特点

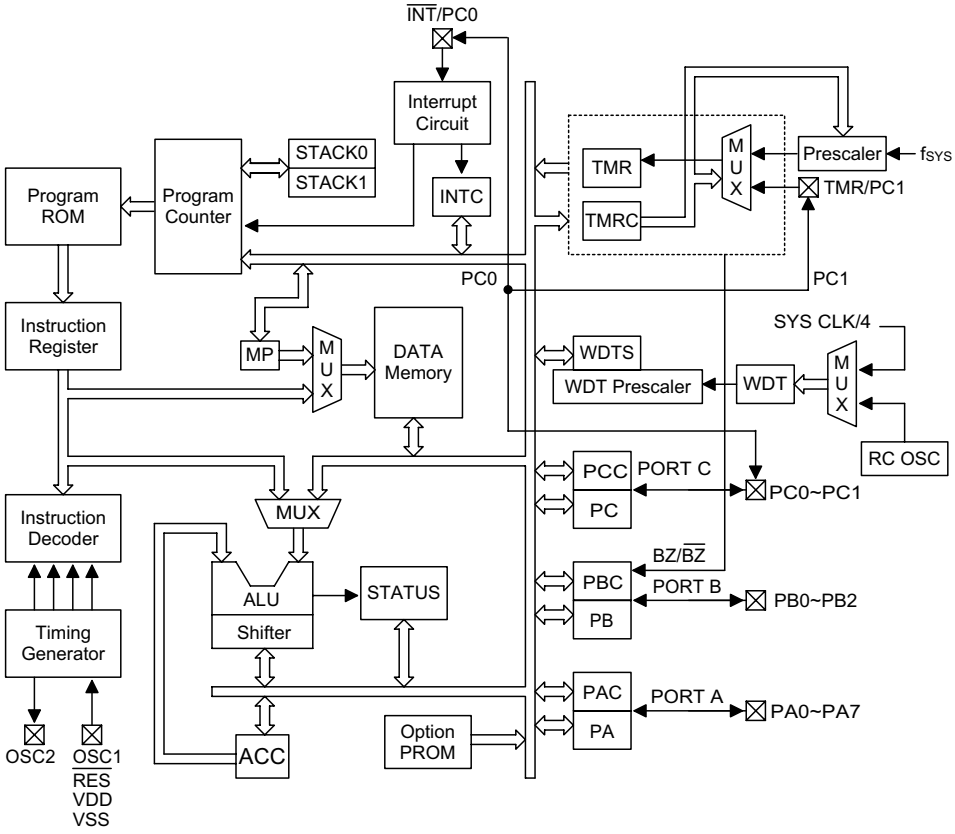
- 工作电压：3.3V ~ 5.5V (fsys = 4MHz)
  4.5V ~ 5.5V (fsys = 8MHz)
- 13 个双向输入/输出口
- 中断输入与一个输入/输出口共用
- 8 位带溢出中断的可编程定时/计数器和八级前置分频器
- 石英晶振或 RC 振荡器
- 看门狗定时器
- 1024×14 程序存储器 PROM
- 64×8 位的数据存储器 RAM
- 支持 PFD 蜂鸣器输出端
- 暂停和唤醒功能降低了功耗
- 在 V_{DD}=5V, 系统时钟为 8MHz 时，指令周期为 0.5 us
- 执行所有指令需 1 或 2 个机器周期
- 查表指令可读 14 位数据
- 二级堆栈
- 位操作指令
- 63 条功能强大的指令
- 低电压复位功能
- 16-pin SSOP 封装
  18-pin DIP/SOP 封装

概述

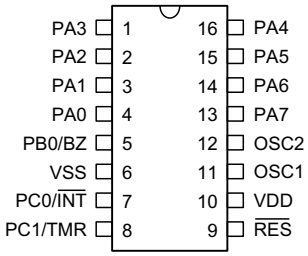
HT48R06A-1 为八位高性能精简指令集单片机。专门为多输入/输出的产品而设计的。这些器件适用于遥控、电扇 / 电灯控制、洗衣机控制、电子秤、玩具及各种各样次级系统控制。暂停功能降低了功耗。

可编程的程序和功能存储器使得这种单片机满足程序员升级产品的要求。

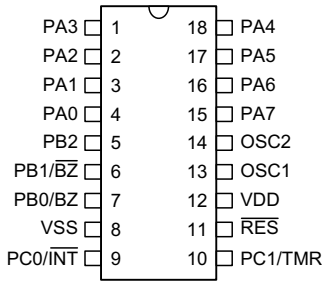
系统框图



引脚图



HT48R06A-1
−16 SSOP-A



HT48R06A-1
−18 DIP-A/SOP-A

引脚说明

引脚编号	引脚名称	输入输出	掩模选项	说明
1~4 18~15	PA0~PA7	I/O	上拉电阻* 唤醒功能	8 位双向输入/输出端口。每一位能被掩膜选项设置为唤醒输入。软件指令确定 CMOS 输出，或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。
5 6 7	PB2 PB1/$\overline{BZ}$ PB0/BZ	I/O	上拉电阻* I/O 或 BZ/$\overline{BZ}$	3 位双向输入输/出端口。每一位能被掩膜选项设置为唤醒输入。软件指令确定 CMOS 输出，或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。PB0 和 PB1 与 BZ 和$\overline{BZ}$共享一个引脚。一旦 PB0 和 PB1 被选为蜂鸣器驱动输出，输出信号来自内部 PFD 发生器（与定时/计数器共享）。
8	V_{SS}	—	—	电源负极，GND。
9 10	PC0/$\overline{INT}$ PC1/TRM	I/O	上拉电阻*	双向 I/O 口。软件指令确定 CMOS 输出，或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。外部中断和定时器输入与 PC0 和 PC1 共享一个引脚。外部中断输入是电平由高到低的变化激励的。
11	$\overline{RES}$	I	—	斯密特触发复位输入端，低电平有效。
12	V_{DD}	—	—	电源正极。
13 14	OSC1 OSC2	I O	石英晶振或 RC	OSC1 和 OSC2 被连接到一个 RC 或一个石英晶振（由掩膜选项确定）来产生内部系统时钟。在 RC 方式下 OSC2 是一个系统时钟四分频的输出端。

*所有的上拉电阻由一个选择位控制。

极限参数*

工作电压 …….. V_{SS} -0.3 至 V_{SS} +5.5 伏　　储存温度 …… -50℃至 125℃

输入信号电压 …... V_{SS} - 0.3 至 V_{DD} +0.3 伏　　工作温度 ……. -40℃至 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。

直流特性　　Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DD1}	工作电压	—	Fsys=4MHZ	3.3	—	5.5	V
I_{DD2}	工作电流	—	Fsys=8MHZ	4.5	—	5.5	V
I_{DD1}	工作电流 (石英晶振振荡器)	3.3V	无负载 sys=4MHZ	—	1	2	mA
		5V		—	2	4	
I_{DD2}	工作电流 (RC 振荡器)	3.3V	无负载 fsys=4MHZ	—	1	2	mA
		5V		—	2	4	
I_{DD3}	工作电流 (石英晶振振荡器)	5V	无负载 fsys=8MHZ	—	5	10	mA
I_{STB1}	静态电流 （WDT 允许）	3.3V	无负载 系统 HALT	—	—	5	μA
		5V		—	—	10	
I_{STB2}	静态电流 （WDT 禁止）	3.3V	无负载 系统 HALT	—	—	1	μA
		5V		—	—	2	

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{IL1}	I/O 端口输入低电平,TMR 和 $\overline{INT}$	3.3V	—	0		0.3V_{DD}	V
		5V		0		0.3V_{DD}	
V_{IH1}	I/O 端口输入高电平,TMR 和 $\overline{INT}$	3.3V	—	0.7V_{DD}	—	V_{DD}	V
		5V		0.7V_{DD}	—	V_{DD}	
V_{IL2}	输入低电平($\overline{RES}$)	3.3V	—	0	—	0.4V_{DD}	V
		5V		0	—	0.4V_{DD}	
V_{IH2}	输入高电平($\overline{RES}$)	3.3V	—	0.9V_{DD}	—	V_{DD}	V
		5V		0.9V_{DD}	—	V_{DD}	
I_{OL}	I/O 端口灌电流	3.3V	V_{OL}=0.1V_{DD}	4	8	—	mA
		5V	V_{OL}=0.1V_{DD}	10	20	—	
I_{OH}	I/O 端口源电流	3.3V	V_{OL}=0.9V_{DD}	-2	-4	—	mA
		5V	V_{OL}=0.9V_{DD}	-5	-10	—	
R_{PH}	I/O 端口上拉电阻	3.3V	—	40	60	80	KΩ
		5V	—	10	30	50	
V_{LVR}	低电平复位	—	—	2.7	3.0	3.3	V

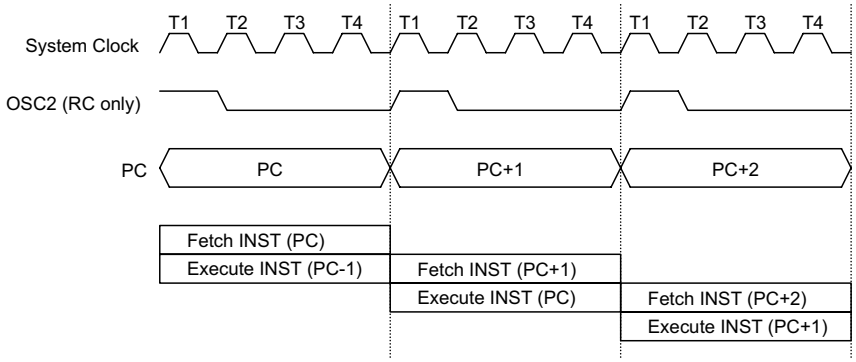
交流特性 Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		VDD	条件				
fsys1	系统时钟（石英晶振振荡器）	3.3V	—	400	—	4000	KHz
		5V	—	400	—	8000	
fsys2	系统时钟（RC 体振荡器）	3.3V	—	400	—	4000	KHz
		5V	—	400	—	4000	
ftimer	定时器 I/P 频率 (TMR)	3.3V	—	0	—	4000	KHz
		5V	—	0	—	4000	
t_{WDTOSC}	看门狗振荡器	3.3V	—	43	86	168	μs
		5V	—	35	65	130	
t_{WDT1}	看门狗溢出周期(RC)	3.3V	无 WDT 前置分频器	11	22	43	ms
		5V		9	17	35	
t_{WDT2}	看门狗溢出周期(系统时钟)	—	无 WDT 前置分频器	—	1024	—	t_{SYS}
t_{RET}	外部复位低电平的脉冲宽度	—	—	1	—	—	μs
t_{SST}	系统启动定时器延时	—	上电或复位或从 HALT 状态唤醒	—	1024	—	Tsys
t_{INT}	中断脉冲宽度	—	—	1	—	—	μs

功能说明

指令执行时序

HT48R06A-1 系统时钟由石英晶振振荡器或 RC 振荡器产生。系统内部对此频率进行四分频，产生四个不重迭的时钟周期。一个指令周期包含了四个系统时钟周期。



指令读取与执行是以流水线方式来进行的。这种方式允许在一个指令周期进行读取指令操作，而在下一个指令周期里进行解码与执行该指令。这种流水线方式能在一个指令周期里有效地执行一个指令。但是，如果指令是要改变程序计数器，就需要两个指令周期来完成这一条指令。

程序计数器（PC）

程序计数器控制存放在程序存储器中的要被执行的指令序列。程序计数器可寻址程序存储器的所有地址。

通过访问一个程序存储单元来取出指令代码后，PC 的值便会加 1。然后程序计数器便会指向下一条指令代码所在的程序存储单元。

当执行跳转、条件跳转、装载 PCL 寄存器、子程序调用、初始复位、中断或从一个子程序返回，PC 会通过装载指令的相应地址来执行程序转移。

通过指令实现条件跳转，一旦条件满足，那么在当前指令执行期间取出的下一条指令会被放弃，而替代它的是一个假指令周期(dummy cycle)来获取正确的指令，接着就执行这条指令。否则就执行下一条指令。

程序计数器的低位字节（PCL；06H）是可读写的寄存器。将数据赋值到 PCL 会执行一个短跳转。这种跳转只能在 256 个地址范围内。

当一个控制转移发生时，就需要有一个附加的假指令周期。

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**
外部中断	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
定时/计数器	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
短跳转	**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　　S9～S0 ： 堆栈寄存器位
　　#9～#0 ： 指令代码位　　@7～@0 ： PCL 位

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

表格地址

注意：*8～*0 ： 表格地址　　P8 ： 当前程序计数器
　　@7～@0 ： 表格指针

程序存储器（PROM）

程序存储器被用来存放要执行的指令代码。还包括数据、表格、中断入口。一共为 1024 x 14 位。可以由程序计数器或表格指针来寻址。

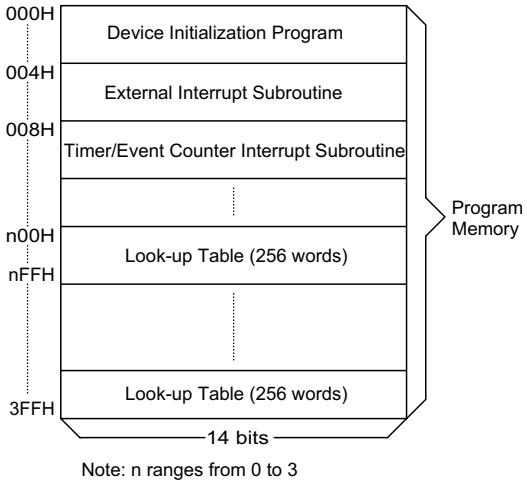
在程序存储器中某几个地址被保留作为特殊用途。

- 地址 000H

此区域保留给程序初始化之用。当系统复位时，程序会从 000H 地址开始执行。

- 地址 004H

此区域保留给外部中断服务使用。当中断是开放的，且堆栈未满，则一旦端被正确电平触发，就能产生中断，程序会从 004H 地址开始执行外部中断服务程序。



Note: n ranges from 0 to 3

- 地址 008H

此区域保留给定时/计数器中断服务使用。当中断是开放的，且堆栈未满，则一旦定时/计数器发生溢出时，就能产生中断，程序会从 008H 地址开始执行中断服务程序。

- 表格地址

ROM 内的任何地址都可被用来作为查表地址使用。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)]。TABRDL [m]是查表最后一页的数据。[m]为数据被存放的地址。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]。这个字的高位字节传送到 TBLH（08H）。只有表格中的低位字节被送到目标地址中，而表格中的高位字节的被传送到 TBLH 的低部位，剩余的二位读出为 0。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。在访问表格以前，通过对 TBLP 寄存器赋值来指明表格地址。TBLH 为只能读出而不能存储。如果主程序和 ISR（中断服务程序）二者都使用查表指令，那么在主程序中的 TBLH 的内容可能会被 ISR 中的查表指令改变而产生错误。换句话说，应该避免在主程序和 ISR（中断服务程序）中同时使用查表指令。但是，如果主程序和 ISR 二者都必须使用查表指令，那么中断应该在查表指令前被禁止，直到 TBLH 被备份好。查表指令要花两个指令周期来完成这一条指令的操作。按照程序员的需要，这些区域可以作为正常的程序存储器来使用。

堆栈寄存器（STACK）

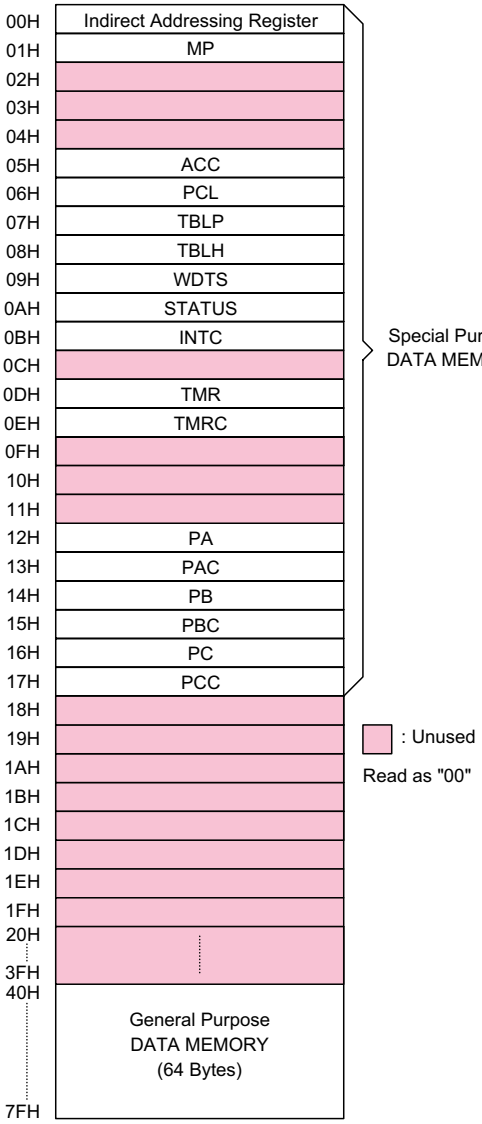
堆栈寄存器（STACK）是一个用来保存 PC 值的特殊存储单元。HT48R06A-1 的堆栈为 2 级。堆栈寄存器既不是数据存储器的一部分，也不是程序存储器的一部分，而且它既不能读出，也不能写入。任何一级堆栈的使用是由堆栈指针（SP）来确定。堆栈指针也不能读出与写入。一旦发生子程序调用或中断响应时，程序计数器（PC）的值会被压入堆栈。在子程序调用结束或从中断返回时，通过一条返回指令（RET 或 RETI），堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。

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，该中断还是会被禁止，直到堆栈指针（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失落。只有最后的 2 个返回地址会被保留着。

数据存储器（RAM）

数据存储器 RAM 由 81×8 位组成。它可分成两个功能组：特殊功能寄存器和通用数据存储器（64×8）。这两个功能组的大部单元可以读写，而某些单元只能读出，不能写入。

特殊功能寄存器包括间接寻址寄存器（00H），存储器指针寄存器（MP；01H），累加器（ACC；05H），PC 低位字节寄存器（PCL；06H），表格指针寄存器（TBLP；07H），表格高位字节寄存器（TBLH；08H），看门狗寄存器（WDTS；09H），状态寄存器（STACK；0AH），中断控制寄存器（INTC；0BH），定时/计数器（TMR；0DH），定时/计数器控制寄存器（TMRC；0EH），I/O 寄存器（PA；12H，PB；14H，PC；16H）和 I/O 控制寄存器（PA；13H，PB；15H，PC；17H）。40H 以前的剩余单元都被保留为将来进一步扩展使用。读取这些被保留单元的值，都将返回 00H。通用数据存储器地址 60H-7FH 用来保存程序数据和控制信息使用。



所有的 RAM 区单元都能直接执行算术、逻辑、递增、递减和移位等运算。除了某些特殊的位以外，RAM 中的每一位都可以由 SET[m].i 和 CLR[m].i 指令来置位和清零。它们都可通过存储器指针寄存器（MP；01H） 间接寻址来存取。

间接寻址寄存器

地址 00H 是作为间接寻址寄存器。它没有物理上结构。任何对[00H]的读/写操作，都会访问由 MP(01H)所指向的 RAM 单元。间接地读取[00H]，将会返回 00H 的值，而间接地写入 00H 单元，则不会产生任何结果。

存储器指针寄存器 MP 的有效位是 7 位，MP 的第 7 位是没有定义的。若读它，将返回“1”。而任何对 MP 写的操作只能将低 7 位数据传送到 MP。

累加器 （ACC）

累加器（ACC）与算术逻辑单元（ALU）运算相联系。它对应于 RAM 的地址 05H，并能与立即数进行操作，在存储器间的数据传送都必须经过累加器。

算术逻辑单元 （ALU）

算术逻辑单元是执行 8 位算术逻辑运算的电路。它提供有下列功能：

- 算术运算 （ADD，ADC，SUB，SBC，DAA）
- 逻辑运算 （AND，OR，XOR，CPL）
- 移位运算 （ RL，RR，RLC，RRC ）
- 递增和递减运算 （ INC，DEC ）
- 分支转移 （ SZ，SNZ，SIZ，SDZ 等）

算术逻辑单元 ALU 不仅会保存运算的结果而且会改变状态寄存器。

状态寄存器 （STATUS）

符号	位	功 能
C	0	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。它也可被一个循环移位指令影响。
AC	1	在加法运算中低四位产生了进位或减法运算中在低四位不产生借位，AC 被置位；反之，AC 被清除。
Z	2	算术运算或逻辑运算的结果为零则 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，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 溢出，TO 被置位。
—	6	未定义，读出为零
—	7	未定义，读出为零

8 位的状态寄存器(0AH)，由零标志位(Z)、进位标志位(C)、辅助进位标志位(AC)、溢出标志位(OV)、掉电标志位(PD)、看门狗定时器溢出标志位(TO)组成。该寄存器不仅记录状态信息，而且还控制运算顺序。

除了 TO 和 PD 以外，状态寄存器中的位都可用指令来改变，这种情况与其它寄存器一样。任何写到状态寄存器的数据不会改变 TO 或 PD 标志位。但是与状态寄存器有关的运算会导致状态寄存器的改变。

系统上电，看门狗定时器溢出或执行“CLR WDT”或“HALT”指令，能改变看门狗定时器溢出标志位（TO）。系统上电，或执行“CLR WDT”或“HALT”指令，能改变掉电标志位（PD）。

Z，OV，AC 和 C 标志位都反映了当前的运算状态。

另外，在进入中断子程序或执行子程序调用时，状态寄存器的内容不会自动压入堆栈。如果状态寄存器的内容是重要的，而且子程序会改变状态寄存器的内容，那么程序员必须事先将其保存好，以免被破坏。

中 断 （INT）

HT48R06A 提供一个外部中断和内部定时/计数器中断。中断控制寄存器（INTC；0BH）包含了中断控制位，用来设置中断允许/禁止及中断请求标志。

一旦有中断子程序被服务，所有其它的中断将被禁止（通过清除 EMI 位）。这种机制能防止中断嵌套。这时如有其它中断请求发生，这个中断请求的标志会被记录下来。如果在一个中断服务程序中有另一个中断需要服务的话，程序员可以设置 EMI 位及 INTC 所对应的位来允许中断嵌套服务。如果堆栈已满，该中断请求将不会被响应。即使相关的中断被允许，也要到堆栈指针发生递减时才会响应。如果需要立即得到中断服务，则必须避免让堆栈饱和。

寄存器	位	符号	功 能
INTC（0BH）	0	EMI	主中断控制位 1=允许，0=禁止
	1	EEI	外部中断控制位 1=允许，0=禁止
	2	ETI	定时/计数器中断控制位 1=允许，0=禁止
	3	—	未使用位，读出为零
	4	EIF	外部中断请求标志位 1=有效，0=无效
	5	TF	定时/计数器中断请求位 1=有效，0=无效
	6	—	未使用位，读出为零
	7	—	未使用位，读出为零

中断控制寄存器 INTC （0BH）

所有的中断都具有唤醒功能。当一个中断被服务时，会产生一个控制传送：通过将程序计数器（PC）压入堆栈，然后转移到中断服务程序的入口。只有程序计数器的内容能压入堆栈。如果寄存器和状态寄存器的内容会被中断服务程序改变，从而破坏主程序的预定控制，那么程序员必须事先将这些数据保存起来。

外部中断是由 $\overline{\text{INT}}$ 脚上的电平由高到低的变化触发的，相关的中断请求位（EIF，INTC 的第 4 位）被置位。当中断是被允许，堆栈也没有满，一个外部中断触发时，那么将会产生地址 04H 的子程序调用。中断请求标志（EIF）位和 EMI 位将也会被清除来禁止中断的再次发生。

内部定时/计数器中断发生时，会设置定时/计数器中断请求标志位（TF，INTC 的第五位），中断的请求是由定时器溢出产生的。当中断是允许的，堆栈又未满，并且 TF 已被置位，就会产生地址 08H 的子程序调用。该中断请求标志位（TF）被复位并且 EMI 位将被清除，以便禁止再次中断的发生。

单片机在执行中断子程序期间，其他的中断响应会被暂停，直到 RETI 指令被执行或是 EMI 位和相关的中断控制位都被置为 1（当堆栈是未满时）。若要从中断子程序返回时，只要执行 RET 或 RETI 指令即可。RETI 指令将会自动置位 EMI 位来允许中断服务，而 RET 则不能自动置位 EMI。

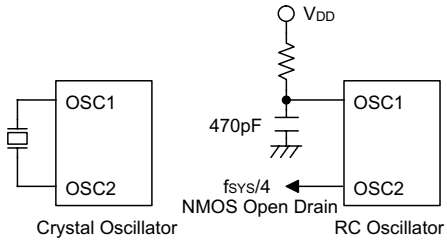
如果中断在二个连续的 T2 脉冲的上升沿间发生，中断响应被允许的话，那么在二个 T2 脉冲后，该中断会被服务。如果同时发生中断服务请求，那么下列表中列出了中断服务优先等级。这种优先级也可以通过 EMI 位的复位来屏蔽。

中 断 源	优 先 级	中断
外部中断	1	04H
定时/计数器溢出	2	08H

中断控制寄存器（INTC）其 RAM 地址是 0BH，由定时/计数器中断请求标志位（TF）、外部中断请求标志位（EIF）、定时/计数器允许位（ETI）、外部中断允许位（EEI）和主中断控制允许位（EMI）组成。EMI、EEI 和 ETI 是用来控制中断的允许/禁止的状态的。这些位防止正在进行中断服务中的中断请求。一旦中断请求标志位（EEI，ETI）被置位，它们将在 INTC 中被保留下来，直到相关的中断被服务或由软件指令来清除。

建议不要在中断子程序中使用“CALL”指令来调用子程序，因为它可能会破坏原来的控制序列，而中断经常随机发生或某一个确定的应用程序可能要求立即服务。基于上述情况，如果只剩下一个堆栈，若此时中断不能很好地被控制，而且在这个中断服务程序中又执行了 CALL 子程序调用，则会造成堆栈溢出而破坏原先的控制序列。

振荡器的配置



通过掩膜选项，可以选择石英晶振振荡器或 RC 振荡器。两者都可作为系统时钟。不管用哪种振荡器，其信号都支持系统时钟。HALT 模式会停止系统振荡器并忽视任何外部信号，由此来节省电源。

如果采用 RC 振荡器，那么在 VDD 与 OSC1 之间要接一个外接电阻。其阻值范围为 51K~1M。在 OSC2 端可获得系统时钟四分频信号，它可以用于同步系统外部逻辑。RC 振荡器是一种低成本的方案，但是振荡频率由于 VDD，温度及芯片自身参量的漂移，而产生误差。因此，要求精确度高的振荡器频率的场合，RC 振荡器是不适用的。

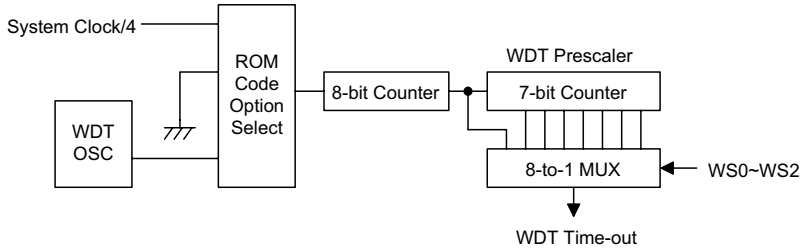
如果选用石英晶体振荡器，在 OSC1 与 OSC2 之间连接一个石英晶体，被用来提供石英振荡器所要的反馈（Feedback）和相移（Phase Shift)。除此以外，不再需要其他外部元件。另外，在 OSC1 与 OSC2 之间接一个谐振器（Resonator) 来取代石英振荡器用来得到参考频率，但是需要在 OSC1 与 OSC2 外接二个电容器（如果振荡器频率小 1MHz）。

WDT 振荡器是一个独立的 RC 振荡器，不需要外部元件。甚至系统进入省电模式时，系统时钟停止了，但 WDT 振荡器仍然以大约 65us（5V）的周期工作。由掩膜选项，WDT 振荡器可以被禁止来节省电能。

看门狗定时器 （WDT）

WDT 的时钟源是一个专用的 RC 振荡器（WDT 振荡器）或是由指令时钟（系统时钟 4 分频）来实现的，由掩膜选项来决定。WDT 是用来防止程序不正常运行或是跳到未知或不希望去的地址，而导致不可预见的结果。WDT 可以被掩膜选项禁止。如果 WDT 定时器被禁止，所有与 WDT 有关的执行都导致一个空操作。

如果设置了内部 WDT 振荡器（以 65us/5V 为周期的 RC 振荡器）的话，WDT 的值会先除以 256（8 级）来产生大约 16. 6ms/5V 的额定溢出时间，这个溢出时间会因为温度，VDD，以及芯片参数的变化而变化。如果启动 WDT 的前置分频器，则可实现更长的溢出时间。写数据到 WS2,WS1,WS0 (WDTS 的 2、1、0 位）会产生不同的溢出时间，如果 WS2，WS1，WS0 的值都为 1，其分频因子最多为 1：128，并且溢出时间最长约为 2.2s/5V。如果 WDT 被禁止，那么 WDT 的时钟来源可来自指令时钟，其运作与 WDT 振荡器一样。但当在 HALT 状态时，来源于指令时钟的 WDT 会在暂停模式时停止计数并失去保护功能。在这种情况下，只能由外部逻辑来重新启动系统。WDTS 的高四位及其第 3 位保留给程序员定义标志来使用，程序员可以利用这些标志来指示某些特殊的状态。



如果单片机工作在干扰很大的环境中，那么强烈建议使用片上的 RC 振荡器（WDT OSC），因为 HALT 模式会使系统时钟终止运作。

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

看门狗定时器前置分频寄存器

在正常运作下，WDT 溢出会使系统复位并置位 TO 状态位。但在 HALT 模式下，溢出只产生一个“热复位”，并只能使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$\overline{RES}$端），用软件指令和 HALT 指令三种。软件指令由 CLR WDT 和另一组指令 CLR WDT1 及 CLR WDT2 组成。这两组指令中，只能选取其中一种。选择的方式由掩膜选项的 CLR WDT 次数选项决定。如果“CLR WDT”被选择（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

暂停模式 （HALT）

暂停模式是由 HALT 指令来实现的，产生如下结果：

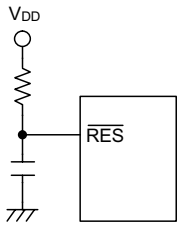
- 关闭系统振荡器，但 WDT 振荡器继续工作（选 WDT 振荡器选）。
- RAM 及寄存器的内容保持不变。
- WDT 和 WDT 前置分频器被清除并再次重新计数（如果 WDT 的时钟来自 WDT 振荡器）。
- 所有的 I/O 端口都保持其原先状态。
- PD 标志位被置位，TO 标志位被清除。

由于外部复位、中断、外部输入一个下降沿的信号到 PA 口或 WDT 溢出，可使系统脱离暂停状态。外部复位能使系统初始化而 WDT 溢出使系统“热复位”。测试 TO 和 PD 状态后，系统复位的原因就可以被确定。PD 标志位是由系统上电复位和执行 CLR WDT 指令被清除，而它的置位是由于执行了 HALT 指令。如果 WDT 产生溢出，会使 TO 标志位置位，还能产生唤醒使得程序计数的 PC 和堆栈指针 SP 复位。其他都保持原状态。

PA 端口的唤醒和中断方式被看作为正常运行的继续。PA 口的每一位都可以通过掩膜选项来单独设定为唤醒系统。唤醒是来自于 I/O 口的信号变化，程序会重新继续执行下一条命令。如果唤醒是来自于中断，那么二种情况可能发生：如果相关的中断被禁止或中断是允许的，但堆栈已满，那么程序将继续执行下一条指令，如果中断允许并且堆栈未满，那么这个中断响应就发生了。如果在进入 HALT 模式以前，中断请求标志位被置“1”，那么相关的中断唤醒功能被禁止。一旦唤醒事件发生，要花 1024tsys（系统时钟周期）后，系统重新正常运行。换句话说，在唤醒后被插入了一个等待时间。如果唤醒是来自于中断响应，那么实际的中断程序执行就被延迟了一个或一个以上的周期。但是如果唤醒导致下一条指令执行，那么在一个等待周期结束后指令就立即被执行。

为了减少功耗，在进入 HALT 模式之前必须要小心处理 I/O 端口状态。

复 位（RESET）



有三种方法可以产生复位

- 在正常运行时期由 $\overline{\text{RES}}$ 脚产生复位。
- 在 HALT 期间 $\overline{\text{RES}}$ 脚产生复位。
- 正常运行时，WDT 溢出复位。

在 HALT 期间 WDT 溢出是不同于其它的复位操作条件，因为它可以“热复位”，结果只能使程序计数器 PC 和堆栈指针 SP 复位，而别的电路均保持原来的状态。在其他复位条件下，某些寄存器保持不变。当复位条件被满足时，大多数的寄存器被复位到“初始状态”，通过测试 PD 标志位和 TO 标志位，程序能分辨不同的系统复位。

TO	PD	复 位 条 件
0	0	上电时复位 $\overline{\text{RES}}$
u	u	正常动作时 $\overline{\text{RES}}$ 复位
0	1	$\overline{\text{RES}}$ 时唤醒 HALT
1	u	正常动作时 WDT 定时溢出
1	1	WDT 时唤醒 HALT

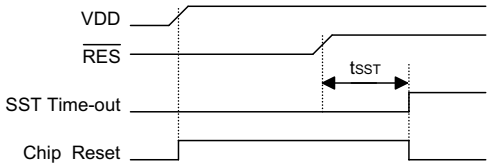
注释：“u”表示未变化。

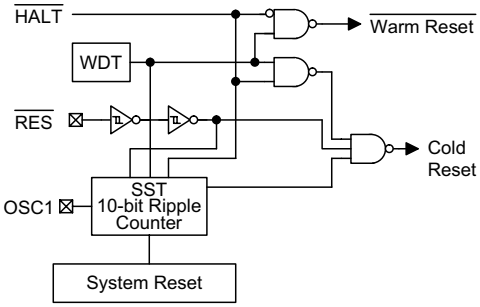
为保证系统振荡器起振并稳定运行，SST（系统启动定时器）当系统复位（上电，WDT 定时溢出或 $\overline{\text{RES}}$）或从 HALT 状态被唤醒时，提供额外延迟 1024 个系统时钟脉冲。

当系统复位时，SST 延迟被加到复位周期中。任何来自 HALT 的唤醒都将允许 SST 延迟。

系统复位时各功能单元的状态如下所示：

PC	000H
中断	禁止
前置分频器	清除
WDT	清除。在主系统复位后，WDT 开始计数
定时/计数器	停止
输入/输出端口	输入模式
堆栈指针 SP	指向堆栈顶部





特殊功能寄存状态表

寄存器	上电复位	正常运行期间		HALT 状态	
		WDT 溢出溢出	RES 端复位	RES 端复位	WDT 溢出溢出
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
PC	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	---- -111	---- -111	---- -	---- -111	---- -uuu
PBC	---- -111	---- -111	---- -	---- -111	---- -uuu
PC	---- --11	---- --11	---- --11	---- --11	---- -- uu
PCC	---- --11	---- --11	---- --11	---- --11	---- -- uu

注意：1. “*” 表示 “热复位。
2. “U” 表示不变化。
3. “×” 表示不确定。

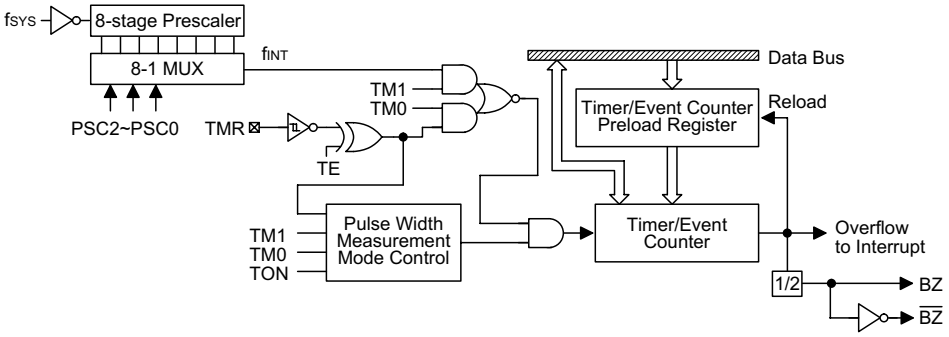
定时/计数器

这个单片机提供一个定时/计数器。定时/计数器包含一个 8 位可编程的向上计数的计数器，时钟可以来自外部的时钟源或是系统时钟。

如果采用内部系统时钟，那么只有一个参考时基信号。这个内部时钟源来自 fsys。

而外部时钟输入，允许程序员去计量外部事件、测量时间长度、脉冲宽度或产生一个精确的时基信号。

有两个寄存器与定时/计数器相关联，即 TMR（[0DH]）和 TMRC（[0EH]）。有两个物理寄存器对应 TMR 的位置。写入 TMR 会将初始值装入到定时/计数器的预置寄存器中，而读 TMR 则会获得定时/计数器的内容。TMRC 是定时/计数器控制寄存器。



TM0 和 TM1 位定义操作模式。事件计数模式用来记录外部事件，这意味着时钟来源来自外部 TMR 引脚。定时器模式是作为一个普通的定时器功能，时钟源来自 fINT 时钟。脉冲宽度测量模式能用来计算外部引脚 TMR 上的高电平或低电平的宽度。计数是基于 fINT 时钟。

在事件计数或定时器模式中，一旦定时/计数器开始计数，它将会从当前定时/计数器中的数值开始计数到 FFH。一旦产生溢出，计数器会从定时/计数器预置寄存器重新装载并且同时产生相应的中断请求状态位（TF ； INTC 的第 5 位）。

在脉冲宽度测量中，将 TON 和 TE 位置为“1”，如果 TMR 接收到从低到高的电平跳变（或从高到低的变化，如果 TE 位被清零），就开始计数直到 TMR 返回到原来的电平并且复位 TON 位。测量的结果被保留在定时/计数器中，甚至电平跳变再一次发生也不会改变。换句话说，一次只能测量一个周期。直到 TON 再次被置位，接到跳变信号，那么测量过程会再次执行。要注意在这个操作模式中，定时/计数器的启动计数不是根据逻辑电平，而是依据信号的边缘跳变触发。一旦发生计数器溢出，计数器会从定时/计

数器的预置寄存器重新装入，并发出中断请求，这种情况与另外两个模式一样。要使得计数运行，只要将定时器启动位（TON；TMRC 的第 4 位）置成为 1。在脉宽测量模式中，TON 在测量周期结束后自动被清除。但在另外两个模式中，TON 只能由指令来复位。定时/计数器溢出是唤醒的信号之一。不管任何模式，若写 0 到 ETI 位即可禁止相应的中断服务。

在定时/计数器为 0FF 的状态下，写数据到定时/计数器的预置寄存器之中，同时也会将数据装入定时/计数器中。但若是定时/计数器已经开启，写到定时/计数器的数据只会被保留在定时/计数器的预置寄存器中，直到定时/计数器发生计数溢出为止，再由预置寄存器加载新的值。当定时/计数器的数据被读取时，会禁止时钟输入以防出错。因为禁止时钟输入可能导致计数错误，所以程序员必须仔细加以考虑。

TMRC 的 0~2 位被用于定义定时/计数器的内部时钟源的前置分频因子。定义如表所示。定时/计数器的溢出信号被用于产生驱动蜂鸣器的 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}= fsys/2 001：f_{INT}= fsys/4 010：f_{INT}= fsys/8 011：f_{INT}= fsys/16 100：f_{INT}= fsys/32 101：f_{INT}= fsys/64 110：f_{INT}= fsys/128 111：f_{INT}= fsys/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 = 电平由商到低有效
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

TMR 寄存器

输入/输出端口

单片机具有 13 个双向输入输出口，标号从 PA 到 PC，其分别对应的 RAM 的[12H]，[14H]和[16H]。所有的 I/O 端口都能被作为输入或输出使用。就输入而言这些端口不具有锁存功能，即，输入数据必须在“MOV A，[m]”(m=12H,14H 或 16H) 指令的 T2 上升沿被准备好。对输出而言，所有的数据被锁存并保持不变，直到输出锁存器重新被改写。

每个 I/O 口都有其自己的控制寄存器（PAC，PBC，PCC），用来控输入/输出的设置。由于使用了控制寄存器，CMOS 输出、斯密特触发输入可以在软件下动态地进行配置。要设置为输入功能，相应的控制寄存器必须被写入“1”。信号源的输入也取决于控制寄存器。如果控制寄存器的某位值为“1”那么输入信号是读取这个引脚（PAD）的状态，但是如果控制寄存器的某位值为“0”，那么锁存器的内容将会被送到内部总线。后者，可以在“读改写”指令中发生。

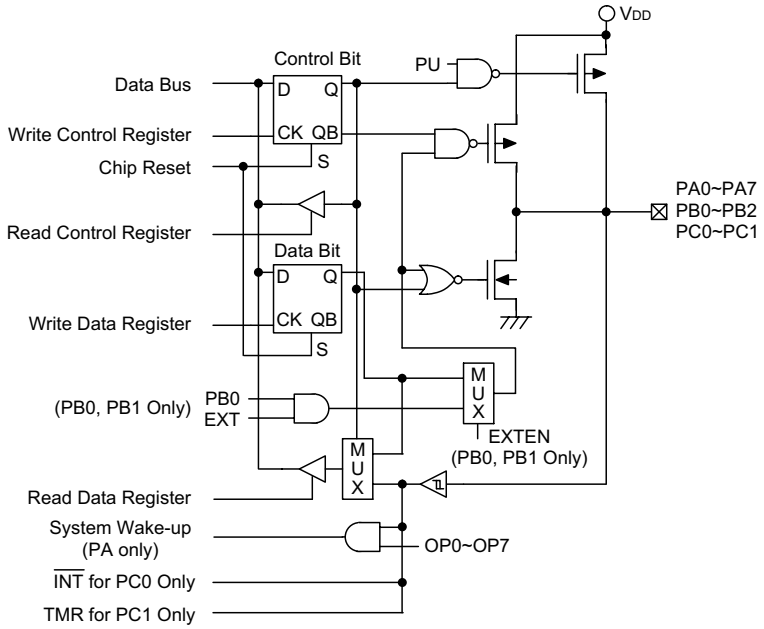
对于输出功能，只有 CMOS 需要设置。这些控制寄存器是对应的内存地址为 13H，15H，17H。

芯片复位后，这些输入/输出口都会是高电平或浮空状态（取决于上拉电阻的选项）。每一个输入/输出锁存位都能被 SET [m].i 或 CLR [m].i 指令置位或清除，m=12H，14H 或 16H。

某些指令会首先输入数据然后进行输出操作。例如，SET [m].i ，CLR [m].i，CPL [m]和 CPLA [m]指令，读取输入口的状态到 CPU，执行这个被定义的操作（位操作），然后将数据写回锁存器或累加器。

PA 口的每一个口都具有唤醒系统的能力。PC 端口的高 6 位和 PB 端口的高 5 位在物理上是不存在的；读这些位将返回“0”，而写这些位结果为无操作。请看应用注释。

所有的 I/O 口都可以有上拉电阻的选择。一旦选择上拉电阻，所有的 I/O 口都具有上拉电阻。否则没有上拉电阻。必须要注意的是：没有上拉电阻的 I/O 口工作在输入模式会产生浮空状态。



EXT=BZ for PB0 only, EXT=$\overline{BZ}$ for PB1 only, control=PB0 data register

PB0 和 PB1 分别与 BZ 和 $\overline{BZ}$ 共享引脚。如果 BZ/$\overline{BZ}$ 的选项被选择，PB0/PB1 在输出模式时的输出信号将是由定时/计数器的溢出信号产生的 PFD 信号。在输入模式始终保持它的原来的功能。一旦 BZ/$\overline{BZ}$ 的选项被选择，蜂鸣器的输出信号只受 PB0 数据寄存器控制。PB0/PB1 的 I/O 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	D_0	0	1
PB1 数据	×	D	×	×	×	×	×	D_1	×	×
PB0 Pad 状态	I	I	I	I	D	0	B	D_0	0	B
PB1 Pad 状态	I	D	0	B	I	I	I	D_1	0	B

注释：I：输入；O：输出；D，D0，D1 ：数据；

B：蜂鸣器的选项，BZ 或 $\overline{BZ}$ ；×：任意值；

C：CMOS 输出；

PC0 和 PC1 分别与 $\overline{INT}$ 和 TMR 共享引脚。为了避免在浮空状态下功耗太大，建议将未用的或没有联结到外部的 I/O 口由软件指令设置成输出引脚。

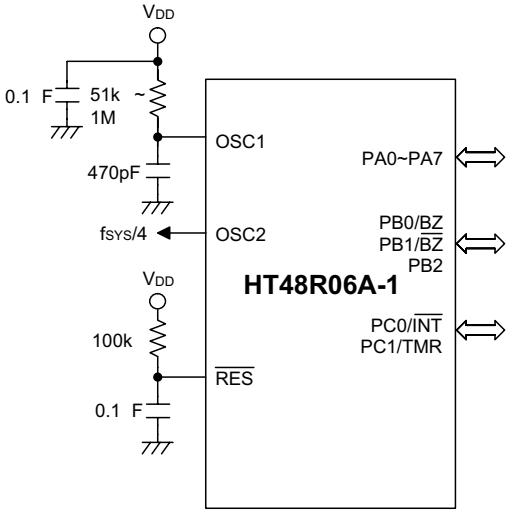
掩膜选项

下表示出了该单片机的各掩膜选项。为了满足系统功能，所有的掩膜选项必须定义正确。

编号	选　　项
1	WDT 时钟源：WDTOSC/F_{tid}
2	WDT 允许/禁止：允许/禁止
3	LVD 允许/禁止：允许/禁止
4	CLR WDT 指令：一条或二条清除 WDT 指令
5	系统振荡器：RC/石英晶振
6	上拉电阻（PA~PC）：无上拉电阻或有上拉电阻
7	BZ 选项：允许/禁止
8	PA0～PA7 唤醒功能：允许/禁止
9	锁定：不锁定/锁定

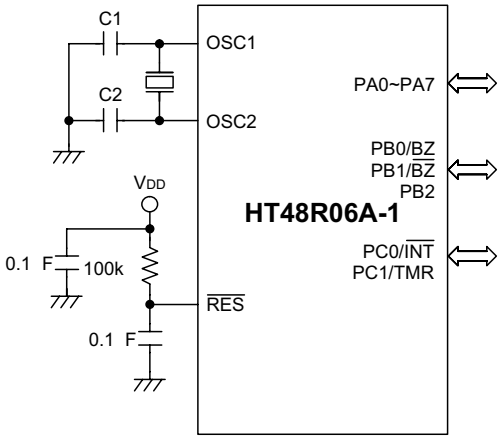
应用电路

RC oscillator for multiple I/O applications



Note: The resistance and capacitance for reset circuit should be designed to ensure that the VDD is stable and remains in a valid range of the operating voltage before bringing $\overline{RES}$ to high.

Crystal oscillator for multiple I/O applications



Note: C_1=C_2=300pF if f_{SYS}<1MHz
Otherwise, C_1=C_2=0.

注：本文译至 July 02，2001 的 48R06A-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 Piaza，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