

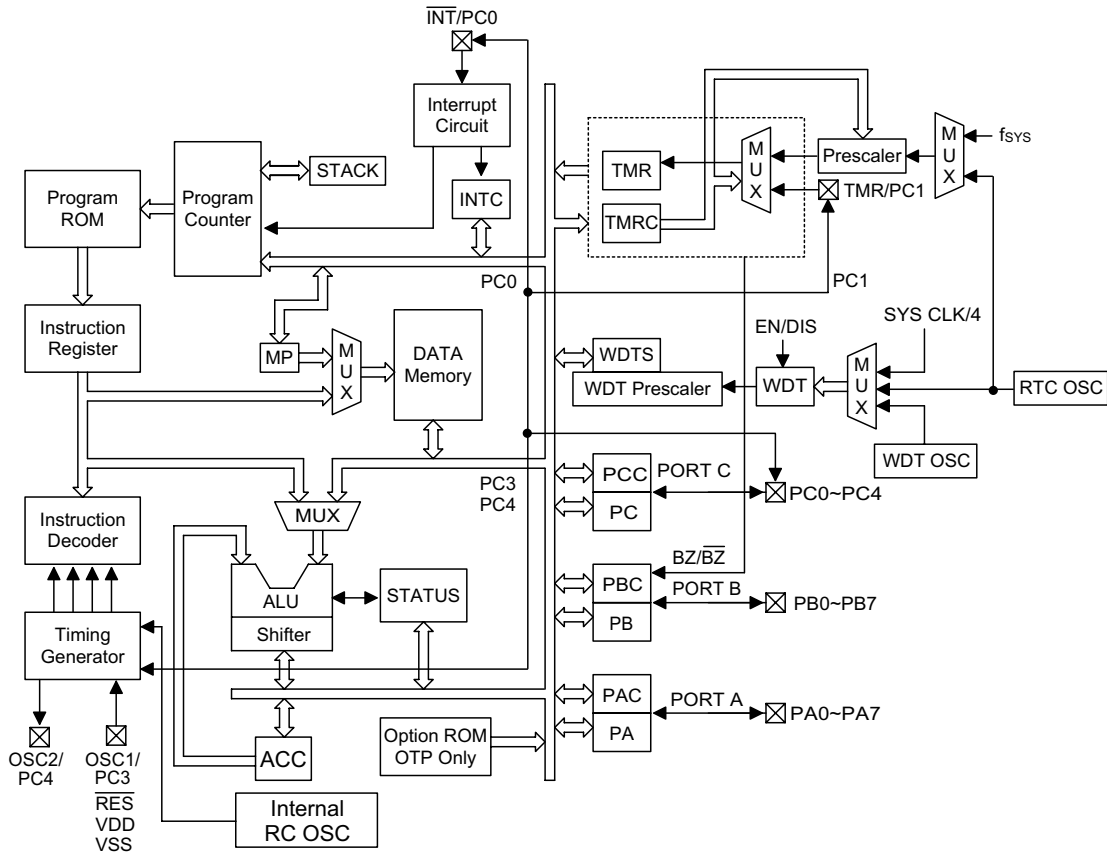
## 特点

- 工作电压：  
     $f_{\text{sys}} = 4\text{MHz}$ : 3.3V~5.5V  
     $f_{\text{sys}} = 8\text{MHz}$ : 4.5V~5.5V
- 21 个双向 I/O 端口（最大值）
- 1 个与中断输入复用管脚的端口
- 8 位可编程带溢出中断和 8 阶预置值的定时计数器
- 片内带有晶振和 RC 震荡器，可外接 RC 震荡器
- 用于定时 32768Hz 晶振
- 看门狗定时器
- 1024×14 程序存储器 ROM
- 64×8 数据存储器 RAM
- 有一对蜂鸣器驱动器并支持 PFD
- HALT 和唤醒功能以减少功耗
- 在  $V_{\text{DD}}=5\text{V}$  系统时钟为 8MHz 时的指令周期为 0.5 秒
- 所有指令在 1 至 2 个指令周期内完成
- 14 位读表格指令
- 4 层子程序嵌套
- 位操作指令
- 63 条指令
- 低电压复位功能
- 24 引脚 SKDIP/SOP 封装

## 简介

HT48R10A-1 为八位高性能的精简指令集单片机，适用于多路输入/输出口的产品，如遥控器、电扇 / 灯光控制器、洗衣机控制器、电子秤、玩具及各种各样子系统控制器。内部含有省电模式以降低功耗。

系统框图:



引脚分配:

PB5	1	24	PB6
PB4	2	23	PB7
PA3	3	22	PA4
PA2	4	21	PA5
PA1	5	20	PA6
PA0	6	19	PA7
PB3	7	18	OSC2/PC4
PB2	8	17	OSC1/PC3
PB1/BZ	9	16	VDD
PB0/BZ	10	15	RES
VSS	11	14	PC2
PC0/INT	12	13	PC1/TMR

HT48R10A-1  
-24 SKDIP-A/SOP-A

**引脚说明:**

引脚号	输入/输出	掩模选择	说明
PA.0~PA.7	输入/输出	上拉电阻* 唤醒模式 CMOS/斯密特触发输入	8 位双向输入/输出口。每一位可掩模选择为唤醒输入。由软件设置该端口为 CMOS 输出，斯密特触发或带上拉电阻 CMOS 输入。（由一位上拉选择项决定）
PB.0/BZ PB.1/B $\bar{Z}$ PB2~PB7	输入/输出	上拉电阻* PB0 或 BZ PB1 或 B $\bar{Z}$	8 位双向输入/输出端口。软件设置该端口为 CMOS 输出或带上拉电阻的斯密特触发输入（由掩模选项决定）。PB0 和 PB1 也可分别作为 BZ 和 B $\bar{Z}$ 。当 PB0 和 PB1 作为蜂鸣器输出时，输出信号可由内部的 PFD 发生器（同时作为定时计数器）产生。
V <sub>SS</sub>	——	——	接地端
PC.0/INT PC.1/TMR PC.2	输入/输出	上拉电阻*	双向输入/输出端口。由软件设置为 CMOS 输出或斯密特触发带上拉电阻输入。PC0 和 PC1 与外部中断和定时器输入端共用管脚。外部中断下降沿有效。
RES	输入	——	斯密特触发复位输入端，低电平有效。
V <sub>DD</sub>	——	——	正向电源端
OSC1/PC3 OSC2/PC4	输入/输出	上拉电阻* 晶振或 RC 振荡或 Int.RC+I/O 或 Int.RC+RTC	根据掩模选择决定 OSC1、OSC2 是否连接 RC 或石英振荡器供系统时钟使用。在 RC 模式下，OSC2 是一个系统时钟四分频的输出端。在这两个端口作为 RTC 晶振（32768Hz）或 I/O 口时，其系统时钟来自内部 RC 振荡器，该 RC 振荡器的频率有四种可选择为 3.2MHz，1.6MHz，800kHz，400kHz。如果选择作为输入/输出端，则允许带上拉电阻。否则 PC3 和 PC4 用作内部寄存器（不带上拉电阻）。

注意：“\*”每一个 I/O 口（PA、PB、PC）的上拉电阻在掩模选择时可以位为单位选择；  
由一位掩模选项决定端口 A 为 CMOS 或斯密特触发结构。

**极限参数:**

电源电压.....V<sub>SS</sub>-0.3V~V<sub>SS</sub>+5.5V

储藏温度.....-50℃~125℃

输入电压.....V<sub>SS</sub>-0.3V~V<sub>DD</sub>+0.3V

工作温度.....-40℃~85℃

注意：这是器件的极限参数。超越上述的极限参数范围可能会对器件造成严重的损害。器件功能并非必然可于未列于规格中的其他情况而且长期暴露于极限状况下可能会影响器件的可靠性。

**D. C. 电气特性**

Ta=25℃

符号	参数	测试条件		最小	标准	最大	单位
		V <sub>DD</sub>	条件				
V <sub>DD1</sub>	工作电压	—	f <sub>sys</sub> =4MHz	3.3	—	5.5	V
V <sub>DD2</sub>	工作电压	—	f <sub>sys</sub> =8MHz	4.5	—	5.5	V
I <sub>DD1</sub>	工作电流（石英振荡器）	3.3V	无负载	—	1	2	mA
		5V	f <sub>sys</sub> =4MHz	—	3	5	mA
I <sub>DD2</sub>	工作电流（RC 振荡器）	3V	无负载	—	1	2	mA
		5V	f <sub>sys</sub> =4MHz	—	3	5	mA
I <sub>DD3</sub>	工作电流（石英振荡器）	5V	无负载 f <sub>sys</sub> =8MHz	—	4	8	mA
I <sub>STB1</sub>	静态电流(看门狗允许, RTC 关闭)	3.3V	无负载	—	—	5	μA
		5V	暂停模式	—	—	10	μA

符号	参数	测试条件		最小	标准	最大	单位
		V <sub>DD</sub>	条件				
I <sub>STB2</sub>	静态电流(看门狗不允许, RTC 关闭)	3.3V	无负载 暂停模式	—	—	1	μA
		5V		2	μA		
I <sub>STB3</sub>	静态电流(看门狗不允许, RTC 打开)	3.3V	无负载 暂停模式	—	—	5	μA
		5V		10	μA		
V <sub>IL1</sub>	输入/输出口的低电位输入电压	—	—	0	—	0.3V <sub>DD</sub>	V
V <sub>IH1</sub>	输入/输出口的高电位输入电压	—	—	0.7V <sub>DD</sub>	—	V <sub>DD</sub>	V
V <sub>IL2</sub>	低电位输入电压( $\overline{\text{RES}}$ )	—	—	0	—	0.4V <sub>DD</sub>	V
V <sub>IH2</sub>	高电位输入电压( $\overline{\text{RES}}$ )	—	—	0.9V <sub>DD</sub>	—	V <sub>DD</sub>	V
I <sub>OL</sub>	输入/输出灌电流	3.3V	V <sub>OL</sub> =0.1V <sub>DD</sub>	4	8	—	mA
		5V	V <sub>OL</sub> =0.1V <sub>DD</sub>	10	20	—	mA
I <sub>OH</sub>	输入/输出源电流	3.3V	V <sub>OH</sub> =0.9V <sub>DD</sub>	-2	-4	—	mA
		5V	V <sub>OH</sub> =0.9V <sub>DD</sub>	-5	-10	—	mA
R <sub>PH</sub>	上拉电阻	3.3V	—	40	60	80	KΩ
		5V	—	10	30	50	KΩ
V <sub>LVR</sub>	最低复位电压	—	选 3.3V	2.7	3.0	3.3	V

**A. C. 特性**

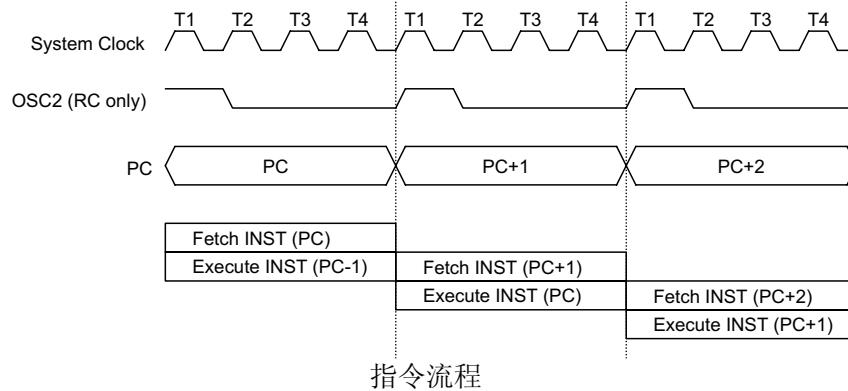
Ta=25°C

符号	参数	测试条件		最小	标准	最大	单位
		V <sub>DD</sub>	条件				
f <sub>SYs1</sub>	系统时钟 (石英振荡器)	3.3V	—	400	—	4000	kHz
		5V	—	400	—	8000	kHz
f <sub>SYs2</sub>	系统时钟 (RC 振荡器)	3.3V	—	400	—	4000	kHz
		5V	—	400	—	8000	kHz
f <sub>SYs3</sub>	系统时钟 (内部 RC 振荡器)	3.3V	选择 3.2MHz	1600	2500	3500	kHz
		5V		2000	3200	4500	kHz
F <sub>timeR</sub>	定时器 I/P 频率 (TMR)	3.3V	—	0	—	4000	kHz
		5V	—	0	—	8000	kHz
t <sub>WDTOSC</sub>	看门狗振荡器	3.3V	—	43	86	168	μs
		5V	—	36	72	144	μs
t <sub>WDT1</sub>	看门狗溢出周期 (WDT 振荡)	3.3V	WDT 无预置值	11	22	43	ms
		5V		9	18	37	ms
t <sub>WDT2</sub>	看门狗溢出周期 (系统时钟)	—	WDT 无预置值	—	1024	—	t <sub>sys</sub>
t <sub>WDT3</sub>	看门狗溢出周期 (RTC 振荡)	—	WDT 无预置值	—	7.812	—	ms
t <sub>RES</sub>	外部复位的低电平脉冲宽度	—	—	1	—	—	μs
t <sub>SST</sub>	系统启动时定时器周期	—	上电, 复位或由 HALT 唤醒	—	1024	—	t <sub>sys</sub>
t <sub>INT</sub>	中断脉冲宽度	—	—	1	—	—	μs

## 系统功能说明

### 指令系统

单片机的系统时钟由石英振荡器或 RC 振荡器产生。该时钟在芯片内部被分为四个互不重叠的 T 状态。一个指令周期包括四个系统时钟周期。指令的提取和执行是流水线操作的，这种方式允许在一个指令周期内进行读取指令操作，而在下一个指令周期里进行解码与执行该指令。因此，流水线结构使多数指令都能在一个周期内执行完成。但如果该指令改变了程序计数器 PC 的值，则需要两个指令周期来执行。



### 程序计数器\_PC

程序计数器 (PC) 控制程序存储器 ROM 中指令执行的流程，它可寻址整个 ROM 的范围。在访问了 ROM 的某个单元并取出指令后，PC 的内容自动加 1，即指向下一个 ROM 单元。在执行 JUMP，条件跳跃，向 PCL 赋值，调用子程序，由于程序返回，内部外部中断，或中断返回等操作时，PC 是通过载入与指令相关的地址而非下一条指令的地址来控制程序的执行的。

当遇到条件跳跃指令且符合条件时，在当前执行过程中取出的下一条指令将被抛弃，并代之以一个空周期，这样才能获得正确的指令。若不符合条件则继续执行下一条指令。

程序计数器的低八位 PCL 是一个可读写的寄存器（位于 RAM 的 06H）。为 PCL 赋值将产生一个短跳转动作，跳转的范围在当前页内。

当一个控制传送的动作发生时，也需要一个空周期。

模 式	程序计数器									
	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
初始化复位	0	0	0	0	0	0	0	0	0	0
跳步	PC+2									
装载 PCL	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
跳转，子程序调用	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
从子程序返回	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

程序计数器

注意：\*9 ~ \*0 : 程序计数器位  
#9 ~ #0 : 指令代码位

S9 ~ S0 : 堆栈寄存器位  
@7 ~ @0 : PCL 位

### 程序存储器\_ROM

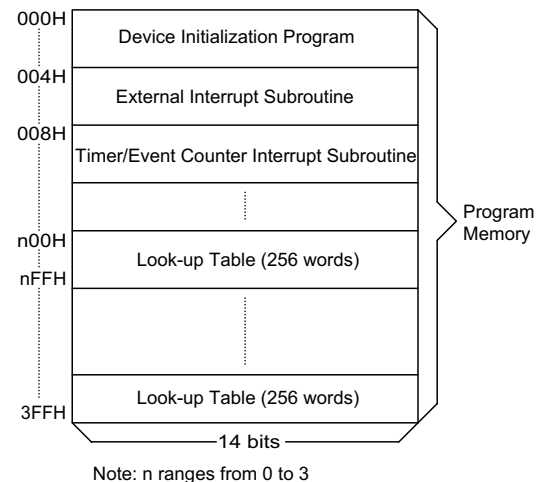
程序存储器用于存放要执行的指令代码以及一些数据、表格和中断入口。程序存储器有 1024 × 14 bit，由程序计数器 PC 和表格指针寻址。

以下的程序存储器空间是系统保留供特殊用途的：

- 000H  
该单元为初始化程序保留。芯片复位后，程序总是由 000H 开始执行。
- 004H  
该单元为外部中断服务程序保留。如果 INT 引脚有信号输入，中断允许且堆栈未滿，程序将转至 004H 处执行。

- 008H  
该单元为定时计数器中断服务程序保留。如果定时计数器溢出引起定时中断，且中断允许、堆栈不满，则程序转至 008H 处执行。
- 表格区

PROM 中的任何位置都可存放表格。查表指令为 TABRDC [m] 与 TABRDL [m]。TABRDC [m] 是查表当前页的数据 [1 页=256 个字 (word)]。在执行 TABRDC [m] 指令 (或 TABRDL [m] 指令) 后，将会传送当前页 (或最后一页) 上的一个字的低位字节到 [m]，而这个字的高位字节传送到 TBLH (08H)。只有表格中的低位字节被定义到目标地址中，而高位字节传送到表格的高位字节寄存器 (TBLH) 并且保留 TBLH 的高 2 位为 “0” 不变。TBLH 为只读寄存器。而表格指针 (TBLP; 07H) 是可以读写的寄存器，用来指明表格地址。单片机会将读出的低位字节送入数据存储器的特殊单元，而将高位字节送入特殊寄存器 TBLH (08H)，而其余的两位为 “0”。表格指针 TBLP (07H) 是读写寄存器，它指出表格在 PROM 中的位置，所以在查表之前，要先将表格的位置写入 TBLP 中。TBLH 是只读寄存器，它的值是不可更改的。如果主程序和 ISR (中断服务程序) 同时执行读表格的操作，主程序中 TBLH 的值将会因 ISR 中的读表格指令的执行而发生变化，产生错误。也就是说，要避免在主程序和中断服务程序中同时使用读表格指令。但如果必须这样做的话，我们可以在读表格指令前先禁止中断，在保留了 TBLH 的值后再开放中断以避免发生错误。所有与表格有关的指令都需要两个指令周期的执行时间。



指 令	表格地址									
	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC [m]	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	@7	@6	@5	@4	@3	@2	@1	@0

附注: \*9~\*0: 表格地址字节  
@7~@0: 表格指针字节  
P9~P8: 当前程序指针字节

### 堆栈寄存器 STACK

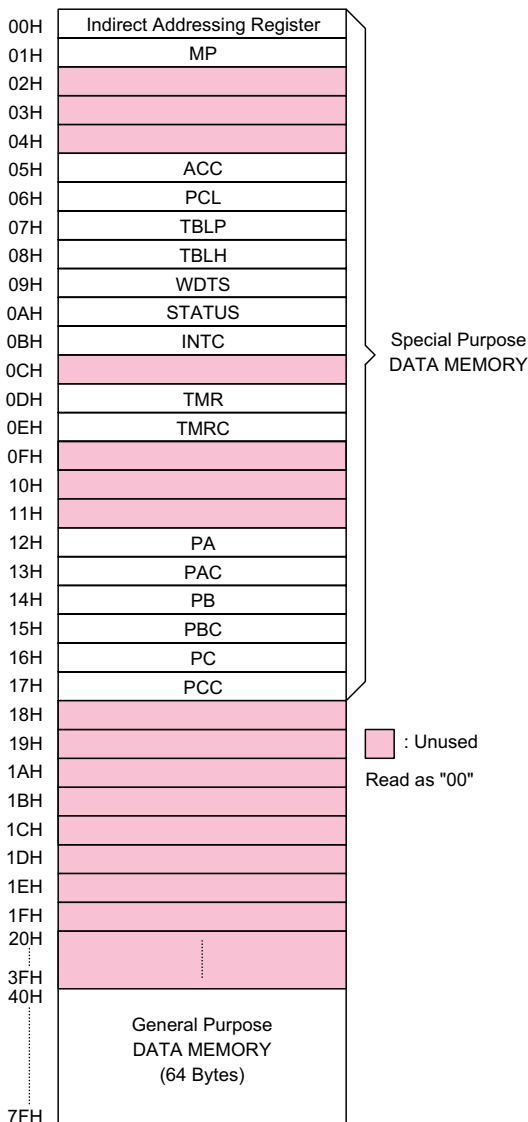
该处是存储器中的特定空间，仅用于保存 PC 的值。该单片机的堆栈有四层，堆栈寄存器既不是数据存储器的一部分，也不是程序存储器的一部分，而且它既不能读出，也不能写入。任何一级堆栈的使用是通过堆栈指针 (SP) 来实现的，堆栈指针也不能读出与写入。一旦发生子程序调用或中断响应时，程序计数器 (PC) 的值会被压入堆栈，在子程序调用结束或中断响应结束时 (执行指令 RET 或 RETI)，堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。

如果堆栈满了，并且发生了不可屏蔽的中断，那么中断请求标志将会被记录下来，但是，该中断的响应还是会被禁止，一旦堆栈指针 (由 RET 或 RETI) 发生递减时，中断服务才会被响应。这个功能防止堆栈溢出，使得程序员易于使用这种结构。同样地，堆栈已满，接着又执行一个子程序调用 (CALL)，那么堆栈会产生溢出，而使首先进入堆栈的内容将会丢失。只有最后的四个返回地址会被保留着。

### 数据寄存器 RAM

RAM 由 81×8bits 构成，分为两个功能区间：特殊功能寄存器和通用数据存储区 (64×8)。这些空间多数是可读写的，但也有只读的。特殊功能寄存器包括间接地址寄存器 (00H)，定时计数器 (TMR; 0DH)，定时计数控制寄存器 (TMRC; 0EH)，程序计数器低位字节寄存器 (PCL; 06H)，存储器指针寄存器 (MP; 01H)，累加器 (ACC; 05H)，表格指针 (TBLP; 07H)，表格高位字节寄存器 (TBLH; 08H)，状态寄存器 (STATUS; 0AH)，中断控制寄存器 (INTC; 0BH)，看门狗选择设置寄存器 (WDTS; 09H)，I/O 寄存器 (PA; 12H, PB; 14H, PC; 16H) 和 I/O 控制寄存器 (PAC; 13H, PBC; 15H, PCC; 17H)。其余在 40H 之前的空间由系统保留供以后扩展使用，他们的值为 00H。通用数据寄存器由 40H 到 7FH，供存储数据和控制信息。

所有的 RAM 区都能直接完成算术、逻辑、递增、递减和循环操作。除了一些特殊位外，RAM 的每一位都可通过 “SET[m].i” 置位或由 “CLR[m].i” 复位。这些寄存器可由 MP 来读取。



### 间接定址寄存器(Indirect addressing register)

地址 00H 是一个间接定址寄存器，并无实际的物理区存在，间接读此地址得到 00H，直接对此地址写值是无作用的。任何对间接定址寄存器的读写动作是依据 MP (01H) 所指的地址操作，亦即对间接定址寄存器做读写的目的地为标志寄存器所指的地址。

MP (01H) 是一个 7 位的存储器指针寄存器。MP 的最高位未定义，如读该位得到结果是“1”。对 MP 的写操作只能将低 7 位数据写入 MP 中。

### 累加器(Accumulator)

累加器 (ACC) 与算术逻辑单元 (ALU) 有密切关系，同样也是对应至 RAM 的地址 05H，作为运算的立即数据，存储器之间的数据传送必须经过 ACC。

### 算术逻辑单元(ALU)

算术逻辑单元 (ALU) 为执行八位算术及逻辑运算的电路，提供有下列的功能：

- 算术运算 (ADD, ADC, SUB, SBC, DAA)
- 逻辑运算 (AND, OR, XOR, CPL)
- 移位 (PL, RR, RLC, RRC)
- 递增及递减 (INC, DEC)
- 进位及无进位减法
- 分支判断 (SZ, SNZ, SIZ, SDZ 等)

ALU 不仅可以储存数据运算的结果，还会改变状态寄存器。

### 状态寄存器(Status register)

八位的状态寄存器包含 0 标志 (Z)，进位标志 (C)，辅助进位标志 (AC)，溢出标志 (OV)，暂停标志 (PD)，看门狗定时溢出标志 (TO)。它也用来记录状态信息和控制操作流程。

如下图所示：

符号	位	功 能
C	0	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产生借位，那么 C 被置位；反之，C 被清除。它也可被循环移位指令影响。
AC	1	在加法运算中低四位产生了进位或减法运算中低四位不产生借位，AC 被置位；反之，AC 被清除。
Z	2	算术运算或逻辑运动的结果为零则 Z 被置位；反之，Z 被清除。
OV	3	如果运算结果向最高位进位，但最高位并不产生进位输出，或那么 OV 被置位；反之，OV 被清除
PD	4	系统上电或执行了 CLR WDT 指令，PD 被清除。执行 HALT 指令 PD 被置位。
TO	5	系统上电或执行了 CLR WDT 指令，或 HALT 指令，TO 被清除。WTD 定时溢出，TO 被置位。
—	6, 7	未定义，读出为零

除了 PD, TO 两个标志外, 状态寄存器的其他位均能改变状态, 而任何对状态寄存器的写操作都不能改变 PD、TO 的值。PD, TO 两标志只受看门狗计时溢出、系统上电 (POWER-UP)、清除看门狗计时“CLR WDT”和暂停 (HALT) 指令的影响。标志位 Z, OV, AC 和 C 反映的是最近一次操作时的状态。

在进入中断程序或响应子程序时, 状态寄存器不会被自动压栈。如果状态寄存器的内容重要或子程序将影响状态寄存器的内容, 程序员须事先将 STATUS 的内容保存好。

## 中断

HT48R10A-1 提供了一个外部中断, 一个内部定时器/计数器中断。其内部中断控制寄存器 (INTC; 0BH) 包含了中断控制位, 用来置位允许/禁止模式以及中断请求标志。

只要有中断子程序被服务, 剩余的中断全部都被禁止 (通过清除 EMI 位), 这种做法的目的在于防止进一步的中断服务 (防止多层中断服务)。在这期间, 可能会发生其它的中断请求, 但只有中断请求标志会被记录下来。如果在中断服务程序中另一个中断需要服务, 程序员可以置位 EMI 及 INTC 所对应的位, 以便进行嵌套中断服务。如果堆栈已满, 即使有关的中断允许, 该中断申请并不会被响应, 一直到堆栈指针 (SP) 发生递减后才会响应。但如果需要立即服务, 应避免堆栈饱和。

这些外部或内部中断全都具有唤醒的能力, 当有中断被服务, 此时会将程序计数器值压入堆栈, 然后再转至中断服务程序的入口。但要记住这时只有程序计数器的内容被压入堆栈, 如果寄存器和状态寄存器的内容被中断程序改变, 而该中断服务程序会破坏预期的控制序列的话, 程序员就应该事先将这些数据储存起来。

外部中断是由  $\overline{\text{INT}}$  脚下降沿触发的, 其相关的中断申请标志位 (EIF; INTC 的第四个位) 接着就会被置位。如果外部中断允许, 而且此时堆栈尚未用满, 而外部中断被触发时, 会发生地址 04H 子程序的调用。该中断申请标志 (EIF) 和 EMI 位也因此而清除, 以便禁止其他中断。

内部定时/计数器中断是由置位定时/计数器中断申请标志 (TF; INTC 的第四位) 的方式来初始化的, 而该中断申请标志是由定时器溢出产生的。如果中断允许, 而此时堆栈尚未用满, 而且 TF 也已置位的话, 会发生地址 08H 子程序的调用此时不仅其相关的中断申请标志 (TF) 会被清除, 而且 EMI 位也会被清除, 用以禁止任何进一步的中断。

执行中断子程序期间, 其他的中断请求会被保持到执行 RETI 指令或是将 EMI 位和其相关中断控制位设为“1” (此时堆栈尚未用满) 为止。若欲从此中断子程序返回, 只要执行 RET 或 RETI 指令即可。其中, RETI 会设定 EMI 位, 用以允许中断服务, 而 RET 则不会。

如果 T2 脉冲的上升沿之前发生中断的话, 在下两个指令周期脉冲时, 该中断会被服务。而如果同时发生中断申请, 其优先级会依照下表所显示, 可以通过设定各中断相关的控制位来改变优先级。

编号	中断来源	优先级	中断向量
a	外部中断	1	04H
b	定时/计数器中断	2	08H

默认中断优先级

中断控制寄存器 (INTC), 是由外部中断请求标志位 (EIF)、外部中断允许位 (EEI)、定时计数器中断请求标志位 (TF)、定时计数器中断允许位 (ETI) 以及总中断允许位 (EMI) 组成的, 其地址为数据存储器的 0BH。EMI、EEI、ETI 用来控制中断的允许/禁止状态。这些控制位可以防止正在进行中断服务时的中断申请。一旦中断申请标志 (TF、EIF) 置位后, 仍会继续保留在 INTC 寄存器中, 直到全部中断都被响应或用软件指令清除为止。

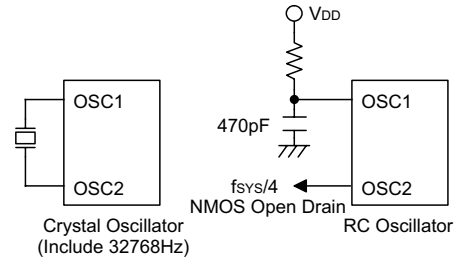
寄存器	位	符号	功能
INTC (0BH)	0	EMI	主中断控制 (1=使能)
	1	EEI	外部中断控制 (1=使能)
	2	ETI	定时/计数器中断控制 (1=使能)
	3	—	空位 (读为 0)
	4	EIF	外部中断请求标志 (1=激活)
	5	TF	定时/计数器中断请求标志 (1=激活)
	6	—	空位 (读为 0)
	7	—	空位 (读为 0)

INTC 寄存器

我们并不建议在中断服务中使用“CALL”指令来调用子程序，这是因为中断通常随时都可能发生而且需要立刻给予服务。如果只剩下一个堆栈，而此堆栈无法完全控制中断的话，一旦中断产生，原先的控制序列很可能因为在中断子程序中执行“CALL”指令调用子程序使堆栈溢出而发生混乱。

### 振荡器

HT48R10A-1 有三种振荡电路：外部石英振荡器电路，外部 RC 振荡器和内部 RC 振荡器，这三种振荡器都是针对系统时钟设计的，可在掩模时选择。不管所选的是哪一种振荡器，其信号都可以支持系统的时钟。若欲终止系统振荡器的运作，可进入暂停模式。进入暂停模式还可以阻止外部信号的进入，以保持省电。



如果使用 RC 型振荡器，在 OSC1 和 V<sub>DD</sub> 之间需要接一个外部电阻，其电阻值可为 51KΩ 到 1MΩ。OSC2 上会输出系统频率的四分频信号，即可用于同步外部逻辑。一般 RC 型振荡器提供了最有效的解决方式，可是，振荡频率会随着 V<sub>DD</sub>、温度和芯片自身的不同而不同。因此，在用于需要非常精确振荡频率的计时操作场合，我们并不建议使用 RC 型振荡器。

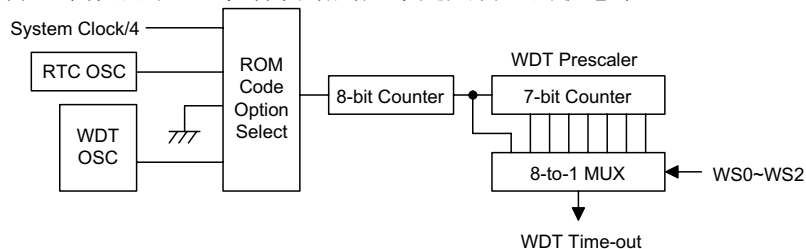
反之，如果选的是石英振荡器，在 OSC1 和 OSC2 之间需要连接一个石英，用来提供石英振荡器所需的回馈和相移，但除此之外就不再需要其它任何外部元件了。另外，在 OSC1 和 OSC2 接脚之间也可使用谐振器来取代石英振荡器，但是在 OSC1 和 OSC2 需要多连接两个电容。如果选用内部 RC 振荡器，OSC1 和 OSC2 可被用作通用 I/O 口线或 32768HZ 的石英振荡器 (RTC OSC)。此时，内部 RC 振荡器的频率可为 3.2MHz, 1.6 MHz, 800kHz, 400kHz (根据选择)。

WDT 振荡器为一个 IC 内部自由振荡的 RC 型振荡器，并不需要连接任何外部元件。当系统进入暂停模式时，系统时钟将关闭，但 WDT 振荡器仍会工作。(其振荡周期大约为 72 μs)。在掩模制造时，如欲节省电源，可关闭 WDT 振荡器。

### 看门狗计时器(Watchdog Timer)

看门狗计时器的时钟来源有三种，RC 时钟从看门狗振荡器(RC)引出，RTC 振荡器或是指令时钟（系统时钟 4 分频）引出，只能通过掩模选项选择时钟来源。看门狗主要用来避免程序运行故障和程序跳入一死循环而导致不可预测的结果。看门狗定时器可用掩模来使能或屏蔽，如果在屏蔽状态，所有对 WDT 的操作都是没有作用的。只有在内部 RC+RTC 模式时 RTC 时钟才允许工作。

我们先由内部看门狗振荡器分析 time-out 周期：（假定 WDT 来源是从内部看门狗振荡器引出），内部看门狗振荡器本身为一 RC 结构振荡器，周期大约是 72 μs/5V，它会先经过一个除 256 (8 级) 的电路而得到大约 18.4ms/5V 时间 (time-out) 的周期，这个周期会随温度、V<sub>DD</sub> 和晶片的结构而改变。如果要得到更长的 timeout 周期，可以利用看门狗预置寄存器来完成。设置 WDTS 的 2、1、0 位，即 WS2、WS1、WS0 位，可以产生不同的频率。举例来说，如果 WS2、WS1、WS0 的值都为 1，则 time-out 的周期被拉长 128 倍，那么用户将得到一个大约 2.4s/5V 秒的 WDT time-out 周期。如果 WDT 被屏蔽，在 HALT 时将停止计数，丧失保护系统的功能，此时只能从指令周期来获得时钟信号，靠外部 Reset 功能来重新启动电路。看门狗预置寄存器的 3~7 位保留给用户来定义自己的状态字。



如果系统运用在强干扰的环境中，建议用户选择芯片内部看门狗振荡器或是 32kHz 的石英振荡器(RTC OSC)，因为如果程序不正常时又刚好执行到 HALT 指令，选用系统时钟除以四为来源的 WDT 会使系统时钟停止，看门狗也就失去了保护的作用。

在正常运作下，WDT 溢出会使系统复位并设置 TO 状态位。但在 HALT 模式下，溢出只产生一个热复位，并只能使 PC 程序计数器和堆栈指针 SP 复位到零。要清除 WDT 的值（包括 WDT 前置分频器）可以有三种方法：外部复位（低电平输入到 RES 端），用软件指令和 HALT 指令三种。软件指令由 CLR WDT 和 CLR WDT1 及 CLR WDT2 二组指令组成。这两组指令中，只能选取其中一种。选择的方式由掩模选

项的 CLR WDT 次数选项决定。如果选择“CLR WDT”（即 CLR WDT 次数为 1），那么只要执行 CLR WDT 指令就会清除 WDT。在选择 CLR WDT1 和 CLR WDT2 的情况下（即 CLR WDT 次数为 2），那么要执行二条件指令才会清除 WDT，否则，WDT 会由于溢出而使系统复位。

WS2	WS1	WS0	分频率
0	0	0	1: 1
0	0	1	1: 2
0	1	0	1: 4
0	1	1	1: 8
1	0	0	1: 16
1	0	1	1: 32
1	1	0	1: 64
1	1	1	1: 128

### 暂停模式

暂停模式是由 HALT 指令来启动的，其可执行下列功能：

- 关闭系统振荡器，但看门狗定时器振荡器仍会继续运行（如果选择 WDT 振荡器）
- 不改变 RAM 和寄存器的内容
- 清除并重新计数看门狗定时器（如果 WDT 时钟来源为 WDT 振荡器）
- 所有输入/输出口都维持其原有状态
- 置位 PD 标志并且清除 TO 标志

若欲离开暂停模式，可以执行外部复位、或是外部中断、或是 PA 口下降沿的信号、或是看门狗定时器溢出。其中，外部复位会造成系统初始化（System Initialization），看门狗定时器溢出则会发生热复位。在检测到 TO 和 PD 标志之后，即可了解系统复位的原因。可通过系统上电或是执行“CLR WDT”指令来清除 PD 标志。而若要置位 PD 标志，则可执行 HALT 指令。如果发生看门狗定时器超时，不仅会置位 TO 标志，还会产生唤醒，但只复位程序计数器 PC 和状态指针 SP，而其它部分则继续维持其原有的状态。

PA 口唤醒和中断这两种方式可以视为正常运行的继续，PA 口上每个位在掩模制造时都可以单独选择是否用来唤醒系统。一旦从输入/输出口启动唤醒之后，程序即从下一条指令重新开始运行。但如果是从中断被唤醒的话，此时可能会发生两种情况。如果禁止中断或允许中断而堆栈已用满，程序将会从下一条指令重新开始运行。但如果该中断被允许，堆栈尚未用满，则会产生一般中断响应。如果在进入 HALT 模式之前，中断请求标志位被置“1”，与中断相关的唤醒功能被禁止。

若发生唤醒事件，必需额外花费 1024tSYS（系统时钟周期）的时间，才能重新正常运行。换言之，唤醒之后即会插入一个等待周期。如果唤醒是由于中断响应的话，实际中断子程序的执行会延迟大约一个以上的周期。如果唤醒事件导致下一条指令执行，一旦插入的等待周期执行完成之后，会立即执行该指令。另外，为减少电源损耗，在进入暂停模式之前，应小心处理所有的输入/输出管脚。RTC 振荡器在 HALT 模式下仍然运行（如果选用 RTC 振荡器）。

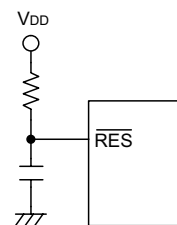
### 初始复位(Reset)

总共有三种方法会产生初始复位，如下所示：

- 正常操作时由  $\overline{\text{RES}}$  引脚发生复位
- 在暂停模式由  $\overline{\text{RES}}$  引脚发生复位
- 正常操作时由看门狗定时器超时发生复位

暂停模式中的看门狗定时器超时与其它系统复位状况不同，因为看门狗定时器超时会执行热复位“warm reset”，热复位只清零程序计数器 PC 和状态标志 SP，并保持其它电路原有的状态。少数寄存器的值在其它复位状态下不会改变，大部分寄存器在一旦符合复位条件时，会复位成初始的状态。通过检测 PD 和 TO 这两个标志，程序即可区别出各种不同的“芯片复位”状态。

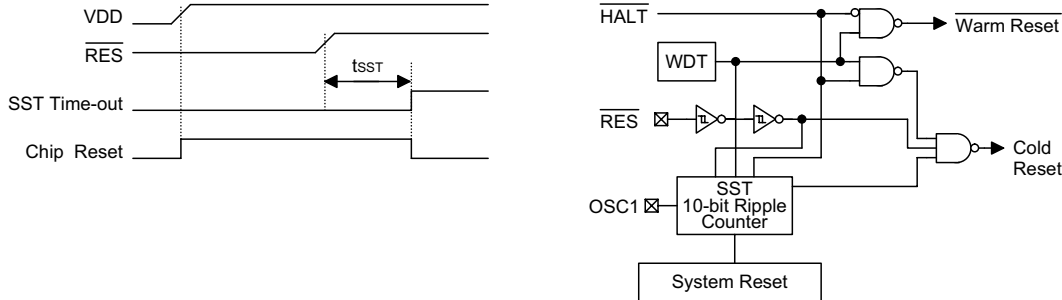
为了保证系统振荡器的启动和稳定运行，系统复位（包括上电复位、看门狗定时器溢出或由  $\overline{\text{RES}}$  端复位）或由暂停状态唤醒时，启动定时器（SST）提供了一个额外的延迟时间（Delay）共 1024 个系统时钟脉冲。



TO	PD	复位条件
0	0	电源上电复位
u	u	正常运作时由RES发生复位
0	1	由RES唤醒暂停模式
1	u	正常运作时发生看门狗定时器超时
1	1	由看门狗定时器唤醒暂停模式

“u”表示不变

系统复位时，SST 加在复位周期中。由暂停模式唤醒后就会启动 SST 延迟。



有关功能的系统复位状态如下所示：

程序计数器 (PC)	000H
中断	禁止
预分频器	清除
看门狗定时器	清除、复位后看门狗定时器开始计数
定时/计数器	关闭
输入/输出口	输入模式
堆栈指针	指向堆栈的顶端

有关寄存器的状态如下：

寄存器	复位(通电)	WDT 超时 (正常运作)	RES复位 (正常运作)	RES复位 (暂停模式)	WDT 超时 (暂停模式)*
TMR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMRC	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u uuuu
Program Counter	000H	000H	000H	000H	000H
MP	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
STATUS	--00 xxxx	--lu uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu
WDTS	0000 0111	0000 0111	0000 0111	0000 0111	uuuu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PC	---1 1111	---1 1111	---1 1111	---1 1111	---u uuuu
PCC	---1 1111	---1 1111	---1 1111	---1 1111	---u uuuu

附注：“\*”表示“热复位” “u”表示“不变” “x”表示“未知”

### 定时/计数器(Time /event counter)

HT48R10A-1 有一组定时/计数器，为一 8 位可编程向上计数器。它的时钟来源可以是外部信号输入、RTC 或由内部系统时钟 4 分频得到。

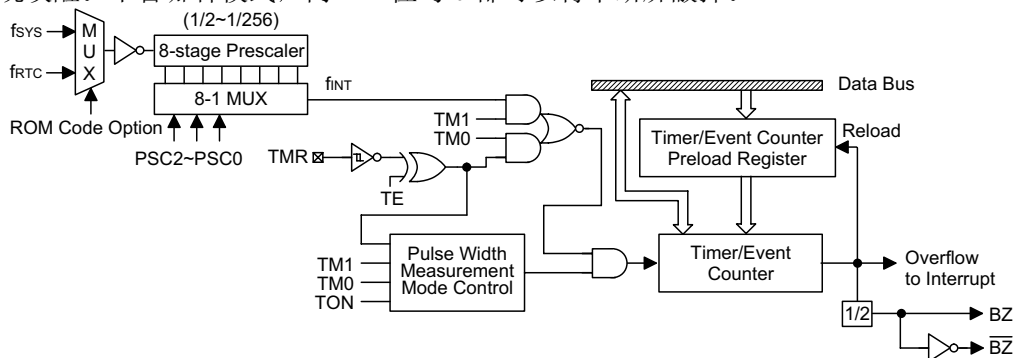
当使用内部时钟源时，定时/计数器有两个参考时钟基准。内部时钟源可在掩模时选择为来自  $f_{SYS}$ （任何情况下可选）或来自  $f_{RTC}$ （仅在系统振荡为内部 RC+RTC 模式时可选）。外部的时钟输入允许用户计量外部事件、测量时间间隔、脉冲宽度或者产生一个累计的时钟信号及 PFD 信号。

符号	位	功能
PSC0-PSC2	0-2	定义前置分频器因子, PSC2, PSC1, PSC0= 000: $f_{INT} = f_{SYS}/2$ 或 $f_{RTC}/2$ 001: $f_{INT} = f_{SYS}/4$ 或 $f_{RTC}/4$ 010: $f_{INT} = f_{SYS}/8$ 或 $f_{RTC}/8$ 011: $f_{INT} = f_{SYS}/16$ 或 $f_{RTC}/16$ 100: $f_{INT} = f_{SYS}/32$ 或 $f_{RTC}/32$ 101: $f_{INT} = f_{SYS}/64$ 或 $f_{RTC}/64$ 110: $f_{INT} = f_{SYS}/128$ 或 $f_{RTC}/128$ 111: $f_{INT} = f_{SYS}/256$ 或 $f_{RTC}/256$
TE	3	定义定时/计数器 TMR 的边缘触发方式: 0 = 电平由低到高有效 1 = 电平由高到低有效
TON	4	允许/禁止计数(1=允许 0=禁止)
—	5	未用, 读出为 0
TM0 TM1	6 7	定义操作模式: 01=事件计数模式(外部时钟) 10=定时模式(内部时钟) 11=脉冲宽度测量模式 00=未用

TMRC 寄存器

TM0, TM1 位定义了工作模式。事件计数模式是用来记录外部事件的，此时时钟来源于外部 TMR 脚。作为一个常用模式，计时模式时钟来源于指令周期。在外部 TMR 信号的范围内，脉宽测量模式可以测量高/低电平的宽度，其时钟来源于指令周期。

无论在定时或者事件计数模式，一旦开始计数，定时/计数器会从寄存器当前值向上计到 FFH。当溢出发生时，预置的值被重新载入同时产生一个中断请求标志(TF; bit 5 of INTC)。作为脉宽测量时，TON 与 TE 位是等同的。只要 TMR 脚得到一个由低到高 (TE 位是 0 时由高到低) 的电平跳跃，它就会开始计数，直到 TMR 脚电平恢复并对 TON 置位。测到的结果被送入寄存器，如果此时有电平跳跃来到也不会改变它的值。换句话说，只能计一次脉冲宽度。对 TON 置位以后，可以继续测量。注意，在该模式下，定时/计数器是跳变触发而不是电平触发。当计数器溢出时，会从定时/计数器的预置寄存器中重新载入数据，处理中断的方式与其他两种模式一样。为了工作在计数模式，TON (bit 4 of TMRC) 要被置 1。测量脉宽时，一次测量结束 TON 位自动被清除。而在另外两种模式，必须用指令来清除。定时/计数器的溢出也可将系统唤醒。不管哪种模式，向 ETI 位写 0 都可以将中断屏蔽掉。



定时/计数器

在定时/计数器停止计数时，如果写数据到定时/计数器的预置寄存器中，会将该数据重装入定时/计数器里。但如果在计数器工作时这么做，新的数据只能存入定时/计数器的预置寄存器，直到发生溢出时才会将数据从定时/计数器预置寄存器送到定时/计数器寄存器。当读定时计数器时，时钟将被禁止以避免错误。当时钟被禁止时会导致计数错误，程序员必须注意到这一点。

TMRC 的 bit0-bit2 能用来定义使用内部时钟的定时/计数器的预置时钟频率。定义如上表所示。定时/计数器的溢出信号可用作产生 PFD 信号来驱动蜂鸣器。

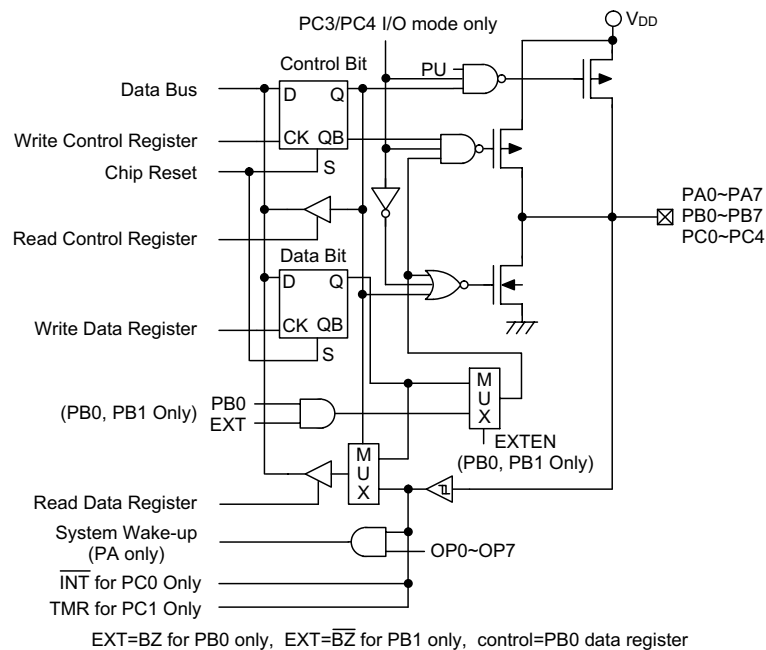
## 输入/输出口

HT48R10A-1 有 21 个双向输入/输出口，标识为 PA~PC，分别对应 RAM 中的[12H]、[14H]和[16H]。所有输入/输出口都可以进行输入/输出操作。输入不锁存，输出锁存。也就是说，输入信号必须在 MOV A, [m] (m=12H, 14H 或 16H) 指令的 T2 上升沿到来时准备好。输出锁存保持不变直到下一个数据写入前。

每一个输入/输出口都有自己的控制寄存器 (PAC, PBC, PCC)。使用这些寄存器可以由软件动态控制 CMOS 输出或 Schmitt 触发输入。作输入时，对应锁存的控制寄存器必须置 1。输入信号也依靠控制寄存器来控制。如果控制寄存器是 1，就会读管脚状态，如果是 0，数据将被输入到内部总线上。后者在‘读-修改-写’指令中可能实现。输出功能时，只能采用 CMOS 输出。控制寄存器对应 RAM 的 13H, 15H, 17H。

在系统复位之后，这些输入/输出线若不是高电平即为浮接（视有无上拉电阻而定）。这些输出锁存的每个位都可通过 SET [m].i 置位或是以 CLR [m].i 清除 (m=12H,14H,16H)。

有些指令会先输入数据，然后才输出运行结果。举例来说：“SET [m].i”，“CLR [m].i”，“CPL [m]”，“CPLA[m]”这些指令会先将整个口状态读入 CPU 中，接着执行所定义的运算（位操作），最后再将执行的结果写入锁存或是累加器中。



PA 的每一个口都可作为唤醒端。PC 的高 3 位实际上是没有的。读这三位将会返回 0，而写则产生一个空操作。

对所有的输入/输出口都有上拉选项（字节操作）。一旦选择了上拉选项，所有的输入/输出口就加了上拉电阻。否则，不加上拉电阻。必须注意在输入模式下，若 I/O 口不加上拉电阻会产生悬空状态。

PB0 和 PB1 端的第二功能是 BZ 和  $\overline{BZ}$ 。如果选择为 BZ/ $\overline{BZ}$  输出，PB0/PB1 在输出模式时的输出 PFD 信号，该 PFD 信号来自定时/计数器的溢出信号。在输入模式时保持其原来的功能。选择 BZ/ $\overline{BZ}$  后，蜂鸣器输出信号仅受 PB0 数据寄存器控制。PB0/PB1 的输入/输出功能如下所示。

PB0 I/O	I	I	I	I	O	O	O	O	O	O
PB1 I/O	I	O	O	O	I	I	I	O	O	O
PB0/PB1 模式	×	C	B	B	C	B	B	C	B	B
PB0 数据	×	×	0	1	D	0	1	D0	0	1
PB1 数据	×	D	×	×	×	×	×	D1	×	×
PB0 Pad 状态	I	I	I	I	D	0	B	D0	0	B
PB1 Pad 状态	I	D	0	B	I	I	I	D1	0	B

注释: I: 输入; O: 输出; D, D0, D1 : 数据;  
 B: 蜂鸣器的选项,  $\overline{BZ}$  或 BZ ; ×: 任意值;  
 C: CMOS 输出;

PC1 与  $\overline{INT}$  和 PC0 与 TMR 共用一个端口。

在系统振荡为“内部 RC+I/O”时, PC3 与 OSC1 及 PC4 与 OSC2 共用一个端口。一旦选择“内部 RC+I/O”模式后, PC3 和 PC4 可用作通用 I/O 口。否则, PC3 和 PC4 没有上拉电阻和输入输出功能。

我们建议用软件将未使用和没有外接的输入/输出端口置为输出端, 因为这些端口在输入悬浮的状态下将增加系统的耗电量。

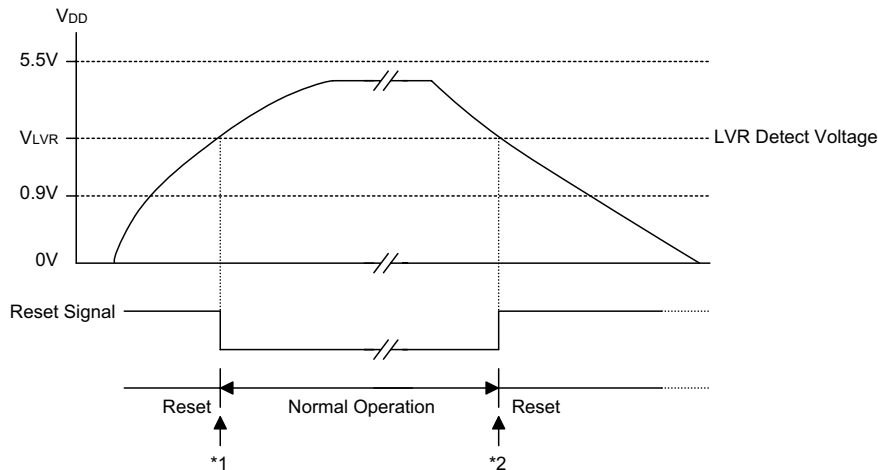
### 低电压复位 (LVR)

为了监控器件的工作电压, HT48R10A-1 提供低电压复位电路。如果器件的工作电压在 0.9 伏~ $V_{LVR}$  伏之间, 例如电池电压的变化, 那么 LVR 会自动使器件产生内部复位

LVR 具有下列功能说明:

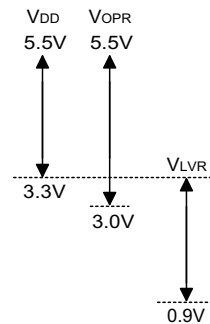
- 低电压 (0.9 伏~ $V_{LVR}$  伏) 的状态必须持续 1ms 以上。如果低电压的状态没持续 1ms 以上, 那么 LVR 会忽视它而不去执行复位功能。
- LVR 通过与  $\overline{RES}$  信号的“或”的功能来执行系统复位。

$V_{DD}$  与  $V_{LVR}$  之间的关系如下所示:



注: \*1. 在进入一般操作前, SST 提供了额外的 1024 个系统时钟脉冲以确定系统是稳定振荡的。

\*2. 因为低电压必须保持 1ms 以上才有效, 所以进入复位状态后有 1ms 延时。



附注: V<sub>OPR</sub> 是在系统时钟为 4MHz 时, 使得芯片正常运行的电压域值

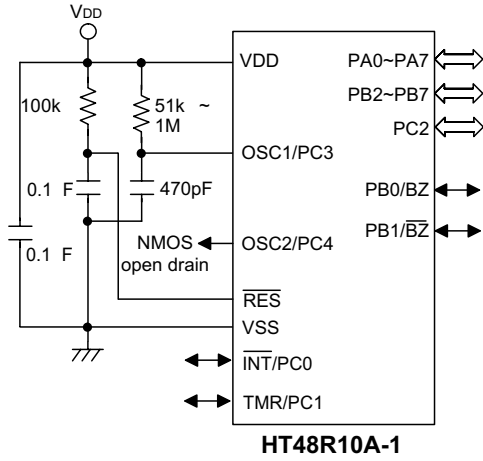
**掩模选项**

下表标出了 HT48R10A-1 的所有掩模选项。在系统运行前必须定义所有选项的值。

项目	选择
1	WDT 时钟源: $WDT_{osc}/f_{TID}$ / $RTC_{osc}/disable$
2	CLRWDT 指令: 1 或 2 条指令
3	定时/计数器 计数时钟源: $f_{SYS}$ 或 $RTC_{osc}$
4	PA 唤醒
5	PA CMOS/斯密特 输入
6	PA 上拉电阻 允许/不允许
7	PB 上拉电阻 允许/不允许
8	PC 上拉电阻 允许/不允许
9	BZ/ $\overline{BZ}$ 允许/不允许
10	LVR 允许/不允许
11	系统振荡选择: 外部 RC, 外部振荡器, 内部 RC+RTC 或内部 RC+PC3/PC4
12	内部 RC 频率选择 3.2MHz, 1.6MHz, 800kHz 或 400kHz

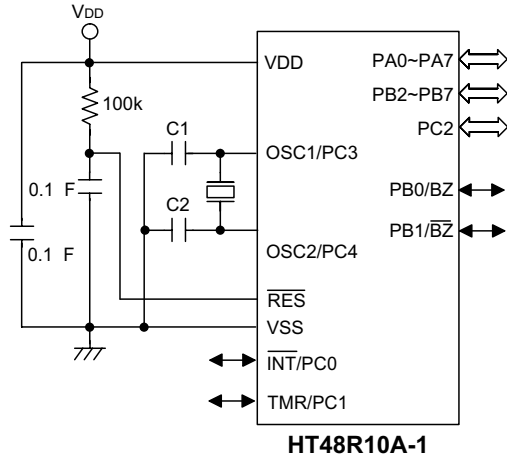
应用电路

RC 振荡器供多输入/输出端口使用



HT48R10A-1

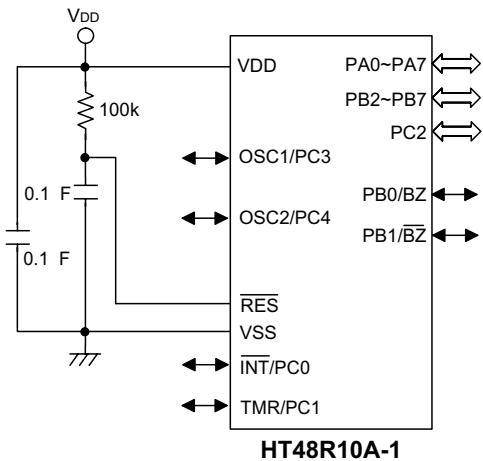
晶体或陶瓷谐振供多输入/输出端使用



HT48R10A-1

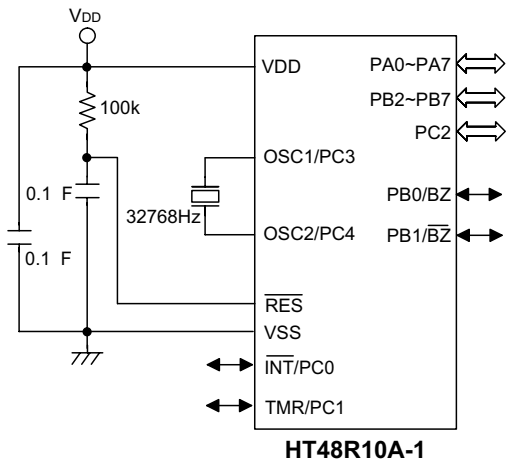
注意：如果  $F_{sys} < 1\text{MHz}$  则  $C1 = C2 = 300\text{pF}$   
 否则  $C1 = C2 = 0$

内部 RC 振荡器供多输入/输出使用



HT48R10A-1

内部 RC 振荡和 RTC 供多输入输出使用



HT48R10A-1

注意：电阻和电容值选取的原则是使 VDD 保持稳定并在  $\overline{\text{RES}}$  置为高以前把工作电压保持在允许的范围內。

注：本文译至 July 02, 2001 的 48r10a-1.pdf 文档

Holtek Semiconductor Inc. (Headquarters)  
No.3, Creation Rd. II, Science-based Industrial Park, Hsinchu, Taiwan  
Tel: 886-3-563-1999  
Fax: 886-3-563-1189

Holtek Semiconductor Inc. (Sales Office)  
11F, No.576, Sec.7 Chung Hsiao E.Rd., Taiwan  
Tel: 886-2-2782-9635  
Fax: 886-2-2782-9636  
Fax: 886-2-2782-7129(Internation at sales hotline)

Holtek Semiconductor (Hong Kong) Ltd.  
RM.711, Tower 2, Cheung Sha Wan Plaza, 833 Cheung Sha Wan Rd., Kowloon, Hong Kong  
Tel: 852-2-745-8288  
Fax: 852-2-742-8657

Holtek Semiconductor (Shanghai) Inc.  
7th, Floor, Building 2, No.889, YiShan Rd., Shanghai, China  
Tel: 021-6485-5560  
Fax: 021-6485-0313

**Holmate Technology Corp.**  
48531 Warm Springs Boulevard, Suite 413, Fremont, CA 94539  
Tel: 510-252-9880  
Fax: 510-252-9885

HOLTEK SEMICONDUCTOR INC. 版权所有 ©

However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com.tw>.