 引脚共用功能选择寄存器

封装中有限的引脚个数会对某些单片机功能造成影响。然而，引脚功能共用和引脚功能选择，使得小封装单片机具有更多不同的功能。单片机包含端口 A 输出功能选择寄存器，记为 PxSn，以及输入功能选择寄存器，记为 IFS，可以用来选择共用引脚的特定功能。

要注意的最重要一点是，确保所需的引脚共用功能被正确地选择和取消。对于大部分共用功能，要选择所需的引脚共用功能，首先应通过相应的引脚共用控制寄存器正确地选择该功能，然后再配置相应的外围功能设置以使能外围功能。但是，在设置相关引脚控制位时，一些数字输入引脚如 INTn、xTCK 等，与对应的通用 I/O 口共用同一个引脚共用设置选项。要选择这个引脚功能，除了上述的必要的引脚共用控制和外围功能设置外，还必须将其对应的端口控制寄存器位设置为输入。要正确地取消引脚共用功能，首先应除能外围功能，然后再修改相应的引脚共用控制寄存器以选择其它的共用功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAS0	PAS07	PAS06	PAS05	PAS04	PAS03	PAS02	PAS01	PAS00
PAS1	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
PBS0	PBS07	PBS06	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00
PBS1	—	—	PBS15	PBS14	PBS13	PBS12	PBS11	PBS10
PCS0	—	—	—	—	PCS03	PCS02	PCS01	PCS00
IFS	INT1PS	INT0PS	PTPI2PS	PTPI1PS	PTPI0PS	STPIPS	PTCKPS	STCKPS

引脚共用功能选择寄存器列表

- **PAS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAS07	PAS06	PAS05	PAS04	PAS03	PAS02	PAS01	PAS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **PAS07~PAS06**：PA3 引脚共用功能选择
00：PA3/PTPI
01：PA3/PTPI
10：PA3/PTPI
11：PWM4O

Bit 5~4 **PAS05~PAS04**：PA2 引脚共用功能选择
00：PA2
01：PA2
10：PA2
11：PWM3O

Bit 3~2 **PAS03~PAS02**：PA1 引脚共用功能选择
00：PA1/PTPI
01：PA1/PTPI
10：PA1/PTPI
11：PWM2O

Bit 1~0 **PAS01~PAS00**：PA0 引脚共用功能选择
00：PA0/STPI
01：PA0/STPI
10：PA0/STPI
11：STP

- **PAS1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **PAS17~PAS16**：PA7 引脚共用功能选择
00：PA7/PTPI
01：PWM3O
10：PTP
11：AN6

Bit 5~4 **PAS15~PAS14**：PA6 引脚共用功能选择
00：PA6/STCK
01：PA6/STCK
10：PWM4O
11：AN5

Bit 3~2 **PAS13~PAS12**：PA5 引脚共用功能选择
00：PA5
01：PWM4OB
10：AN4
11：VREF

Bit 1~0 **PAS11~PAS10**：PA4 引脚共用功能选择
00：PA4/PTPI/PTCK
01：PA4/PTPI/PTCK
10：PWM3OB
11：AN3

- **PBS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PBS07	PBS06	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **PBS07~PBS06**：PB3 引脚共用功能选择
00：PB3
01：STP
10：SCOM3
11：AN7

Bit 5~4 **PBS05~PBS04**：PB2 引脚共用功能选择
00：PB2/PTPI/STCK
01：PB2/PTPI/STCK
10：PWM2OB
11：AN2

Bit 3~2 **PBS03~PBS02**：PB1 引脚共用功能选择
00：PB1/PTPI/INT1
01：PB1/PTPI/INT1
10：PWM1OB
11：AN1

Bit 1~0 **PBS01~PBS00**：PB0 引脚共用功能选择
00：PB0/PTPI/INT0
01：PB0/PTPI/INT0
10：PWM0OB
11：AN0

- **PBS1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	PBS15	PBS14	PBS13	PBS12	PBS11	PBS10
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 未定义，读为“0”

Bit 5~4 **PBS15~PBS14**：PB6 引脚共用功能选择
00：PB6
01：PB6
10：PWM0O
11：PTP

Bit 3~2 **PBS13~PBS12**：PB5 引脚共用功能选择
00：PB5/PTCK
01：PB5/PTCK
10：PTPB
11：PWM1O

Bit 1~0 **PBS11~PBS10**：PB4 引脚共用功能选择
00：PB4/STPI
01：PB4/STPI
10：SCOM2
11：PWM2O

- **PCS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PCS03	PCS02	PCS01	PCS00
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义，读为“0”

Bit 3~2 **PCS03~PCS02**：PC1 引脚共用功能选择
00：PC1/INT1
01：PC1/INT1
10：PWM1O
11：SCOM1

Bit 1~0 **PCS01~PCS00**：PC0 引脚共用功能选择
00：PC0/INT0
01：PC0/INT0
10：PWM0O
11：SCOM0

- **IFS 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	INT1PS	INT0PS	PTPI2PS	PTPI1PS	PTPI0PS	STPIPS	PTCKPS	STCKPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **INT1PS**：INT1 输入源引脚选择
0：PB1
1：PC1

Bit 6 **INT0PS**：INT0 输入源引脚选择
0：PB0
1：PC0

Bit 5~3 **PTPI2PS~PTPI0PS**：PTPI 输入源引脚选择
000：PA7
001：PB0
010：PB1
011：PB2
100：PA4
101：PA3
110：PA1
111：PC2

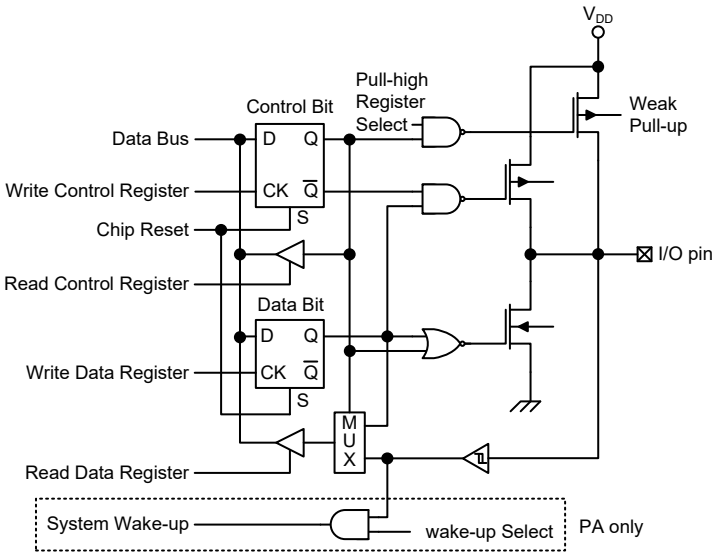
Bit 2 **STPIPS**：STPI 输入源引脚选择
0：PA0
1：PB4

Bit 1 **PTCKPS**：PTCK 输入源引脚选择
0：PA4
1：PB5

Bit 0 **STCKPS**：STCK 输入源引脚选择
0：PB2
1：PA6

24.6 输入 / 输出引脚结构

下图为输入 / 输出引脚逻辑功能的内部结构图。输入 / 输出引脚的准确逻辑结构图可能与此图不同，这里只是为了方便对 I/O 引脚逻辑功能的理解提供一个参考。由于存在诸多的引脚共用结构，在此不方便提供所有类型引脚功能结构图。



逻辑功能输入 / 输出端口结构

24.7 编程注意事项

在编程中，最先要考虑的是端口的初始化。复位之后，所有的输入 / 输出数据及端口控制寄存器都将被设为逻辑高。所有输入 / 输出引脚默认为输入状态，而其电平则取决于其它相连接电路以及是否选择了上拉电阻。如果端口控制寄存器将某些引脚设定为输出状态，这些输出引脚会有初始高电平输出，除非端口数据寄存器在程序中被预先设定。设置哪些引脚是输入及哪些引脚是输出，可通过设置正确的值到对应的端口控制寄存器，或使用指令“SET [m].i”及“CLR [m].i”来设定端口控制寄存器中个别的位。注意，当使用这些位控制指令时，系统即将产生一个读 - 修改 - 写的操作。单片机需要先读入整个端口上的数据，修改个别的位，然后重新把这些数据写入到输出端口。

PA 口的每个引脚都带唤醒功能。单片机处于休眠或空闲模式时，有很多方法可以唤醒单片机，其中之一就是通过 PA 任一引脚电平从高到低转换的方式，可以设置 PA 口一个或多个引脚具有唤醒功能。

25. 定时器模块 – TM

控制和测量时间在任何单片机中都是一个很重要的部分。该单片机提供几个定时器模块 (简称 TM)，来实现和时间有关的功能。定时器模块是包括多种操作的定时单元，提供的操作有：定时 / 计数器，捕捉输入，比较匹配输出，单脉冲输出以及 PWM 输出等功能。每个定时器模块有两个独立中断。每个 TM 外加的输入输出引脚，扩大了定时器的灵活性，便于用户使用。

这里只介绍各种 TM 的共性，更多详细资料请参考标准型和周期型定时器章节。

25.1 简介

该单片机包含两个 TM，每个 TM 可被划分为一个特定的类型，即标准型 TM 和周期型 TM。虽然性质相似，但不同 TM 特性复杂度不同。本章介绍标准型和周期型 TM 的共性，更多详细资料分别见后面各章。两种类型 TM 的特性和区别见下表。

TM 功能	STM	PTM
定时 / 计数器	√	√
捕捉输入	√	√
比较匹配输出	√	√
PWM 输出	√	√
单脉冲输出	√	√
PWM 对齐方式	边沿对齐	边沿对齐
PWM 调节周期 & 占空比	占空比或周期	占空比或周期

TM 功能概要

25.2 TM 操作

不同类型的 TM 提供从简单的定时操作到 PWM 信号产生等多种功能。理解 TM 操作的关键是比较 TM 内独立运行的计数器的值与内部比较器的预置值。当计数器的值与比较器的预置值相同时，则比较匹配，TM 中断信号产生，清零计数器并改变 TM 输出引脚的状态。用户选择内部时钟或外部时钟来驱动内部 TM 计数器。

25.3 TM 时钟源

驱动 TM 计数器的时钟源很多。通过设置 xTM 控制寄存器的 xTCK2~xTCK0 位，选择所需的时钟源，“x”代表 S 或者 P 类型的 TM。该时钟源来自系统时钟 f_{SYS} 的分频比或内部高速时钟 f_H 或 f_{SUB} 时钟源或外部 xTCK 引脚。xTCK 引脚时钟源用于允许外部信号作为 TM 时钟源或用于事件计数。

25.4 TM 中断

标准型 TM 或周期型 TM 都有两个内部中断，分别是内部比较器 A 或比较器 P，当比较匹配发生时产生 TM 中断。当 TM 中断产生时，计数器清零并改变 TM 输出引脚的状态。

25.5 TM 外部引脚

无论哪种类型的 TM，都有两个 TM 输入引脚，分别为 xTCK 和 xTPI。xTM 输入引脚 xTCK 作为 xTM 时钟源输入脚，通过设置 xTMC0 寄存器中的 xTCK2~xTCK0 位进行选择。外部时钟源可通过该引脚来驱动内部 TM。xTCK 输入引脚可选择上升沿有效或下降沿有效。xTCK 引脚还可分别用作 xTM 单脉冲输出模式的外部触发引脚。

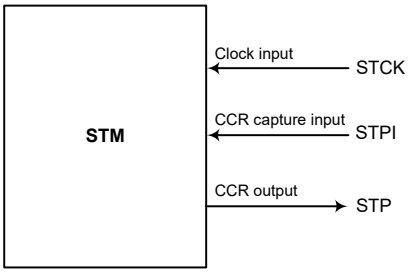
另一种 xTM 输入引脚 xTPI 作为捕捉输入脚，其有效边沿有上升沿、下降沿和双沿，通过设置 xTMC1 寄存器中的 xTIO1~xTIO0 位来选择有效边沿类型。对于周期型 TM，除 PTPI 引脚外，PTCK 引脚也可在 PTM 捕捉输入模式中作为捕捉输入引脚。

每个 TM 都有一个或两个输出引脚 xTP 和 xTPB。当 TM 工作在比较匹配输出模式且比较匹配发生时，xTP 引脚会由 TM 控制切换到高电平或低电平或翻转。xTPB 信号是 xTP 输出的反相信号。xTP 和 xTPB 输出引脚也被 TM 用来产生 PWM 输出波形。

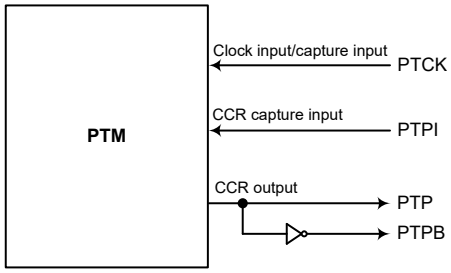
因 TM 输入和输出引脚与其它功能共用，TM 输入和输出功能需要事先通过相关引脚共用功能选择位进行设置。更多引脚共用功能选择详见引脚共用功能章节。

STM		PTM	
输入	输出	输入	输出
STCK, STPI	STP	PTCK, PTPI	PTP, PTPB

TM 外部引脚



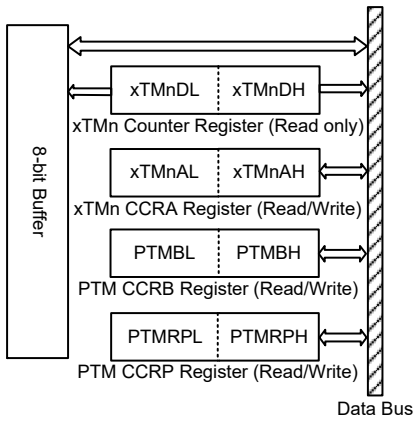
STM 功能引脚方框图



PTM 功能引脚方框图

25.6 编程注意事项

TM 计数寄存器和捕捉 / 比较寄存器 CCRA 和 CCRP 寄存器以及 PTM 的 CCRB 寄存器，都含有低字节和高字节结构。高字节可直接访问，低字节则仅能通过一个内部 8-bit 的缓存器进行访问。读写这些成对的寄存器需通过特殊的方式。值得注意的是 8-bit 缓存器的存取数据及相关低字节的读写操作仅在其相应的高字节读取操作执行时发生。CCRA、CCRB 和 CCRP 寄存器访问方式如下图所示，读写这些成对的寄存器需通过上述的特殊方式。建议使用“MOV”指令按照以下步骤访问 CCRA、CCRB 和 CCRP 低字节寄存器，即 xTMnAL、PTMRBL 和 PTMRPL，否则可能导致无法预期的结果。



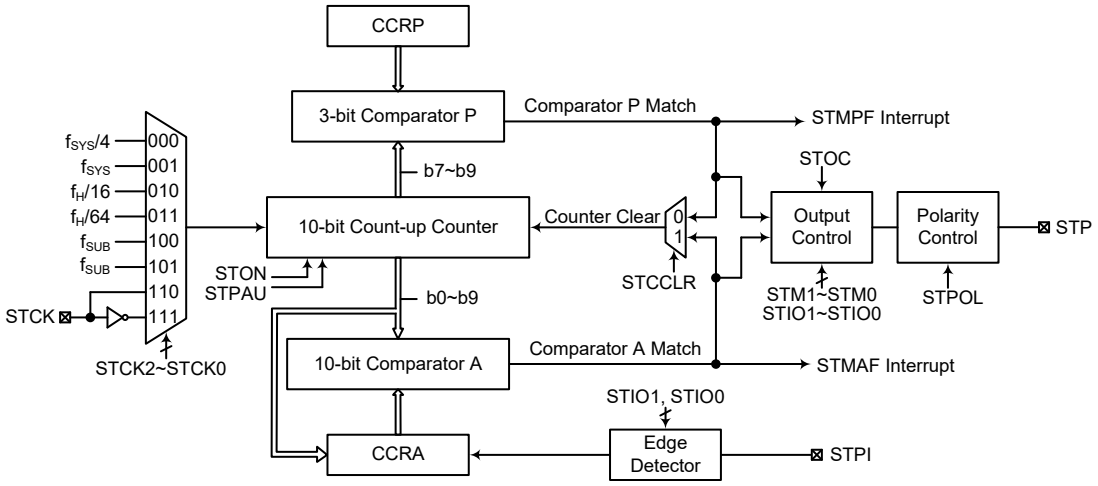
读写流程如下步骤所示：

- 写数据至 CCRA、CCRB 或 CCRP
  - ♦ 步骤 1. 写数据至低字节寄存器 xTMnAL、PTMBL 或 PTMRPL
    - – 注意，此时数据仅写入 8-bit 缓存器。
  - ♦ 步骤 2. 写数据至高字节寄存器 xTMnAH、PTMBH 或 PTMRPH
    - – 注意，此时数据直接写入高字节寄存器，同时锁存在 8-bit 缓存器中的数据写入低字节寄存器。

- 由计数器寄存器、CCRA、CCRB 或 CCRP 中读取数据
  - ♦ 步骤 1. 由高字节寄存器 xTMnDH、xTMnAH、PTMBH 或 PTMRPH 读取数据
    - – 注意，此时高字节寄存器中的数据直接读取，同时由低字节寄存器读取的数据锁存至 8-bit 缓存器中。
  - ♦ 步骤 2. 由低字节寄存器 xTMnDL、xTMnAL、PTMBL 或 PTMRPL 读取数据
    - – 注意，此时读取 8-bit 缓存器中的数据。

26. 标准型 TM – STM

标准型 TM 包括 5 种工作模式，即比较匹配输出、定时 / 事件计数器、捕捉输入、单脉冲输出和 PWM 输出模式。标准型 TM 由两个外部输入脚控制并驱动一个外部输出脚。



注：STM 外部引脚与其它功能共用引脚，因此在使用 STM 之前应该合理配置相关引脚共用功能选择寄存器以确保使能 STM 引脚功能。对于 STCK 和 STPI 输入引脚还需设置相应的端口控制寄存器，将该引脚设置为输入口。

10 位标准型 TM 方框图

26.1 标准型 TM 操作

标准型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 10 位向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 是 3 位宽度，与计数器的高 3 位比较；而 CCRA 是 10 位的，与计数器的所有位比较。

通过应用程序改变 10 位计数器值的唯一方法是使 STON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 STM 中断信号。标准型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。

26.2 标准型 TM 寄存器介绍

标准型 TM 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 10 位计数器的值，一对读 / 写寄存器存放 10 位 CCRA 的值，剩下两个控制寄存器设置不同的操作和控制模式以及 CCRP 的 3 个位。

寄存器名称	位							
	7	6	5	4	3	2	1	0
STMC0	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
STMC1	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
STMDL	D7	D6	D5	D4	D3	D2	D1	D0
STMDH	—	—	—	—	—	—	D9	D8
STMAL	D7	D6	D5	D4	D3	D2	D1	D0
STMAH	—	—	—	—	—	—	D9	D8

10-bit 标准型 TM 寄存器列表

- **STMC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **STPAU**：STM 计数器暂停控制位
0：运行
1：暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，STM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其当前计数值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **STCK2~STCK0**：STM 计数时钟选择位
000：$f_{SYS}/4$
001：f_{SYS}
010：$f_H/16$
011：$f_H/64$
100：f_{SUB}
101：f_{SUB}
110：STCK 上升沿时钟
111：STCK 下降沿时钟
此三位用于选择 STM 的时钟源。外部引脚时钟源能被选择在上升沿或下降沿有效。f_{SYS} 是系统时钟，f_H 和 f_{SUB} 是其它的内部时钟源，细节方面请参考工作模式和系统时钟章节。

Bit 3 **STON**：STM 计数器 On/Off 控制位
0：Off
1：On
此位控制 STM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 STM。清零此位将停止计数器并关闭 STM 减少耗电。当此位经由低到高转换时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其当前计数值，直到此位再次改变为高电平。
若 STM 处于比较匹配输出模式时或 PWM 输出模式或单脉冲输出模式时，当 STON 位经由低到高转换时，STM 输出脚将复位至 STOC 位指定的初始值。

Bit 2~0 **STRP2~STRP0**：STM CCRP 3-bit 寄存器，与 STM 计数器 bit 9 ~ bit 7 比较
比较器 P 匹配周期 =
000：1024 个 STM 时钟
001：128 个 STM 时钟
010：256 个 STM 时钟
011：384 个 STM 时钟
100：512 个 STM 时钟
101：640 个 STM 时钟
110：768 个 STM 时钟
111：896 个 STM 时钟

此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 STCCLR 位设定为 0 时，选中该比较结果清除内部计数器。STCCLR 位设为 0，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较值是 128 时钟周期的倍数。CCRP 被清零时，会使得计数器在最大值溢出。

- **STMC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **STM1~STM0**：STM 工作模式选择位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 输出模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 STM 需要的工作模式。为了确保操作可靠，STM 应在 STM1 和 STM0 位有任何改变前先关掉。在定时 / 计数器模式，STM 输出脚状态未定义。

Bit 5~4 **STIO1~STIO0**：STM 外部引脚功能选择位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 输出模式 / 单脉冲输出模式
00：PWM 输出无效状态
01：PWM 输出有效状态
10：PWM 输出
11：单脉冲输出
捕捉输入模式
00：在 STPI 上升沿输入捕捉
01：在 STPI 下降沿输入捕捉
10：在 STPI 双沿输入捕捉
11：输入捕捉除能
定时 / 计数器模式
未使用
此两位用于决定在满足特定条件时 STM 外部引脚如何改变状态。这两位值的选择取决于 STM 运行在哪种模式下。
在比较匹配输出模式下，STIO1 和 STIO0 位决定当比较器 A 比较匹配输出发生时 STM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 STM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。STM 输出脚的初始值通过 STMC1 寄存器的 STOC 位设置取得。注意，由 STIO1 和 STIO0 位得到的输出电平必须与通过 STOC 位设置的初始值不同，否则当比较匹配发生时，STM 输出脚将不会发生变化。在 STM 输出脚改变状态后，通过 STON 位由低到高电平的转换复位至初始值。
在 PWM 输出模式，STIO1 和 STIO0 用于决定比较匹配条件发生时怎样改变 STM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 STM 关闭时改变 STIO1 和 STIO0 的值是很有必要的。若在 STM 运行时改变 STIO1 和 STIO0 值，PWM 输出的值是无法预料的。

Bit 3 **STOC**：STM STP 输出控制位
比较匹配输出模式
0：初始低
1：初始高

PWM 输出模式 / 单脉冲输出模式
0：低有效
1：高有效
这是 STM 输出脚输出控制位。它取决于 STM 此时正运行于比较匹配输出模式还是 PWM 输出模式，或单脉冲输出模式。若 STM 处于定时 / 计数器模式，则该位无效。在比较匹配输出模式时，其决定比较匹配发生前 STM 输出脚的逻辑电平值。在 PWM 输出模式时，其决定 PWM 信号是高有效还是低有效。在单脉冲输出模式，其决定 STON 位由低变高时 STM 输出脚的逻辑电平。

Bit 2 **STPOL**：STM STP 输出极性控制位
0：同相
1：反相
此位控制 STP 输出脚的极性。此位为高时 STM 输出脚反相，为低时 STM 输出脚同相。若 STM 处于定时 / 计数器模式时该位无效。

Bit 1 **STDPX**：STM PWM 周期 / 占空比控制位
0：CCRP – 周期；CCRA – 占空比
1：CCRP – 占空比；CCRA – 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **STCCLR**：选择 STM 计数器清零条件位
0：比较器 P 匹配
1：比较器 A 匹配
此位用于选择清除计数器的方法。标准型 TM 包括两个比较器 – 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。STCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。STCCLR 位在 PWM 输出模式、单脉冲输出模式和捕捉输入模式时未使用。

- **STMDL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：STM 计数器低字节寄存器 bit 7 ~ bit 0
STM 10-bit 计数器 bit 7 ~ bit 0

- **STMDH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8**：STM 计数器高字节寄存器 bit 1 ~ bit 0
STM 10-bit 计数器 bit 9 ~ bit 8

- **STMAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：STM CCRA 低字节寄存器 bit 7 ~ bit 0
STM 10-bit CCRA bit 7 ~ bit 0

- **STMAH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”
Bit 1~0 **D9~D8**：STM CCRA 高字节寄存器 bit 1 ~ bit 0
STM 10-bit CCRA bit 9 ~ bit 8

26.3 标准型 TM 工作模式

标准型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时 / 计数器模式。通过设置 STMC1 寄存器的 STM1 和 STM0 位选择任意模式。

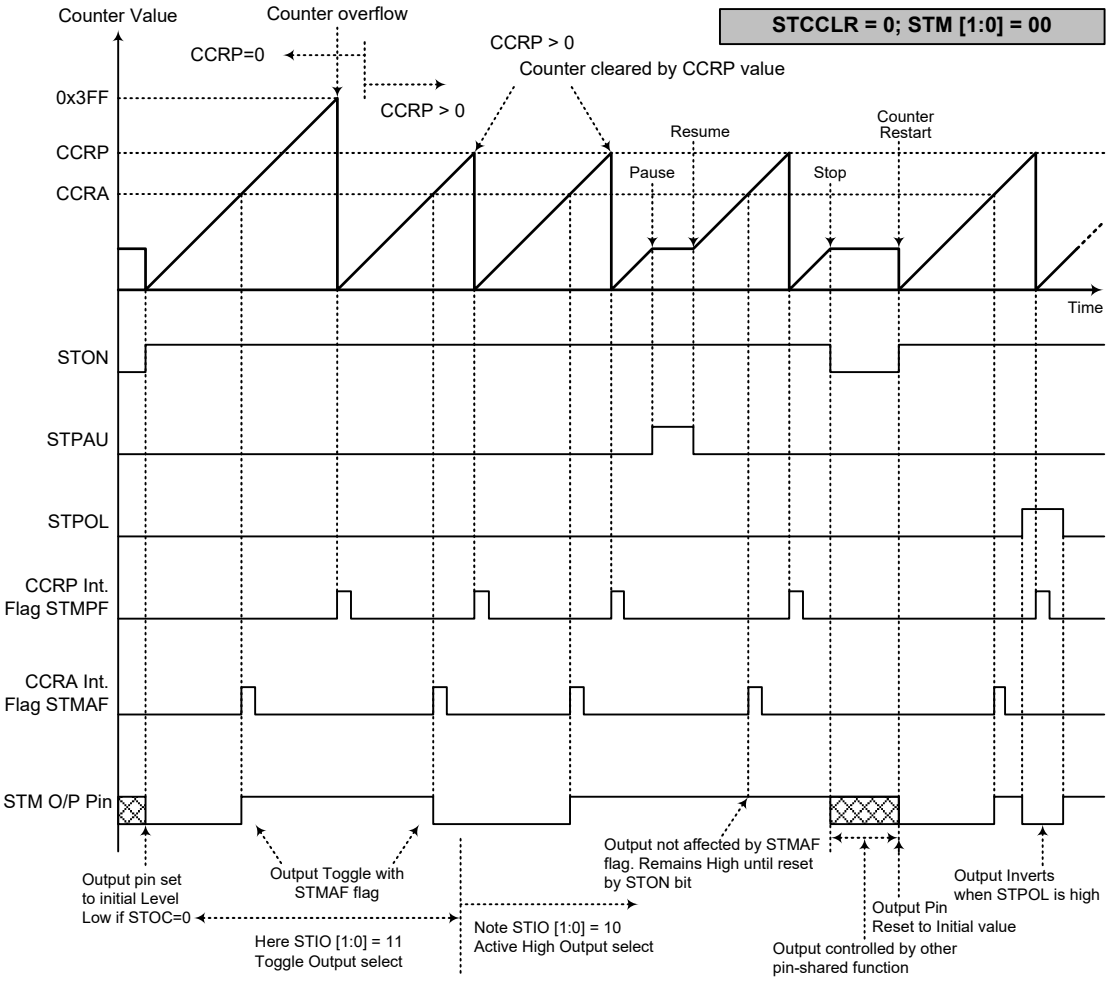
26.3.1 比较匹配输出模式

为使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 STCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 STMAF 和 STMPF 将分别置位。

如果 STMC1 寄存器的 STCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅产生 STMAF 中断请求标志。所以当 STCCLR 为高时，不会产生 STMPF 中断请求标志。在比较匹配输出模式下，CCRA 不能设为“0”。

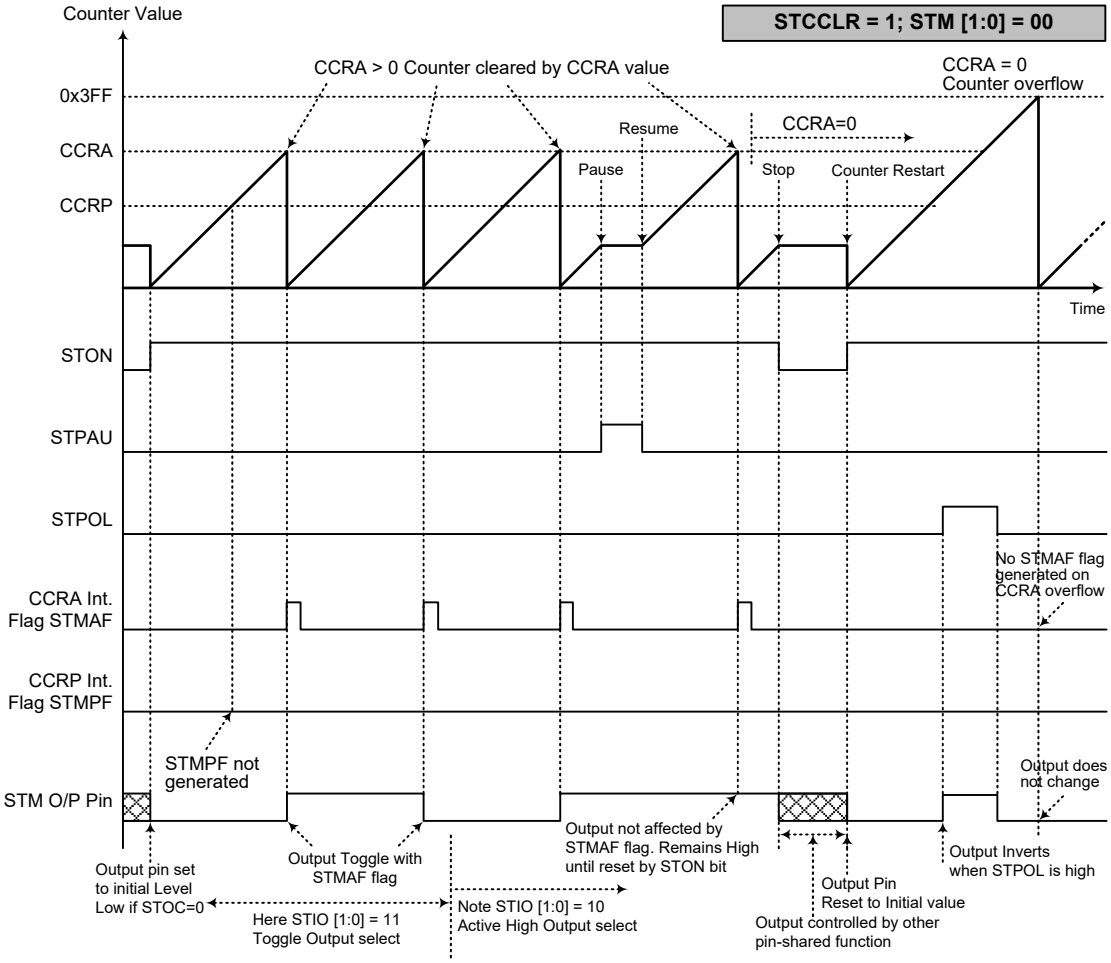
如果 CCRA 位都清除为零，当计数器的值达到 10 位最大值 3FFH 时将溢出，但此时不会产生 STMAF 中断请求标志。

正如该模式名所言，当比较匹配发生后，STM 输出脚状态改变。当比较器 A 比较匹配发生后 STMAF 标志产生时，STM 输出脚状态改变。比较器 P 比较匹配发生时产生的 STMPF 标志不影响 STM 输出脚。STM 输出脚状态改变方式由 STMC1 寄存器中 STIO1 和 STIO0 位决定。当比较器 A 比较匹配发生时，STIO1 和 STIO0 位决定 STM 输出脚输出高，低或翻转当前状态。在 STON 位由低到高后，STM 输出脚初始状态为 STOC 位所指定的电平。注意，若 STIO1 和 STIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – **STCCLR=0**

注：1. STCCLR=0，比较器 P 匹配将清除计数器
2. STM 输出脚仅由 STMAF 标志位控制
3. 在 STON 上升沿 STM 输出脚复位至初始值



比较匹配输出模式 – STCCLR=1

注：1. STCCLR=1，比较器 A 匹配将清除计数器
2. STM 输出脚仅由 STMAF 标志位控制
3. 在 STON 上升沿 STM 输出脚复位至初始值
4. 当 STCCLR=1 时，不会产生 STMPF 标志

26.3.2 定时 / 计数器模式

为使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“11”。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 STM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 STM 输出脚用作普通 I/O 脚或其它功能。

26.3.3 PWM 输出模式

为使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”。STM 的 PWM 输出功能在马达控制，加热控制，照明控制等方面十分有用。给 STM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 输出模式中，STCCLR 位不影响 PWM 周期。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 STMC1 寄存器的 STDPX 位。所以 PWM 波形由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。STMC1 寄存器中的 STOC 位决定 PWM 波形的极性，STIO1 和 STIO0 位使能 PWM 输出或将 STM 输出脚置为逻辑高或逻辑低。STPOL 位对 PWM 输出波形的极性取反。

- **10-bit STM，PWM 输出模式，边沿对齐模式，STDPX=0**

CCRP	1~7	0
Period	CCRP×128	1024
Duty	CCRA	

若 f_{SYS}=16MHz，STM 时钟源选择 $f_{SYS}/4$，CCRP=2，CCRA=256，

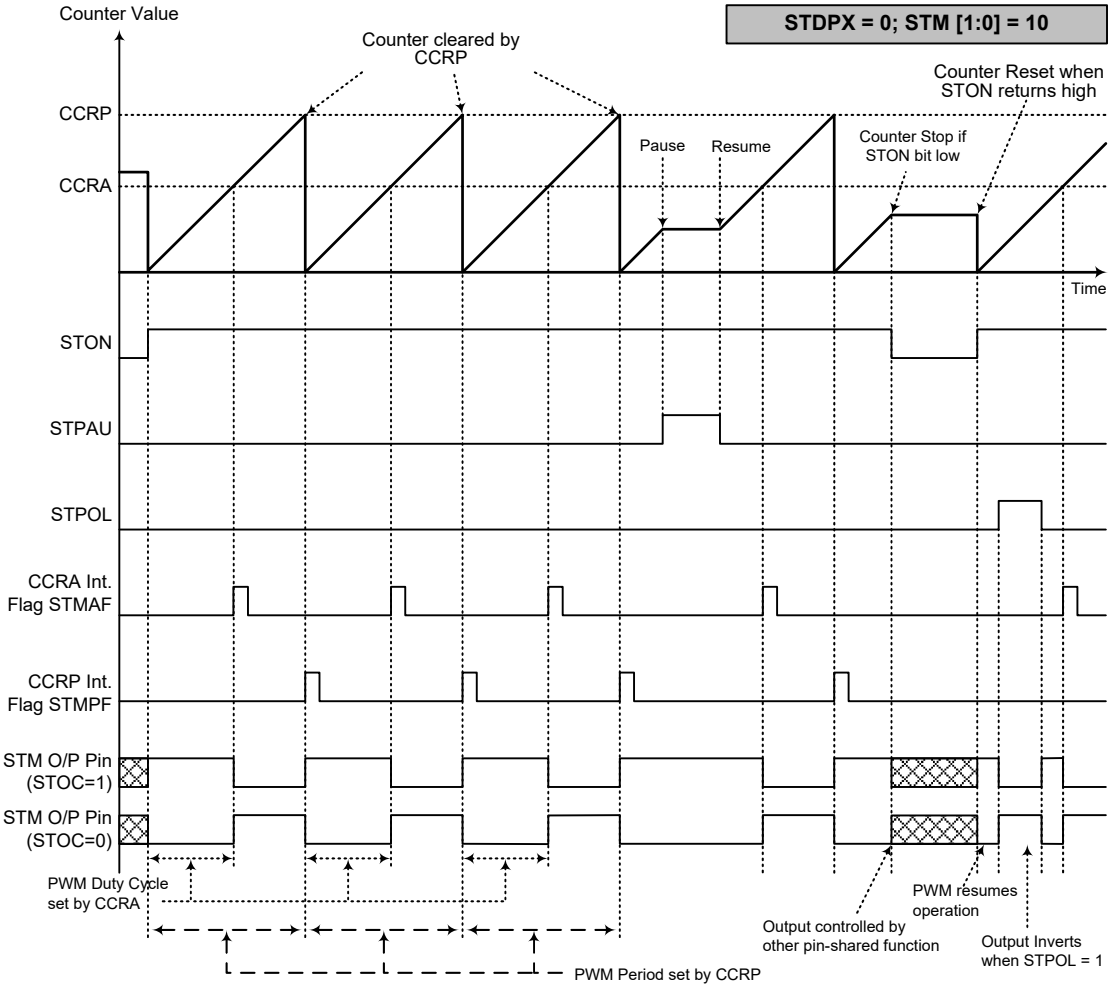
STM PWM 输出频率 = $(f_{SYS}/4)/(2\times256)=f_{SYS}/2048$=7.8125kHz，duty=128/(2×256)=25%，

若由 CCRA 寄存器定义的 Duty 值等于或大于 Period 值，PWM 输出占空比为 100%。

- **10-bit STM，PWM 输出模式，边沿对齐模式，STDPX=1**

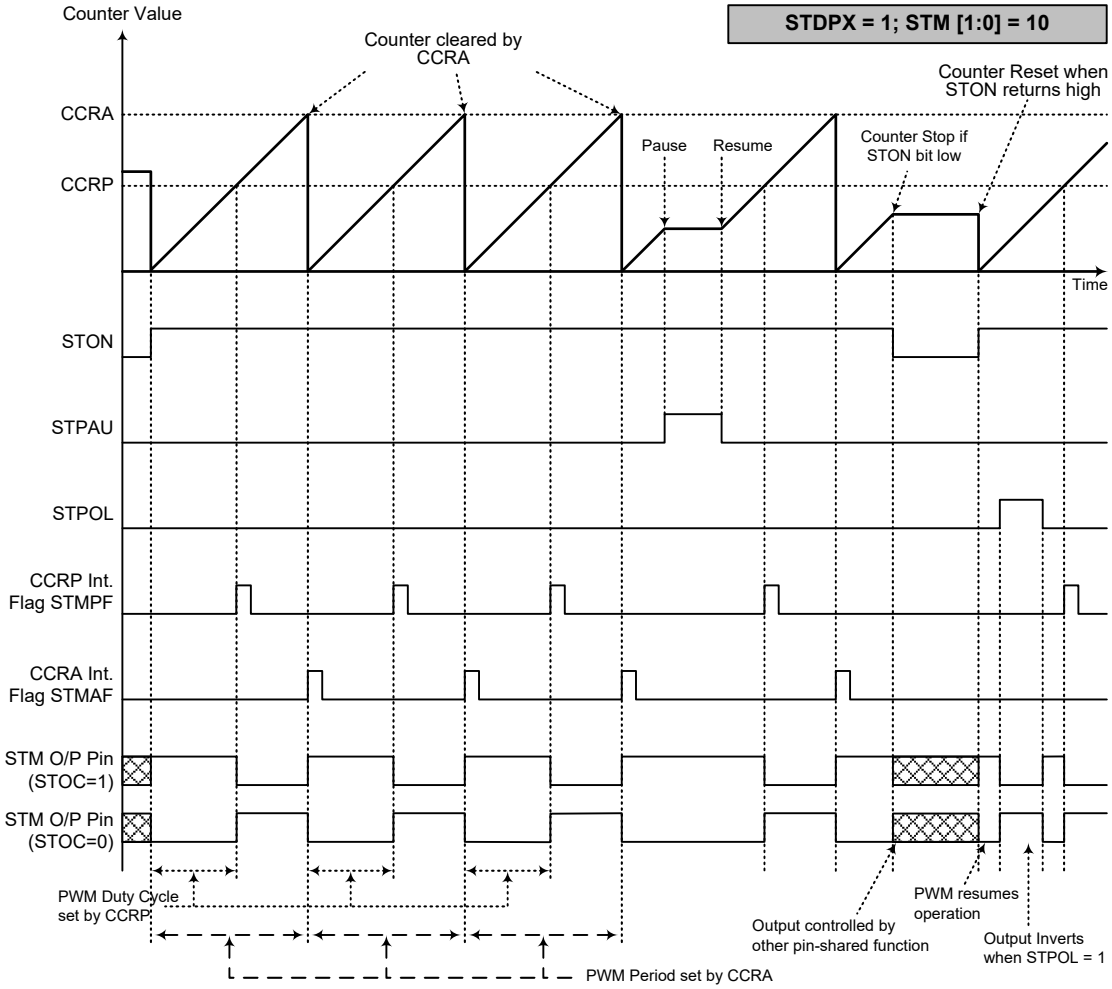
CCRP	1~7	0
Period	CCRA	
Duty	CCRP×128	1024

PWM 的输出周期由 CCRA 寄存器的值与 STM 的时钟共同决定，PWM 的占空比由 CCRP×256 (除了 CCRP 为“0”外) 的值决定。



PWM 输出模式 – STDPX=0

注：1. STDPX=0，CCRP 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 STIO[1:0]=00 或 01，PWM 输出功能不变
4. STCCLR 位不影响 PWM 操作



PWM 输出模式 – STDPX=1

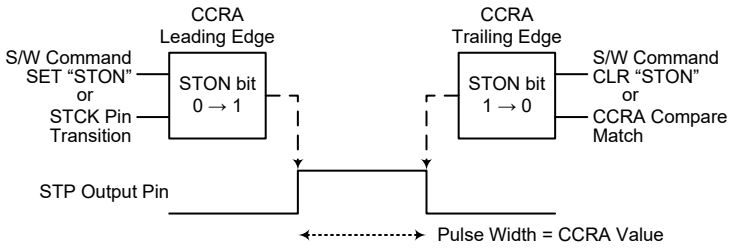
注：1. STDPX=1，CCRA 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 STIO[1:0]=00 或 01，PWM 输出功能不变
4. STCCLR 位不影响 PWM 操作

26.3.4 单脉冲输出模式

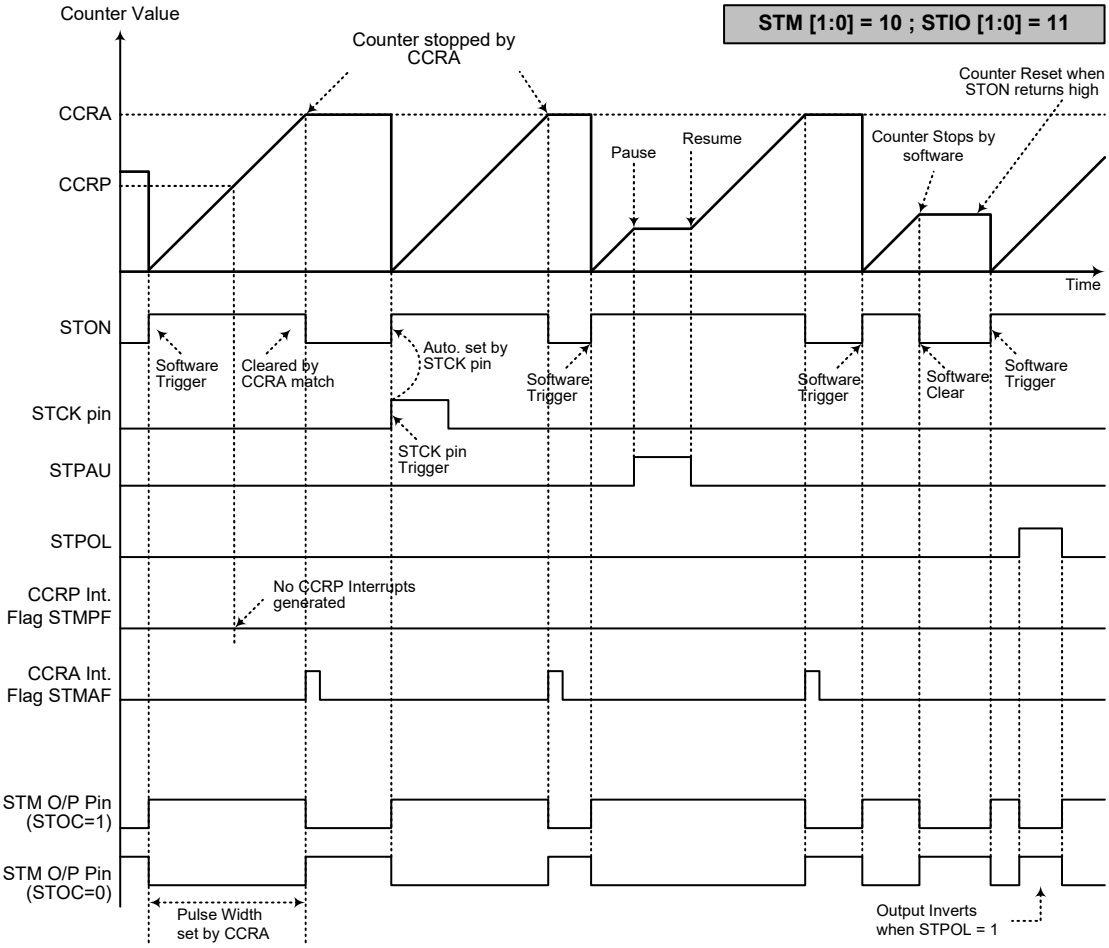
为使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”，同时 STIO1 和 STIO0 位需要设置为“11”。正如模式名所言，单脉冲输出模式，在 STM 输出脚将产生一个脉冲输出。

脉冲输出可以通过应用程序控制 STON 位由低到高的转变来触发脉冲前沿输出。而处于单脉冲输出模式时，STON 位可在 STCK 脚发生有效边沿跳转时自动由低转变为高，进而开始单脉冲输出。当 STON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 STON 位保持高电平。通过应用程序使 STON 位清零或比较器 A 比较匹配发生时，产生脉冲后沿。

然而，比较器 A 比较匹配发生时，会自动清除 STON 位并产生单脉冲输出边沿转换。CCRA 的值通过这种方式控制脉冲宽度。比较器 A 比较匹配发生时，也会产生 STM 中断。STON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲模式中，CCRP 寄存器，STCCLR 和 STDPX 位未使用。



单脉冲产生示意图



单脉冲输出模式

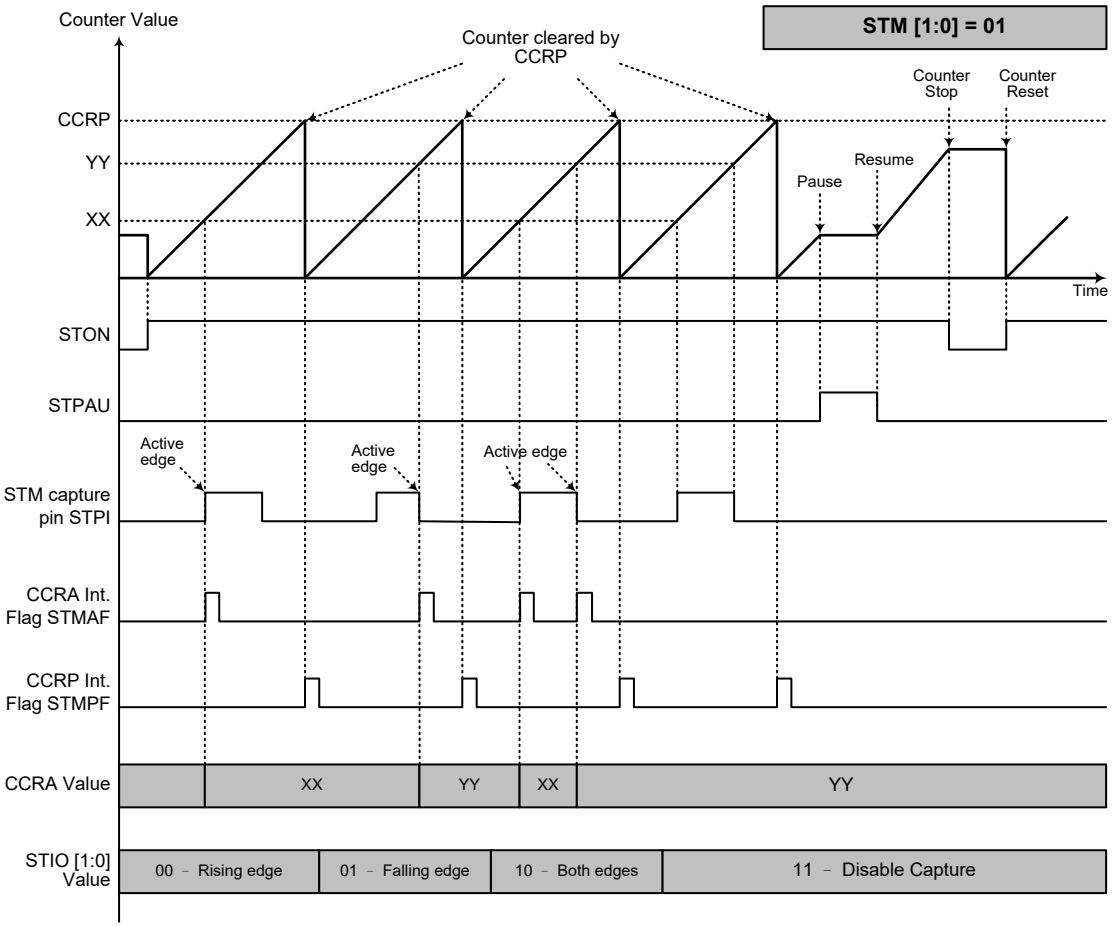
注：1. 通过 CCRA 匹配停止计数器
2. CCRP 未使用
3. 通过 STCK 脚或设置 STON 位为高来触发脉冲
4. STCK 脚有效沿会自动置位 STON
5. 单脉冲输出模式中，STIO[1:0] 需置位“11”，且不能更改

26.3.5 捕捉输入模式

为使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。STPI 脚上的外部信号，通过设置 STMC1 寄存器的 STIO1 和 STIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。通过应用程序将 STON 位由低到高转变时，计数器启动。

当 STPI 脚出现有效边沿转换时，计数器当前值被锁存到 CCRA 寄存器，并产生 STM 中断。无论 STPI 引脚发生哪种边沿转换，计数器将继续工作直到 STON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 STM 中断。记录 CCRP 溢出中断信号的值可以测量脉宽。通过设置 STIO1 和 STIO0 位选择 STPI 引脚为上升沿，下降沿或双沿有效。如果 STIO1 和 STIO0 位都设置为高，无论 STPI 引脚发生哪种边沿转换都不会产生捕捉操作，但计数器仍会继续运行。STCCLR 和 STDPX 位在此模式中未使用。

有几点注意事项须留意。如果捕捉脉宽小于 2 个定时器时钟周期，则可能会被硬件忽略。当计数器的值被有效捕捉边沿锁存到 CCRA 寄存器后，再过 0.5 个定时器时钟周期，STMAF 标志位将被置高。从接收到有效捕捉边沿，到开始将计数器值锁存到 CCRA 寄存器的动作，这之间的延迟时间小于 1.5 个定时器时钟周期。

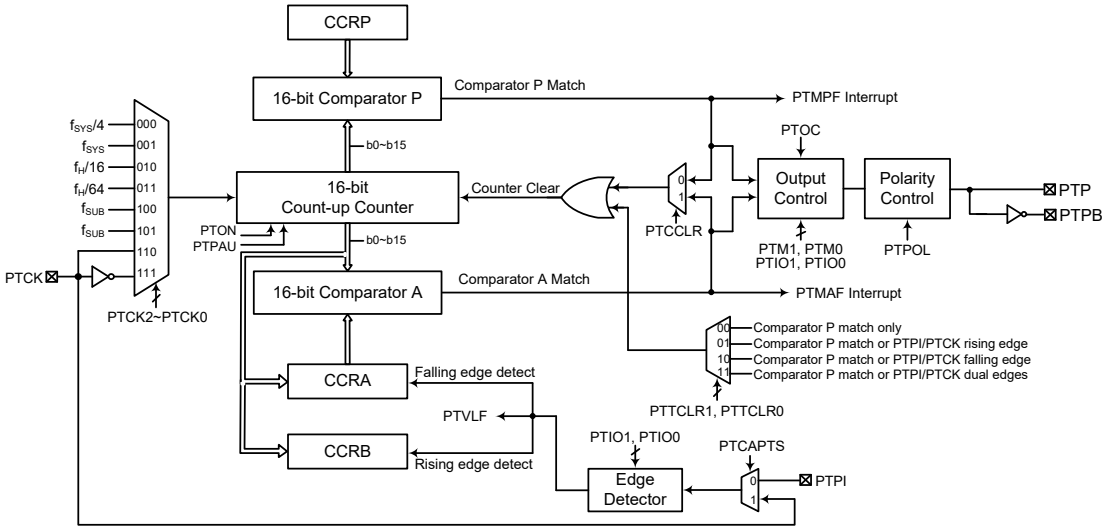


捕捉输入模式

注：1. STM[1:0]=01 并通过 STIO1 和 STIO0 位设置有效边沿
2. STM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 中
3. STCCLR 位未使用
4. 无输出功能 – STOC 和 STPOL 位未使用
5. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大
6. 捕捉输入模式需在有 STM 计数时钟的情况下才可使用

27. 周期型 TM – PTM

周期型 TM 包括 5 种工作模式，即比较匹配输出，定时 / 事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。周期型 TM 由两个外部输入脚控制并驱动两个外部输出脚。



注：1. PTM 外部引脚与其他功能共用引脚，因此在使用 PTM 功能前，应合理设置相关引脚共用功能选择寄存器以选择所需的 PTM 引脚功能。对于 PTCK 和 PTPI 输入引脚还需设置相应的端口控制寄存器，将该引脚设置为输入口。

2. PTPB 是 PTP 输出的反相信号。

16-bit 周期型 TM 框图

27.1 周期型 TM 操作

周期型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 16 位向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 和 CCRA 是 16 位的，与计数器的所有位比较。

通过应用程序改变 16 位计数器值的唯一方法是使 PTON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 PTM 中断信号。周期型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。

27.2 周期型 TM 寄存器介绍

周期型 TM 的所有工作由一系列寄存器控制。一对只读寄存器用来存放 16 位计数器的值，三对读 / 写寄存器存放 16 位 CCRA、CCRP 值和 CCRB 值。剩下三个控制寄存器设置不同的操作和控制模式。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PTMC0	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
PTMC1	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	PTCAPTS	PTCCLR
PTMC2	—	—	—	—	—	PTTCLR1	PTTCLR0	PTVLF
PTMDL	D7	D6	D5	D4	D3	D2	D1	D0
PTMDH	D15	D14	D13	D12	D11	D10	D9	D8
PTMAL	D7	D6	D5	D4	D3	D2	D1	D0

寄存器名称	位							
	7	6	5	4	3	2	1	0
PTMAH	D15	D14	D13	D12	D11	D10	D9	D8
PTMBL	D7	D6	D5	D4	D3	D2	D1	D0
PTMBH	D15	D14	D13	D12	D11	D10	D9	D8
PTMRPL	D7	D6	D5	D4	D3	D2	D1	D0
PTMRPH	D15	D14	D13	D12	D11	D10	D9	D8

16-bit 周期型 TM 寄存器列表

- **PTMC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **PTPAU**：PTM 计数器暂停控制位
0：运行
1：暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，PTM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其当前计数值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **PTCK2~PTCK0**：PTM 计数时钟选择位
000：$f_{SYS}/4$
001：f_{SYS}
010：$f_H/16$
011：$f_H/64$
100：f_{SUB}
101：f_{SUB}
110：PTCK 上升沿时钟
111：PTCK 下降沿时钟
此三位用于选择 PTM 的时钟源。外部引脚时钟源能被选择在上升沿或下降沿有效。f_{SYS} 是系统时钟，f_H 和 f_{SUB} 是其它的内部时钟源，细节方面请参考工作模式和系统时钟章节。

Bit 3 **PTON**：PTM 计数器 On/Off 控制位
0：Off
1：On
此位控制 PTM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 PTM。清零此位将停止计数器并关闭 PTM 减少耗电。当此位经由低到高转换时，内部计数器将复位清零；当此位经由高到低的转换时，内部计数器将保持其当前计数值，直到此位再次改变为高电平。
若 PTM 处于比较匹配输出模式、PWM 输出模式或单脉冲输出模式，当 PTON 位经由低到高的转换时，PTM 输出脚将复位至 PTOC 位指定的初始值。

Bit 2~0 未定义，读为“0”

- **PTMC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	PTCAPTS	PTCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **PTM1~PTM0**：PTM 工作模式选择位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 输出模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 PTM 需要的工作模式。为了确保操作可靠，PTM 应在 PTM1 和 PTM0 位有任何改变前先关掉。在定时 / 计数器模式，PTM 输出脚状态未定义。

Bit 5~4 **PTIO1~PTIO0**：PTM 外部引脚功能选择位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 输出模式 / 单脉冲输出模式
00：PWM 输出无效状态
01：PWM 输出有效状态
10：PWM 输出
11：单脉冲输出
捕捉输入模式
PTTCLR[1:0]=00B：
00：在 PTPI 或 PTCK 上升沿输入捕捉，计数器值将锁存至 CCRA
01：在 PTPI 或 PTCK 下降沿输入捕捉，计数器值将锁存至 CCRA
10：在 PTPI 或 PTCK 双沿输入捕捉，计数器值将锁存至 CCRA
11：输入捕捉除能
PTTCLR[1:0]=01B、10B 或 11B：
00：在 PTPI 或 PTCK 上升沿输入捕捉，计数器值将锁存至 CCRB
01：在 PTPI 或 PTCK 下降沿输入捕捉，计数器值将锁存至 CCRA
10：在 PTPI 或 PTCK 双沿输入捕捉，下降沿计数器值将锁存至 CCRA，上升沿计数器值将锁存至 CCRB
11：输入捕捉除能
定时 / 计数器模式
未使用
此两位用于决定在满足特定条件时 PTM 外部引脚如何改变状态。这两位值的选择取决于 PTM 运行在哪种模式下。
在比较匹配输出模式下，PTIO1 和 PTIO0 位决定当从比较器 A 比较匹配输出发生时 PTM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 PTM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。PTM 输出脚的初始值通过 PTMC1 寄存器的 PTOC 位设置取得。注意，由 PTIO1 和 PTIO0 位得到的输出电平必须与通过 PTOC 位设置的初始值不同，否则当比较匹配发生时，PTM 输出脚将不会发生变化。在 PTM 输出脚改变状态后，通过 PTON 位由低到高电平的转换复位至初始值。
在 PWM 输出模式，PTIO1 和 PTIO0 决定比较匹配条件发生时怎样改变 PTM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 PTM 关闭后才可改变 PTIO1 和 PTIO0 位的值。若在 PTM 运行时改变 PTIO1 和 PTIO0 的值，PWM 输出的值将无法预料。

Bit 3 **PTOC**：PTM PTP 输出控制位
比较匹配输出模式
0：初始低
1：初始高

PWM 输出模式 / 单脉冲输出模式
0：低有效
1：高有效
这是 PTM 输出脚输出控制位。它取决于 PTM 此时正运行于比较匹配输出模式还是 PWM 输出模式 / 单脉冲输出模式。若 PTM 处于定时 / 计数器模式，则其无效。在比较匹配输出模式时，其决定比较匹配发生前 PTM 输出脚的逻辑电平值。在 PWM 输出模式时，其决定 PWM 信号是高有效还是低有效。在单脉冲输出模式，其决定 PTON 位由高变为低时 PTM 输出脚的逻辑电平值。

Bit 2 **PTPOL**：PTM PTP 输出极性控制位
0：同相
1：反相
此位控制 PTP 输出脚的极性。此位为高时 PTM 输出脚反相，为低时 PTM 输出脚同相。若 PTM 处于定时 / 计数器模式时其无效。

Bit 1 **PTDPX**：PTM PWM 周期 / 占空比控制位
0：CCRP – 周期；CCRA – 占空比
1：CCRP – 占空比；CCRA – 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **PTCCLR**：PTM 计数器清零条件选择位
0：PTM 比较器 P 匹配
1：PTM 比较器 A 匹配
此位用于选择清除计数器的方法。周期型 TM 包括两个比较器 – 比较器 A 和比较器 P。这两个比较器每个都可以用于清除内部计数器。PTCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。PTCCLR 位在 PWM 输出模式、单脉冲输出模式或捕捉输入模式时未使用。

- **PTMC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PTTCLR1	PTTCLR0	PTVLF
R/W	—	—	—	—	—	R/W	R/W	R
POR	—	—	—	—	—	0	0	0

Bit 7~3 未定义，读为“0”

Bit 2~1 **PTTCLR1~PTTCLR0**：选择 PTM 计数器清零条件位 (仅适用于捕捉输入模式时)
00：比较器 P 比较匹配
01：比较器 P 比较匹配或 PTCK/PTPI 上升沿
10：比较器 P 比较匹配或 PTCK/PTPI 下降沿
11：比较器 P 比较匹配或 PTCK/PTPI 双沿
注意，PTTCLR1~PTTCLR0 仅在 PTM 处于捕捉输入模式时可用。

Bit 0 **PTVLF**：PTM 计数器值锁存边沿标志位
0：下降沿触发计数器值锁存
1：上升沿触发计数器值锁存
当 PTTCLR1~PTTCLR0 位为 00B，忽略该标志位状态。

- **PTMDL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PTM 计数器低字节寄存器 bit 7 ~ bit 0
PTM 16-bit 计数器 bit 7 ~ bit 0

- **PTMDH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PTM 计数器高字节寄存器 bit 7 ~ bit 0
PTM 16-bit 计数器 bit 15 ~ bit 8

- **PTMAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PTM CCRA 低字节寄存器 bit 7 ~ bit 0
PTM 16-bit CCRA bit 7 ~ bit 0

- **PTMAH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PTM CCRA 高字节寄存器 bit 7 ~ bit 0
PTM 16-bit CCRA bit 15 ~ bit 8

- **PTMBL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PTM CCRB 低字节寄存器 bit 7 ~ bit 0
PTM 16-bit CCRB bit 7 ~ bit 0

- **PTMBH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PTM CCRB 高字节寄存器 bit 7 ~ bit 0
PTM 16-bit CCRB bit 15 ~ bit 8

- **PTMRPL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PTM CCRP 低字节寄存器 bit 7 ~ bit 0
PTM 16-bit CCRP bit 7 ~ bit 0

- **PTMRPH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PTM CCRP 高字节寄存器 bit 15 ~ bit 8
PTM 16-bit CCRP bit 15 ~ bit 8

27.3 周期型 TM 工作模式

周期型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式、捕捉输入模式或定时 / 计数器模式。通过设置 PTMC1 寄存器的 PTM1 和 PTM0 位选择任意模式。

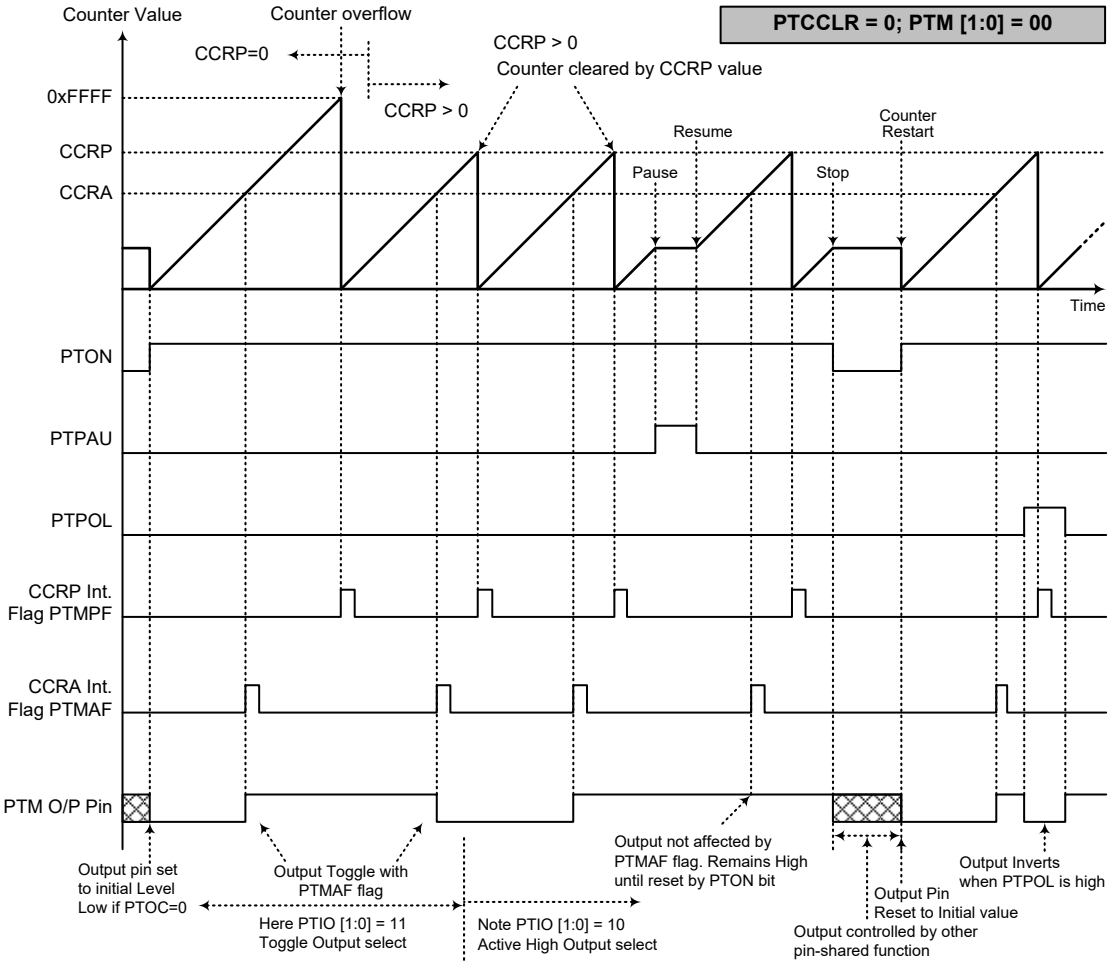
27.3.1 比较匹配输出模式

为使 PTM 工作在此模式，PTMC1 寄存器中的 PTM1 和 PTM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 PTCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 PTMAF 和 PTMPF 将分别置位。

如果 PTMC1 寄存器的 PTCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅产生 PTMAF 中断请求标志。所以当 PTCCLR 为高时，不会产生 PTMPF 中断请求标志。在比较匹配输出模式下，CCRA 不能设为“0”。

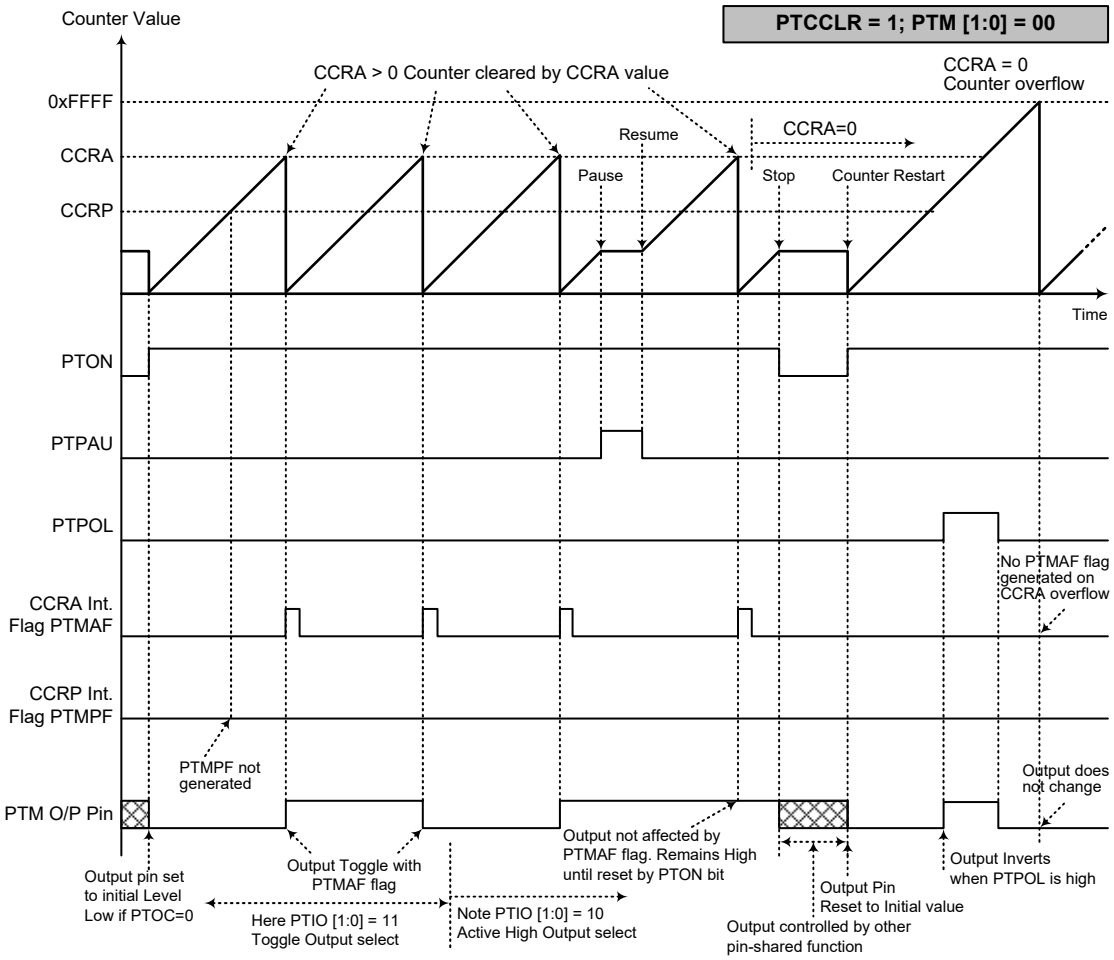
如果 CCRA 位都清除为零，当计数器的值达到 16 位最大值 FFFFH 时将溢出，但此时不会产生 PTMAF 中断请求标志。

正如该模式名所言，当比较匹配发生后，PTM 输出脚状态改变。当比较器 A 比较匹配发生后 PTMAF 标志产生时，PTM 输出脚状态改变。比较器 P 比较匹配发生时产生的 PTMPF 标志不影响 PTM 输出脚。PTM 输出脚状态改变方式由 PTMC1 寄存器中 PTIO1 和 PTIO0 位决定。当比较器 A 比较匹配发生时，PTIO1 和 PTIO0 位决定 PTM 输出脚输出高、低或翻转当前状态。在 PTON 位由低到高电平的变化后，PTM 输出脚初始状态为 PTOC 位所指定的电平。注意，若 PTIO1 和 PTIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – **PTCCLR=0**

注：1. PTCCLR=0，比较器 P 匹配将清除计数器
2. PTM 输出脚仅由 PTMAF 标志位控制
3. 在 PTON 上升沿 PTM 输出脚复位至初始值



比较匹配输出模式 – **PTCCLR=1**

注：1. PTCCLR=1，比较器 A 匹配将清除计数器
2. PTM 输出脚仅由 PTMAF 标志位控制
3. 在 PTON 上升沿 PTM 输出脚复位至初始值
4. 当 PTCCLR=1 时，不会产生 PTMPF 标志

27.3.2 定时 / 计数器模式

为使 PTM 工作在此模式，PTMC1 寄存器中的 PTM1 和 PTM0 位需要设置为“11”。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 PTM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 PTM 输出脚用作普通 I/O 脚或其它功能。

27.3.3 PWM 输出模式

为使 PTM 工作在此模式，PTMC1 寄存器中的 PTM1 和 PTM0 位需要设置为“10”。PTM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 PTM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 输出模式中，PTCCLR 位不影响 PWM 周期。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 PTMC1 寄存器的 PTDPX 位。所以 PWM 波形由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。PTMC1 寄存器中的 PTOC 位决定 PWM 波形的极性，PTIO1 和 PTIO0 位使能 PWM 输出或将 PTM 输出脚置为逻辑高或逻辑低。PTPOL 位对 PWM 输出波形的极性取反。

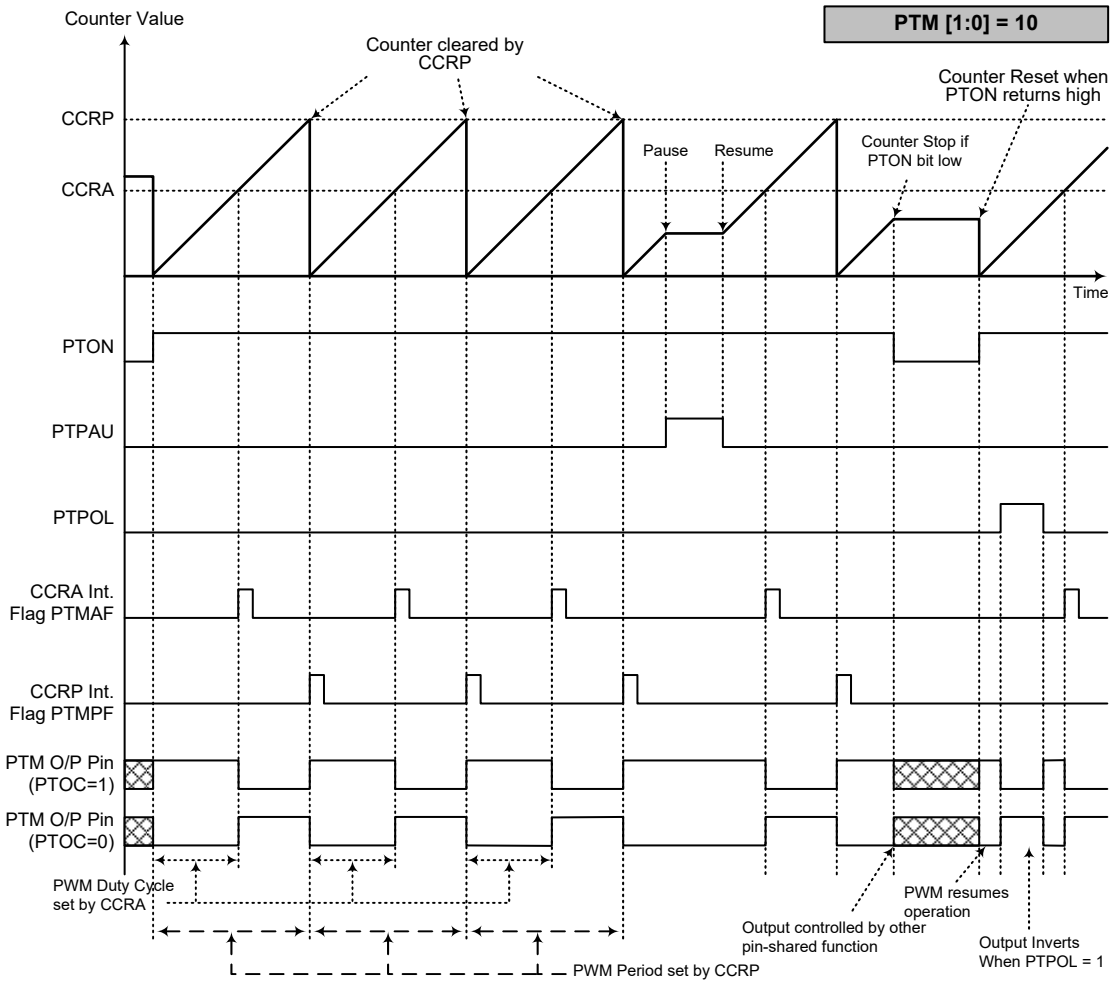
- **16-bit PTM，PWM 输出模式，边沿对齐模式**

CCRP	1~65535	0
Period	1~65535	65536
Duty	CCRA	

若 f_{SYS}=8MHz，PTM 时钟源选择 f_{SYS}/4，CCRP=512，CCRA=128，

PTM PWM 输出频率 = (f_{SYS}/4)/512=f_{SYS}/2048=4kHz，duty=128/512=25%。

若由 CCRA 寄存器定义的 Duty 值等于或大于 Period 值，PWM 输出占空比为 100%。



PWM 输出模式

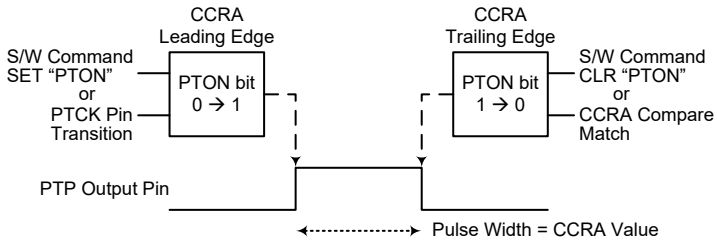
注：1. CCRP 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 PTIO[1:0]=00 或 01，PWM 功能不变
4. PTCCLR 位对 PWM 功能无影响

27.3.4 单脉冲输出模式

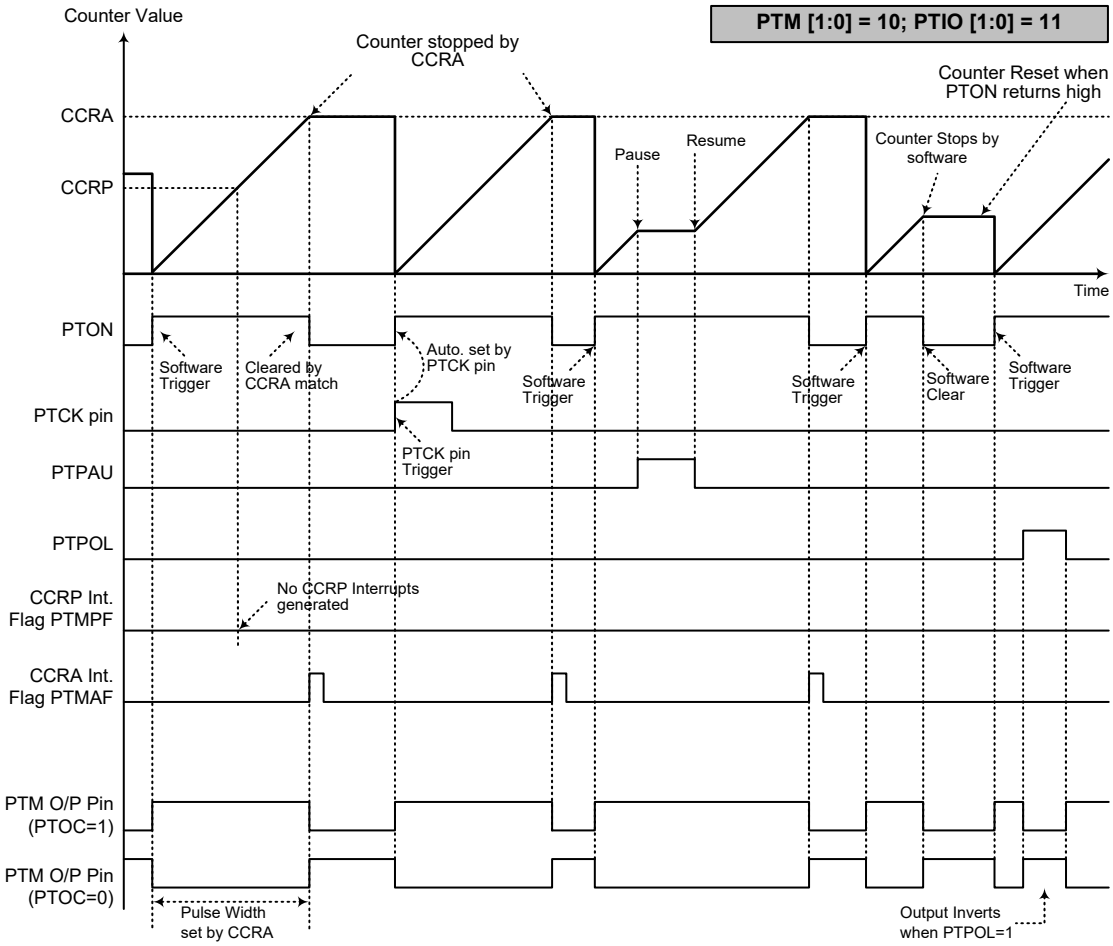
为使 PTM 工作在此模式，PTMC1 寄存器中的 PTM1 和 PTM0 位需要设置为“10”，同时 PTIO1 和 PTIO0 位需要设置为“11”。正如模式名所言，单脉冲输出模式，在 PTM 输出脚将产生一个脉冲输出。

脉冲输出可以通过应用程序控制 PTON 位由低到高的转变来触发。而处于单脉冲输出模式时，PTON 位可在 PTCK 脚发生有效边沿跳转时自动由低转变为高，进而开始单脉冲输出。当 PTON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 PTON 位保持高电平。通过应用程序使 PTON 位清零或比较器 A 比较匹配发生时，产生脉冲后沿。

然而，比较器 A 比较匹配发生时，会自动清除 PTON 位并产生单脉冲输出边沿跳转。CCRA 的值通过这种方式控制脉冲宽度。比较器 A 比较匹配发生时，也会产生 PTM 中断。PTON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲输出模式中，CCRP 寄存器、PTCCLR 和 PTDPX 位未使用。



单脉冲产生示意图



单脉冲输出模式

注：1. 通过 CCRA 匹配停止计数器
2. CCRP 未使用
3. 通过 PTCK 脚或设置 PTON 位为高来触发脉冲
4. PTCK 脚有效沿会自动置高位 PTON
5. 单脉冲输出模式中，PTIO[1:0] 需置位“11”，且不能更改

27.3.5 捕捉输入模式

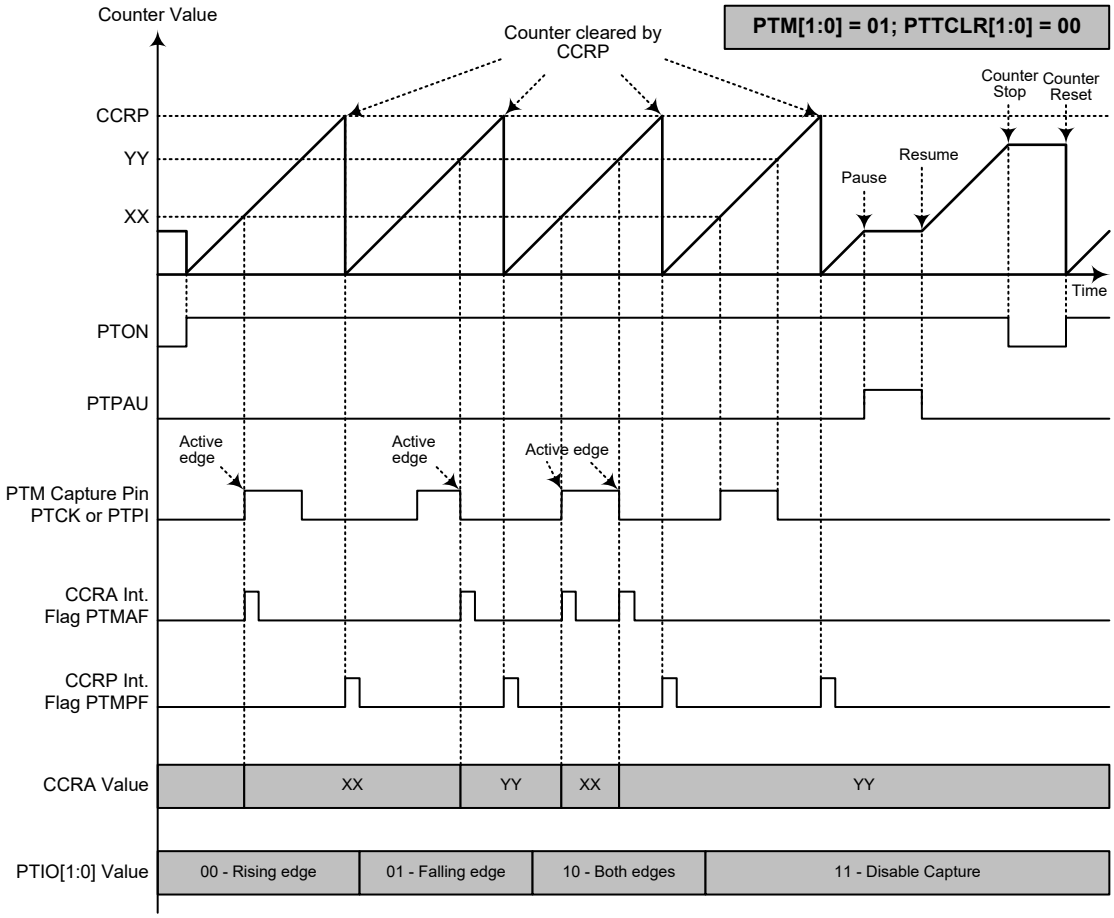
为使 PTM 工作在此模式，PTMC1 寄存器中的 PTM1 和 PTM0 位需要设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。PTPI 或 PTCK 引脚上的外部信号，通过设置 PTMC1 寄存器的 PTCAPTS 位选择。可通过设置 PTMC1 寄存器的 PTIO1 和 PTIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。通过应用程序将 PTON 位由低到高转变时，计数器启动。

PTIO1 和 PTIO0 位决定触发产生中断且被锁存的有效边沿。PTTCLR1 和 PTTCLR0 位决定计数器复位至零的条件。当前计数器值是锁存至 CCRA 还是 CCRB 取决于 PTIO1~PTIO0 位和 PTTCLR1~PTTCLR0 位的设置。PTIO1~PTIO0 位和 PTTCLR1~PTTCLR0 位的设置是相互独立且不相互影响的。

当 PTPI 或 PTCK 引脚出现有效边沿转换时，计数器当前值被锁存到 CCRA 或 CCRB 寄存器，并产生 PTM 中断。无论 PTPI 或 PTCK 引脚发生哪种边沿转换，计数器将继续工作直到 PTON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的

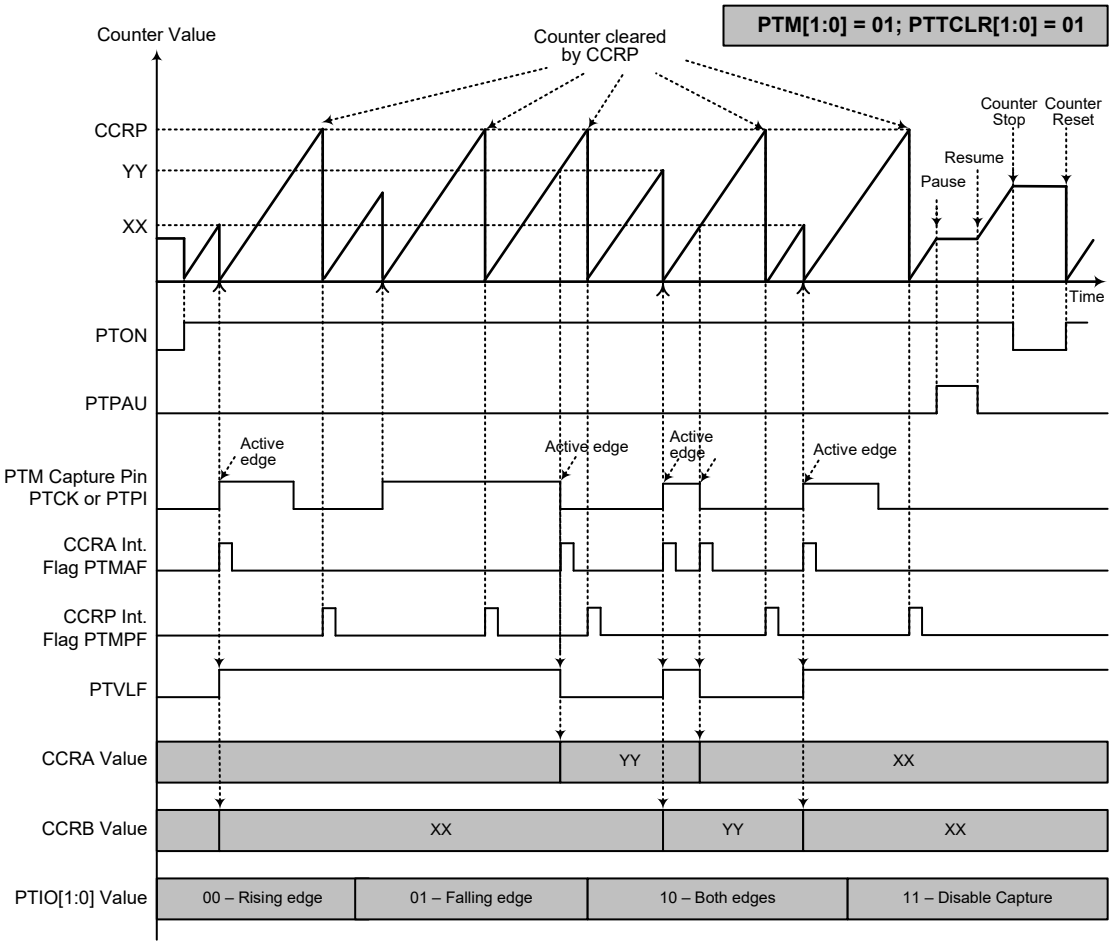
值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 PTM 中断。记录 CCRP 溢出中断信号的值可以测量长脉宽。通过设置 PTIO1 和 PTIO0 位选择 PTPI 或 PTCK 引脚为上升沿，下降沿或双沿有效。如果 PTIO1 和 PTIO0 位都设置为高，无论 PTPI 或 PTCK 引脚发生哪种边沿转换都不会产生捕捉操作，但计数器仍会继续运行。PTCCLR，PTOC 和 PTPOL 位在此模式中未使用。

有几点注意事项须留意。如果 PTCK 用作捕捉输入源，则不能将其选作 PTM 的时钟源。如果捕捉脉宽小于 2 个定时器时钟周期，则可能会被硬件忽略。当计数器的值被有效捕捉边沿锁存到 CCRA 或 CCRB 寄存器后，再过 0.5 个定时器时钟周期，PTMAF 标志位将被置高。从接收到有效捕捉边沿，到开始将计数器值锁存到 CCRA 或 CCRB 寄存器的动作，这之间的延迟时间小于 1.5 个定时器时钟周期。



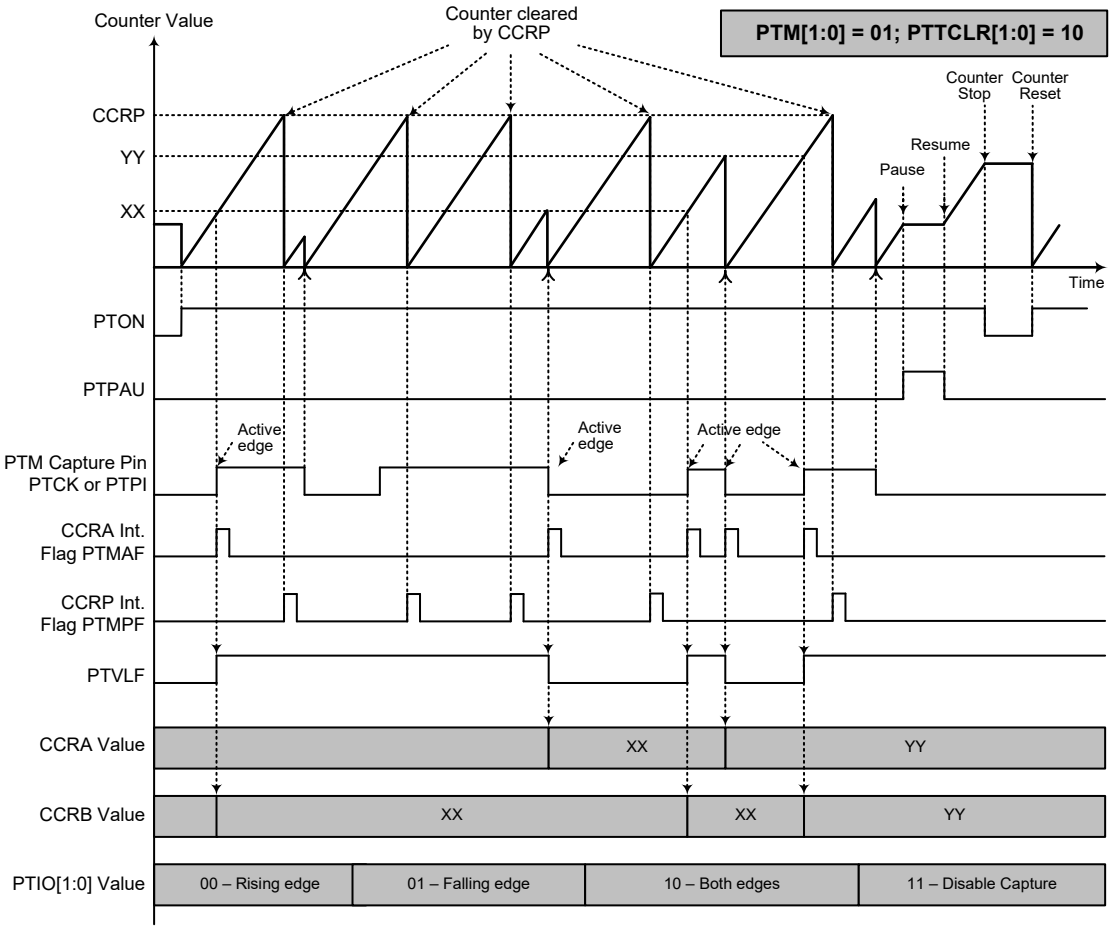
捕捉输入模式 – PTTCLR[1:0]=00

注：1. PTM [1:0]=01，PTTCLR[1:0]=00 并通过 PTIO[1:0] 位设置有效边沿
2. PTM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 中
3. 比较器 P 比较匹配，计数器清零
4. PTCCLR 位未使用
5. 无输出功能 – PTOC 和 PTPOL 位未使用
6. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大
7. 当 PTTCLR[1:0]=00 时忽略 PTVLF 位的状态
8. 捕捉输入模需在有 PTM 计数时钟的情况下才可使用



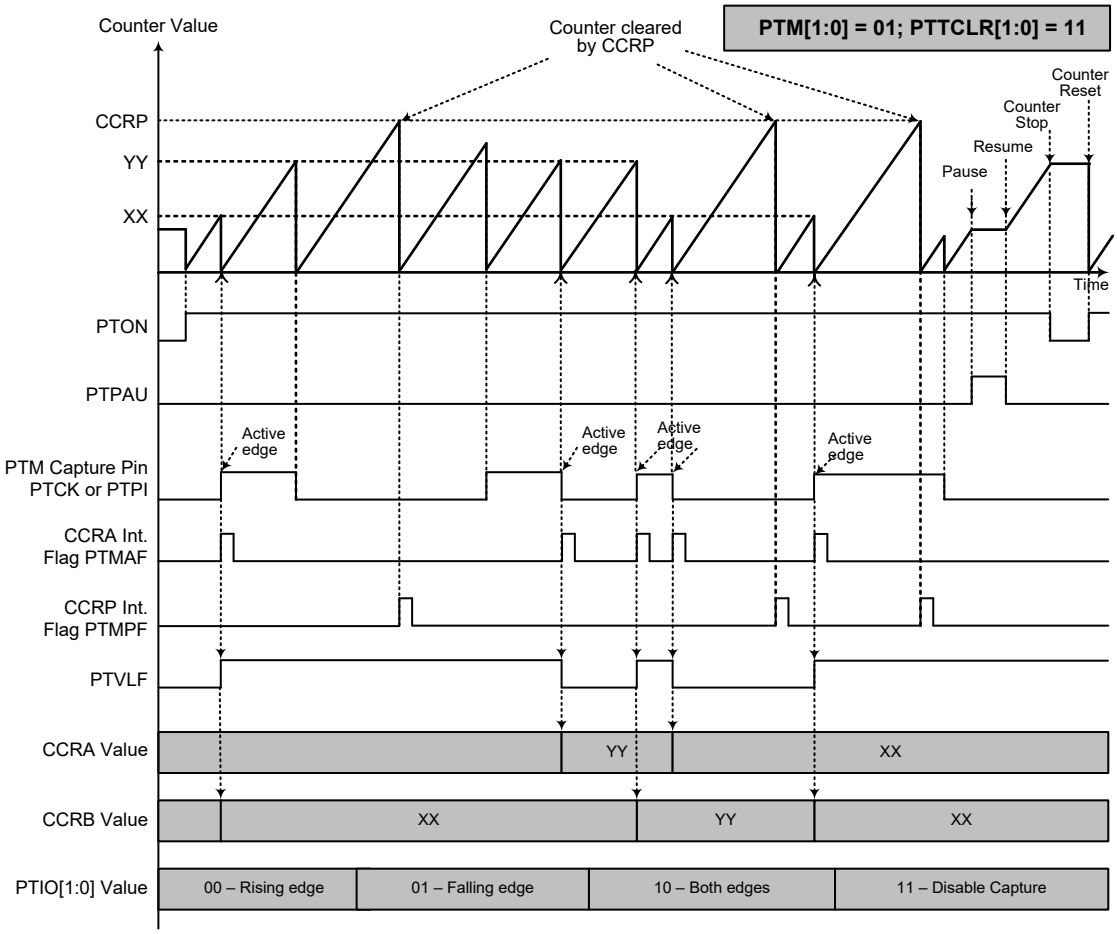
捕捉输入模式 – PTTCLR[1:0]=01

注：1. PTM[1:0]=01，PTTCLR[1:0]=01 并通过 PTIO[1:0] 位设置有效边沿
2. PTM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 或 CCRB 中
3. 比较器 P 比较匹配或 PTM 捕捉输入上升沿时，计数器清零
4. PTCCLR 位未使用
5. 无输出功能 – PTOC 和 PTPOL 位未使用
6. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大
7. 捕捉输入模需在有 PTM 计数时钟的情况下才可使用



捕捉输入模式 – PTTCLR[1:0]=10

注：1. PTM[1:0]=01，PTTCLR[1:0]=10 并通过 PTIO[1:0] 位设置有效边沿
2. PTM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 或 CCRB 中
3. 比较器 P 比较匹配或 PTM 捕捉输入下降沿时，计数器清零
4. PTCCLR 位未使用
5. 无输出功能 – PTOC 和 PTPOL 位未使用
6. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大
7. 捕捉输入模需在有 PTM 计数时钟的情况下才可使用

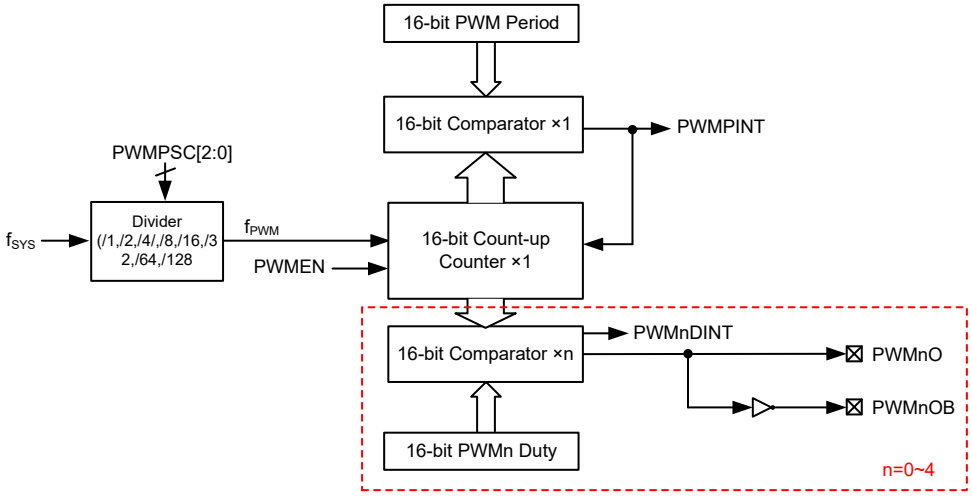


捕捉输入模式 – PTTCLR[1:0]=11

注：1. PTM[1:0]=01，PTTCLR[1:0]=11 并通过 PTIO[1:0] 位设置有效边沿
2. PTM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 或 CCRB 中
3. 比较器 P 比较匹配或 PTM 捕捉输入上升沿或下降沿时，计数器清零
4. PTCCLR 位未使用
5. 无输出功能 – PTOC 和 PTPOL 位未使用
6. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大
7. 捕捉输入模需在有 PTM 计数时钟的情况下才可使用

28. 脉冲宽度调制

该单片机提供一个 16 位的脉冲宽度调制 (PWM) 控制电路。此电路由 PWM 计数器电路所组成，主要是产生 PWM 信号且可线性调整输出占空比。



16-bit PWM 控制电路方框图

28.1 PWM 计数器电路

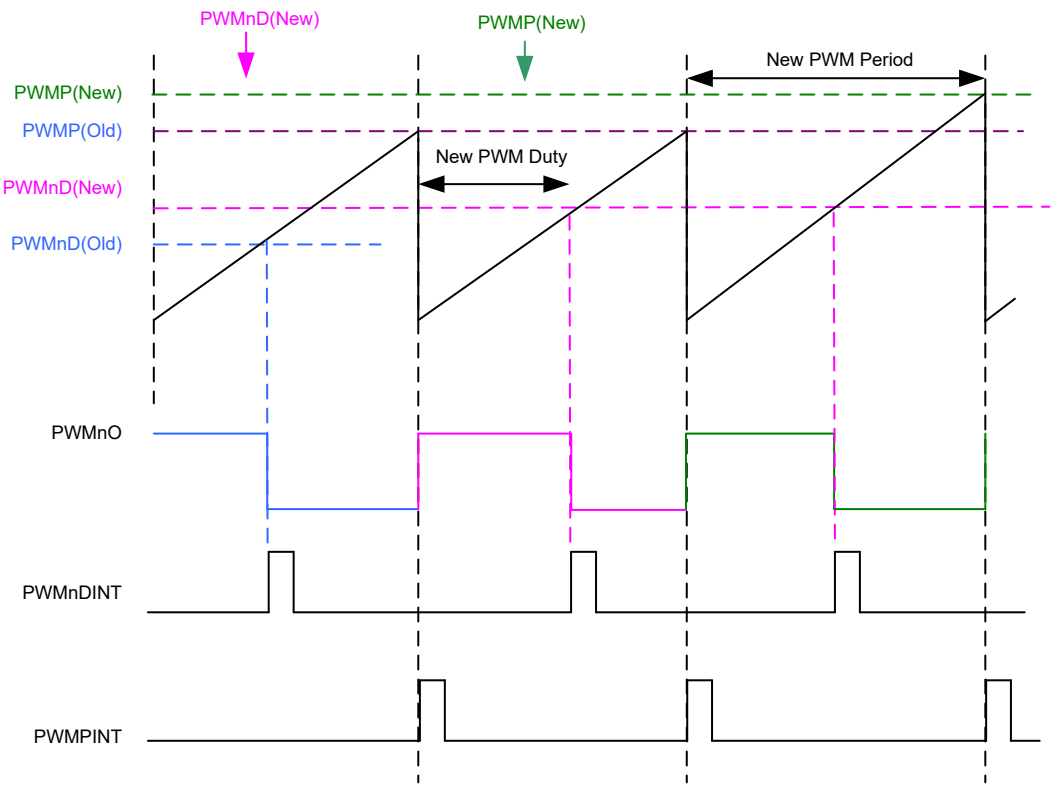
此单片机提供一个 16 位 PWM 发生器。通过给相应的 PWM 寄存器设置特定的 16 位值，PWM 功能可提供占空比和频率可调的波形。

28.2 PWM 寄存器介绍

通过 PWMC 寄存器即可设置 PWM 使能 / 除能以及 PWM 分频选择。通过写入 PWMPL/PWMPH 寄存器与 PWMnDL/PWMnDH 寄存器来设定 PWM 输出的频率与占空比。

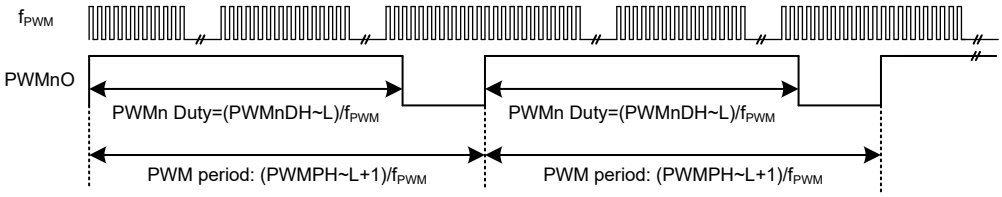
PWMPL/PWMPH 与 PWMnDL/PWMnDH 为 16 位寄存器，写入时需先写入低字节再写入高字节数据才有效，读取时需先读取高字节再读取低字节数据才有效。且写入的 PWMnD/PWMP 会在新的周期才有效。

16 位 PWM 提供 2 种中断信号，分别是 PWMnDINT 与 PWMPINT。当 PWM 计数值与 PWMnD 相同时将会触发 PWMnDINT 中断信号。当 PWM 计数值与 PWMP 相同时将会清除 PWM 计数器同时触发 PWMPINT 中断信号。对应时序可参考下图。



PWM 时序图

PWMnO Period 和 Duty 的计算公式如下：



注：PWMPH 和 PWMPL 寄存器不能同时设置为 00H。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PWMC	PWMEN	—	—	—	—	PWMPSC2	PWMPSC1	PWMPSC0
PWMPL	D7	D6	D5	D4	D3	D2	D1	D0
PWMPH	D15	D14	D13	D12	D11	D10	D9	D8
PWMCL	D7	D6	D5	D4	D3	D2	D1	D0
PWMCH	D15	D14	D13	D12	D11	D10	D9	D8
PWM0DL	D7	D6	D5	D4	D3	D2	D1	D0
PWM0DH	D15	D14	D13	D12	D11	D10	D9	D8
PWM1DL	D7	D6	D5	D4	D3	D2	D1	D0
PWM1DH	D15	D14	D13	D12	D11	D10	D9	D8
PWM2DL	D7	D6	D5	D4	D3	D2	D1	D0
PWM2DH	D15	D14	D13	D12	D11	D10	D9	D8

寄存器名称	位							
	7	6	5	4	3	2	1	0
PWM3DL	D7	D6	D5	D4	D3	D2	D1	D0
PWM3DH	D15	D14	D13	D12	D11	D10	D9	D8
PWM4DL	D7	D6	D5	D4	D3	D2	D1	D0
PWM4DH	D15	D14	D13	D12	D11	D10	D9	D8

PWM 寄存器列表

- **PWMC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PWMEN	—	—	—	—	PWMPSC2	PWMPSC1	PWMPSC0
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 **PWMEN**：PWM 使能控制
0：除能
1：使能

Bit 6~3 未定义，读为“0”

Bit 2~0 **PWMPSC2~PWMPSC0**：PWM 预分频器时钟源 f_{PWM} 选择
000：f_{SYS}
001：$f_{SYS}/2$
010：$f_{SYS}/4$
011：$f_{SYS}/8$
100：$f_{SYS}/16$
101：$f_{SYS}/32$
110：$f_{SYS}/64$
111：$f_{SYS}/128$

- **PWMDnL 寄存器 (n=0~4)**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PWM 占空比低字节寄存器 bit 7 ~ bit 0

- **PWMDnH 寄存器 (n=0~4)**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PWM 占空比高字节寄存器 bit 7 ~ bit 0
PWM Period=(PWMPH~L+1)/f_{PWM} (PWMPH 和 PWMPL 寄存器不能同时设置为 00H)
PWMn Duty=PWMnDH~L/f_{PWM}
注：修改 PWMnDL 和 PWMnDH 会在下一个周期反应在输出的 PWMnO 信号上。

- **PWMPL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PWM 周期低字节寄存器 bit 7 ~ bit 0

- **PWMPH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PWM 周期高字节寄存器 bit 7 ~ bit 0

PWM Period=(PWMPH~L+1)/f_{PWM} (PWMPH 和 PWMPL 寄存器不能同时设置为 00H)

PWMn Duty=PWMnDH~L/f_{PWM}

注：修改 PWMPL 和 PWMPH 会在下一个周期反应在输出的 PWMnO 信号上。

- **PWMCL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**：PWM 计数器低字节寄存器 bit 7 ~ bit 0

- **PWMCH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D15~D8**：PWM 计数器高字节寄存器 bit 7 ~ bit 0

29. A/D 转换器

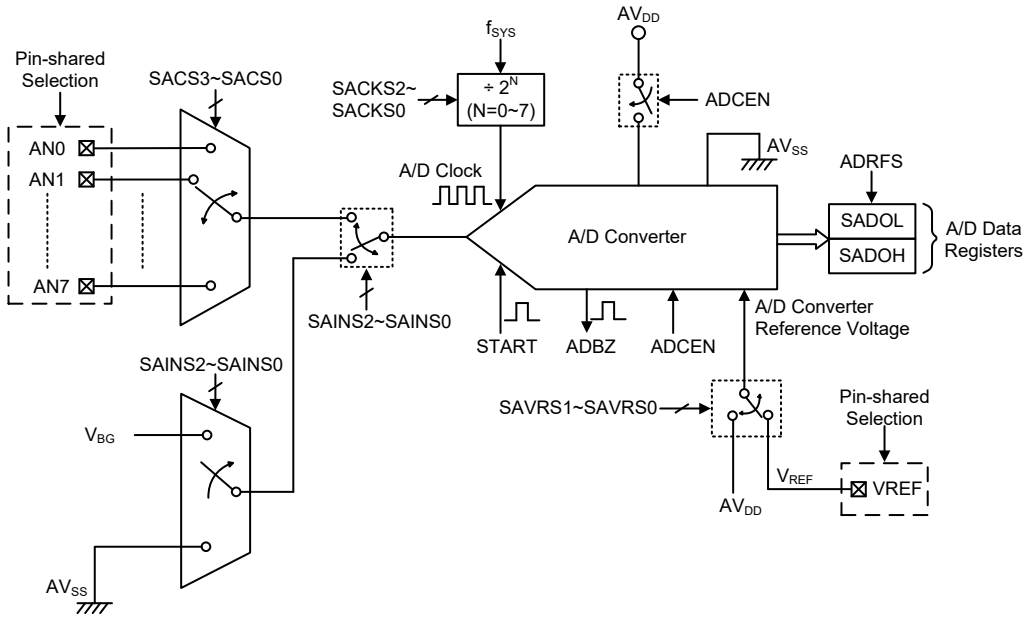
对于大多数电子系统而言，处理现实世界的模拟信号是共同的需求。为了完全由单片机来处理这些信号，首先需要通过 A/D 转换器将模拟信号转换成数字信号。将 A/D 转换器电路集成入单片机，可有效的减少外部器件，随之而来，具有降低成本和减少器件空间需求的优势。

29.1 A/D 简介

该单片机包含一个多通道的 A/D 转换器，它可以直接接入外部模拟信号 (例如传感器或其它控制信号) 或内部模拟信号 (例如 Bandgap 参考电压 V_{BG}) 并直接将这些信号转换成 12 位的数字量。选择转换外部模拟信号由 SAINS2~SAINS0 位和 SACS3~SACS0 位共同控制。应注意的是，若通过 SAINS2~SAINS0 位选择内部模拟信号，则外部通道模拟输入将被自动关闭以避免冲突。关于 A/D 输入信号的详细描述请参考“A/D 转换器输入信号”章节。

外部输入通道	内部信号	A/D 通道选择位
8：AN0~AN7	2：V_{BG}, AV_{SS}	SAINS2~SAINS0, SACS3~SACS0

下图显示了 A/D 转换器内部结构和相关的寄存器。



A/D 转换器结构

29.2 A/D 转换寄存器介绍

A/D 转换器的所有工作由一系列寄存器控制。一对只读寄存器用来存放 12 位 A/D 转换数据的值。两个控制寄存器 SADC0 和 SADC1 用于设置 A/D 转换器的操作和控制功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SADOL (ADRFS=0)	D3	D2	D1	D0	—	—	—	—

寄存器名称	位							
	7	6	5	4	3	2	1	0
SADOL (ADRFS=1)	D7	D6	D5	D4	D3	D2	D1	D0
SADOH (ADRFS=0)	D11	D10	D9	D8	D7	D6	D5	D4
SADOH (ADRFS=1)	—	—	—	—	D11	D10	D9	D8
SADC0	START	ADBZ	ADCEN	ADRFS	SACS3	SACS2	SACS1	SACS0
SADC1	SAINS2	SAINS1	SAINS0	SAVRS1	SAVRS0	SACKS2	SACKS1	SACKS0

A/D 转换器寄存器列表

29.2.1 A/D 转换器数据寄存器 – SADOL, SADOH

对于具有 12 位 A/D 转换器的芯片，需要两个数据寄存器存放转换结果，一个高字节寄存器 SADOH 和一个低字节寄存器 SADOL。在 A/D 转换完毕后，单片机可以直接读取这些寄存器以获得转换结果。由于寄存器只使用了 16 位中的 12 位，其数据存储格式由 SADC0 寄存器的 ADRFS 位控制，如下表所示。D0~D11 是 A/D 换转数据结果位。未使用的位读为“0”。当 A/D 转换器除能时，数据寄存器的内容不变。

ADRFS	SADOH								SADOL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

A/D 数据寄存器

29.2.2 A/D 转换器控制寄存器 – SADC0, SADC1

寄存器 SADC0 和 SADC1 用来控制 A/D 转换器的功能和操作。这些 8 位的寄存器定义包括选择连接至内部 A/D 转换器的模拟通道，数字化数据格式，A/D 时钟源，并控制和监视 A/D 转换器的忙碌状态。由于每个单片机只包含一个实际的模数转换电路，因此这些外部模拟信号中的每一个都需要分别被发送到转换器。SADC0 寄存器中的 SACS3~SACS0 位用于选择哪个外部模拟输入通道被连接到内部 A/D 转换器。SADC1 寄存器中的 SAINS2~SAINS0 位用于选择外部模拟输入通道或内部模拟信号被连接到内部 A/D 转换器。

引脚共用功能选择寄存器的相关位用来定义 I/O 端口中的哪些引脚为 A/D 转换器的模拟输入，哪些引脚不作为 A/D 转换输入。当引脚作为 A/D 输入时，其原来的 I/O 或其它引脚共用功能消失，此外，其内部上拉电阻也将自动断开。

- **SADC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	START	ADBZ	ADCEN	ADRFS	SACS3	SACS2	SACS1	SACS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **START**：启动 A/D 转换位
0→1→0：启动 A/D 转换
此位用于启动 A/D 转换过程。通常此位为低，但如果设为高再被清零，将启动 A/D 转换过程。

Bit 6 **ADBZ**：A/D 转换忙碌标志位
0：A/D 转换结束或未开始转换
1：A/D 转换中
此位用于表明 A/D 转换过程是否完成。当 START 位由低变为高再变为低时，ADBZ 位为高，表明 A/D 转换已开始。A/D 转换结束后，此位被清零。

Bit 5 **ADCEN**：A/D 转换器使能 / 除能控制位
0：除能
1：使能
此位控制 A/D 内部功能。该位被置高将使能 A/D 转换器。如果该位设为低将关闭 A/D 转换器以降低功耗。当 A/D 转换器除能时，A/D 数据寄存器 SADOH 和 SADOL 的值保持不变。

Bit 4 **ADRFS**：A/D 转换数据格式选择位
0：A/D 转换数据格式 → SADOH=D[11:4]，SADOL=D[3:0]
1：A/D 转换数据格式 → SADOH=D[11:8]，SADOL=D[7:0]
此位控制存放在两个 A/D 数据寄存器中的 12 位 A/D 转换结果的格式。细节方面请参考 A/D 数据寄存器章节。

Bit 3~0 **SACS3~SACS0**：A/D 转换器输入通道选择位
0000：AN0
0001：AN1
0010：AN2
0011：AN3
0100：AN4
0101：AN5
0110：AN6
0111：AN7
1000~1111：未定义，输入浮空

- **SADC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	SAINS2	SAINS1	SAINS0	SAVRS1	SAVRS0	SACKS2	SACKS1	SACKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 **SAINS2~SAINS0**：A/D 输入信号选择位
000：外部来源 – 外部模拟通道输入，ANn
001：内部来源 – 内部 Bandgap 参考电压 V_{BG}
010~100：内部来源 – 接地，AV_{SS}
101~111：外部来源 – 外部模拟通道输入，ANn
当选择转换内部模拟信号时，无论 SACS3~SACS0 为何值，外部通道输入信号都会自动关闭。此举预防了外部通道输入与内部模拟信号连接从而导致的不可预期的后果。

Bit 4~3 **SAVRS1~SAVRS0**：A/D 转换器参考电压选择位
00：来自 VREF 引脚
01：来自内部 AV_{DD}
1x：来自 VREF 引脚
这两位用于选择 A/D 转换器参考电压。当选中内部参考电压源时，外部 VREF 引脚的参考电压输入会被自动断开。

Bit 2~0 **SACKS2~SACKS0**：A/D 时钟源选择位
000：保留，不可使用
001：$f_{SYS}/2$
010：$f_{SYS}/4$
011：$f_{SYS}/8$
100：$f_{SYS}/16$
101：$f_{SYS}/32$
110：$f_{SYS}/64$
111：$f_{SYS}/128$

29.3 A/D 转换器操作

SADC0 寄存器中的 START 位，用于打开 A/D 转换器。当单片机设置此位从逻辑低到逻辑高，然后再到逻辑低，就会开始一个模数转换周期。

SADC0 寄存器中的 ADBZ 位用于表明模数转换过程是否正在进行。A/D 转换成功启动后，ADBZ 位会被单片机自动置为“1”。在转换周期结束后，ADBZ 位会自动置为“0”。此外，也会置位中断控制寄存器内相应的 A/D 中断请求标志位，如果中断使能，就会产生对应的内部中断信号。A/D 内部中断信号将引导程序跳转到相应的 A/D 内部中断地址。如果 A/D 内部中断被禁止，可以让单片机轮询 SADC0 寄存器中的 ADBZ 位，检查此位是否被清除，作为另一种侦测 A/D 转换周期结束的方法。

A/D 转换器的时钟源为系统时钟 f_{SYS} 或其分频，而分频系数由 SADC1 寄存器中的 SACKS2~SACKS0 位决定。虽然 A/D 时钟源是由系统时钟 f_{SYS} 和 SACKS2~SACKS0 位决定，但可选择的 A/D 时钟源则有一些限制。在 A/D 转换器电气特性中规定了不同情况下允许的 A/D 时钟周期 t_{ADCK} 的范围，所以在选择系统时钟速度时必须小心。例如，在 V_{DD}=2.0V~5.5V 时 t_{ADCK} 的范围为 0.5μs~10μs，如果系统时钟速度为 8MHz 时，SACKS2~SACKS0 位不能设为“000”，“001”或“111”。必须保证设置的 A/D 转换时钟周期不小于时钟周期的最小值或大于时钟周期的最大值，否则将会产生不准确的 A/D 转换值。使用者可以参考下面的表格，被标上星号 * 的数值是不允许的，因为它们超出了 A/D 转换时钟周期规定的范围。

f_{SYS}	A/D 时钟周期 (t_{ADCK})							
	SACKS[2:0] =000 (f_{SYS})	SACKS[2:0] =001 (f_{SYS}/2)	SACKS[2:0] =010 (f_{SYS}/4)	SACKS[2:0] =011 (f_{SYS}/8)	SACKS[2:0] =100 (f_{SYS}/16)	SACKS[2:0] =101 (f_{SYS}/32)	SACKS[2:0] =110 (f_{SYS}/64)	SACKS[2:0] =111 (f_{SYS}/128)
1MHz	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *	128μs *
2MHz	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *
4MHz	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *
8MHz	125ns *	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *
12MHz	83ns *	167ns *	333ns *	667ns	1.33μs	2.67μs	5.33μs	10.67μs *
16MHz	62.5ns *	125ns *	250ns *	500ns	1μs	2μs	4μs	8μs

A/D 时钟周期范例

SADC0 寄存器中的 ADCEN 位用于控制 A/D 转换电路电源的开启和关闭。该位必须置高以开启 A/D 转换器电源。当设置 ADCEN 位为高开启 A/D 转换器内部电路时，在 A/D 转换成功开启前需一段延时。即使通过相关引脚共用控制位选择无引脚作为 A/D 输入，如果 ADCEN 设为“1”，那么仍然会产生功耗。因此在功耗敏感的应用中，当未使用 A/D 转换器功能时，建议设置 ADCEN 为低以减少功耗。

29.4 A/D 转换器参考电压

A/D 转换器参考电压来自正电源电压 AV_{DD} 或外部参考源引脚 VREF，通过 SAVRS1 和 SAVRS0 位选择。当 SAVRS1~SAVRS0 位为“01”时，A/D 转换器参考电压来自 AV_{DD}。当 SAVRS1~SAVRS0 位为“01”以外的任意值时，A/D 转换器参考电压来自 VREF 引脚。由于 VREF 引脚与其它功能共用，当选择 VREF 引脚作为参考电压源时，需先正确设置引脚共用选择位将 VREF 引脚配置为参考电压输入功能。然而，当内部 A/D 转换器电源被选作参考电压源时，相关的引脚共用控制位不可选择 VREF 参考电压输入功能，避免 VREF 引脚电压跟内部参考电压 AV_{DD} 一起接入 A/D 转换器。模拟输入值一定不能超过所选的参考电压值。

29.5 A/D 转换器输入信号

所有的 A/D 外部模拟输入引脚都与 I/O 口及其它功能共用。使用 PxSn 寄存器中的相

应位，可以将它们设置为 A/D 转换器模拟输入脚或其它共用功能。如果对应的引脚作为 A/D 转换输入，那么它原来的引脚功能将除能。通过这种方式，引脚的功能可由程序来控制，灵活地切换引脚功能。如果将引脚设为 A/D 输入，则通过寄存器编程设置的所有上拉电阻会自动断开。请注意，端口控制寄存器不需要为使能 A/D 输入而先设定为输入模式，当 A/D 输入功能选择位使能 A/D 输入时，端口控制寄存器的状态将被重置。

另外还有内部模拟信号可作为 A/D 转换器的模拟输入信号，来自 Bandgap 参考电压 V_{BG}，通过设置 SAINS2~SAINS0 位来选择。若 SAINS2~SAINS0 位为“000”或“101~111”，则选择转换外部模拟输入信号，具体通道编号由 SACS3~SACS0 位决定。若选择内部模拟信号时，此时无论 SACS 位的值配置为多少，外部通道信号输入将被自动关闭，此举将预防外部通道输入将与内部模拟信号连接从而导致的不可预期的后果。

SAINS[2:0]	SACS[3:0]	输入信号	描述
000, 101~111	0000~0111	AN0~AN7	外部模拟通道输入 ANn
	1000~1111	—	输入浮空，未选择外部通道
001	xxxx	V_{BG}	内部 Bandgap 参考电压
010~100	xxxx	AV_{SS}	接地

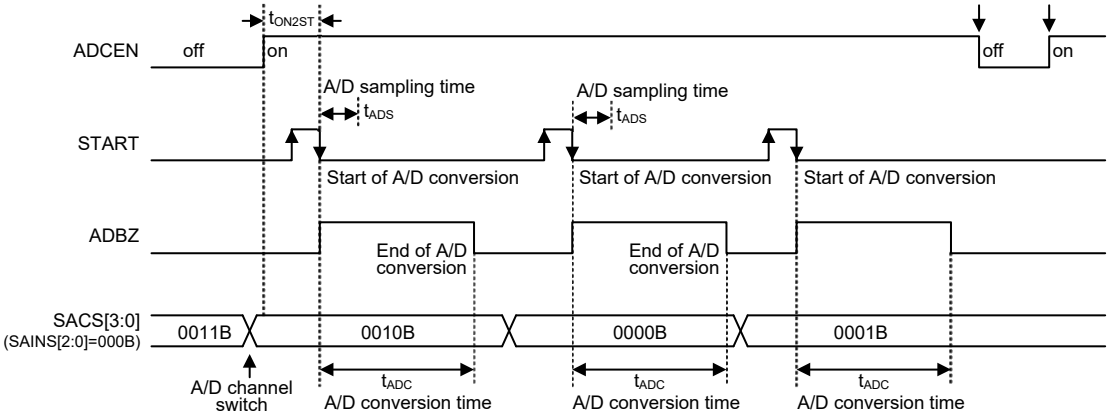
A/D 转换器输入信号选择

29.6 A/D 转换率及时序图

一个完整的 A/D 转换包含两部分，数据采样和数据转换。数据采样时间定义为 t_{ADS}，需要 4 个 A/D 时钟周期，而数据转换需要 12 个 A/D 时钟周期。所以一个完整的 A/D 转换时间 t_{ADC}，一共需要 16 个 A/D 时钟周期。

最大 A/D 转换率 = 1/(A/D 时钟周期 × 16)

下列时序图表示一个外部通道输入信号模数转换过程中不同阶段的图形与时序。由应用程序控制开始 A/D 转换过程后，单片机的内部硬件就会开始进行转换，在这个过程中，程序可以继续其它功能。A/D 转换时间为 $16t_{ADCK}$，t_{ADCK} 为 A/D 时钟周期。



A/D 转换时序图

29.7 A/D 转换步骤

下面概述实现 A/D 转换过程的各个步骤。

- 步骤 1

  通过 SADC1 寄存器中的 SACKS2~SACKS0 位，选择所需的 A/D 转换时钟。

- 步骤 2
  将 SADC0 寄存器中的 ADCEN 位置高使能 A/D 转换器。
- 步骤 3
  通过配置 SAINS2~SAINS0 位，选择连接至内部 A/D 转换器的信号。
  若选择外部通道输入，接着执行步骤 4。
  若选择内部模拟信号，接着执行步骤 5。
- 步骤 4
  若已通过 SAINS2~SAINS0 位选择 A/D 输入信号来自外部通道输入，接着应设置相关的引脚共用控制位将该引脚规划为 A/D 输入引脚。通过设置 SACS3~SACS0 位选择哪个外部通道接至 A/D 转换器。接着执行步骤 6。
- 步骤 5
  选择内部模拟信号前，应正确设置 SACS3~SACS0 位，将外部通道输入切换到无通道输入。然后再设置 SAINS2~SAINS0 位选择所需的内部模拟信号。接着执行步骤 6。
- 步骤 6
  通过 SADC1 寄存器中的 SAVRS1~SAVRS0 位选择参考电压。此步骤之注意事项请参考 A/D 转换器参考电压选择章节。
- 步骤 7
  设置 SADC0 寄存器中的 ADRFS 位选择 A/D 转换器输出数据格式。
- 步骤 8
  如果要使用中断，则中断控制寄存器需要正确地设置，以确保 A/D 中断功能是有效的。总中断控制位 EMI 以及 A/D 转换器中断位 ADE 需要预先置位为“1”。
- 步骤 9
  现在可以通过设置 SADC0 寄存器中的 START 位从“0”到“1”再回到“0”，开始模数转换的过程。
- 步骤 10
  如果 A/D 转换正在进行中，ADBZ 位会被置为逻辑高。A/D 转换完成后，ADBZ 位会被置为逻辑低，并可从 SADOH 和 SADOL 寄存器中读取输出数据。

注：若使用轮询 SADC0 寄存器中 ADBZ 位的状态的方法来检查转换过程是否结束时，则中断使能的步骤可以省略。

29.8 编程注意事项

在编程时，如果 A/D 转换器未使用，通过设置 SADC0 寄存器中的 ADCEN 为低，关闭 A/D 内部电路以减少电源功耗。此时，不考虑输入脚的模拟电压，内部 A/D 转换器电路不产生功耗。如果 A/D 转换器输入脚用作普通 I/O 脚，必须特别注意，输入电压为无效逻辑电平也可能增加功耗。

29.9 A/D 转换功能

单片机含有一组 12 位的 A/D 转换器，它们转换的最大值可达 FFFH。由于模拟输入最大值等于实际 A/D 转换器参考电压值，V_{REF}，因此每一位可表示 $V_{REF}/4096$ 的模拟输入值。

$$1\ LSB=V_{REF}/4096$$

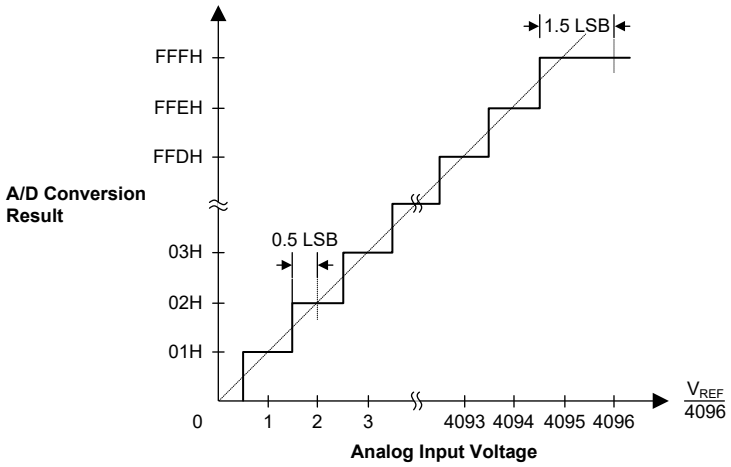
通过下面的等式可估算 A/D 转换器输入电压值：

$$\text{A/D 输入电压} = \text{A/D 数字输出值} \times V_{REF}/4096$$

下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。除了数字化数值

0，其后的数字化数值会在精确点之前的 0.5 LSB 处改变，而数字化数值的最大值将在 V_{REF} 之前的 1.5 LSB 处改变。

注意，这里的 V_{REF} 电压指代的是通过 SAVRS1~SAVRS0 位选择的实际 A/D 转换器参考电压。



理想的 A/D 转换功能

29.10 A/D 转换应用范例

下面两个范例程序用来说明怎样使用 A/D 转换。第一个范例是轮询 SADC0 寄存器中的 ADBZ 位来判断 A/D 转换是否完成；第二个范例则使用中断的方式判断。

范例 1：使用查询 ADBZ 的方式来检测转换结束

```
clr ADE                    ; disable ADC interrupt
mov a, 0BH
mov SADC1, a               ; select input signal from external channel input,
                           ; reference voltage from A/D internal power and fSYS/8 as
                           ; A/D clock
mov a, 03h                 ; set PBS0 register to configure pin AN0
mov PBS0, a
mov a,20H                  ; enable A/D converter and select AN0 as
mov SADC0,a                ; the A/D external channel input
:
start_conversion:
clr START                  ; high pulse on start bit to initiate conversion
set START                  ; reset A/D
clr START                  ; start A/D
:
polling_EOC:
sz  ADBZ                   ; poll the SADC0 register ADBZ bit to detect end of A/D
                           ; conversion
jmp polling_EOC            ; continue polling
:
mov a,SADOL                ; read low byte conversion result value
mov SADOL_buffer,a         ; save result to user defined register
mov a,SADOH                ; read high byte conversion result value
mov SADOH_buffer,a         ; save result to user defined register
:
jmp start_conversion       ; start next A/D conversion
```

范例 2：使用中断的方式来检测转换结束

```
clr ADE                    ; disable ADC interrupt
mov a, 0BH
mov SADC1, a               ; select input signal from external channel input,
                           ; reference voltage from A/D internal power and fSYS/8 as
                           ; A/D clock
mov a, 03h                 ; set PBS0 register to configure pin AN0
mov PBS0, a
mov a,20H                  ; enable A/D converter and select AN0 as
mov SADC0,a                ; the A/D external channel input
:
Start_conversion:
clr START                  ; high pulse on START bit to initiate conversion
set START                  ; reset A/D
clr START                  ; start A/D
clr ADF                    ; clear ADC request flag
set ADE                    ; enable ADC interrupt
set EMI                    ; enable global interrupt
:
:
ADC_ISR:                   ; ADC interrupt service routine
mov acc_stack,a            ; save ACC to user defined memory
mov a,STATUS
mov status_stack,a         ; save STATUS to user defined memory
:
mov a,SADOL                ; read low byte conversion result value
mov SADOL_buffer,a         ; save result to user defined register
mov a,SADOH                ; read high byte conversion result value
mov SADOH_buffer,a         ; save result to user defined register
:
EXIT_INT_ISR:
mov a,status_stack
mov STATUS,a               ; restore STATUS from user defined memory
mov a,acc_stack            ; restore ACC from user defined memory
reti
```

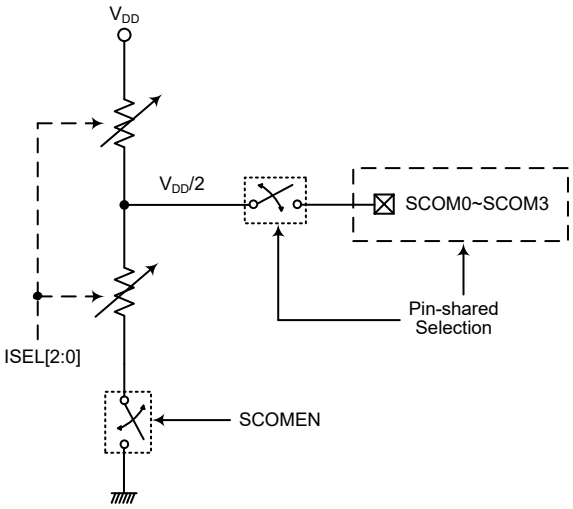
30. 软件控制的 LCD 驱动器

该单片机具有驱动外部 LCD 面板的能力。LCD 驱动的 COM 脚 SCOM0~SCOM3 与 I/O 口共用。LCD 的 COM 控制信号通过应用程序实现。

30.1 LCD 操作

单片机通过设置输入 / 输出口作为 COM 引脚以驱动外部的液晶面板。LCD 驱动功能是由 SCOMC 寄存器来控制，该寄存器除了可设置 LCD 的开启和关闭外还可控制 SCOMn 引脚 R 型偏压电流，使得 LCD COM 驱动产生电压 V_{SS}、$V_{DD}/2$ 和 V_{DD}，从而实现 1/2 偏压操作。

SCOMC 寄存器中的 SCOMEN 位是 LCD 驱动的主控制位，LCD 的 SCOMn 引脚可通过正确配置相关的引脚共用功能选择寄存器用于 LCD 驱动。需注意的是，端口控制寄存器不需要先设置为输出以使能 LCD 驱动操作。



软件控制的 LCD 驱动结构

30.2 LCD 偏压电流控制

LCD COM 驱动器可以提供多种驱动电流选择以适应不同 LCD 面板的需求。通过设置 SCOMC 寄存器中 ISEL2~ISEL0 位可以配置不同的偏压电流。所有的 COM 引脚与 I/O 口共用，通过相应的引脚共用功能选择位将这些引脚选作 SCOM 引脚。

- **SCOMC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	ISEL2	ISEL1	ISEL0	SCOMEN	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7~5 **ISEL2~ISEL0**：SCOM 典型偏压电流选择 (@V_{DD}=5V)
000：25μA
001：50μA
010：100μA
011：200μA
100：5μA
101~111：12.5μA

Bit 4 **SCOMEN**：软件控制 LCD 驱动使能控制位
0：除能
1：使能
当 SCOMEN 位为“1”时，SCOMn 引脚可通过配置相应的引脚共用选择位使能。若 SCOMEN 为“0”，则 SCOMn 的输出固定在 V_{DD} 电平。应注意相应的引脚共用选择位应在 SCOMn 功能使能之前预先合理设置。

Bit 3~0 未定义，读为“0”

31. 低电压检测 – LVD

此单片机具有低电压检测功能，即 LVD。该功能使能用于监测电源电压 V_{DD}，若低于一定值可提供一个警告信号。此功能在电池类产品中非常有用，在电池电压较低时产生警告信号。低电压检测也可产生中断信号。

31.1 LVD 寄存器

低电压检测功能由 LVDC 寄存器控制。VLVD2~VLVD0 位用于选择 8 个固定电压中的一个参考点。LVDO 位被置位时低电压情况发生，若 LVDO 位为低表明 V_{DD} 电压工作在当前所设置低电压水平值之上。LVDEN 位用于控制低电压检测功能的开启 / 关闭，设置此位为高使能此功能，反之，关闭内部低电压检测电路。低电压检测会有一定的功耗，在不使用时可考虑关闭此功能，此举在功耗要求严格的电池供电应用中值得考虑。

- **LVDC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	TLVD1	TLVD0	LVDO	LVDEN	VBGEN	VLVD2	VLVD1	VLVD0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **TLVD1~TLVD0**：产生 LVD 中断的低电压最短保持时间 t_{LVD} 选择
00：(1~2)×t_{LIRC}
01：(3~4)×t_{LIRC}
10：(7~8)×t_{LIRC}
11：(1~2)×t_{LIRC}

Bit 5 **LVDO**：LVD 输出标志位
0：未检测到低电压
1：检测到低电压

Bit 4 **LVDEN**：低电压检测控制位
0：除能
1：使能
在快速模式、低速模式或空闲模式下，LVD 使能 / 除能由 LVDEN 位控制。在休眠模式下，LVD 一直关闭。

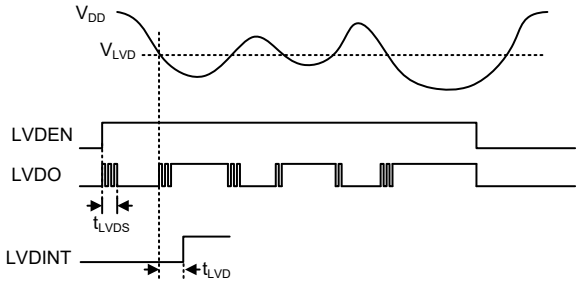
Bit 3 **VBGEN**：Bandgap 缓冲器控制位
0：除能
1：使能
注意，当 LVD 或 LVR 使能或当 VBGEN 置位时，Bandgap 会自动使能。

Bit 2~0 **VLVD2~VLVD0**：LVD 参考点电压选择位
000：1.8V
001：2.0V
010：2.4V
011：2.7V
100：3.0V
101：3.3V
110：3.6V
111：4.0V

31.2 LVD 操作

通过比较电源电压 V_{DD} 与存储在 LVDC 寄存器中的预置参考电压值的结果，低电压检测功能工作。其设置的范围为 1.8V~4.0V。当电源电压 V_{DD} 低于预置电压值时，LVDO 位被置为高，表明低电压产生。当单片机进入休眠模式时，即使 LVDEN 位为高，低电压检测器也会自动除能。低电压检测器使能后，读取 LVDO 位前，电路稳定需要一定的延时

t_{LVDS}。注意，V_{DD} 电压可能上升或下降比较缓慢，在 V_{LVD} 电压值附近时，LVDO 位可能有多种变化。



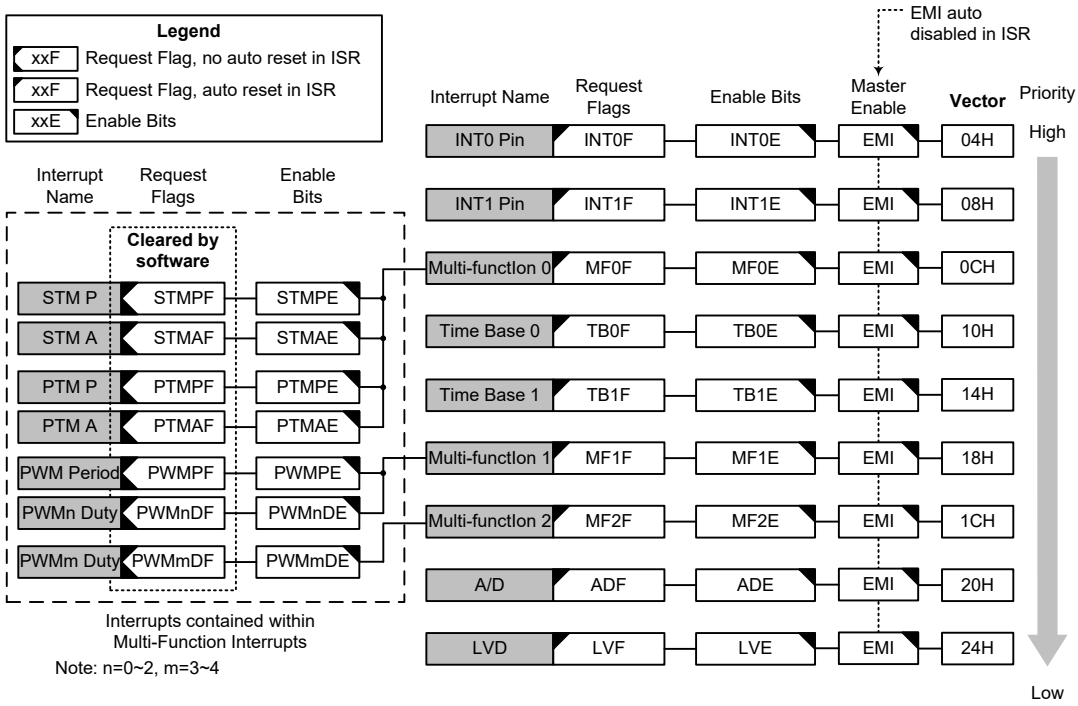
LVD 操作

低电压检测器也有自己的中断功能。它是除了轮询 LVDO 位之外的另一种检测低电压的方法。中断条件产生置位 LVDO 并延时 t_{LVD} 后，中断产生。此种情况下，若 V_{DD} 降至小于 LVD 预置电压值时，中断请求标志位 LVF 将被置位，中断产生，单片机将从空闲模式中被唤醒。若不要求低电压检测的唤醒功能使能，在单片机进入空闲模式前应将 LVF 标志置为高。

32. 中断

中断是单片机一个重要功能。当外部事件或内部功能如发生定时器模块或 A/D 转换器转换结束等，并且产生中断时，系统会暂时中止当前的程序而转到执行相对应的中断服务程序。此单片机提供多个外部中断和内部中断功能，外部中断由 INT0~INT1 引脚动作产生，而内部中断由各种内部功能产生，如定时器模块、时基和 A/D 转换器等等。

各个中断使能位以及相应的请求标志位，以优先级的次序显示在下图。一些中断源有自己的向量，而有些中断则共用多功能中断向量。



中断结构

32.1 中断寄存器

中断控制基本上是在一定单片机条件发生时设置请求标志位，应用程序中中断使能位的设置是通过位于专用数据存储器中的一系列寄存器控制的。寄存器总的分为三类。第一类是 INTC0~INTC2 寄存器，用于设置基本的中断；第二类是 MFI0~MFI2 寄存器，用于设置多功能中断；最后一种有 INTEG 寄存器，用于设置外部中断边沿触发类型。

寄存器中含有中断控制位和中断请求标志位。中断控制位用于使能或除能各种中断，中断请求标志位用于存放当前中断请求的状态。它们都按照特定的模式命名，前面表示中断类型的缩写，紧接着的字母“E”代表使能 / 除能位，“F”代表请求标志位。

功能	使能位	请求标志	注释
总中断	EMI	—	—
INTn 引脚	INTnE	INTnF	n=0~1
时基	TBnE	TBnF	n=0~1
PWM	PWMPE	PWMPF	—
	PWMnDE	PWMnDF	n=0~4
A/D 转换器	ADE	ADF	—
LVD	LVE	LVF	—
多功能中断	MFnE	MFnF	n=0~2
STM	STMPE	STMPF	—
	STMAE	STMAF	—
PTM	PTMPE	PTMPF	—
	PTMAE	PTMAF	—

中断寄存器位命名模式

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	MF0F	INT1F	INT0F	MF0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	TB1F	TB0F	MF2E	MF1E	TB1E	TB0E
INTC2	—	—	LVF	ADF	—	—	LVE	ADE
MFI0	PTMAF	PTMPF	STMAF	STMPF	PTMAE	PTMPE	STMAE	STMPE
MFI1	PWM2DF	PWM1DF	PWM0DF	PWMPF	PWM2DE	PWM1DE	PWM0DE	PWMPE
MFI2	—	—	PWM4DF	PWM3DF	—	—	PWM4DE	PWM3DE

中断寄存器列表

- **INTEG 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4　未定义，读为“0”

Bit 3~2　**INT1S1~INT1S0**：INT1 中断边沿控制位
00：除能
01：上升沿
10：下降沿
11：双沿

Bit 1~0 **INT0S1~INT0S0**：INT0 中断边沿控制位
00：除能
01：上升沿
10：下降沿
11：双沿

- **INTC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	MF0F	INT1F	INT0F	MF0E	INT1E	INT0E	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 未定义，读为“0”

Bit 6 **MF0F**：多功能中断 0 请求标志位
0：无请求
1：中断请求

Bit 5 **INT1F**：INT1 请求标志位
0：无请求
1：中断请求

Bit 4 **INT0F**：INT0 请求标志位
0：无请求
1：中断请求

Bit 3 **MF0E**：多功能中断 0 中断控制位
0：除能
1：使能

Bit 2 **INT1E**：INT1 中断控制位
0：除能
1：使能

Bit 1 **INT0E**：INT0 中断控制位
0：除能
1：使能

Bit 0 **EMI**：总中断控制位
0：除能
1：使能

- **INTC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	MF2F	MF1F	TB1F	TB0F	MF2E	MF1E	TB1E	TB0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **MF2F**：多功能中断 2 请求标志位
0：无请求
1：中断请求

Bit 6 **MF1F**：多功能中断 1 请求标志位
0：无请求
1：中断请求

Bit 5 **TB1F**：时基 1 请求标志位
0：无请求
1：中断请求

Bit 4 **TB0F**：时基 0 请求标志位
0：无请求
1：中断请求

Bit 3 **MF2E**：多功能中断 2 中断控制位
0：除能
1：使能

Bit 2 **MF1E**：多功能中断 1 中断控制位
0：除能
1：使能

Bit 1 **TB1E**：时基 1 中断控制位
0：除能
1：使能

Bit 0 **TB0E**：时基 0 中断控制位
0：除能
1：使能

- **INTC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVF	ADF	—	—	LVE	ADE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 未定义，读为“0”

Bit 5 **LVF**：LVD 请求标志位
0：无请求
1：中断请求

Bit 4 **ADF**：A/D 转换器请求标志位
0：无请求
1：中断请求

Bit 3~2 未定义，读为“0”

Bit 1 **LVE**：LVD 中断控制位
0：除能
1：使能

Bit 0 **ADE**：A/D 转换器中断控制位
0：除能
1：使能

- **MFI0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PTMAF	PTMPF	STMAF	STMPF	PTMAE	PTMPE	STMAE	STMPE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **PTMAF**：PTM 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 6 **PTMPF**：PTM 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 5 **STMAF**：STM 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 4 **STMPF**：STM 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 3 **PTMAE**：PTM 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 2 **PTMPE**：PTM 比较器 P 匹配中断控制位
0：除能
1：使能

Bit 1 **STMAE**：STM 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **STMPE**：STM 比较器 P 匹配中断控制位
0：除能
1：使能

- **MFI1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PWM2DF	PWM1DF	PWM0DF	PWMPF	PWM2DE	PWM1DE	PWM0DE	PWMPE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **PWM2DF**：PWM 占空比 2 中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 6 **PWM1DF**：PWM 占空比 1 中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 5 **PWM0DF**：PWM 占空比 0 中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 4 **PWMPF**：PWM 周期中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 3 **PWM2DE**：PWM 占空比 2 中断控制位
0：除能
1：使能

Bit 2 **PWM1DE**：PWM 占空比 1 中断控制位
0：除能
1：使能

Bit 1 **PWM0DE**：PWM 占空比 0 中断控制位
0：除能
1：使能

Bit 0 **PWMPE**：PWM 周期中断控制位
0：除能
1：使能

- **MFI2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	PWM4DF	PWM3DF	—	—	PWM4DE	PWM3DE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6　未定义，读为“0”

Bit 5　**PWM4DF**：PWM 占空比 4 中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 4　**PWM3DF**：PWM 占空比 3 中断请求标志位
0：无请求
1：中断请求
注意，当中断响应时此位必须通过应用程序清零。

Bit 3~2　未定义，读为“0”

Bit 1　**PWM4DE**：PWM 占空比 4 中断控制位
0：除能
1：使能

Bit 0　**PWM3DE**：PWM 占空比 3 中断控制位
0：除能
1：使能

32.2 中断操作

若中断事件条件产生，如一个 TM 比较器 P、比较器 A 匹配或 A/D 转换结束等等，相关中断请求标志将置起。中断标志产生后程序是否会跳转至相关中断向量执行是由中断使能位的条件决定的。若使能位为“1”，程序将跳至相关中断向量中执行；若使能位为“0”，即使中断请求标志置起中断也不会发生，程序也不会跳转至相关中断向量执行。若总中断使能位为“0”，所有中断都将除能。

当中断发生时，下条指令的地址将被压入堆栈。相应的中断向量地址加载至 PC 中。系统将从此向量取下条指令。中断向量处通常为“JMP”指令，以跳转到相应的中断服务程序。中断服务程序必须以“RETI”指令返回至主程序，以继续执行原来的程序。

一旦中断子程序被响应，系统将自动清除 EMI 位，所有其它的中断将被屏蔽，这个方式可以防止任何进一步的中断嵌套。其它中断请求可能发生在此期间，虽然中断不会立即响应，但是中断请求标志位会被记录。

如果某个中断服务子程序正在执行时，有另一个中断要求立即响应，那么 EMI 位应在程序进入中断子程序后置位，以允许此中断嵌套。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到 SP 减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，执行优先级如中断结构图所示。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应将相应的标志置起。

32.3 外部中断

通过 INT0~INT1 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，INT0~INT1 引脚发生所选的边沿跳转时，外部中断请求标志 INT0F~INT1F 被置位产生外部中断请求。若要跳转到相应中断向量地址，总中断控制位 EMI 和相应中断使能位 INT0E~INT1E 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。外部中断引脚和通用 I/O 口共用，如果相应寄存器中的中断使能位被置位，并且通过引脚共用寄存器选择外部中断脚，此引脚将被作为外部中断脚使用。此时必须通过设

置端口控制寄存器中的相关位，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断引脚发生所选边沿跳转时，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INT0F~INT1F 会自动复位且 EMI 位会被清零以除能其它中断。注意，即使此引脚被用作外部中断输入，其上拉电阻选择仍保持有效。

寄存器 INTEG 被用来选择触发外部中断的有效的边沿类型。可以选择上升沿还是下降沿或双沿触发都产生外部中断。注意 INTEG 也可以用来除能外部中断功能。

32.4 多功能中断

此单片机中有三个多功能中断，与其它中断不同，它没有独立源，而是由其它现有的中断源构成，即 TM 中断、PWM 周期中断和 PWM 占空比中断。

当多功能中断中任何一种中断请求发生，标志位 MFnF 被置位，多功能中断请求产生。当多功能中断使能，堆栈未满，包括在多功能中断中的任意一个中断发生时，将调用多功能中断向量子程序。当响应中断服务子程序时，相关的多功能请求标志位会自动复位且 EMI 位会自动清零以除能其它中断。

但必须注意的是，在中断响应时，虽然多功能中断标志会自动复位，但多功能中断源的请求标志位不会自动复位，必须由应用程序清零

32.5 TM 中断

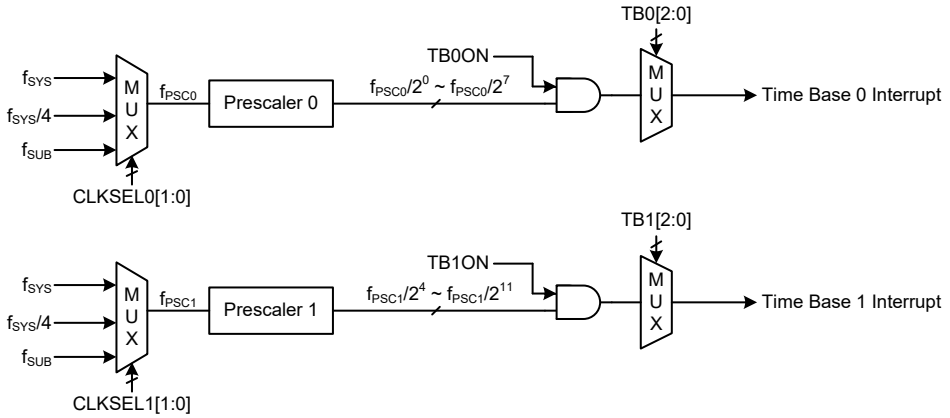
标准型和周期型 TM 各有两个中断，分别来自比较器 P、A 匹配，都属于多功能中断。所有类型的 TM 都有两个中断请求标志位及两个使能位。当 TM 比较器 P、A 匹配情况发生时，相应 TM 中断请求标志被置位，TM 中断请求产生。

若要程序跳转到相应中断向量地址，总中断控制位 EMI、相应 TM 中断使能位和相关多功能中断使能位 MFnE 需先被置位。当中断使能，堆栈未满且 TM 比较器匹配情况发生时，可跳转至相关多功能中断向量子程序中执行。当 TM 中断响应，EMI 将被自动清零以除能其它中断，相关 MFnF 标志也可自动清除，但 TM 中断请求标志需在应用程序中手动清除。

32.6 时基中断

时基中断提供一个固定周期的中断信号，由各自的定时器功能产生溢出信号控制。当各自的中断请求标志 TB0F 或 TB1F 被置位时，中断请求发生。当总中断使能位 EMI 和时基使能位 TB0E 或 TB1E 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未满且时基溢出时，将调用它们各自的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 TB0F 或 TB1F 会自动复位且 EMI 位会被清零以除能其它中断。

时基中断的目的是提供一个固定周期的中断信号。其时钟源 f_{PSC0} 或 f_{PSC1} 来自内部时钟源 f_{SYS}、$f_{SYS}/4$ 或 f_{SUB}，经过一个分频器，分频比可通过配置 TB0C 和 TB1C 寄存器中的相关位选择以获得更长的中断周期。时钟源控制时基中断周期，分别通过 PSC0R 和 PSC1R 寄存器中的 CLKSEL0[1:0] 和 CLKSEL1[1:0] 位进行选择。



时基中断

- **PSC0R 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CLKSEL01	CLKSEL00
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2　未定义，读为“0”

Bit 1~0　**CLKSEL01~CLKSEL00**：预分频器 0 时钟源 f_{PSC0} 选择
00：f_{SYS}
01：$f_{SYS}/4$
1x：f_{SUB}

- **PSC1R 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CLKSEL11	CLKSEL10
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2　未定义，读为“0”

Bit 1~0　**CLKSEL11~CLKSEL10**：预分频器 1 时钟源 f_{PSC1} 选择
00：f_{SYS}
01：$f_{SYS}/4$
1x：f_{SUB}

- **TB0C 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	TB0ON	—	—	—	—	TB02	TB01	TB00
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7　**TB0ON**：时基 0 使能控制位
0：除能
1：使能

Bit 6~3　未定义，读为“0”

Bit 2~0 **TB02~TB00**：时基 0 溢出周期选择位
000：$2^0/f_{PSC0}$
001：$2^1/f_{PSC0}$
010：$2^2/f_{PSC0}$
011：$2^3/f_{PSC0}$
100：$2^4/f_{PSC0}$
101：$2^5/f_{PSC0}$
110：$2^6/f_{PSC0}$
111：$2^7/f_{PSC0}$

- **TB1C 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	TB1ON	—	—	—	—	TB12	TB11	TB10
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 **TB1ON**：时基 1 使能控制位
0：除能
1：使能

Bit 6~3 未定义，读为“0”

Bit 2~0 **TB12~TB10**：时基 1 溢出周期选择位
000：$2^4/f_{PSC1}$
001：$2^5/f_{PSC1}$
010：$2^6/f_{PSC1}$
011：$2^7/f_{PSC1}$
100：$2^8/f_{PSC1}$
101：$2^9/f_{PSC1}$
110：$2^{10}/f_{PSC1}$
111：$2^{11}/f_{PSC1}$

32.7 PWM 中断

PWM 电路包含 6 个中断，分别是 PWM 周期中断和 PWM0~PWM4 占空比中断，都属于多功能中断。当 PWM 周期匹配或 PWM0~PWM4 占空比匹配发生时，相应中断请求标志位 PWMPF、PWM0DF~PWM4DF 被置位，PWM 中断请求产生。

若要跳转到相应中断向量地址，总中断控制位 EMI、相应中断使能位 PWMPE、PWM0DE~PWM4DE 和相关多功能中断使能位 MFnE 需先被置位。当中断使能，堆栈未满并且 PWM 周期匹配、PWM0~PWM4 占空比匹配发生时，可跳转至相关多功能中断向量子程序中执行。当 PWM 中断服务子程序响应时，EMI 将被自动清零以除能其它中断，相关 MFnF 标志也可自动清除，但 PWM 中断请求标志需在应用程序中手动清除。

32.8 A/D 转换器中断

A/D 转换动作的结束控制 A/D 转换器中断。当 A/D 转换器中断请求标志 ADF 被置位，即 A/D 转换过程完成时，A/D 转换器中断请求发生。若要跳转到相应中断向量地址，总中断控制位 EMI、A/D 转换器中断使能位 ADE 需先被置位。当中断使能，堆栈未满且 A/D 转换动作结束时，将调用 A/D 转换器中断向量子程序。当 A/D 转换器中断响应时，ADF 标志将自动清除，EMI 将被自动清零以除能其它中断。

32.9 LVD 中断

当低电压检测功能检测到一个低电源电压时，LVD 中断请求标志 LVF 被置位，LVD 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、低电压中断使能位 LVE 需先被置位。当中断使能，堆栈未满且低电压条件发生时，可跳转至 LVD 中断向

量子程序中执行。当 LVD 中断响应，EMI 会被自动清零且 EMI 位会被清零以除能其它中断。当响应中断服务子程序时，LVD 中断请求标志位 LVF 会被自动复位且 EMI 位会被清零以除能其它中断。

32.10 中断唤醒功能

每个中断都具有将处于休眠或空闲模式的单片机唤醒的能力。当中断请求标志由低到高转换时唤醒动作产生，其与中断是否使能无关。因此，尽管单片机处于休眠或空闲模式且系统振荡器停止工作，如有外部中断脚上产生外部边沿跳变或低电压可能导致其相应的中断标志被置位，由此产生中断，因此必须注意避免伪唤醒情况的发生。如果要除能中断唤醒功能，单片机进入休眠或空闲模式前相应中断请求标志应被置起。中断唤醒功能不受中断使能位的影响。

32.11 编程注意事项

通过禁止相关中断使能位，可以屏蔽中断请求，然而，一旦中断请求标志位被设定，它们会被保留在中断控制寄存器内，直到相应的中断服务子程序执行或请求标志位被软件指令清零。

建议在中断服务子程序中不要使用“CALL 子程序”指令。中断通常发生在不可预料的情况或是需要立刻执行的某些应用。假如只剩下一层堆栈且没有控制好中断，当“CALL 子程序”在中断服务子程序中执行时，将破坏原来的控制序列。

所有中断在休眠或空闲模式下都具有唤醒功能，当中断请求标志发生由低到高的转变时都可产生唤醒功能。若要避免相应中断产生唤醒动作，在单片机进入休眠或空闲模式前需先将相应请求标志置为高。

当进入中断服务程序，系统仅将程序计数器的内容压入堆栈，如果中断服务程序会改变状态寄存器或其它的寄存器的内容而破坏控制流程，应事先将这些数据保存起来。

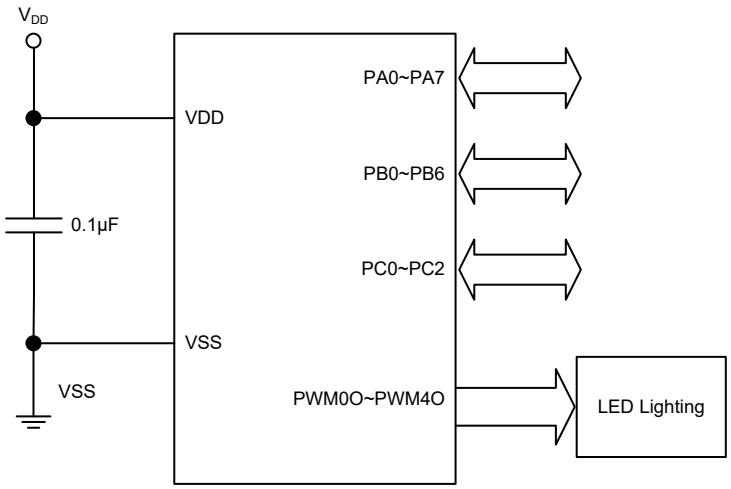
若从中断子程序中返回可执行 RET 或 RETI 指令。除了能返回至主程序外，RETI 指令还能自动设置 EMI 位为高，允许进一步中断。RET 指令只能返回至主程序，清零 EMI 位，除能进一步中断。

33. 配置选项

配置选项在烧写程序时写入芯片。通过 BX-IDE 的软件开发环境，使用者在开发过程中可以选择配置选项。所有位必须按系统的需要定义，具体内容可参考下表：

序号	选项
I/O 共用引脚选项	
1	$\overline{\text{RES}}$ 引脚复位功能选择： 00：I/O 口或其它功能 01：$\overline{\text{RES}}$ 引脚 10~11：由 RSTC 寄存器控制
振荡器选项	
2	HIRC 频率选择 – f_{HIRC}： 8MHz，12MHz 或 16MHz

34. 应用电路


V_{DD}
VDD
0.1μF
VSS
VSS
PA0~PA7
PB0~PB6
PC0~PC2
PWM0O~PWM4O
LED Lighting

35. 指令集

35.1 简介

任何单片机成功运作的核心在于它的指令集，此指令集为一组程序指令码，用来指导单片机如何去执行指定的工作。在芯群单片机中，提供了丰富且灵活的指令，共超过六十条，程序设计者可以事半功倍地实现它们的应用。

为了更加容易理解各种各样的指令码，接下来按功能分组介绍它们。

35.2 指令周期

大部分的操作均只需要一个指令周期来执行。分支、调用或查表则需要两个指令周期。一个指令周期相当于四个系统时钟周期，因此如果在 8MHz 的系统时钟振荡器下，大部分的操作将在 0.5μs 中执行完成，而分支或调用操作则将在 1μs 中执行完成。虽然需要两个指令周期的指令通常指的是 JMP、CALL、RET、RETI 和查表指令，但如果牵涉到程序计数器低字节寄存器 PCL 也将多花费一个周期去加以执行。即指令改变 PCL 的内容进而导致直接跳转至新地址时，需要多一个周期去执行，例如“CLR PCL”或“MOV PCL, A”指令。对于跳转指令必须注意的是，如果比较的结果牵涉到跳转动作将多花费一个周期，如果没有则需一个周期即可。

35.3 数据的传送

单片机程序中数据传送是使用最为频繁的操作之一，使用三种 MOV 的指令，数据不但可以从寄存器转移至累加器 (反之亦然)，而且能够直接移动立即数到累加器。数据传送最重要的应用之一是从输入端口接收数据或传送数据到输出端口。

35.4 算术运算

算术运算和数据处理是大部分单片机应用所必需具备的能力，在芯群单片机内部的指令集中，可直接实现加与减的运算。当加法的结果超出 255 或减法的结果少于 0 时，要注意正确的处理进位和借位的问题。INC、INCA、DEC 和 DECA 指令提供了对一个指定地址的值加一或减一的功能。

35.5 逻辑和移位运算

标准逻辑运算例如 AND、OR、XOR 和 CPL 全都包含在芯群单片机内部的指令集中。大多数牵涉到数据运算的指令，数据的传送必须通过累加器。在所有逻辑数据运算中，如果运算结果为零，则零标志位将被置位，另外逻辑数据运用形式还有移位指令，例如 RR、RL、RRC 和 RLC 提供了向左或向右移动一位的方法。不同的移位指令可满足不同的应用需要。移位指令常用于串行端口的程序应用，数据可从内部寄存器转移至进位标志位，而此位则可被检验，移位运算还可应用在乘法与除法的运算组成中。

35.6 分支和控制转换

程序分支是采取使用 JMP 指令跳转至指定地址或使用 CALL 指令调用子程序的形式，两者之不同在于当子程序被执行完毕后，程序必须马上返回原来的地址。这个动作是由放置在子程序里的返回指令 RET 来实现，它可使程序跳回 CALL 指令之后的地址。在 JMP 指令中，程序则只是跳到一个指定的地址而已，并不需如 CALL 指令般跳回。一个非常有用的分支指令是条件跳转，跳转条件是由数据存储器或指定位来加以决定。遵循跳转条件，程序将继续执行下一条指令或略过且跳转至接下来的指令。这些分支指令是程序走向的关键，跳转条件可能是外部开关输入，或是内部数据位的值。

35.7 位运算

提供数据存储器中单个位的运算指令是芯群单片机的特性之一。这特性对于输出端口位的设置尤其有用，其中个别的位或端口的引脚可以使用“SET [m].i”或“CLR [m].i”指令来设定其为高位或低位。如果没有这特性，程序设计师必须先读入输出口的 8 位数据，处理这些数据，然后再输出正确的新数据。这种读入 - 修改 - 写出的过程现在则被位运算指令所取代。

35.8 查表运算

数据的储存通常由寄存器完成，然而当处理大量固定的数据时，它的存储量常常造成对个别存储器的不便。为了改善此问题，芯群单片机允许在程序存储器中建立一个表格作为数据可直接存储的区域，只需要一组简易的指令即可对数据进行查表。

35.9 其它运算

除了上述功能指令外，其它指令还包括用于省电的“HALT”指令和使程序在极端电压或电磁环境下仍能正常工作的看门狗定时器控制指令。这些指令的使用则请查阅相关的章节。

36. 指令集概要

当要操作的数据存储器位于数据存储器 Sector 0 时，下表说明了与数据存储器存取有关的指令。

36.1 惯例

x：立即数
m：数据存储器地址
A：累加器
i：第 0~7 位
addr：程序存储器地址

助记符	说明	指令周期	影响标志位
算术运算			
ADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	1	Z, C, AC, OV, SC
ADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	1[注]	Z, C, AC, OV, SC
ADD A, x	ACC 与立即数相加，结果放入 ACC	1	Z, C, AC, OV, SC
ADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	1	Z, C, AC, OV, SC
ADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	1[注]	Z, C, AC, OV, SC
SUB A, x	ACC 与立即数相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	1[注]	Z, C, AC, OV, SC, CZ
SBC A, x	ACC 与立即数、进位标志相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	1[注]	Z, C, AC, OV, SC, CZ
DAA [m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	1[注]	C
逻辑运算			
AND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	1	Z
OR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	1	Z
XOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	1	Z
ANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	1[注]	Z
ORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	1[注]	Z
XORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	1[注]	Z
AND A, x	ACC 与立即数做“与”运算，结果放入 ACC	1	Z
OR A, x	ACC 与立即数做“或”运算，结果放入 ACC	1	Z
XOR A, x	ACC 与立即数做“异或”运算，结果放入 ACC	1	Z
CPL [m]	对数据存储器取反，结果放入数据存储器	1[注]	Z
CPLA [m]	对数据存储器取反，结果放入 ACC	1	Z
递增和递减			
INCA [m]	递增数据存储器，结果放入 ACC	1	Z
INC [m]	递增数据存储器，结果放入数据存储器	1[注]	Z
DECA [m]	递减数据存储器，结果放入 ACC	1	Z
DEC [m]	递减数据存储器，结果放入数据存储器	1[注]	Z

助记符	说明	指令周期	影响标志位
移位			
RRA [m]	数据存储器右移一位，结果放入 ACC	1	无
RR [m]	数据存储器右移一位，结果放入数据存储器	1注	无
RRCA [m]	带进位将数据存储器右移一位，结果放入 ACC	1	C
RRC [m]	带进位将数据存储器右移一位，结果放入数据存储器	1注	C
RLA [m]	数据存储器左移一位，结果放入 ACC	1	无
RL [m]	数据存储器左移一位，结果放入数据存储器	1注	无
RLCA [m]	带进位将数据存储器左移一位，结果放入 ACC	1	C
RLC [m]	带进位将数据存储器左移一位，结果放入数据存储器	1注	C
数据传送			
MOV A,[m]	将数据存储器送至 ACC	1	无
MOV [m],A	将 ACC 送至数据存储器	1注	无
MOV A, x	将立即数送至 ACC	1	无
位运算			
CLR [m].i	清除数据存储器的位	1注	无
SET [m].i	置位数据存储器的位	1注	无
转移			
JMP addr	无条件跳转	2	无
SZ [m]	如果数据存储器为零，则跳过下一条指令	1注	无
SZA [m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	1注	无
SNZ [m]	如果数据存储器不为零，则跳过下一条指令	1注	无
SZ [m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	1注	无
SNZ [m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	1注	无
SIZ [m]	递增数据存储器，如果结果为零，则跳过下一条指令	1注	无
SDZ [m]	递减数据存储器，如果结果为零，则跳过下一条指令	1注	无
SIZA [m]	递增数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1注	无
SDZA [m]	递减数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1注	无
CALL addr	子程序调用	2	无
RET	从子程序返回	2	无
RET A, x	从子程序返回，并将立即数放入 ACC	2	无
RETI	从中断返回	2	无
查表			
TABRD [m]	读取特定页的 ROM 内容，并送至数据存储器和 TBLH	2注	无
TABRDL [m]	读取最后页的 ROM 内容，并送至数据存储器和 TBLH	2注	无
ITABRD [m]	读表指针 TBLP 自加，读取特定页的 ROM 内容，并送至数据存储器和 TBLH	2注	无
ITABRDL [m]	读表指针 TBLP 自加，读取最后页的 ROM 内容，并送至数据存储器和 TBLH	2注	无
其它指令			
NOP	空指令	1	无
CLR [m]	清除数据存储器	1注	无
SET [m]	置位数据存储器	1注	无

助记符	说明	指令周期	影响标志位
CLR WDT	清除看门狗定时器	1	TO, PDF
SWAP [m]	交换数据存储器的高低字节，结果放入数据存储器	1 注	无
SWAPA [m]	交换数据存储器的高低字节，结果放入 ACC	1	无
HALT	进入暂停模式	1	TO, PDF

注：1. 对跳转指令而言，如果比较的结果牵涉到跳转即需 2 个周期，如果没有发生跳转，则只需一个周期。
2. 任何指令若要改变 PCL 的内容将需要 2 个周期来执行。

36.2 扩展指令集

扩展指令用来提供更大范围的数据存储器寻址。当被存取的数据存储器位于 Sector 0 之外的任何数据存储器 Sector，扩展指令可直接存取数据存储器而无需使用间接寻址，此举不仅可节省 Flash 存储器空间的使用，同时可提高 CPU 执行效率。

助记符	说明	指令周期	影响标志位
算术运算			
LADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	2	Z, C, AC, OV, SC
LADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	2注	Z, C, AC, OV, SC
LADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	2	Z, C, AC, OV, SC
LADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	2注	Z, C, AC, OV, SC
LSUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	2	Z, C, AC, OV, SC, CZ
LSUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	2注	Z, C, AC, OV, SC, CZ
LSBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	2	Z, C, AC, OV, SC, CZ
LSBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	2注	Z, C, AC, OV, SC, CZ
LDAA [m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	2注	C
逻辑运算			
LAND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	2	Z
LOR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	2	Z
LXOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	2	Z
LANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	2注	Z
LORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	2注	Z
LXORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	2注	Z
LCPL [m]	对数据存储器取反，结果放入数据存储器	2注	Z
LCPLA [m]	对数据存储器取反，结果放入 ACC	2	Z
递增和递减			
LINCA [m]	递增数据存储器，结果放入 ACC	2	Z
LINC [m]	递增数据存储器，结果放入数据存储器	2注	Z
LDECA [m]	递减数据存储器，结果放入 ACC	2	Z
LDEC [m]	递减数据存储器，结果放入数据存储器	2注	Z
移位			
LRRA [m]	数据存储器右移一位，结果放入 ACC	2	无
LRR [m]	数据存储器右移一位，结果放入数据存储器	2注	无
LRRCA [m]	带进位将数据存储器右移一位，结果放入 ACC	2	C
LRRC [m]	带进位将数据存储器右移一位，结果放入数据存储器	2注	C
LRLA [m]	数据存储器左移一位，结果放入 ACC	2	无
LRL [m]	数据存储器左移一位，结果放入数据存储器	2注	无
LRLCA [m]	带进位将数据存储器左移一位，结果放入 ACC	2	C
LRLC [m]	带进位将数据存储器左移一位，结果放入数据存储器	2注	C
数据传送			
LMOV A,[m]	将数据存储器送至 ACC	2	无
LMOV [m],A	将 ACC 送至数据存储器	2注	无

助记符		说明	指令周期	影响标志位
位运算				
LCLR	[m].i	清除数据存储器的位	2 注	无
LSET	[m].i	置位数据存储器的位	2 注	无
转移				
LSZ	[m]	如果数据存储器为零，则跳过下一条指令	2 注	无
LSZA	[m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	2 注	无
LSNZ	[m]	如果数据存储器不为零，则跳过下一条指令	2 注	无
LSZ	[m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	2 注	无
LSNZ	[m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	2 注	无
LSIZ	[m]	递增数据存储器，如果结果为零，则跳过下一条指令	2 注	无
LSDZ	[m]	递减数据存储器，如果结果为零，则跳过下一条指令	2 注	无
LSIZA	[m]	递增数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	2 注	无
LSDZA	[m]	递减数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	2 注	无
查表				
LTABRD	[m]	读取特定页的 ROM 内容，并送至数据存储器和 TBLH	3 注	无
LTABRDL	[m]	读取最后页的 ROM 内容，并送至数据存储器和 TBLH	3 注	无
LITABRD	[m]	读表指针 TBLP 自加，读取特定页的 ROM 内容，并送至数据存储器和 TBLH	3 注	无
LITABRDL	[m]	读表指针 TBLP 自加，读取最后页的 ROM 内容，并送至数据存储器和 TBLH	3 注	无
其它指令				
LCLR	[m]	清除数据存储器	2 注	无
LSET	[m]	置位数据存储器	2 注	无
LSWAP	[m]	交换数据存储器的高低字节，结果放入数据存储器	2 注	无
LSWAPA	[m]	交换数据存储器的高低字节，结果放入 ACC	2	无

注：1. 对扩展跳转指令而言，如果比较的结果牵涉到跳转即需 3 个周期，如果没有发生跳转，则只需两个周期。

2. 任何扩展指令若要改变 PCL 的内容将需要 3 个周期来执行。

37. 指令定义

ADC A, [m] Add Data Memory to ACC with Carry
指令说明 将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。
功能表示 ACC ← ACC + [m] + C
影响标志位 OV、Z、AC、C、SC

ADCM A, [m] Add ACC to Data Memory with Carry
指令说明 将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。
功能表示 [m] ←ACC + [m] + C
影响标志位 OV、Z、AC、C、SC

ADD A, [m] Add Data Memory to ACC
指令说明 将指定的数据存储器和累加器内容相加，结果存放到累加器。
功能表示 ACC ← ACC + [m]
影响标志位 OV、Z、AC、C、SC

ADD A, x Add immediate data to ACC
指令说明 将累加器和立即数相加，结果存放到累加器。
功能表示 ACC ← ACC + x
影响标志位 OV、Z、AC、C、SC

ADDM A, [m] Add ACC to Data Memory
指令说明 将指定的数据存储器和累加器内容相加，结果存放到指定的数据存储器。
功能表示 [m] ←ACC + [m]
影响标志位 OV、Z、AC、C、SC

AND A, [m] Logical AND Data Memory to ACC
指令说明 将累加器中的数据和指定数据存储器内容做逻辑与，结果存放到累加器。
功能表示 ACC ← ACC “AND” [m]
影响标志位 Z

AND A, x Logical AND immediate data to ACC
指令说明 将累加器中的数据和立即数做逻辑与，结果存放到累加器。
功能表示 ACC ← ACC“AND”x
影响标志位 Z

ANDM A, [m] Logical AND ACC to Data Memory
指令说明 将指定数据存储器内容和累加器中的数据做逻辑与，结果存放到数据存储器。
功能表示 [m] ← ACC“AND”[m]
影响标志位 Z

CALL addr Subroutine call
指令说明 无条件地调用指定地址的子程序，此时程序计数器先加 1 获得下一个要执行的指令地址并压入堆栈，接着载入指定地址并从新地址继续执行程序，由于此指令需要额外的运算，所以为一个 2 周期的指令。
功能表示 Stack ← Program Counter + 1
Program Counter ← addr
影响标志位 无

CLR [m] Clear Data Memory
指令说明 将指定数据存储器的内容清零。
功能表示 [m] ← 00H
影响标志位 无

CLR [m].i Clear bit of Data Memory
指令说明 将指定数据存储器的第 i 位内容清零。
功能表示 [m].i ← 0
影响标志位 无

CLR WDT Clear Watchdog Timer
指令说明 WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。
功能表示 WDT cleared
TO & PDF ← 0
影响标志位 TO、PDF

CPL [m] Complement Data Memory
指令说明 将指定数据存储器中的每一位取逻辑反，
相当于从 1 变 0 或 0 变 1。
功能表示 [m] ← $[\overline{m}]$
影响标志位 Z

CPLA [m] Complement Data Memory with result in ACC
指令说明 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1，而结果被储存回累加器且数据存储器中的内容不变。
功能表示 ACC←$[\overline{m}]$
影响标志位 Z

DAA [m] Decimal-Adjust ACC for addition with result in Data Memory
指令说明 将累加器中的内容转换为 BCD (二进制转成十进制) 码。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对原值加“6”，否则原值保持不变；如果高四位的值大于“9”或 C=1，那么 BCD 调整就执行对原值加“6”。BCD 转换实质上是根据累加器和标志位执行 00H，06H，60H 或 66H 的加法运算，结果存放到数据存储器。只有进位标志位 C 受影响，用来指示原始 BCD 的和是否大于 100，并可以进行双精度十进制数的加法运算。
功能表示 [m] ← ACC + 00H 或
[m] ← ACC + 06H 或
[m] ← ACC + 60H 或
[m] ← ACC + 66H
影响标志位 C

DEC [m] Decrement Data Memory
指令说明 将指定数据存储器内容减 1。
功能表示 [m] ← [m] – 1
影响标志位 Z

DECA [m] Decrement Data Memory with result in ACC
指令说明 将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器的内容不变。
功能表示 ACC ← [m] – 1
影响标志位 Z

HALT Enter power down mode
指令说明 此指令终止程序执行并关掉系统时钟，RAM 和寄存器的内容保持原状态，WDT 计数器和分频器被清“0”，暂停标志位 PDF 被置位 1，WDT 溢出标志位 TO 被清 0。
功能表示 TO ← 0
PDF ← 1
影响标志位 TO、PDF

INC [m] Increment Data Memory
指令说明 将指定数据存储器的内容加 1。
功能表示 [m] ← [m] + 1
影响标志位 Z

INCA [m] Increment Data Memory with result in ACC
指令说明 将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。
功能表示 ACC ← [m] + 1
影响标志位 Z

JMP addr Jump unconditionally
指令说明 程序计数器的内容无条件地由被指定的地址取代，程序由新的地址继续执行。当新的地址被加载时，必须插入一个空指令周期，所以此指令为 2 个周期的指令。
功能表示 Program Counter ← addr
影响标志位 无

MOV A, [m] Move Data Memory to ACC
指令说明 将指定数据存储器的内容复制到累加器。
功能表示 ACC← [m]
影响标志位 无

MOV A, x Move immediate data to ACC
指令说明 将 8 位立即数载入累加器。
功能表示 ACC ← x
影响标志位 无

MOV [m], A Move ACC to Data Memory
指令说明 将累加器的内容复制到指定的数据存储器。
功能表示 [m] ← ACC
影响标志位 无

NOP No operation
指令说明 空操作，接下来顺序执行下一条指令。
功能表示 无操作
影响标志位 无

OR A, [m] Logical OR Data Memory to ACC
指令说明 将累加器中的数据和指定的数据存储器内容逻辑或，结果存放到累加器。
功能表示 ACC ← ACC“OR”[m]
影响标志位 Z

OR A, x Logical OR immediate data to ACC
指令说明 将累加器中的数据和立即数逻辑或，结果存放到累加器。
功能表示 ACC ← ACC“OR”x
影响标志位 Z

ORM A, [m] Logical OR ACC to Data Memory
指令说明 将存在指定数据存储器中的数据和累加器逻辑或，结果放到数据存储器。
功能表示 [m] ← ACC“OR”[m]
影响标志位 Z

RET Return from subroutine
指令说明 将堆栈寄存器中的程序计数器值恢复，程序由取回的地址继续执行。
功能表示 Program Counter←Stack
影响标志位 无

RET A, x Return from subroutine and load immediate data to ACC
指令说明 将堆栈寄存器中的程序计数器值恢复且累加器载入指定的立即数，程序由取回的地址继续执行。
功能表示 Program Counter ← Stack
ACC←x
影响标志位 无

RETI Return from interrupt

指令说明 将堆栈寄存器中的程序计数器值恢复且中断功能通过设置 EMI 位重新使能。EMI 是控制中断使能的主控制位。如果在执行 RETI 指令之前还有中断未被响应，则这个中断将在返回主程序之前被响应。

功能表示 Program Counter ←Stack

EMI ← 1

影响标志位 无

RL [m] Rotate Data Memory left

指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。

功能表示 [m].(i+1) ← [m].i (i=0~6)

[m].0 ← [m].7

影响标志位 无

RLA [m] Rotate Data Memory left with result in ACC

指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，结果送到累加器，而指定数据存储器的内容保持不变。

功能表示 ACC.(i+1) ← [m].i (i=0~6)

ACC.0 ←[m].7

影响标志位 无

RLC [m] Rotate Data Memory Left through Carry

指令说明 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位。

功能表示 [m].(i+1) ← [m].i (i=0~6)

[m].0 ← C

C ← [m].7

影响标志位 C

RLC A [m] Rotate Data Memory left through Carry with result in ACC

指令说明 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。

功能表示 ACC.(i+1) ← [m].i (i=0~6)

ACC.0 ← C

C ← [m].7

影响标志位 C

RR [m] Rotate Data Memory right

指令说明 将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。

功能表示 [m].i ← [m].(i+1) (i=0~6)
[m].7 ← [m].0

影响标志位 无

RRA [m] Rotate Data Memory right with result in ACC

指令说明 将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。

功能表示 ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← [m].0

影响标志位 无

RRC [m] Rotate Data Memory right through Carry

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。

功能表示 [m].i ← [m].(i+1) (i=0~6)
[m].7← C
C ← [m].0

影响标志位 C

RRCA [m] Rotate Data Memory right through Carry with result in ACC

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。

功能表示 ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← C
C ← [m].0

影响标志位 C

SBC A, [m] Subtract Data Memory from ACC with Carry

指令说明 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – [m] – $\overline{C}$

影响标志位 OV、Z、AC、C、SC、CZ

SBC A, x Subtract immediate data from ACC with Carry

指令说明 将累加器减去立即数以及进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – [m] – $\overline{C}$

影响标志位 OV、Z、AC、C、SC、CZ

SBCM A, [m] Subtract Data Memory from ACC with Carry and result in Data Memory

指令说明 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 [m] ← ACC – [m] – $\overline{C}$

影响标志位 OV、Z、AC、C、SC、CZ

SDZ [m] Skip if Decrement Data Memory is 0

指令说明 将指定的数据存储器的内容减 1，判断是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 [m] ← [m] – 1，如果 [m]=0 跳过下一条指令执行

影响标志位 无

SDZA [m] Skip if decrement Data Memory is zero with result in ACC

指令说明 将指定数据存储器内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 ACC ← [m] – 1，如果 ACC=0 跳过下一条指令执行

影响标志位 无

SET [m] Set Data Memory

指令说明 将指定数据存储器的每一位设置为 1。

功能表示 [m] ← FFH

影响标志位 无

SET [m].i Set bit of Data Memory
指令说明 将指定数据存储器的第 i 位置位为 1。
功能表示 [m].i ← 1
影响标志位 无

SIZ [m] Skip if increment Data Memory is 0
指令说明 将指定的数据存储器的内容加 1，判断是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示 [m] ←[m] + 1，如果 [m]=0 跳过下一条指令执行
影响标志位 无

SIZA [m] Skip if increment Data Memory is zero with result in ACC
指令说明 将指定数据存储器的内容加 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示 ACC ←[m] + 1，如果 ACC=0 跳过下一条指令执行
影响标志位 无

SNZ [m].i Skip if bit i of Data Memory is not 0
指令说明 判断指定数据存储器的第 i 位，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果为 0，则程序继续执行下一条指令。
功能表示 如果 [m].i≠0，跳过下一条指令执行
影响标志位 无

SNZ [m] Skip if Data Memory is not 0
指令说明 指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定存储器，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果为 0，则程序继续执行下一条指令。
功能表示 如果 [m]≠0，跳过下一条指令执行
影响标志位 无

SUB A, [m] Subtract Data Memory from ACC

指令说明 将累加器的内容减去指定的数据存储器的数据，把结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – [m]

影响标志位 OV、Z、AC、C、SC、CZ

SUBM A, [m] Subtract Data Memory from ACC with result in Data Memory

指令说明 将累加器的内容减去指定数据存储器的数据，结果存放到指定的数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 [m] ← ACC – [m]

影响标志位 OV、Z、AC、C、SC、CZ

SUB A, x Subtract immediate Data from ACC

指令说明 将累加器的内容减去立即数，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – x

影响标志位 OV、Z、AC、C、SC、CZ

SWAP [m] Swap nibbles of Data Memory

指令说明 将指定数据存储器的低 4 位和高 4 位互相交换。

功能表示 [m].3~[m].0 ↔ [m].7~[m].4

影响标志位 无

SWAPA [m] Swap nibbles of Data Memory with result in ACC

指令说明 将指定数据存储器的低 4 位与高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。

功能表示 ACC.3~ACC.0 ← [m].7~[m].4

ACC.7~ACC.4 ← [m].3~[m].0

影响标志位 无

SZ [m] Skip if Data Memory is 0

指令说明 指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定数据存储器的内容是否为 0，若为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果 [m]=0，跳过下一条指令执行

影响标志位 无

SZA [m] Skip if Data Memory is 0 with data movement to ACC

指令说明 将指定数据存储器内容复制到累加器，并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 ACC ← [m]，如果 [m]=0，跳过下一条指令执行

影响标志位 无

SZ [m].i Skip if bit i of Data Memory is 0

指令说明 判断指定数据存储器的第 i 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果 [m].i=0，跳过下一条指令执行

影响标志位 无

TABRD [m] Read table (specific page) to TBLH and Data Memory

指令说明 将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

TABRDL [m] Read table (last page) to TBLH and Data Memory

指令说明 将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

ITABRD [m] Increment table pointer low byte first and read table (specific page) to TBLH and data memory

指令说明 自加表格指针低字节 TBLP，将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定的数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

ITABRDL [m] Increment table pointer low byte first and read table (last page) to TBLH and data memory

指令说明 自加表格指针低字节 TBLP，将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定的数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

XOR A, [m] Logical XOR Data Memory to ACC

指令说明 将累加器的数据和指定的数据存储器内容逻辑异或，结果存放到累加器。

功能表示 ACC ← ACC“XOR”[m]

影响标志位 Z

XORM A, [m] Logical XOR ACC to Data Memory

指令说明 将累加器的数据和指定的数据存储器内容逻辑异或，结果放到数据存储器。

功能表示 [m] ← ACC“XOR”[m]

影响标志位 Z

XOR A, x Logical XOR immediate data to ACC

指令说明 将累加器的数据与立即数逻辑异或，结果存放到累加器。

功能表示 ACC ← ACC“XOR”x

影响标志位 Z

37.1 扩展指令定义

扩展指令被用来直接存取存储在任何数据存储器 Sector 中的数据。

LADC A, [m] Add Data Memory to ACC with Carry
指令说明 将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。
功能表示 ACC ← ACC + [m] + C
影响标志位 OV、Z、AC、C、SC

LADCM A, [m] Add ACC to Data Memory with Carry
指令说明 将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。
功能表示 [m] ←ACC + [m] + C
影响标志位 OV、Z、AC、C、SC

LADD A, [m] Add Data Memory to ACC
指令说明 将将指定的数据存储器和累加器内容相加，结果存放到累加器。
功能表示 ACC ← ACC + [m]
影响标志位 OV、Z、AC、C、SC

LADDM A, [m] Add ACC to Data Memory
指令说明 将指定的数据存储器和累加器内容相加，结果存放到指定的数据存储器。
功能表示 [m] ←ACC + [m]
影响标志位 OV、Z、AC、C、SC

LAND A, [m] Logical AND Data Memory to ACC
指令说明 将累加器中的数据和指定数据存储器内容做逻辑与，结果存放到累加器。
功能表示 ACC ← ACC “AND” [m]
影响标志位 Z

LANDM A, [m] Logical AND ACC to Data Memory
指令说明 将指定数据存储器内容和累加器中的数据做逻辑与，结果存放到数据存储器。
功能表示 [m] ← ACC “AND” [m]
影响标志位 Z

LCLR [m] Clear Data Memory
指令说明 将指定数据存储器的内容清零。
功能表示 [m] ← 00H
影响标志位 无

LCLR [m].i Clear bit of Data Memory
指令说明 将指定数据存储器的第 i 位内容清零。
功能表示 [m].i ← 0
影响标志位 无

LCPL [m] Complement Data Memory
指令说明 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1。
功能表示 [m] ← $[\overline{m}]$
影响标志位 Z

LCPLA [m] Complement Data Memory with result in ACC
指令说明 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1，结果被存放回累加器且数据寄存器的内容保持不变。
功能表示 ACC←$[\overline{m}]$
影响标志位 Z

LDAA [m] Decimal-Adjust ACC for addition with result in Data Memory
指令说明 将累加器中的内容转换为 BCD (二进制转成十进制) 码。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对低四位加“6”，否则低四位保持不变；如果高四位的值大于“9”或 C=1，那么 BCD 调整就执行对高四位加“6”。BCD 转换实质上是根据累加器和标志位执行 00H，06H，60H 或 66H 的加法运算，结果存放到数据存储器。只有进位标志位 C 受影响，用来指示原始 BCD 的和是否大于 100，并可以进行双精度十进制数的加法运算。
功能表示 [m] ← ACC + 00H 或
[m] ← ACC + 06H 或
[m] ← ACC + 60H 或
[m] ← ACC + 66H
影响标志位 C

LDEC [m] Decrement Data Memory
指令说明 将指定数据存储器的内容减 1。
功能表示 [m] ← [m] – 1
影响标志位 Z

LDECA [m] Decrement Data Memory with result in ACC
指令说明 将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器的内容不变。
功能表示 ACC ← [m] – 1
影响标志位 Z

LINC [m] Increment Data Memory
指令说明 将指定数据存储器的内容加 1。
功能表示 [m] ← [m] + 1
影响标志位 Z

LINCA [m] Increment Data Memory with result in ACC
指令说明 将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。
功能表示 ACC ← [m] + 1
影响标志位 Z

LMOV A, [m] Move Data Memory to ACC
指令说明 将指定数据存储器的内容复制到累加器中。
功能表示 ACC← [m]
影响标志位 无

LMOV [m], A Move ACC to Data Memory
指令说明 将累加器的内容复制到指定数据存储器。
功能表示 [m] ← ACC
影响标志位 无

LOR A, [m] Logical OR Data Memory to ACC
指令说明 将累加器中的数据和指定的数据存储器内容逻辑或，结果存放到累加器。
功能表示 ACC ← ACC “OR” [m]
影响标志位 Z

LORM A, [m] Logical OR ACC to Data Memory
指令说明 将存在指定数据存储器中的数据和累加器逻辑或，
结果放到数据存储器。
功能表示 [m] ← ACC “OR” [m]
影响标志位 Z

LRL [m] Rotate Data Memory left
指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
功能表示 [m].(i+1) ← [m].i (i=0~6)
[m].0 ← [m].7
影响标志位 无

LRLA [m] Rotate Data Memory left with result in ACC
指令说明 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，
结果送到累加器，而指定数据存储器的内容保持不变。
功能表示 ACC.(i+1) ← [m].i (i=0~6)
ACC.0 ←[m].7
影响标志位 无

LRLC [m] Rotate Data Memory Left through Carry
指令说明 将指定数据存储器的内容连同进位标志左移 1 位，
第 7 位取代进位标志且原本的进位标志移到第 0 位。
功能表示 [m].(i+1) ← [m].i (i=0~6)
[m].0 ← C
C ← [m].7
影响标志位 C

LRLC A [m] Rotate Data Memory left through Carry with result in ACC
指令说明 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位
取代进位标志且原本的进位标志移到第 0 位，移位结果送
回累加器，但是指定数据寄存器的内容保持不变。
功能表示 ACC.(i+1) ← [m].i (i=0~6)
ACC.0 ← C
C ← [m].7
影响标志位 C

LRR [m] Rotate Data Memory right

指令说明 将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。

功能表示 [m].i ← [m].(i+1) (i=0~6)
[m].7 ← [m].0

影响标志位 无

LRRA [m] Rotate Data Memory right with result in ACC

指令说明 将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。

功能表示 ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← [m].0

影响标志位 无

LRRC [m] Rotate Data Memory right through Carry

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。

功能表示 [m].i ← [m].(i+1) (i=0~6)
[m].7← C
C ← [m].0

影响标志位 C

LRRCA [m] Rotate Data Memory right through Carry with result in ACC

指令说明 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。

功能表示 ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← C
C ← [m].0

影响标志位 C

LSBC A, [m] Subtract Data Memory from ACC with Carry

指令说明 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – [m] – $\overline{C}$

影响标志位 OV、Z、AC、C、SC、CZ

LSBCM A, [m] Subtract Data Memory from ACC with Carry and result in Data Memory

指令说明 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 $[m] \leftarrow ACC - [m] - \overline{C}$

影响标志位 OV、Z、AC、C、SC、CZ

LSDZ [m] Skip if Decrement Data Memory is 0

指令说明 将指定的数据存储器的内容减 1，判断是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 [m] ← [m] – 1，如果 [m]=0 跳过下一条指令执行

影响标志位 无

LSDZA [m] Skip if decrement Data Memory is zero with result in ACC

指令说明 将指定数据存储器内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 ACC ← [m] – 1，如果 ACC=0 跳过下一条指令执行

影响标志位 无

LSET [m] Set Data Memory

指令说明 将指定数据存储器的每一个位置位为 1。

功能表示 [m] ← FFH

影响标志位 无

LSET [m].i Set bit of Data Memory

指令说明 将指定数据存储器的第 i 位置位为 1。

功能表示 [m].i ← 1

影响标志位 无

LSIZ [m] Skip if increment Data Memory is 0

指令说明 将指定的数据存储器的内容加 1，判断是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 [m] ←[m] + 1，如果 [m]=0 跳过下一条指令执行

影响标志位 无

LSIZA [m] Skip if increment Data Memory is zero with result in ACC

指令说明 将指定数据存储器的内容加 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 ACC ←[m] + 1，如果 ACC=0 跳过下一条指令执行

影响标志位 无

LSNZ [m].i Skip if bit i of Data Memory is not 0

指令说明 判断指定数据存储器的第 i 位，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果为 0，则程序继续执行下一条指令。

功能表示 如果 [m].i≠0，跳过下一条指令执行

影响标志位 无

LSNZ [m] Skip if Data Memory is not 0

指令说明 指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定数据存储器，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果为 0，则程序继续执行下一条指令。

功能表示 如果 [m]≠0，跳过下一条指令执行

影响标志位 无

LSUB A, [m] Subtract Data Memory from ACC

指令说明 将累加器的内容减去指定的数据存储器的数据，把结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 ACC ← ACC – [m]

影响标志位 OV、Z、AC、C、SC、CZ

LSUBM A, [m] Subtract Data Memory from ACC with result in Data Memory

指令说明 将累加器的内容减去指定数据存储器的数据，结果存放到指定的数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。

功能表示 [m] ← ACC – [m]

影响标志位 OV、Z、AC、C、SC、CZ

LSWAP [m] Swap nibbles of Data Memory

指令说明 将指定数据存储器的低 4 位和高 4 位互相交换。

功能表示 [m].3~[m].0 ↔ [m].7~[m].4

影响标志位 无

LSWAPA [m] Swap nibbles of Data Memory with result in ACC

指令说明 将指定数据存储器的低 4 位和高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。

功能表示 ACC.3~ACC.0 ← [m].7~[m].4

ACC.7~ACC.4 ← [m].3~[m].0

影响标志位 无

LSZ [m] Skip if Data Memory is 0

指令说明 指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定数据存储器的内容是否为 0，若为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果 [m]=0，跳过下一条指令执行

影响标志位 无

LSZA [m] Skip if Data Memory is 0 with data movement to ACC

指令说明 将指定数据存储器内容复制到累加器，并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 ACC ← [m]，如果 [m]=0，跳过下一条指令执行

影响标志位 无

LSZ [m].i Skip if bit i of Data Memory is 0

指令说明 判断指定数据存储器的第 i 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 3 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。

功能表示 如果 [m].i=0，跳过下一条指令执行

影响标志位 无

LTABRD [m] Move the ROM code (specific page) to TBLH and data memory

指令说明 将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

LTABRDL [m] Read table (last page) to TBLH and Data Memory

指令说明 将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

LITABRD [m] Increment table pointer low byte first and read table (specific page) to TBLH and data memory

指令说明 自加表格指针低字节 TBLP，将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定的数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

LITABRDL [m] Increment table pointer low byte first and read table (last page) to TBLH and data memory

指令说明 自加表格指针低字节 TBLP，将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定的数据存储器且将高字节移至 TBLH。

功能表示 [m] ← 程序代码 (低字节)
TBLH ← 程序代码 (高字节)

影响标志位 无

LXOR A, [m]	Logical XOR Data Memory to ACC
指令说明	将累加器的数据和指定的数据存储器内容逻辑异或，结果存放到累加器。
功能表示	ACC ← ACC“XOR”[m]
影响标志位	Z

LXORM A, [m]	Logical XOR ACC to Data Memory
指令说明	将累加器的数据和指定的数据存储器内容逻辑异或，结果放到数据存储器。
功能表示	[m] ← ACC“XOR”[m]
影响标志位	Z

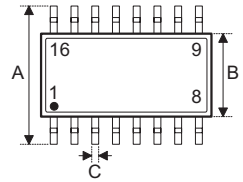
38. 封装信息

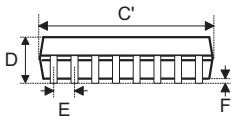
请注意，这里提供的封装信息仅作为参考。由于这个信息经常更新，提醒用户咨询芯群网站 以获取最新版本的封装信息。

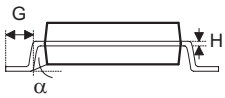
封装信息的相关内容如下所示，点击可链接至芯群网站相关信息页面。

- 封装信息 (包括外形尺寸、包装带和卷轴规格)
- 封装材料信息
- 纸箱信息

38.1 16-pin NSOP (150mil) 外形尺寸



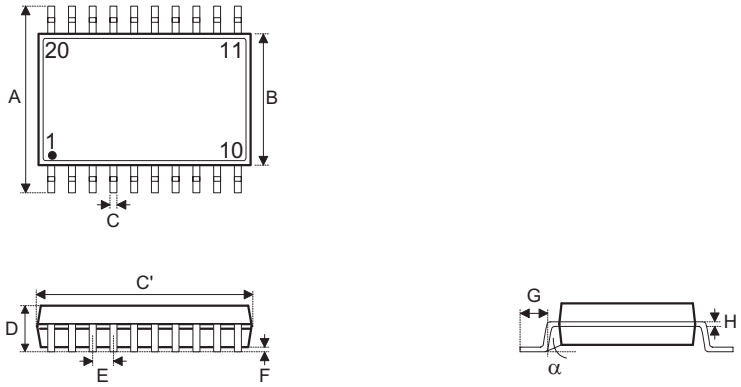




符号	尺寸 (单位：inch)		
	最小值	典型值	最大值
A		0.236 BSC	
B		0.154 BSC	
C	0.012	—	0.020
C’		0.390 BSC	
D	—	—	0.069
E		0.050 BSC	
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符号	尺寸 (单位：mm)		
	最小值	典型值	最大值
A		6.00 BSC	
B		3.90 BSC	
C	0.31	—	0.51
C’		9.90 BSC	
D	—	—	1.75
E		1.27 BSC	
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°

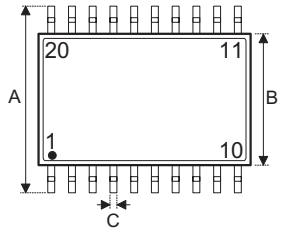
38.2 20-pin NSOP (150mil) 外形尺寸

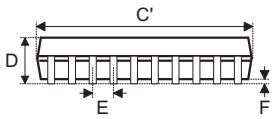


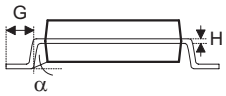
符号	尺寸 (单位：inch)		
	最小值	典型值	最大值
A	0.228	0.236	0.244
B	0.146	0.154	0.161
C	0.009	—	0.012
C’	0.382	0.390	0.398
D	—	—	0.069
E	0.032 BSC		
F	0.002	—	0.009
G	0.020	—	0.031
H	0.008	—	0.010
α	0°	—	8°

符号	尺寸 (单位：mm)		
	最小值	典型值	最大值
A	5.80	6.00	6.20
B	3.70	3.90	4.10
C	0.23	—	0.30
C’	9.70	9.90	10.10
D	—	—	1.75
E	0.80 BSC		
F	0.05	—	0.23
G	0.50	—	0.80
H	0.21	—	0.25
α	0°	—	8°

38.3 20-pin SOP (300mil) 外形尺寸



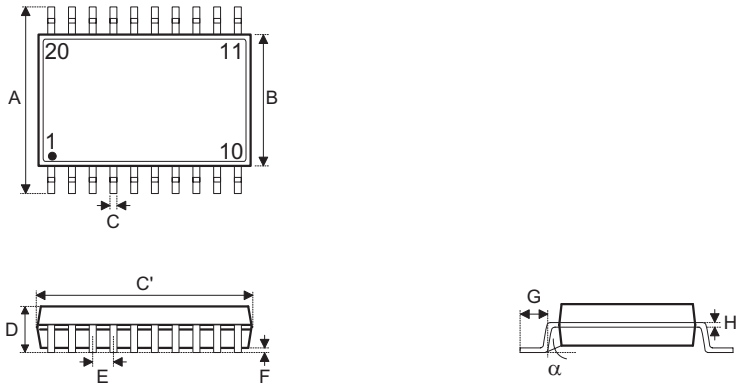




符号	尺寸 (单位：inch)		
	最小值	典型值	最大值
A		0.406 BSC	
B		0.295 BSC	
C	0.012	—	0.020
C’		0.504 BSC	
D	—	—	0.104
E		0.050 BSC	
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

符号	尺寸 (单位：mm)		
	最小值	典型值	最大值
A		10.30 BSC	
B		7.50 BSC	
C	0.31	—	0.51
C’		12.80 BSC	
D	—	—	2.65
E		1.27 BSC	
F	0.10	—	0.30
G	0.40	—	1.27
H	0.20	—	0.33
α	0°	—	8°

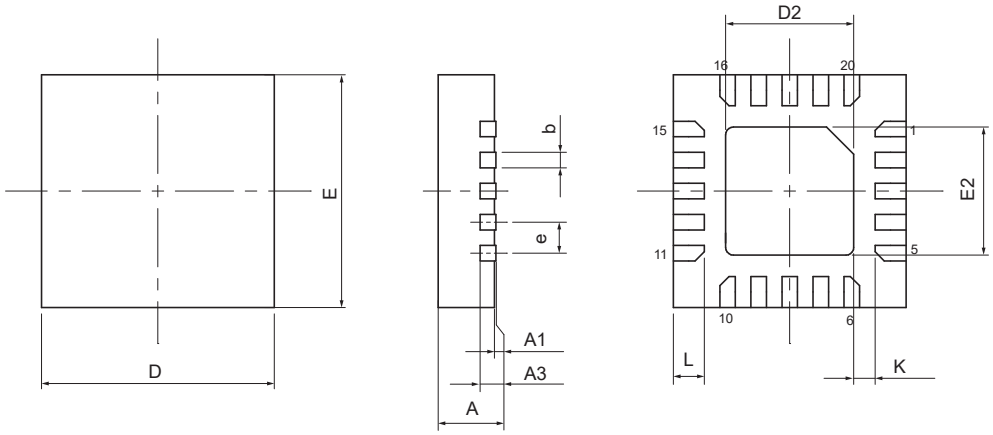
38.4 20-pin SSOP (150mil) 外形尺寸



符号	尺寸 (单位：inch)		
	最小值	典型值	最大值
A		0.236 BSC	
B		0.154 BSC	
C	0.008	—	0.012
C’		0.341 BSC	
D	—	—	0.069
E		0.025 BSC	
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符号	尺寸 (单位：mm)		
	最小值	典型值	最大值
A		6.00 BSC	
B		3.90 BSC	
C	0.20	—	0.30
C’		8.66 BSC	
D	—	—	1.75
E		0.635 BSC	
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

38.5 SAW Type 20-pin QFN (4mm×4mm×0.75mm) 外形尺寸



符号	尺寸 (单位：inch)		
	最小值	典型值	最大值
A	0.028	0.030	0.031
A1	0.000	0.001	0.002
A3		0.008 REF	
b	0.008	0.010	0.012
D		0.157 BSC	
E		0.157 BSC	
e		0.020 BSC	
D2	0.075	—	0.083
E2	0.075	—	0.083
L	0.012	0.016	0.020
K	0.008	—	—

符号	尺寸 (单位：mm)		
	最小值	典型值	最大值
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3		0.203 REF	
b	0.20	0.25	0.30
D		4.00 BSC	
E		4.00 BSC	
e		0.50 BSC	
D2	1.90	—	2.10
E2	1.90	—	2.10
L	0.30	0.40	0.50
K	0.20	—	—