

## SH69P848A

### 2K 一次性编程 10 位 ADC 型 4 位单片机

#### 特性

- 基于 SH6610D 的 10 位 ADC 型 4 位单片机
- OTPROM: 2K X 16 位
- RAM: 253 X 4 位
  - 61 个系统控制寄存器
  - 192 个数据存储寄存器
- 工作电压:
  - $f_{OSC} = 30kHz - 4MHz$ ,  $V_{DD} = 2.4V - 5.5V$
  - $f_{OSC} = 4MHz - 10MHz$ ,  $V_{DD} = 4.5V - 5.5V$
- 5 个双向 I/O 端口和一个输出端口 (PORTE.1)
- 8 层堆栈 (包括中断)
- 两个 8 位自动重载定时/计数器
- 预热计数器
- 中断源:
  - 模/数中断
  - 内部中断 (定时器 0, 定时器 1)
  - 外部中断: PORTB & PORTD (下降沿)
- 振荡器 (代码选项)
  - 外部 RC 振荡器: 400kHz - 10MHz
  - 内部 RC 振荡器: 4MHz  $\pm$  5%
  - 外部时钟: 30kHz - 10MHz
- 指令周期时间 ( $4/f_{OSC}$ )
- 两种低功耗工作模式: HALT 和 STOP
- 复位
  - 内建看门狗定时器 (WDT) (代码选项)
  - 内建上电复位 (POR)
  - 内建低电压复位 (LVR)
- 内建低电压复位功能, 两种监测电平 (代码选项)
- 3 通道 10 位模/数转换器 (ADC)
- 读取存储器数据列表功能
- 1 通道 (8 + 2) 位脉宽调制输出 (PWM)
- OTP 类型代码保护
- 提供 8 引脚 SOP 封装

#### 概述

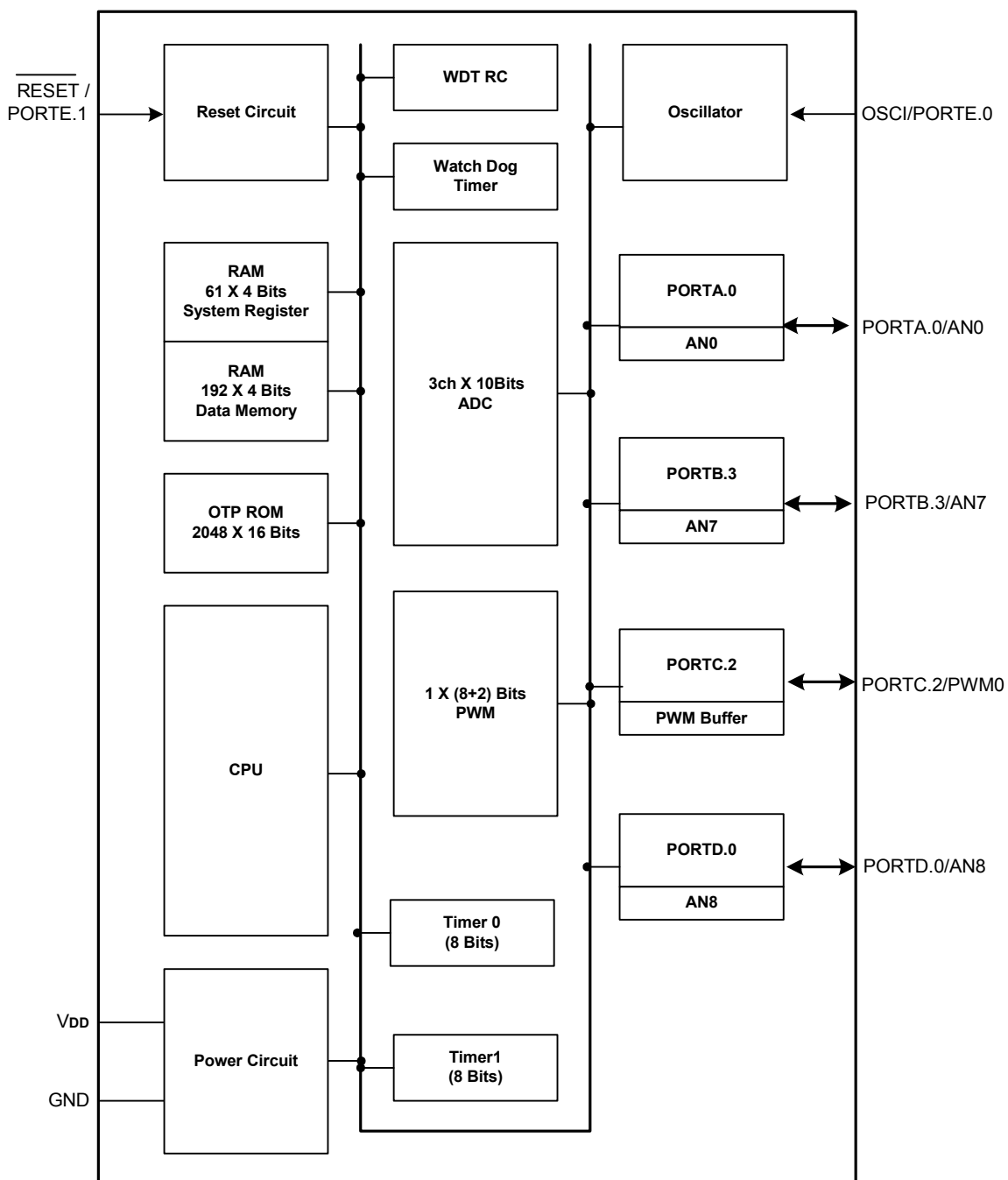
SH69P848A 是一种先进的 CMOS 4 位单片机。该器件集成了 SH6610D CPU 内核, RAM, ROM, 定时器, 10 位模/数转换器, 1 通道 (8+2) 位高速脉宽调制输出, 振荡器时钟电路, 看门狗定时器, 低电压复位功能且支持省电方式以节约电能消耗。SH69P848A 适用于充电器的应用。

#### 引脚配置





方框图





引脚描述

引脚编号	引脚命名	引脚性质	说明
1	PORTC.2 /PWM0	I/O O	可编程 I/O 脉宽调制输出
2	$\overline{\text{RESET}}$ /PORTE.1	I O	复位引脚（低电压有效，施密特触发输入） 开漏输出（代码选项）
3	GND	P	接地引脚
4	PORTA.0 /AN0	I/O I	可编程 I/O ADC 通道 AN0 输入
5	PORTB.3 - /AN7	I/O I I	可编程 I/O 外部中断输入（下降沿有效） ADC 通道 AN7 输入
6	V <sub>DD</sub>	P	电源引脚
7	OSCI /PORTE.0	I I/O	时钟输入引脚，连接到谐振器的外部电阻 可编程 I/O
8	PORTD.0 - /AN8	I/O I I	可编程 I/O 外部中断输入（下降沿有效） ADC 通道 AN8 输入

其中，I: 输入;O: 输出;P: 电源;Z: 高阻

OTP编程引脚说明（OTP编程模式）

引脚编号	引脚命名	引脚性质	共用引脚	说明
6	V <sub>DD</sub>	P	V <sub>DD</sub>	编程电源（+5.5V）
2	V <sub>PP</sub>	P	$\overline{\text{RESET}}$ /PORTE.1	编程高压电源（+11V）
3	GND	P	GND	电源地
7	SCK	I	OSCI /PORTE.0	编程时钟输入引脚
4	SDA	I/O	PORTA.0 /AN0	编程数据引脚

其中，I: 输入;O: 输出;P: 电源;Z: 高阻



## 功能说明

### 1. CPU

CPU包含以下功能模块：程序计数器（PC），算术逻辑单元（ALU），进位标志（CY），累加器，查表寄存器，数据指针（INX, DPH, DPM和DPL），和堆栈。

#### 1.1. PC

程序计数器用于寻址程序 ROM。该计数器有 12 位：页寄存器（PC11），和循环进位计数器（PC10, PC9, PC8, PC7, PC6, PC5, PC4, PC3, PC2, PC1, PC0）。

程序计数器装入与该条指令相关的数据。对于目标地址大于 2K 的 ROM 空间，可通过无条件跳转指令（JMP）设置页寄存器位的值实现跳转。

程序计数器只能寻址 4K 程序 ROM 空间（参考 ROM 说明）。

#### 1.2. ALU 和 CY

ALU 执行算数和逻辑操作。ALU 具有下述功能：

二进制加法/减法（ADC, ADCM, ADD, ADDM, SBC, SBCM, SUB, SUBM, ADI, ADIM, SBI, SBIM）

加法/减法的十进制调整（DAA, DAS）

逻辑操作（AND, ANDM, EOR, EORM, OR, ORM, ANDIM, EORIM, ORIM）

条件跳转（BA0, BA1, BA2, BA3, BAZ, BNZ, BC, BNC）

逻辑移位（SHR）

进位标志（CY）记录 ALU 算术操作中的溢出状态。在中断或子程序调用过程中，进位标志被压入堆栈中并于执行 RTNI 指令时由堆栈中弹出。它不受 RTNW 指令的影响。

#### 1.3. 累加器（AC）

累加器是一个四位寄存器，其中保存了算术逻辑单元的运算结果。它和 ALU 一起，能完成与系统寄存器数据存储器之间的数据传送。

### 2. RAM

内建 RAM 由通用数据存储器 and 系统寄存器组成。由于 RAM 的静态特性，数据存储器能在 CPU 进入 STOP 或者 HALT 方式后保持其中的数据不变。

#### 2.1. RAM 寻址

用一条指令能直接访问数据存储器 and 系统寄存器。下列为存储器空间分配：

系统寄存器：\$000 - \$02F, \$380 - \$38C

数据存储器：\$030 - \$0EF

#### 1.4. 查表寄存器（TBR）

表格数据能被保存在程序存储器中，使用查表指令（TJMP）和常数返回指令（RTNW）查表。在程序 ROM 中，查表寄存器 TBR 和 AC 中存放的是初始地址。TJMP 指令指向地址  $((PC11 - PC8) \times (2^8) + (TBR, AC))$ 。地址由 RTNW 决定，它将查表所得值返回至（TBR, AC）中。ROM 代码的第 7 位-第 4 位返回 TBR 中，同时第 3 位-第 0 位返回 AC 中。

#### 1.5. 数据指针

数据指针能直接寻址数据存储器。指针地址储存在寄存器 DPH（3 位），DPM（3 位）和 DPL（4 位）。最大寻址范围为 3FFH。通过索引寄存器（INX），可以读写由 DPH, DPM 和 DPL 指定的数据存储器。

#### 1.6. 堆栈

堆栈是一组寄存器，在每次子程序调用或中断时能顺序保存 CY 和 PC（11-0）中的值。最高位为 CY 保留，同时它的结构为 13 位 X 8 层。当遇到返回指令（RTNI/RTNW）时，堆栈中的内容将按顺序返回到 PC 中。堆栈中的数据按照先进后出的方式处理。

#### 注意：

堆栈嵌套包括子程序调用和中断请求子程序调用，其最大值为 8 层。如果程序调用和中断请求的数量超过 8 层，堆栈底部将溢出，程序将无法正常运行。



2.2. 系统寄存器配置

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$00	IEAD	IET0	IET1	IEP	读/写	中断允许标志寄存器
\$01	IRQAD	IRQT0	IRQT1	IRQP	读/写	中断请求标志寄存器
\$02	REV0	T0M.2	T0M.1	T0M.0	读/写	第 2-0 位: 定时器 0 模式寄存器 第 3 位: 保留, 在程序中一直保持写 0 操作
\$03	REV0	T1M.2	T1M.1	T1M.0	读/写	第 2-0 位: 定时器 1 模式寄存器 第 3 位: 保留, 在程序中一直保持写 0 操作
\$04	T0L.3	T0L.2	T0L.1	T0L.0	读/写	定时器 0 载入/计数器低位寄存器
\$05	T0H.3	T0H.2	T0H.1	T0H.0	读/写	定时器 0 载入/计数器高位寄存器
\$06	T1L.3	T1L.2	T1L.1	T1L.0	读/写	定时器 1 载入/计数器低位寄存器
\$07	T1H.3	T1H.2	T1H.1	T1H.0	读/写	定时器 1 载入/计数器高位寄存器
\$08	REV0	REV0	REV0	PA.0	读/写	PORTA 数据寄存器 第 1-3 位: 保留, 在程序中一直保持写 0 操作
\$09	PB.3	REV0	REV0	REV0	读/写	PORTB 数据寄存器 第 0-2 位: 保留, 在程序中一直保持写 0 操作
\$0A	REV0	PC.2	REV0	REV0	读/写	PORTC 数据寄存器 第 0-1 位和第 3 位: 保留, 在程序中一直保持写 0 操作
\$0B	REV0	REV0	REV0	PD.0	读/写	PORTD 数据寄存器 第 1-3 位: 保留, 在程序中一直保持写 0 操作
\$0C	-	-	PE.1	PE.0	读/写	PORTE 数据寄存器
\$0D	-	-	-	-	-	保留
\$0E	TBR.3	TBR.2	TBR.1	TBR.0	读/写	查表寄存器
\$0F	INX.3	INX.2	INX.1	INX.0	读/写	索引寄存器
\$10	DPL.3	DPL.2	DPL.1	DPL.0	读/写	索引地址低位寄存器 (4 位)
\$11	-	DPM.2	DPM.1	DPM.0	读/写	索引地址中位寄存器 (3 位)
\$12	-	DPH.2	DPH.1	DPH.0	读/写	索引地址高位寄存器 (3 位)
\$13	T1GO	REV0	-	REV0	读/写	第 3 位: Timer1 功能启动控制寄存器 第 0 位和第 2 位: 保留, 在程序中一直保持写 0 操作
\$14	REV0	-	-	ADCON	读/写	第 0 位: ADC 转换允许设置寄存器 第 3 位: 保留, 在程序中一直保持写 0 操作
\$15	GO/DONE	TADC1	TADC0	ADCS	读/写	第 0 位: ADC 转换时间选择寄存器 第 2-1 位: ADC 时钟周期选择寄存器 第 3 位: ADC 转换控制/状态标志寄存器
\$16	ACR3	ACR2	ACR1	ACR0	读/写	ADC 端口配置控制寄存器
\$17	CH3	CH2	CH1	CH0	读/写	ADC 转换通道选择寄存器
\$18	REV1	REV1	REV1	PACR.0	读/写	PORTA 输入/输出控制寄存器 第 1-3 位: 保留, 在程序中一直保持写 1 操作
\$19	PBCR.3	REV1	REV1	REV1	读/写	PORTB 输入/输出控制寄存器 第 0-2 位: 保留, 在程序中一直保持写 1 操作



2.2. 系统寄存器配置 (续前表)

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$1A	REV1	PCCR.2	REV1	REV1	读/写	PORTC 输入/输出控制寄存器 第 0-1 位和第 3 位: 保留, 在程序中一直保持写 1 操作
\$1B	REV1	REV1	REV1	PDCR.0	读/写	PORTD 输入/输出控制寄存器 第 1-3 位: 保留, 在程序中一直保持写 1 操作
\$1C	-	-	-	PECR.0	读/写	PORTE 输入/输出控制寄存器
\$1D	-	-	-	-	-	保留
\$1E	- WDT	WDT.2 -	WDT.1 -	WDT.0 -	读/写 只读	第 2-0 位: 看门狗定时器控制寄存器 第 3 位: 看门狗定时器溢出标志寄存器 (只读)
\$1F	-	-	-	-	-	保留
\$20	PWM0S	T0CK1	T0CK0	PWM0_EN	读/写	第 0 位: PWM0 输出允许设置寄存器 第 2-1 位: PWM0 时钟选择寄存器 第 3 位: PWM0 占空比的输出模式设置寄存器
\$21	REV0	REV0	REV0	REV0	读/写	保留, 在程序中一直保持写 0 操作
\$22	PP0.3	PP0.2	PP0.1	PP0.0	读/写	PWM0 周期低位寄存器
\$23	PP0.7	PP0.6	PP0.5	PP0.4	读/写	PWM0 周期高位寄存器
\$24	-	-	PDF0.1	PDF0.0	读/写	PWM0 占空比微调设置寄存器 (2 位)
\$25	PD0.3	PD0.2	PD0.1	PD0.0	读/写	PWM0 占空比低位寄存器
\$26	PD0.7	PD0.6	PD0.5	PD0.4	读/写	PWM0 占空比高位寄存器
\$27 - 2C	-	-	-	-	-	保留
\$2D	-	-	A1	A0	只读	ADC 数据 (低 2 位) 寄存器 (只读)
\$2E	A5	A4	A3	A2	只读	ADC 数据 (中 4 位) 寄存器 (只读)
\$2F	A9	A8	A7	A6	只读	ADC 数据 (高 4 位) 寄存器 (只读)
\$380	RDT.3	RDT.2	RDT.1	RDT.0	读/写	ROM 数据查表地址/数据寄存器
\$381	RDT.7	RDT.6	RDT.5	RDT.4	读/写	ROM 数据查表地址/数据寄存器
\$382	RDT.11	RDT.10	RDT.9	RDT.8	读/写	ROM 数据查表地址/数据寄存器
\$383	RDT.15	RDT.14	RDT.13	RDT.12	读/写	ROM 数据查表地址/数据寄存器
\$384	REV0	REV0	REV0	PDIEN.0	读/写	PORTD.0 中断允许标志寄存器 第 1-3 位: 保留, 在程序中一直保持写 0 操作
\$385	-	-	-	PDIF.0	读/写	PORTD.0 中断请求标志寄存器
\$386	PBIEN.3	REV0	REV0	REV0	读/写	PORTB.3 中断允许标志寄存器 第 0-2 位: 保留, 在程序中一直保持写 0 操作
\$387	PBIF.3	-	-	-	读/写	PORTB.3 中断请求标志寄存器
\$388	-	-	-	PPACR.0	读/写	PORTA.0 上拉控制寄存器
\$389	PPBCR.3	-	-	-	读/写	PORTB.3 上拉控制寄存器
\$38A	-	PPCCR.2	-	-	读/写	PORTC.2 上拉控制寄存器
\$38B	-	-	-	PPDCR.0	读/写	PORTD.0 上拉控制寄存器
\$38C	-	-	-	PPECR.0	读/写	PORTE.0 上拉控制寄存器

**3. ROM**

ROM 能寻址 2048 X 16 位程序空间，地址由\$0000 到\$07FF。

**3.1. 矢量地址区 (\$000 到\$004)**

程序顺序执行。从地址\$000 到\$004 的区域是为特殊中断服务程序保留的，作为中断服务的入口地址。

地址	指令	说明
\$000	JMP*	跳转至RESET服务程序
\$001	JMP*	跳转至 ADC 中断服务程序
\$002	JMP*	跳转至 Timer0 中断服务程序
\$003	JMP*	跳转至 Timer1 中断服务程序
\$004	JMP*	跳转至 PORTB/D 中断服务程序

\*JMP 指令能由任意指令代替。

**3.2. ROM 数据读出列表****系统寄存器\$380 - \$383**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$380	RDT.3	RDT.2	RDT.1	RDT.0	读/写	ROM 数据列表地址/数据寄存器
\$381	RDT.7	RDT.6	RDT.5	RDT.4	读/写	ROM 数据列表地址/数据寄存器
\$382	RDT.11	RDT.10	RDT.9	RDT.8	读/写	ROM 数据列表地址/数据寄存器
\$383	RDT.15	RDT.14	RDT.13	RDT.12	读/写	ROM 数据列表地址/数据寄存器

RDT 寄存器由一个 12 位只写地址寄存器 (RDT.11 - RDT.0) 和一个 16 位只读 ROM 表数据读出寄存器组成 (RDT.15 - RDT.0)。为了读出 ROM 表数据，用户应该先写入 ROM 表地址到 RDT 寄存器 (先是高半字节后低半字节)，在一条指令后，指定地址的数据将会自动存入 RDT 寄存器 (写地址的最位寄存器将会启动数据读出动作)。



4. 初始状态

4.1. 系统寄存器初始状态

地址	第 3 位	第 2 位	第 1 位	第 0 位	上电复位 /Reset 引脚复位/低电压复位	WDT 复位
\$00	IEAD	IET0	IET1	IEP	0000	0000
\$01	IRQAD	IRQT0	IRQT1	IRQP	0000	0000
\$02	REV0	T0M.2	T0M.1	T0M.0	0000	uuuu
\$03	REV0	T1M.2	T1M.1	T1M.0	0000	uuuu
\$04	T0L.3	T0L.2	T0L.1	T0L.0	xxxx	xxxx
\$05	T0H.3	T0H.2	T0H.1	T0H.0	xxxx	xxxx
\$06	T1L.3	T1L.2	T1L.1	T1L.0	xxxx	xxxx
\$07	T1H.3	T1H.2	T1H.1	T1H.0	xxxx	xxxx
\$08	REV0	REV0	REV0	PA.0	0000	0000
\$09	PB.3	REV0	REV0	REV0	0000	0000
\$0A	REV0	PC.2	REV0	REV0	0000	0000
\$0B	REV0	REV0	REV0	PD.0	0000	0000
\$0C	-	-	PE.1	PE.0	--10	--10
\$0D	-	-	-	-	----	----
\$0E	TBR.3	TBR.2	TBR.1	TBR.0	xxxx	uuuu
\$0F	INX.3	INX.2	INX.1	INX.0	xxxx	uuuu
\$10	DPL.3	DPL.2	DPL.1	DPL.0	xxxx	uuuu
\$11	-	DPM.2	DPM.1	DPM.0	-xxx	-uuu
\$12	-	DPH.2	DPH.1	DPH.0	-xxx	-uuu
\$13	T1GO	REV0	-	REV0	00-0	0u-u
\$14	REV0	-	-	ADCON	0--0	u--0
\$15	GO/DONE	TADC1	TADC0	ADCS	0000	0uuu
\$16	ACR3	ACR2	ACR1	ACR0	0000	uuuu
\$17	CH3	CH2	CH1	CH0	0000	uuuu
\$18	REV1	REV1	REV1	PACR.0	0000	0000
\$19	PBCR.3	REV1	REV1	REV1	0000	0000
\$1A	REV1	PCCR.2	REV1	REV1	0000	0000
\$1B	REV1	REV1	REV1	PDCR.0	0000	0000
\$1C	-	-	-	PECR.0	---0	---0
\$1D	-	-	-	-	----	----
\$1E	WDT	WDT.2	WDT.1	WDT.0	0000	1000
\$1F	-	-	-	-	----	----
\$20	PWM0S	T0CK1	T0CK0	PWM0_EN	0000	uuu0

说明：x = 不定，u = 未更改，- = 未使用，读出值为'0'。





系统寄存器初始状态（续前表）

地址	第 3 位	第 2 位	第 1 位	第 0 位	上电复位 /Reset 引脚复位/低电压复位	WDT 复位
\$21	REV0	REV0	REV0	REV0	0000	uuu0
\$22	PP0.3	PP0.2	PP0.1	PP0.0	xxxx	uuuu
\$23	PP0.7	PP0.6	PP0.5	PP0.4	xxxx	uuuu
\$24	-	-	PDF0.1	PDF0.0	--xx	--uu
\$25	PD0.3	PD0.2	PD0.1	PD0.0	xxxx	uuuu
\$26	PD0.7	PD0.6	PD0.5	PD0.4	xxxx	uuuu
\$27 - 2C	-	-	-	-	----	----
\$2D	-	-	A1	A0	--xx	--uu
\$2E	A5	A4	A3	A2	xxxx	uuuu
\$2F	A9	A8	A7	A6	xxxx	uuuu
\$380	RDT.3	RDT.2	RDT.1	RDT.0	xxxx	uuuu
\$381	RDT.7	RDT.6	RDT.5	RDT.4	xxxx	uuuu
\$382	RDT.11	RDT.10	RDT.9	RDT.8	xxxx	uuuu
\$383	RDT.15	RDT.14	RDT.13	RDT.12	xxxx	uuuu
\$384	REV0	REV0	REV0	PDIF.0	0000	0000
\$385	-	-	-	PDIF.0	---0	---0
\$386	PBIEN.3	REV0	REV0	REV0	0000	0000
\$387	PBIF.3	-	-	-	0---	0---
\$388	-	-	-	PPACR.0	---0	---0
\$389	PPBCR.3	-	-	-	0---	0---
\$38A	-	PPCCR.2	-	-	-0--	-0--
\$38B	-	-	-	PPDCR.0	---0	---0
\$38C	-	-	-	PPECR.0	---0	---0

说明：x = 不定，u = 未更改，- = 未使用，读出值为'0'。

4.2. 其它初始状态

其它	复位后
程序计数器 (PC)	\$000
CY	不定
累加器 (AC)	不定
数据存储器	不定



### 5. 系统时钟和振荡器

振荡器振荡产生的脉冲为 CPU 和片上电路提供系统时钟。

系统时钟 =  $f_{osc}/4$

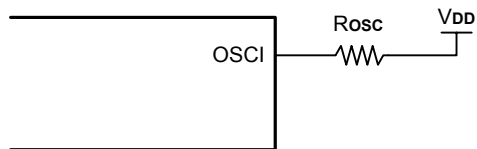
#### 5.1. 指令周期

(1) 对于 4MHz 的振荡器，为 4/4MHz (=1 $\mu$ s)

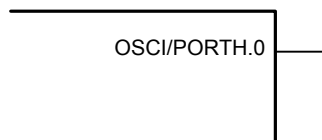
(2) 对于 10MHz 的振荡器，为 4/10MHz (= 0.4 $\mu$ s)

#### 5.2 振荡器类型

(1) RC 振荡器：400kHz - 10MHz

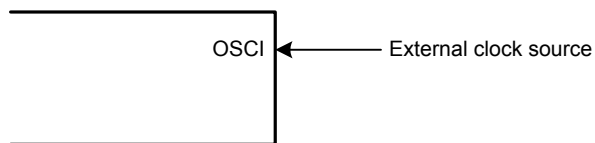


外部 RC



内建 RC ( $f_{osc} = 4\text{MHz}$ , 典型值)

(2) 外部时钟输入：30kHz - 10MHz



#### 注意：

如果选择内建 RC 振荡器，OSCI 引脚用作 PORTE.0。

**6. I/O 端口**

SH69P848A 提供 5 个双向 I/O 端口和一个开漏输出端口 (PORTE.1)。端口数据在寄存器\$08 - \$0C 中。端口控制寄存器 (\$18 - \$1C)控制端口输入或者输出。每个 I/O(不包括开漏输出端口)引脚包含上拉电阻,通过各自端口上拉控制寄存器(\$388 - \$38C)相对应位的值来控制。

- 当端口被选择作为输入端口,写“1”到各自相对的端口上拉控制寄存器(\$388 - \$38C)可以打开上拉电阻,写“0”可以关闭上拉电阻。
- 不论各自端口上拉控制寄存器(\$388 - \$38C)相对应位的值是多少,当端口作为输出端口时,上拉电阻将会自动关闭。
- 当 PORTB/D 被设置为输入端口,端口上的下降沿可以触发端口中断(如果端口中断已经允许)。

**系统寄存器\$08 - \$0C: 端口数据寄存器**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$08	REV0	REV0	REV0	PA.0	读/写	PORTA 数据寄存器
\$09	PB.3	REV0	REV0	REV0	读/写	PORTB 数据寄存器
\$0A	REV0	PC.2	REV0	REV0	读/写	PORTC 数据寄存器
\$0B	REV0	REV0	REV0	PD.0	读/写	PORTD 数据寄存器
\$0C	-	-	PE.1	PE.0	读/写	PORTE 数据寄存器

注: 所有的 REV0 位是保留的,在程序中保持写 0 操作。

**系统寄存器\$18 - \$1C: 输入/输出控制寄存器**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$18	REV1	REV1	REV1	PACR.0	读/写	PORTA 输入/输出控制寄存器
\$19	PBCR.3	REV1	REV1	REV1	读/写	PORTB 输入/输出控制寄存器
\$1A	REV1	PCCR.2	REV1	REV1	读/写	PORTC 输入/输出控制寄存器
\$1B	REV1	REV1	REV1	PDCR.0	读/写	PORTD 输入/输出控制寄存器
\$1C	-	-	-	PECR.0	读/写	PORTE 输入/输出控制寄存器

PA (/B/C/D/E) CR.n, (n = 0, 1, 2, 3)

0: 设置为输入口(初始值)

1: 设置为输出口

注: 所有的 REV1 位是保留的,在程序中保持写 1 操作。

**系统寄存器\$388 - \$38C: 端口上拉控制寄存器**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$388	-	-	-	PPACR.0	读/写	PORTA 上拉控制寄存器
\$389	PPBCR.3	-	-	-	读/写	PORTB 上拉控制寄存器
\$38A	-	PPCCR.2	-	-	读/写	PORTC 上拉控制寄存器
\$38B	-	-	-	PPDCR.0	读/写	PORTD 上拉控制寄存器
\$38C	-	-	-	PPECR.0	读/写	PORTE 上拉控制寄存器

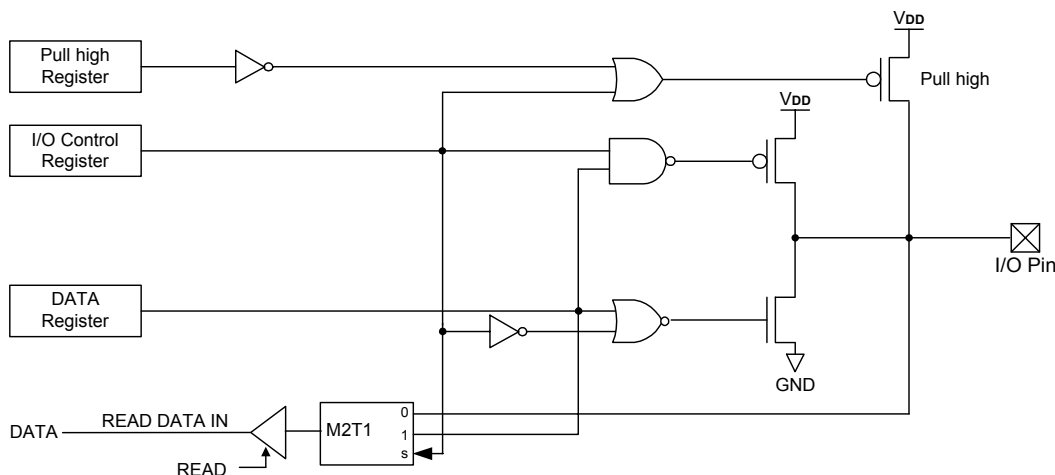
PPA (/B/C/D/E) CR.n, (n = 0, 1, 2, 3)

0: 禁止内部上拉电阻(初始值)

1: 允许内部上拉电阻



I/O 引脚的等效电路



- 在 SH69P848A，每个输出端口包含一个锁存器，用来保存输出数据。在输出模式下写端口数据寄存器（PDR）可以直接传输数据到相应的端口。  
所有输入端口不包含锁存器，所以外部输入电平要一直保持到端口读取动作完成。  
当某个端口被设置为输出口时，对该端口位的读操作会读取锁存器内的数据，而非端口外部电平。
- PORTA.0 可以共用作为 ADC 输入通道（AN0）。
- PORTB.3 可以共用作为 ADC 输入通道（AN7）。
- PORTD.0 可以共用作为 ADC AN8-9 输入通道（AN8）。
- 当 SH69P848A 使用内部 RC 振荡器作为系统振荡器时，OSCI 引脚可以通过代码选项，设置为 PORTE.0。

端口中断

PORTB 和 PORTD 用作外部端口中断源。由于 PORTB 和 PORTD 是位可编程 I/O，因此只有数字信号 V<sub>DD</sub> 到 GND 输入能产生端口中断。模拟输入无法产生中断请求。

中断控制标志映像为系统寄存器的 \$384 - \$387。通过软件可以访问和测试这四个寄存器。芯片复位初始化后所有标志清 0。

端口中断（包括其它外部中断源）可以用来将 CPU 从 HALT 或者 STOP 模式唤醒。

系统寄存器 \$384，\$386：端口中断允许标志寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$384	REV0	REV0	REV0	PDIEN.0	读/写	PORTD.0 中断允许标志位寄存器 第 1-3 位保留，在程序中保持写 0 操作
\$386	PBIEN.3	REV0	REV0	REV0	读/写	PORTB.3 中断允许标志位寄存器 第 0-2 位保留，在程序中保持写 0 操作

PDIEN.n, PBIEN.n (n = 0, 1, 2, 3)

0: 禁止端口中断（初始值）

1: 允许端口中断



系统寄存器\$385, \$387: 端口中断请求寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$385	-	-	-	PDIF.0	读/写	PORTD.0 中断请求标志位寄存器
\$387	PBIF.3	-	-	-	读/写	PORTB.3 中断请求标志位寄存器

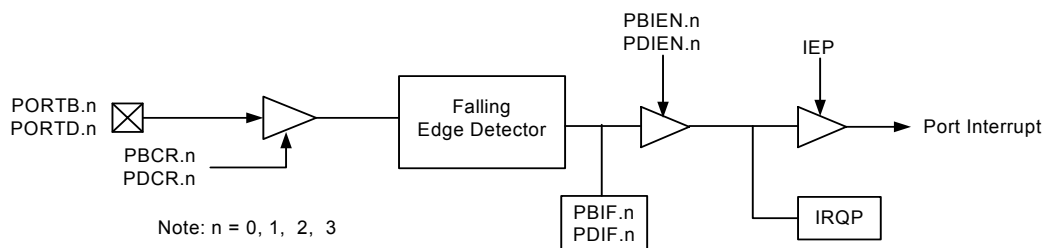
PDIF.n, PBIF.n (n = 0, 1, 2, 3)

0: 没有发生端口中断 (初始值)

1: 已经发生端口中断

该寄存器只能清 0。

以下为端口中断功能模块图, 仅供参考。



端口中断功能块图

端口中断编程注意事项:

1. 设置端口为输入端口, 写 1 到端口上拉控制寄存器相应位。
2. 端口上的下降沿将会置 PIF.n = "1"。如果 PIEN.x = 1, 将产生中断请求 (IRQP = 1)。
3. 端口中断可以从 HALT 或 STOP 模式唤醒 CPU。

ADC 允许寄存器\$14:

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$14	REV0	-	-	ADCON	读/写	第 0 位: ADC 转换允许设置寄存器 第 3 位保留, 在程序中保持写 0 操作
	X	-	-	0	读/写	禁止 ADC 工作
	X	-	-	1	读/写	允许 ADC 工作

当 ADC 被禁止时, PORTA.0, PORTB.3 及 PORTD.0 被用作标准 I/O 端口。当 ADC 被允许, 设置 ADC 端口配置寄存器 (\$16) 来选择 PORTA.0, PORTB.3 和 PORTD.0 中的任何一个作为标准 I/O 端口或 ADC 端口。详细情况请参考 ADC 转换器描述。

PORTC.2 可以与 PWM0 输出共用 (PWM0)。

PWM0 控制寄存器\$20:

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$20	PWM0S	T0CK1	T0CK0	PWM0_EN	读/写	第 0 位: PWM0 输出允许设置寄存器 第 2-1 位: PWM0 时钟选择寄存器 第 3 位: PWM0 占空比的输出模式设置寄存器
	X	X	X	0	读/写	设置 PORTC.2 为 I/O 端口 (初始值)
	X	X	X	1	读/写	设置 PORTC.2 为 PWM0 输出端口和生效 PWM0



## 7. 定时器/计数器 (Timer)

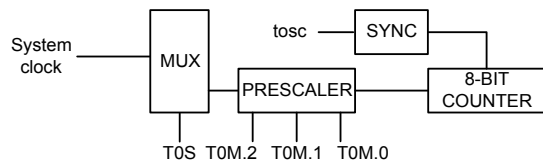
### 7.1. Timer0

SH69P848A 有两个 8 位定时器。

Timer0 有下述特性：

- 8 位递增计数
- 自动重载
- 8 级预分频
- 计数值由 \$FF 到 \$00 时，产生溢出中断请求

Timer0 框图：



Timer0 功能：

- 可编程定时功能
- 计数值可读

#### 7.1.1. Timer0 结构和操作

由一个 8 位只写载入寄存器 (TL0L, TL0H) 和一个 8 位只读计数器 (TC0, TC0H) 构成。每个定时器由低四位和高四位组成。将数据写入载入寄存器 (TL0L, TL0H) 就可以初始化 Timer。

#### 7.1.2. Timer0 模式寄存器

通过设置 Timer0 模式寄存器 TOM 可以使 Timer0 工作在不同的模式。

系统时钟经过预分频器分频后，进入计数器。Timer0 模式寄存器中 TOM.2-0 用于设定分频比。

Timer0 模式寄存器：\$02

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$02	REV0	TOM.2	TOM.1	TOM.0	读/写	第 2-0 位：定时器 0 模式寄存器 第 3 位保留，在程序中保持写 0 操作
	X	0	0	0	读/写	定时器时钟：系统时钟/2 <sup>11</sup>
	X	0	0	1	读/写	定时器时钟：系统时钟/2 <sup>9</sup>
	X	0	1	0	读/写	定时器时钟：系统时钟/2 <sup>7</sup>
	X	0	1	1	读/写	定时器时钟：系统时钟/2 <sup>5</sup>
	X	1	0	0	读/写	定时器时钟：系统时钟/2 <sup>3</sup>
	X	1	0	1	读/写	定时器时钟：系统时钟/2 <sup>2</sup>
	X	1	1	0	读/写	定时器时钟：系统时钟/2 <sup>1</sup>
	X	1	1	1	读/写	定时器时钟：系统时钟/2 <sup>0</sup>

当高 4 位载入寄存器被写入或 Timer 计数值由 \$FF 到 \$00 溢出时，Timer 将自动载入预设值。

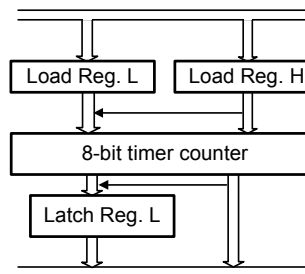
由于寄存器的高 4 位控制 Timer 的读写操作，使用中请根据以下步骤操作：

写操作

- 先写低四位
- 再写高四位以更新计数器

读操作

- 先读高四位
- 再读低四位



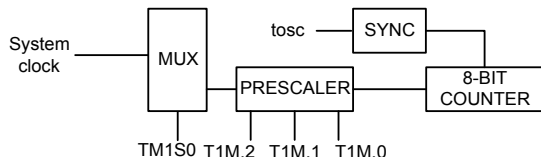


## 7.2. Timer1

Timer1 有下述特性:

- 8 位递增计数
- 自动重载
- 8 级预分频
- 计数值由\$FF 到\$00 时, 产生溢出中断请求

Timer1 框图:



Timer1 功能:

- 可编程定时功能
- 计数值可读

### 7.2.1. Timer1 结构和操作

由一个 8 位只写载入寄存器 (TL1L, TL1H) 和和一个 8 位只读计数器 (TC1L, TC1H) 构成。每个定时器由低四位和高四位组成。将数据写入载入寄存器 (TL1L, TL1H) 就可以初始化 Timer。

当高 4 位载入寄存器被写入或 Timer 计数值由\$FF 到\$00 溢出时, Timer 将自动载入预设值。

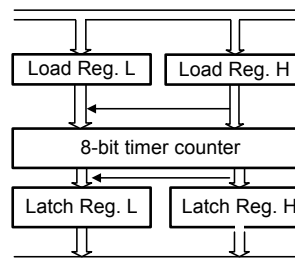
由于寄存器的高 4 位控制 Timer 的读写操作, 使用中请根据以下步骤操作:

写操作

- 先写低四位
- 再写高四位以更新计数器

读操作

- 先读高四位
- 再读低四位



### 7.2.2. Timer1 模式寄存器

通过设置 Timer1 模式寄存器 T1M 可以使 Timer1 工作在不同的模式。

系统时钟经过预分频器分频后, 进入计数器。Timer1 模式寄存器中 T1M.2-0 用于设定分频比。

Timer1 模式寄存器: \$03

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$03	REV0	T1M.2	T1M.1	T1M.0	读/写	第 2-0 位: 定时器 1 模式寄存器 第 3 位保留, 在程序中保持写 0 操作
	X	0	0	0	读/写	定时器时钟: 系统时钟/2 <sup>11</sup> (Default)
	X	0	0	1	读/写	定时器时钟: 系统时钟/2 <sup>9</sup>
	X	0	1	0	读/写	定时器时钟: 系统时钟/2 <sup>7</sup>
	X	0	1	1	读/写	定时器时钟: 系统时钟/2 <sup>5</sup>
	X	1	0	0	读/写	定时器时钟: 系统时钟/2 <sup>3</sup>
	X	1	0	1	读/写	定时器时钟: 系统时钟/2 <sup>2</sup>
	X	1	1	0	读/写	定时器时钟: 系统时钟/2 <sup>1</sup>
	X	1	1	1	读/写	定时器时钟: 系统时钟/2 <sup>0</sup>



## 7.2.3. Timer1 控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$13	T1GO	REV0	-	REV0	读/写	第 3 位: Timer1 开始/暂停寄存器 第 0 位和第 2 位保留, 在程序中保持写 0 操作
	0	X	-	X	读/写	定时器/计数器停止 (读: 状态; 写: 命令)
	1	X	-	X	读/写	定时器/计数器开始 (读: 状态; 写: 命令)

在这个模式里, Timer1 使用的是内部时钟, 当 Timer1 控制寄存器最高位 (\$07) 写操作完成, Timer1 控制寄存器 (\$06 - \$07) 的内容重载。如果 Timer1 控制寄存器 (\$13) T1GO (第 3 位) 置 1, 递增计数器开始计数。如果中断允许寄存器 (\$00) 第 1 位 (IET1) 设置为 1, 递增计数器从 \$FF 到 \$00 溢出时, Timer1 中断产生。

在 Timer1 控制寄存器 (\$13) 内 T1GO (第 3 位) 设置为 1 之后, Timer1 计数器寄存器 (\$06 - \$07) 不再影响递增计数器的工作。只有当 Timer1 控制寄存器 (\$13) 内 T1GO (第 3 位) 设置为 0 且最高位 (\$07) 写入时, Timer1 计数器寄存器 (\$06 - \$07) 才会重新载入递增计数器。

## Timer1 计数寄存器: \$06 - \$07

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$06	T1L.3	T1L.2	T1L.1	T1L.0	读/写	Timer1 载入/计数器低位寄存器
\$07	T1H.3	T1H.2	T1H.2	T1H.0	读/写	Timer1 载入/计数器高位寄存器





### 8. 中断

SH69P848A 有四个中断源:

- ADC 中断
- Timer0 中断
- Timer1 中断
- PORTB-D 中断 (下降沿)

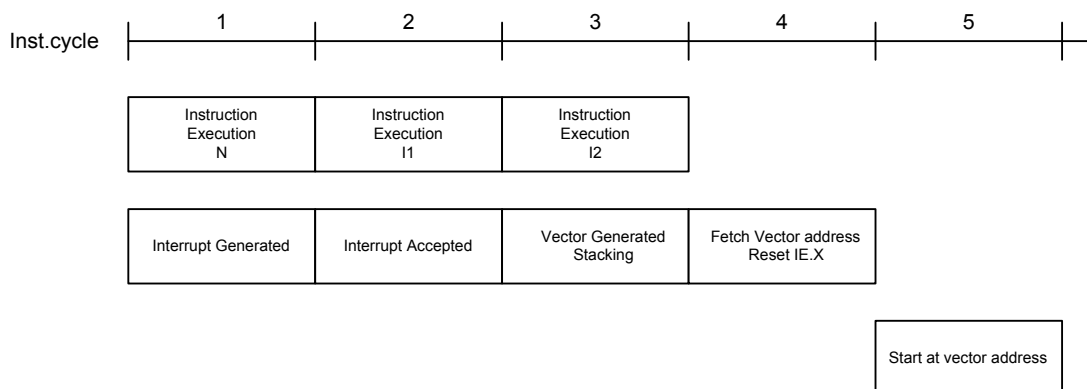
#### 中断控制标志位和中断服务

中断控制标志位为系统寄存器的\$00 和\$01。这两个寄存器能够由软件访问和设置。芯片上电复位后, 这些标志位被清 0。

#### 系统寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$00	IEAD	IET0	IET1	IEP	读/写	中断允许标志寄存器
\$01	IRQAD	IRQT0	IRQT1	IRQP	读/写	中断请求标志寄存器

当 IEx 设置为 1 且有中断请求时 (IRQx 为 1), 中断被激活并且根据中断优先级产生相应的中断矢量地址。当发生中断时, PC 和 CY 标志将被保存在堆栈存储器中, 同时程序跳转至中断服务矢量地址处执行。在中断发生后, 所有中断允许标志 (IEx) 自动复位为 0, 因此在 IRQx = 1 时 IEx 标志再次设置为 1 时, 将可能再次产生中断。



中断服务流程图

#### 中断嵌套

在 CPU 中断服务期间, 用户可以在中断返回前设置任何中断允许标志。中断服务流程图中标示下个中断和将要发生的下一个中断嵌套。如果中断请求已经产生且执行允许 IE 使能的指令 N, 那么在两个指令周期后将执行中断程序。但是, 如果指令 I1 或指令 I2 清除中断请求或允许标志, 那么中断服务将被取消。

#### ADC 中断

系统寄存器\$00 (IEX) 的第 3 位是 ADC 中断使能标志位。当完成 A/D 转换后将产生一个中断请求 (IRQAD = 1), 如果 ADC 中断使能 (IEAD = 1), 将启动一次 ADC 中断服务程序。该 A/D 中断能用来从 HALT 方式唤醒 CPU。

#### 定时器 (Timer0, Timer1) 中断

Timer0 和 Timer1 的时钟输入是以系统时钟 (Timer0 或以外部时钟/事件 T0) 输入为基准的。定时器由\$FF 到\$00 计数溢出, 将产生一个内部中断请求 (IRQT0 或者 IRQT1 = 1), 如果中断使能标志允许 (IET0 或者 IET1 = 1) 则开始定时器中断服务程序。定时器中断同样也能用于从 HALT 方式唤醒 CPU。

#### 端口下降沿中断

只有数字输入端口可以产生端口中断。模拟输入不能产生中断请求。

I/O 端口的任何输入引脚上的下降沿将产生中断请求 (IRQP = 1)。其后的下降沿不会产生中断请求直到所有的的引脚返回到 VDD。

#### 通过位控制端口中断

只有数字输入端口可以产生端口中断。模拟输入不能产生中断请求。

**系统寄存器\$384, \$386: 端口中断允许寄存器**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$384	REV0	REV0	REV0	PDIF.0	读/写	PORTD.0 中断允许标志位寄存器 第 1-3 位保留, 在程序中保持写 0 操作
\$386	PBIEN.3	REV0	REV0	REV0	读/写	PORTB 中断允许标志位寄存器 第 0-2 位保留, 在程序中保持写 0 操作

**系统寄存器\$385, \$387: 端口中断请求寄存器**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$385	-	-	-	PDIF.0	读/写	PORTD.0 中断请求标志位寄存器
\$387	PBIF.3	-	-	-	读/写	PORTB.3 中断请求标志位寄存器



9. 模/数转换器 (ADC)

SH69P848A 内建有一个 3 通道的 10 位逐次逼近型模/数转换器 (ADC)。

ADC 控制寄存器：这些寄存器定义了模/数转换模拟通道数设置，转换通道选择，模/数转换时钟选择，模/数转换启动控制位和结束标志。模/数转换结果寄存器为只读寄存器。

模/数转换的步骤：

- 设置模拟通道数。(任何模拟输入电压值不能大于  $V_{DD}$ )
- 运行模/数转换器，选择需转换的信号通道。
- 设置模/数转换时钟源。
- 置  $\overline{GO/DONE} = 1$ ，启动模/数转换。

系统寄存器\$14：模/数转换端口配置控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$14	REV0	-	-	ADCON	读/写	第 0 位：ADC 转换允许设置寄存器 第 3 位保留，在程序中保持写 0 操作
	X	-	-	0	读/写	禁止 ADC 工作
	X	-	-	1	读/写	允许 ADC 工作

系统寄存器\$16

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$16	ACR3	ACR2	ACR1	ACR0	读/写	ADC 端口配置控制寄存器
	0	0	0	0	读/写	参见下表

设置模拟通道

ACR3	ACR2	ACR1	ACR0	8	7	0
0	0	0	0	PD.0	PB3	PA.0
0	0	0	1	PD.0	PB.3	AN0
0	0	1	0	PD.0	PB.3	AN0
0	0	1	1	PD.0	PB.3	AN0
0	1	0	0	PD.0	PB.3	AN0
0	1	0	1	PD.0	PB.3	AN0
0	1	1	0	PD.0	PB.3	AN0
0	1	1	1	PD.0	PB.3	AN0
1	X	0	0	PD.0	AN7	AN0
1	X	0	1	AN8	AN7	AN0
1	X	1	X	AN8	AN7	AN0

系统寄存器\$17：模/数转换通道控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$17	CH3	CH2	CH1	CH0	读/写	ADC 转换通道选择寄存器
	0	0	0	0	读/写	选择 ADC 通道 AN0
	0	1	1	1	读/写	选择 ADC 通道 AN7
	1	X	X	0	读/写	选择 ADC 通道 AN8

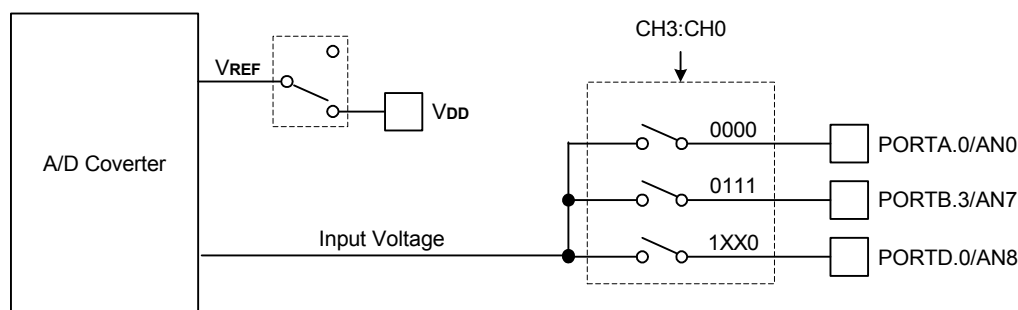


系统寄存器\$2D - \$2F: 模/数转换结果寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$2D	X	X	A1	A0	只读	ADC 数据 (低 2 位) 寄存器 (只读)
\$2E	A5	A4	A3	A2	只读	ADC 数据 (中 4 位) 寄存器 (只读)
\$2F	A9	A8	A7	A6	只读	ADC 数据 (高 4 位) 寄存器 (只读)

系统寄存器\$15: 模/数转换控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$15	GO/ $\overline{\text{DONE}}$	TADC1	TADC0	ADCS	读/写	第 0 位: ADC 转换时间选择寄存器 第 2-1 位: ADC 时钟周期选择寄存器 第 3 位: ADC 转换控制/状态标志寄存器
	X	X	X	0	读/写	模/数转换时间 = 204 $t_{AD}$
	X	X	X	1	读/写	模/数转换时间 = 780 $t_{AD}$
	X	0	0	X	读/写	模/数转换时钟周期 $t_{AD} = t_{OSC}$
	X	0	1	X	读/写	模/数转换时钟周期 $t_{AD} = 4t_{OSC}$
	X	1	0	X	读/写	模/数转换时钟周期 $t_{AD} = 8t_{OSC}$
	X	1	1	X	读/写	模/数转换时钟周期 $t_{AD} = 16t_{OSC}$
	0	X	X	X	读/写	模/数转换完成
	1	X	X	X	读/写	当 $\text{ADCON} = 1$ , 模/数转换正在进行



注意:

- 正确选择模/数转换时钟周期  $t_{AD}$ , 保证  $1\mu s \leq t_{AD} \leq 33.4\mu s$
- 当完成模/数转换后, 将产生模/数转换中断 (如果模/数转换器中断允许)
- 模拟输入通道必须将其对应的  $PXCR$  ( $X = A, B, D$ ) 位作为输入
- 如果 I/O 端口已被选择为模拟输入端, 那么 I/O 功能和上拉电阻被禁止
- 当模/数转换完成后  $\overline{\text{GO/DONE}}$  位由硬件自动清零
- 在转换过程中将  $\overline{\text{GO/DONE}}$  位清零会中止当前的转换过程
- 尚未完成转换过程而将  $\overline{\text{GO/DONE}}$  位清零的模/数转换将不会更新模/数转换结果寄存器内容
- 在下一个模/数转换开始前需要等待  $4t_{OSC}$  的时间
- ADC 能在 HALT 方式下继续工作, 但在执行"STOP"指令后自动停止工作
- ADC 能将 CPU 从 HALT 方式下唤醒 (如果 ADC 中断允许)



10. 脉冲宽度调制 (PWM)

SH69P848A 包含 1 个 8+2 位 PWM 模块。PWM 模块可以产生周期和占空比分别可以调整的脉宽调制波形。PWMC 被用来控制 PWM 模块的工作模式。PWMP 被用来控制 PWM 输出的周期。而 PWMD 被用来控制 PWM 输出波形的占空比。

系统寄存器\$20, \$21: PWM 控制寄存器 (PWMC)

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$20	PWM0S	T0CK1	T0CK0	PWM0_EN	读/写	第 0 位: PWM0 输出允许设置寄存器 第 2-1 位: PWM0 时钟选择寄存器 第 3 位: PWM0 占空比的输出模式设置寄存器
	X	X	X	0	读/写	选择为 I/O 端口 (初始值)
	X	X	X	1	读/写	选择为 PWM0 输出端口
	X	0	0	X	读/写	PWM0 时钟 = t <sub>osc</sub> (初始值)
	X	0	1	X	读/写	PWM0 时钟 = 2t <sub>osc</sub>
	X	1	0	X	读/写	PWM0 时钟 = 4t <sub>osc</sub>
	X	1	1	X	读/写	PWM0 时钟 = 8t <sub>osc</sub>
	0	X	X	X	读/写	PWM0 占空比输出普通模式 (高电平有效) (初始值)
	1	X	X	X	读/写	PWM0 占空比输出负极模式 (低电平有效)

PWM0 输出引脚与 PORTC.2 共用。

系统寄存器\$22 - \$23: PWM 周期控制寄存器 (PWMP)

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$22	PP0.3	PP0.2	PP0.1	PP0.0	读/写	PWM0 周期低位寄存器
\$23	PP0.7	PP0.6	PP0.5	PP0.4	读/写	PWM0 周期高位寄存器

PWM 输出周期 = [PP0.7, PP0.0] X PWM0 时钟。

当[PP0.7, PP0.0] = 00H, 如果 PWM0S 位设置为 0, PWM0 输出低电平。

当[PP0.7, PP0.0] = 00H, 如果 PWM0S 位设置为 1, PWM0 输出高电平。

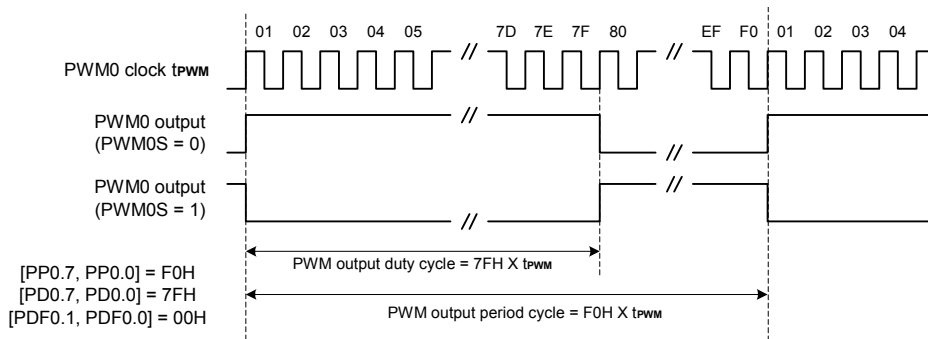
系统寄存器\$24 - \$26: PWM 占空比控制寄存器 (PWMD)

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$24	-	-	PDF0.1	PDF0.0	读/写	PWM0 占空比微调设置寄存器 (2 位)
\$25	PD0.3	PD0.2	PD0.1	PD0.0	读/写	PWM0 占空比低位寄存器
\$26	PD0.7	PD0.6	PD0.5	PD0.4	读/写	PWM0 占空比高位寄存器

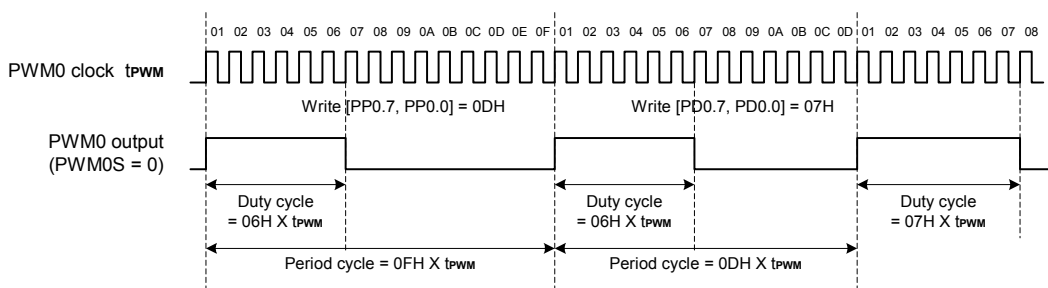
PWM0 输出占空比 = ([PD0.7, PD0.0] + [PDF0.1, PDF0.0]/4) X PWM0 时钟。

如果[PP0.7, PP0.0] ≤ [PD0.7, PD0.0], 当 PWM0S 位设置为 0, PWM0 输出高电平。

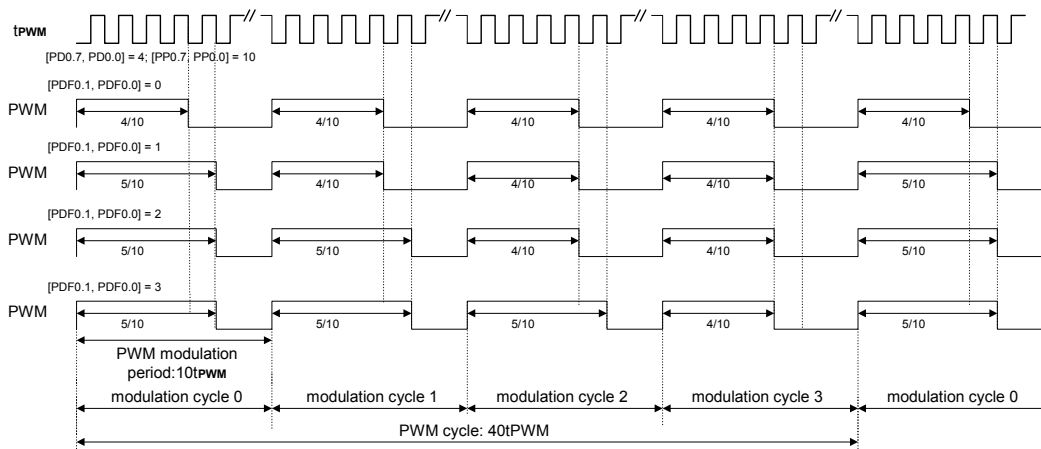
如果[PP0.7, PP0.0] ≤ [PD0.7, PD0.0], 当 PWM0S 位设置为 1, PWM0 输出低电平。



PWM 基本输出波形



PWM 输出周期或者占空比周期变化举例



(8+2) bits PWM 输出波形



在 (8+2) bits PWM 波形中，四个 PWM 周期构成一组（周期 0 - 周期 3），每组中各周期的占空比如下表：

参数	[PDF0.1, PDF0.0] (0-3)	PWM 周期占空比
周期 I (I = 0-3)	$I < [PDF0.1, PDF0.0]$	$([PD0.7, PD0.0] + 1) / [PP0.7, PP0.0]$
	$I \geq [PDF0.1, PDF0.0]$	$[PD0.7, PD0.0] / [PP0.7, PP0.0]$

周期组占空比如下表：

PWM 周期	PWM 周期组	PWM 周期组占空比
$[PP0.7, PP0.0] \times t_{PWM}$	$4 * [PP0.7, PP0.0] \times t_{PWM}$	$(4 \times [PD0.7, PD0.0] + [PDF0.1, PDF0.0]) / (4 \times [PP0.7, PP0.0])$

**编程注意事项：**

1. 选择 PWM 模块时钟源。
2. 通过写适当的值到 PWM 周期控制寄存器 (PWMP) 设置 PWM 周期：首先设置低 4 位，然后设置高 4 位。
3. 通过写适当的值到 PWM 占空比控制寄存器 (PWMD) 设置 PWM 占空比：先设置微调 2 位，然后设置低 4 位，最后设置高 4 位。
4. 通过写 PWM 控制寄存器 (PWMC) 的 PWMS 位选择 PWM 占空比的输出模式。
5. 为了输出适当的 PWM 波形，通过写 PWM 控制寄存器 (PWMC) 中的 PWM\_EN 位为“1”来允许 PWM 模块工作。
6. 如果 PWM 周期或者占空比需要改变，操作流程如同步骤 2 或者步骤 3 说明。修改后的重载入计数器的值在下一个周期开始有效。
7. 读出 PWM0 周期或者占空比控制寄存器的流程与写入的顺序相反，先读出高 4 位，然后读出低 4 位。
8. PWM 能在 HALT 模式下继续工作，并在执行"STOP"指令后自动停止工作。

**11. 低电压复位 (LVR)**

LVR 用于监控电源电压并产生芯片内部复位。它一般用于交流供电电路或有较大负载的电路，这些电路工作时负载的启动会引起器件工作电压暂时低于电路的最低允许工作电压。

LVR 功能的开启通过代码选项选择。

当 LVR 功能允许时其功能如下：

- 当  $V_{DD} \leq V_{LVR}$  时产生系统复位
- 当  $V_{DD} > V_{LVR}$  时释放系统复位

**12. 看门狗定时器 (WDT)**

看门狗定时器是一个递减计数器，拥有独立内建 RC 振荡器作为时钟源，因此虽然在 STOP 模式下仍在运行。当定时器溢出时，WDT 将复位 MCU。通过代码选项可以允许或禁止该功能。

WDT 控制位 (\$1E 第 2 位 - 第 0 位) 用来选择不同的溢出时间。定时器溢出后，WDT 溢出标志 (\$1E 第 3 位) 将由硬件自动设置为“1”。通过读或者写系统寄存器 \$1E，WDT 会在溢出前重新开始计数。

**系统寄存器 \$1E：看门狗定时器 (WDT)**

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$1E	- WDT	WDT.2 -	WDT.1 -	WDT.0 -	读/写 只读	第 2-0 位：看门狗定时器控制寄存器 第 3 位：看门狗定时器溢出标志寄存器（只读）
	X	0	0	0	读/写	WDT 溢出周期为 4096ms
	X	0	0	1	读/写	WDT 溢出周期为 1024ms
	X	0	1	0	读/写	WDT 溢出周期为 256ms
	X	0	1	1	读/写	WDT 溢出周期为 128ms
	X	1	0	0	读/写	WDT 溢出周期为 64ms
	X	1	0	1	读/写	WDT 溢出周期为 16ms
	X	1	1	0	读/写	WDT 溢出周期为 4ms
	X	1	1	1	读/写	WDT 溢出周期为 1ms
	0	X	X	X	只读	未发生 WDT 溢出复位
	1	X	X	X	只读	WDT 溢出，发生 WDT 复位

**注意：**看门狗定时器溢出周期是在  $V_{DD} = 5V$  时参考值。

**13. HALT 和 STOP 模式**

在执行 HALT 指令后，CPU 将进入 HALT 模式。在 HALT 模式下，CPU 将停止工作。但是其周边电路（定时器，ADC，...）将继续工作。

在执行 STOP 指令后，CPU 将进入 STOP 模式。在 STOP 模式下，整个芯片（包括振荡器）将停止工作。除了看门狗定时器电路有效。

在 HALT 模式下，发生任何中断 CPU 将被唤醒。

在 STOP 模式下，发生端口中断 CPU 将被唤醒。

当通过任何中断，CPU 从 HALT/STOP 被唤醒，将会首先执行相关中断服务子程序。然后才会执行 HALT/STOP 的下一条指令。





#### 14. 预热计数器

本芯片内建振荡器预热计数器，它能消除振荡器在下列情况下起振时的不稳定状态：

##### 上电复位及 Reset 引脚复位：

(1) 振荡器频率  $f_{osc} = 30\text{kHz} - 2\text{MHz}$  时，预热计数器预分频比为  $2^{12}$  (4096)

(2) 振荡器频率  $f_{osc} = 2\text{MHz} - 10\text{MHz}$  时，预热计数器预分频比为  $2^{14}$  (16384)

##### 由 STOP 模式唤醒, WDT 复位, LVR 复位：

在 RC 振荡器及外部时钟模式下，预热计数器预分频比为  $2^7$  (128)

#### 15. 代码选项

##### 15.1. 振荡器类型

OP\_OSC [2:0]:

000 = 外部时钟 (初始值)

001 = 内部 RC 振荡器 (4MHz)

010 = 内部 RC 振荡器 (4MHz)

011 = 内部 RC 振荡器 (4MHz)

100 = 外部 RC 振荡器 (400kHz - 10MHz)

##### 15.2. 振荡器范围

OP\_OSC 3:

0 = 2MHz - 10MHz (初始值)

1 = 30kHz - 2MHz

##### 15.3. 看门狗定时器

OP\_WDT:

0 = 允许 (初始值)

1 = 禁止

##### 15.4. 低电压复位

OP\_LVR:

0 = 禁止 (初始值)

1 = 允许

##### 15.5. 低电压复位电压范围

OP\_LVR0:

0 = 高 LVR 电压 (初始值)

1 = 低 LVR 电压



### OTP在系统烧写时注意事项

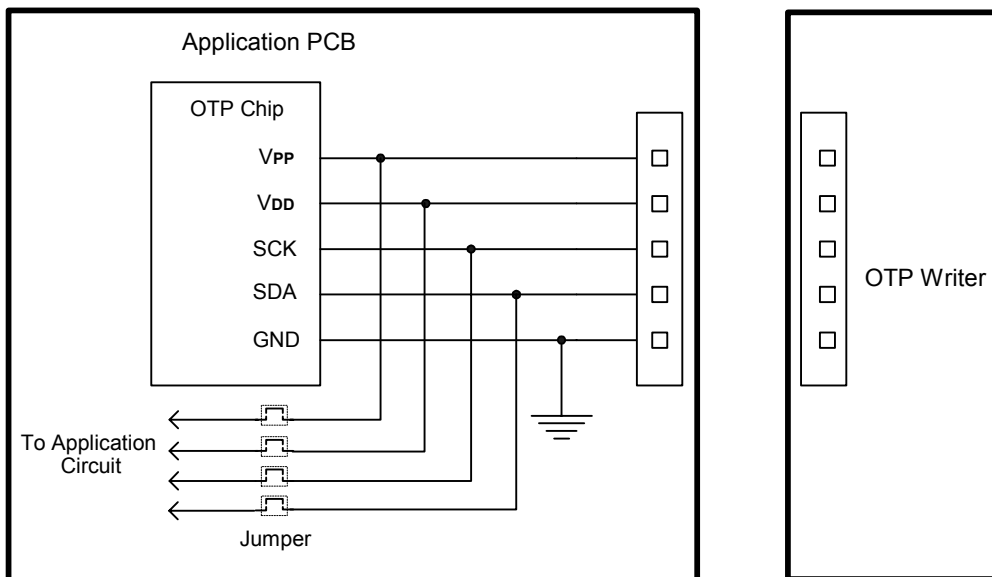
OTP在系统烧写时注意事项只对OTP芯片有效。

对于用户采用COB（Chip on Board）组装方式时，OTP芯片可以使用In System Programming（在系统烧写）方式编程。

为使用在系统烧写方式编程，用户必须在PCB上预留出OTP芯片的编程接口以便连接OTP编程器进行编写。

用户可先将OTP芯片组装到用户PCB上，对OTP芯片编程完成后再将其它器件组装到板上，也可在对OTP芯片编程前将包括OTP芯片在内的所有器件组装在用户PCB上。

为了提高OTP编程的成功率，在编程操作执行时OTP编程信号线直接连接到OTP编程器上，不允许有其它外加电路并联。所以在PCB板上必须预留4组跳线或分割焊盘将OTP编程接口（V<sub>DD</sub>，V<sub>PP</sub>，SDA，SCK）与应用电路分隔开，如下图所示：



具体操作步骤如下：

- (1) 在对组装在 PCB 上的 OTP 芯片编程前将 4 组跳线断开。
- (2) 将 OTP 芯片的编程接口连接到 OTP 编程器并完成代码编程。
- (3) 将用户板与 OTP 烧写器编程器断开，将 4 组跳线短接。

有关 OTP 编程的更多详细资料，请参见 OTP 编程器的用户手册。



## 指令集

所有的指令都是单周期和单字节的指令。具有面向存储器的操作特性。

## 1. 以下为算术和逻辑指令

## 1.1. 累加器类型

助记符	指令代码	功能	标志位改变
ADC X (, B)	00000 0bbb xxx xxxx	$AC \leftarrow Mx + AC + CY$	CY
ADCM X (, B)	00000 1bbb xxx xxxx	$AC, Mx \leftarrow Mx + AC + CY$	CY
ADD X (, B)	00001 0bbb xxx xxxx	$AC \leftarrow Mx + AC$	CY
ADDM X (, B)	00001 1bbb xxx xxxx	$AC, Mx \leftarrow Mx + AC$	CY
SBC X (, B)	00010 0bbb xxx xxxx	$AC \leftarrow Mx + -AC + CY$	CY
SBCM X (, B)	00010 1bbb xxx xxxx	$AC, Mx \leftarrow Mx + -AC + CY$	CY
SUB X (, B)	00011 0bbb xxx xxxx	$AC \leftarrow Mx + -AC + 1$	CY
SUBM X (, B)	00011 1bbb xxx xxxx	$AC, Mx \leftarrow Mx + -AC + 1$	CY
EOR X (, B)	00100 0bbb xxx xxxx	$AC \leftarrow Mx \oplus AC$	
EORM X (, B)	00100 1bbb xxx xxxx	$AC, Mx \leftarrow Mx \oplus AC$	
OR X (, B)	00101 0bbb xxx xxxx	$AC \leftarrow Mx   AC$	
ORM X (, B)	00101 1bbb xxx xxxx	$AC, Mx \leftarrow Mx   AC$	
AND X (, B)	00110 0bbb xxx xxxx	$AC \leftarrow Mx \& AC$	
ANDM X (, B)	00110 1bbb xxx xxxx	$AC, Mx \leftarrow Mx \& AC$	
SHR	11110 0000 000 0000	$0 \rightarrow AC[3], AC[0] \rightarrow CY;$ AC 右移 1 位	CY

## 1.2. 立即数类型

助记符	指令代码	功能	标志位改变
ADI X, I	01000 iiiii xxx xxxx	$AC \leftarrow Mx + I$	CY
ADIM X, I	01001 iiiii xxx xxxx	$AC, Mx \leftarrow Mx + I$	CY
SBI X, I	01010 iiiii xxx xxxx	$AC \leftarrow Mx + -I + 1$	CY
SBIM X, I	01011 iiiii xxx xxxx	$AC, Mx \leftarrow Mx + -I + 1$	CY
EORIM X, I	01100 iiiii xxx xxxx	$AC, Mx \leftarrow Mx \oplus I$	
ORIM X, I	01101 iiiii xxx xxxx	$AC, Mx \leftarrow Mx   I$	
ANDIM X, I	01110 iiiii xxx xxxx	$AC, Mx \leftarrow Mx \& I$	

## 1.3. 十进制调整

助记符	指令代码	功能	标志位改变
DAA X	11001 0110 xxx xxxx	AC, Mx <- 加法的十进制调整	CY
DAS X	11001 1010 xxx xxxx	AC, Mx <- 减法的十进制调整	CY



2. 传输指令

助记符	指令代码	功能	标志位改变
LDA X (, B)	00111 0bbb xxx xxxx	AC ← Mx	
STA X (, B)	00111 1bbb xxx xxxx	Mx ← AC	
LDI X, I	01111 iii xxx xxxx	AC, Mx ← I	

3. 控制指令

助记符	指令代码	功能	标志位改变
BAZ X	10010 xxxx xxx xxxx	PC ← X, 如果 AC = 0	
BNZ X	10000 xxxx xxx xxxx	PC ← X, 如果 AC ≠ 0	
BC X	10011 xxxx xxx xxxx	PC ← X, 如果 CY = 1	
BNC X	10001 xxxx xxx xxxx	PC ← X, 如果 CY ≠ 1	
BA0 X	10100 xxxx xxx xxxx	PC ← X, 如果 AC (0) = 1	
BA1 X	10101 xxxx xxx xxxx	PC ← X, 如果 AC (1) = 1	
BA2 X	10110 xxxx xxx xxxx	PC ← X, 如果 AC (2) = 1	
BA3 X	10111 xxxx xxx xxxx	PC ← X, 如果 AC (3) = 1	
CALL X	11000 xxxx xxx xxxx	ST ← CY, PC + 1 PC ← X (不包括 p)	
RTNW H, L	11010 000h hhh llll	PC ← ST; TBR ← hhhh, AC ← lll	
RTNI	11010 1000 000 0000	CY, PC ← ST	CY
HALT	11011 0000 000 0000		
STOP	11011 1000 000 0000		
JMP X	1110p xxxx xxx xxxx	PC ← X (包括 p)	
TJMP	11110 1111 111 1111	PC ← (PC11-PC8) (TBR) (AC)	
NOP	11111 1111 111 1111	空操作	

其中,

PC	程序计数器	I	立即数
AC	累加器	⊕	逻辑异或
-AC	累加器的补码		逻辑或
CY	进位标志位	&	逻辑与
Mx	数据存储器	bbb	RAM 页
p	ROM 页	B	RAM 页
ST	堆栈	TBR	查表寄存器



电气特性

极限参数\*

直流供电电压.....-0.3V to +7.0V  
 输入信号电压..... GND-0.3V to V<sub>DD</sub>+0.3V  
 工作环境温度.....-40°C to +85°C  
 存储温度.....-55°C to +125°C

\*注释

如果器件的工作条件超过左列“极限参数”的范围，将造成器件永久性破坏。只有当器件工作在说明书所规定的范围内时功能才能得到保障。器件在极限参数列举的条件下工作将会影响到器件工作的可靠性。

直流电气特性 (V<sub>DD</sub> = 2.4 - 5.5V, GND = 0V, T<sub>A</sub> = 25°C, 除非另有说明)

参数	符号	最小值	典型值*	最大值	单位	条件
工作电压	V <sub>DD</sub>	4.5	5.0	5.5	V	30kHz ≤ f <sub>OSC</sub> ≤ 10MHz
		2.4	5.0	5.5	V	30kHz ≤ f <sub>OSC</sub> ≤ 4MHz
低电压复位电压 1	V <sub>LVR1</sub>	2.3	2.5	2.7	V	LVR 有效
低电压复位电压 2	V <sub>LVR2</sub>	3.8	4.0	4.2	V	LVR 有效
工作电流	I <sub>OP</sub>	-	3	4.5	mA	f <sub>OSC</sub> = 10MHz 所有输出引脚无负载, 执行 NOP 指令, WDT 关闭, ADC 禁止, LVR 关闭 V <sub>DD</sub> = 5.0V
		-	2	3	mA	f <sub>OSC</sub> = 4MHz 所有输出引脚无负载, 执行 NOP 指令, WDT 关闭, ADC 禁止, LVR 关闭 V <sub>DD</sub> = 5.0V
待机电流 (HALT)	I <sub>SB1</sub>	-	-	1.5	mA	f <sub>OSC</sub> = 10MHz 所有输出引脚无负载 (执行 HALT 指令), WDT 关闭, ADC 禁止, LVR 关闭 V <sub>DD</sub> = 5.0V
		-	-	1	mA	f <sub>OSC</sub> = 4MHz 所有输出引脚无负载 (执行 HALT 指令), WDT 关闭, ADC 禁止, LVR 关闭 V <sub>DD</sub> = 5.0V
待机电流 (STOP)	I <sub>SB2</sub>	-	-	1	μA	所有输出引脚无负载 (执行 STOP 指令), WDT 关闭, ADC 禁止, LVR 关闭, V <sub>DD</sub> = 5.0V
WDT 电流	I <sub>WDT</sub>	-	-	20	μA	所有输出引脚无负载 (执行 STOP 指令), WDT 关闭, ADC 禁止, LVR 关闭, V <sub>DD</sub> = 5.0V
输入低电压	V <sub>IL1</sub>	GND	-	0.3 X V <sub>DD</sub>	V	I/O 端口
	V <sub>IL2</sub>	GND	-	0.2 X V <sub>DD</sub>	V	RESET, T0, T1, OSCI
输入高电压	V <sub>IH1</sub>	0.7 X V <sub>DD</sub>	-	V <sub>DD</sub>	V	I/O 端口
	V <sub>IH2</sub>	0.8 X V <sub>DD</sub>	-	V <sub>DD</sub>	V	RESET, T0, T1, OSCI
输入漏电流	I <sub>IL</sub>	-1	-	1	μA	I/O 端口, GND < V <sub>IN</sub> < V <sub>DD</sub>
上拉电阻	R <sub>PH</sub>	10	30	50	kΩ	V <sub>DD</sub> = 5.0V
输出漏电流	I <sub>OL</sub>	-1	-	1	μA	I/O 端口, GND < V <sub>OUT</sub> < V <sub>DD</sub>
输出高电压	V <sub>OH</sub>	V <sub>DD</sub> - 0.7	-	-	V	I/O 端口, I <sub>OH</sub> = -10mA, V <sub>DD</sub> = 5.0V
输出低电压	V <sub>OL</sub>	-	-	GND + 0.6	V	I/O 端口, PWM0 & 1, I <sub>OL</sub> = 20mA, V <sub>DD</sub> = 5.0V

\*: 数据在典型值一栏列表中为 5.0V, 25°C, 除非另有说明。

提供给 V<sub>DD</sub> 的最大电流为 100mA。

从 GND 输出的最大电流为 150mA。

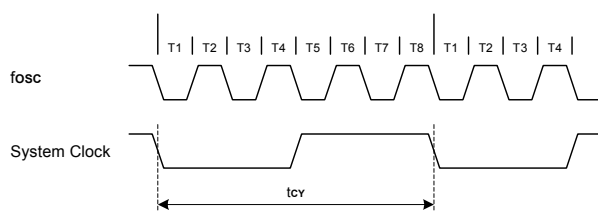


交流电气特性 ( $V_{DD} = 2.4 - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ ,  $f_{OSC} = 30kHz - 10MHz$ , 除非另有说明)

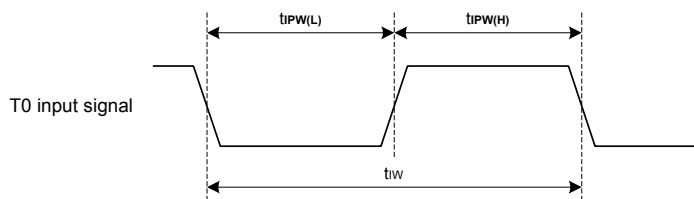
参数	符号	最小值	典型值	最大值	单位	条件
复位脉冲宽度	$t_{RESET}$	10	-	-	$\mu s$	$V_{DD} = 5.0V$
WDT 时间	$t_{WDT}$	1	-	-	ms	$V_{DD} = 5.0V$
频率稳定度 (RC)	$ \Delta F /F$	-	-	15	%	外部 RC 振荡器 $ F(5.0V) - F(2.4V) /F(5.0V)$
内部 RC 频率漂移范围	$f_{OSC}$	3.80	4.00	4.20	MHz	$V_{DD} = 5.0V$ , $T_A = 5^\circ C - 45^\circ C$
指令周期时间	$t_{CY}$	0.4	-	133.4	$\mu s$	$f_{OSC} = 30kHz - 10MHz$
T0 输入宽度	$t_{IW}$	$(t_{CY} + 40)/N$	-	-	ns	$N =$ 预分频比
输入脉冲宽度	$t_{IPW}$	$t_{IW}/2$	-	-	ns	

### 时序波形

(a) 系统时钟时序波形



(b) T0 输入波形



模/数 (A/D) 转换器电气特性 ( $V_{DD} = 2.4 - 5.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ ,  $f_{OSC} = 30kHz - 10MHz$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
精度	$N_R$	-	-	10	bit	$GND \leq V_{AIN} \leq V_{REF}$
参考电压	$V_{REF}$	2.4	-	$V_{DD}$	V	
ADC 输入电压	$V_{AIN}$	GND	-	$V_{REF}$	V	
ADC 输入电阻	$R_{AIN}$	2000	-	-	$k\Omega$	$V_{IN} = 5.0V$
ADC 转换电流	$I_{AD}$	-	500	1000	$\mu A$	ADC 转换器模块工作, $V_{DD} = 5.0V$
非线性误差	$E_{NL}$	-	-	$\pm 2$	LSB	$V_{REF} = V_{DD} = 5.0V$
满刻度误差	$E_F$	-	-	$\pm 1$	LSB	$V_{REF} = V_{DD} = 5.0V$
偏移量误差	$E_Z$	-	-	$\pm 1$	LSB	$V_{REF} = V_{DD} = 5.0V$
总绝对误差	$E_{AD}$	-	$\pm 1$	$\pm 2$	LSB	$V_{REF} = V_{DD} = 5.0V$
ADC 时钟周期	$t_{AD}$	1	-	33.4	$\mu s$	$f_{OSC} = 30kHz - 10MHz$
ADC 转换时间	$t_{CNV1}$	-	204	-	$t_{AD}$	设置 $ADCS = 0$
ADC 转换时间	$t_{CNV2}$	-	780	-	$t_{AD}$	设置 $ADCS = 1$

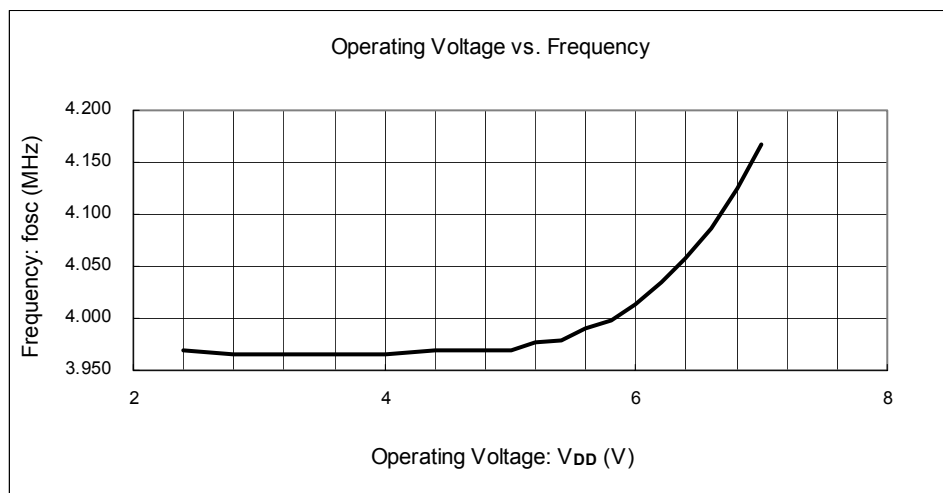
低电压复位电气特性 ( $GND = 0V$ ,  $T_A = 25^\circ C$ ,  $f_{OSC} = 30kHz - 10MHz$ , 除非另有说明)

参数	符号	最小值	典型值	最大值	单位	条件
低电压复位电压 1	$V_{LVR1}$	3.8	-	4.2	V	LVR 有效
低电压复位电压 2	$V_{LVR2}$	2.3	-	2.7	V	LVR 有效

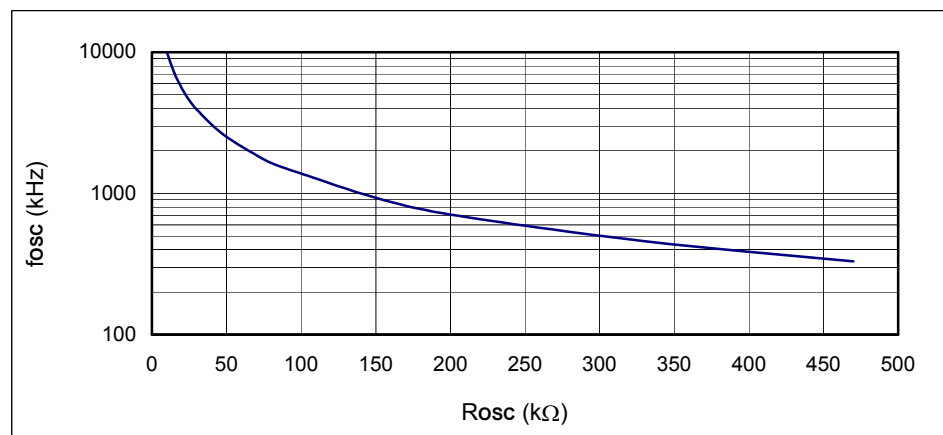


RC 振荡器特性图 (仅供参考)

内部 RC 振荡器特性图 (工作电压 vs. 频率)



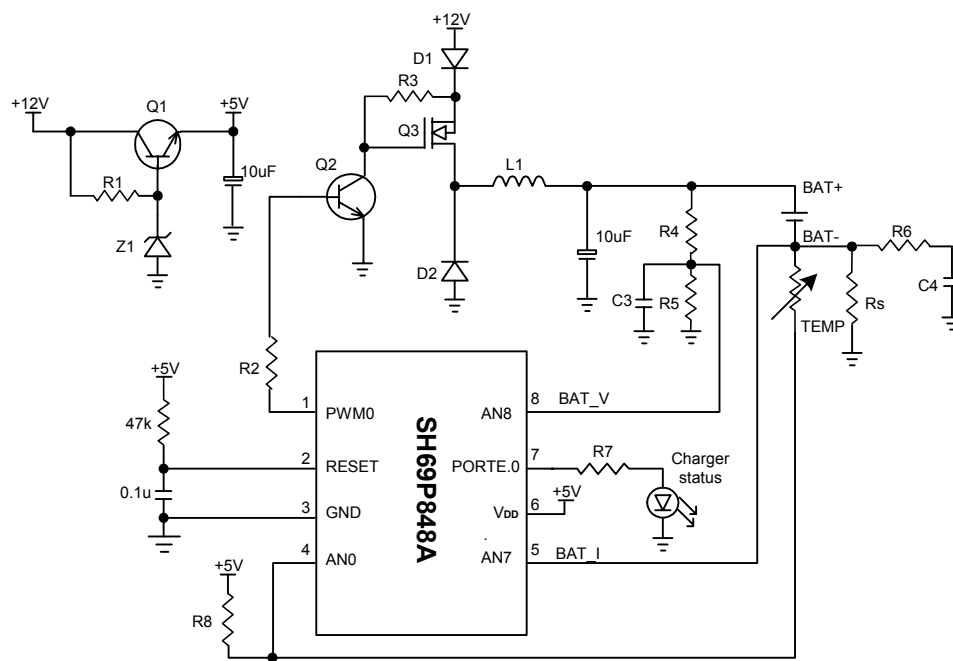
外部 RC 振荡器特性图 (外部电阻 vs. 频率)





## 应用电路（仅供参考）

- (1) 工作电压：5.0V
- (2) 振荡器：内部 RC 4MHz
- (3) AN0: 温度模拟输入  
AN7: 电流模拟输入  
AN8: 电池类型模拟输入  
PWM0: PWM 输出  
PORTE.0: 电池充电状态输出







**SH69P848A**

订购信息

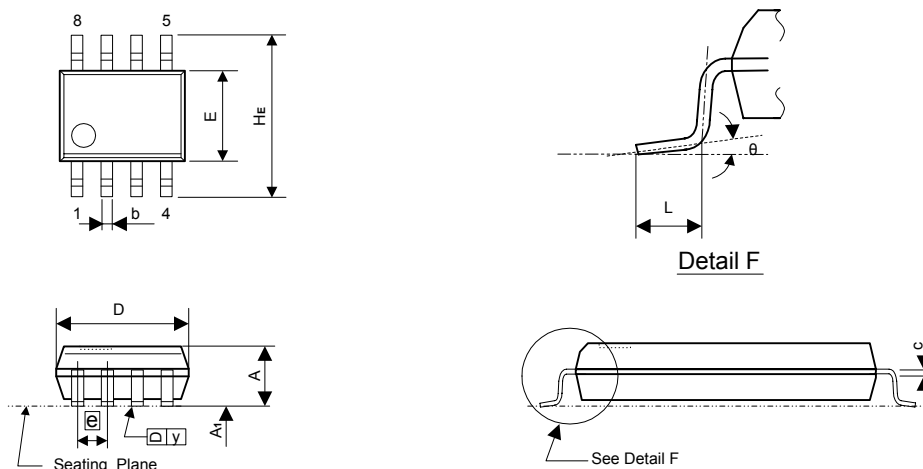
产品编号	封装
SH69P848AM/008MU	8L SOP



封装信息

SOP 8L (W.B.) 外形尺寸

单位: 英寸/毫米



符号	尺寸 (英寸)	尺寸 (毫米)
A	最大值 0.069	最大值 1.75
	最小值 0.053	最小值 1.35
A1	最大值 0.010	最大值 0.25
	最小值 0.004	最小值 0.10
b	典型值 0.016	典型值 0.41
c	典型值 0.008	典型值 0.20
D	最大值 0.196	最大值 4.98
	最小值 0.189	最小值 4.80
E	最大值 0.157	最大值 3.99
	最小值 0.150	最小值 3.81
$\bar{e}$	典型值 0.050	典型值 1.27
HE	最大值 0.244	最大值 6.20
	最小值 0.228	最小值 5.79
L	最大值 0.050	最大值 1.27
	最小值 0.016	最小值 0.41
y	最大值 0.004	最大值 0.10
$\theta$	0° ~ 8°	0° ~ 8°

注意:

- (1) 尺寸 D 的最大值包括末端毛边
- (2) 尺寸 E 不包括树脂凸缘



产品规格更改记录

更改版本	记录	日期
2.0	更新订购信息	2009年3月
1.0	初始版本	2008年2月