

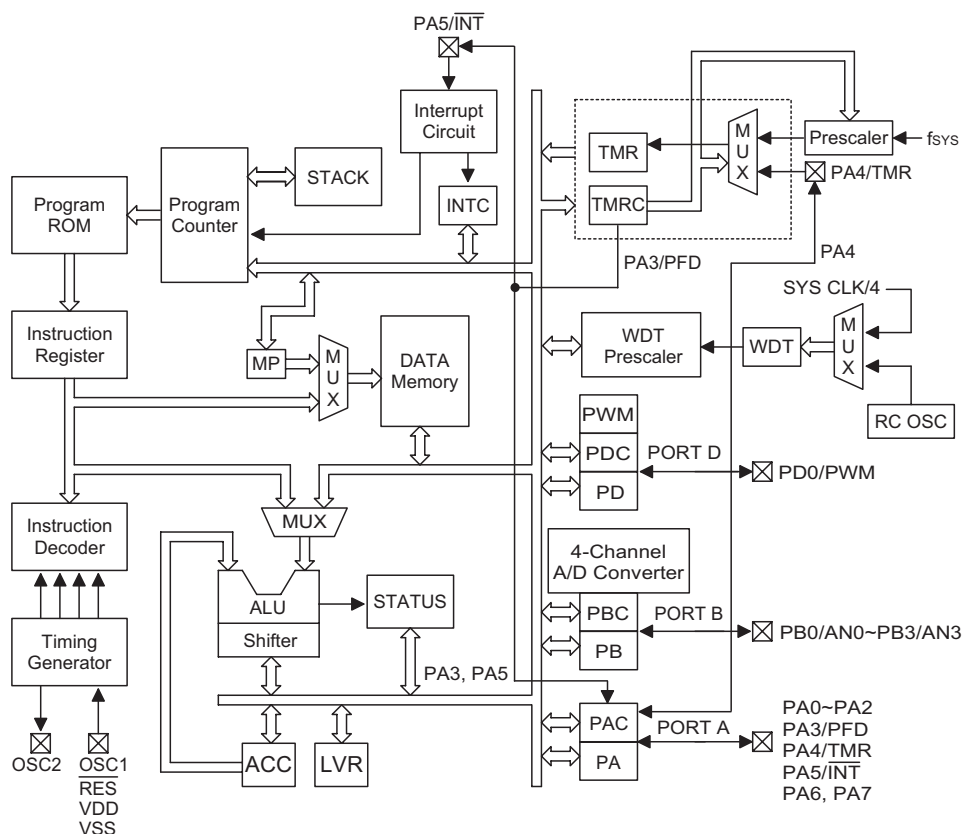
特点

- 工作电压：
 - $f_{\text{SYS}}=4\text{MHz}$: 2.2~5.5V
 - $f_{\text{SYS}}=8\text{MHz}$: 4.5~5.5V
- 13 位双向输入/输出口
- 1 个外部中断输入，与输入/输出共用引脚
- 8 位带溢出中断的可编程定时/计数器，具有 7 级预分频器
- 石英晶体或 RC 振荡器
- 看门狗定时器
- 2048×14 位的程序存储器 PROM
- 64×8 位的数据存储器 RAM
- 具有 PFD（可编程分频器）输出，可以用来驱动蜂鸣器
- 具有暂停和唤醒功能来降低功耗
- 在 $V_{\text{DD}}=5\text{V}$ 且系统时钟为 8MHz 时，指令时钟为 0.5 μs
- 6 层堆栈
- 4 通道 9 位（8 位精度）的 A/D 转换器
- 一个（6+2）位的 PWM（脉宽可调）输出通道，与输入/输出口共用引脚
- 位操作指令
- 查表指令可查 14 位
- 63 条功能强大的指令
- 指令执行时间皆为 1 或 2 个指令周期
- 低电压复位功能
- 18 脚 DIP/SOP 封装

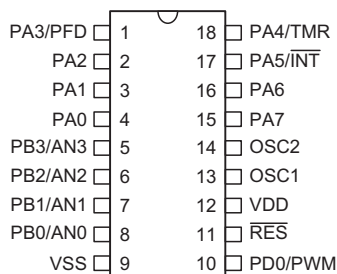
概述

HT46C47 为八位高性能精简指令集单片机。专门为多输入/输出口的产品而设计的。这些器件适用于充电器控制和 A/D 转换应用。具有 HALT 模式可降低功耗。

系统框图

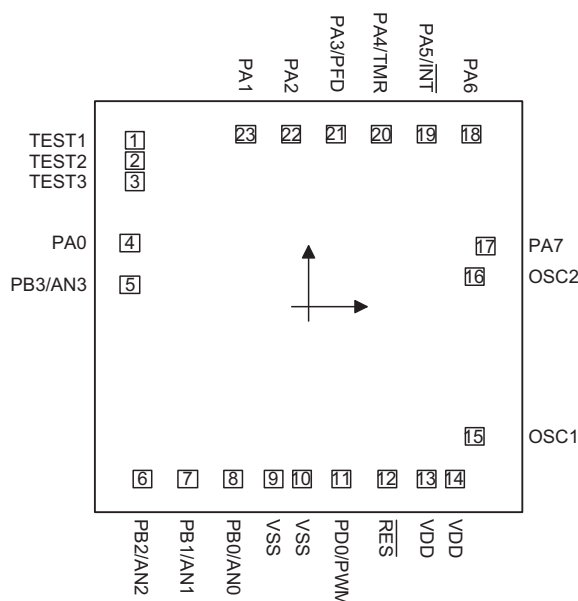


引脚图



HT46C47
- 18 DIP-A/SOP-A

PAD 图



*封装时 IC 衬底必须接地 Vss

引脚说明

引脚名称	输入/输出	掩膜选项	功能说明
PA0-PA2 PA3/PFD PA4/TMR PA5/ $\overline{\text{INT}}$ PA6, PA7	输入/输出	上拉电阻 唤醒功能 PA3 或 PFD	8 位双向输入/输出口。每一位能被掩模选项设置为输入唤醒。可由软件指令决定为 CMOS 输出或斯密特触发输入带或不带上拉电阻（由 pull high 掩模选项决定：位选择）。PFD、TMR 和 $\overline{\text{INT}}$ 与 PA3, PA4 和 PA5 共用引脚。
PB0/AN0 PB1/AN1 PB2/AN2 PB3/AN3	输入/输出	上拉电阻	4 位双向输入/输出口。软件编程决定 CMOS 输出，带或不带上拉电阻的斯密特触发输入（由 pull high 掩模选项决定：位选择）或是 A/D 输入。一旦 PB 口有任何一个口作为 A/D 输入（软件编程控制），I/O 和上拉电阻会自动无效。
V _{SS}	—	—	电源负极，GND。
PD0/PWM	输入/输出	上拉电阻 PD0 或 PWM	双向的 I/O 口。软件编程决定 CMOS 输出，带或不带上拉电阻的斯密特触发输入（由 pull high 掩模选项决定：位选择）。PWM 的输出功能与 PD0 共用引脚（决定于 PWM 掩模选项）
RES	输入	—	斯密特触发复位输入，低电平有效。
V _{DD}	—	—	电源正极。
OSC1 OSC2	输入 输出	石英晶体或 RC	OSC1 和 OSC2 连接至 RC 或 Crystal 振荡器（由掩模选项确定）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。
TEST1 TEST2 TEST3	输入	—	测试模式输入引脚。 正常应用时不必连接。

极限参数

工作电压：V_{SS}-0.3V~V_{SS}+5.5V

储存温度：-50℃~125℃

输入电压：V_{SS}-0.3V~V_{SS}+0.3V

工作温度：-40℃~85℃

注：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
 $T_a=25^{\circ}\text{C}$

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	—	$f_{SYS}=4\text{MHz}$	2.2	—	5.5	V
V_{DD2}	工作电压	—	$f_{SYS}=8\text{MHz}$	4.5	—	5.5	V
I_{DD1}	工作电流 (石英晶振荡器)	3V	无负载, $f_{SYS}=4\text{MHz}$	—	0.6	1.5	mA
		5V	ADC 禁止	—	2	4	mA
I_{DD2}	工作电流 (RC 振荡器)	3V	无负载, $f_{SYS}=4\text{MHz}$	—	0.8	1.5	mA
		5V	ADC 禁止	—	2.5	4	mA
I_{DD3}	工作电流	5V	无负载, $f_{SYS}=8\text{MHz}$ ADC 禁止	—	2.5	5	mA
I_{ADC}	ADC 允许, 其他禁止	3V	无负载	—	0.5	1	mA
		5V		—	1.5	3	mA
I_{STB1}	静态电流 (看门狗允许)	3V	无负载 系统 HALT	—	—	5	μA
		5V		—	—	10	μA
I_{STB2}	静态电流 (看门狗禁止)	3V	无负载 系统 HALT	—	—	1	μA
		5V		—	—	2	μA
V_{AD}	A/D 输入电压	—	—	0	—	V_{DD}	V
V_{IL1}	I/O 口、TMR、 $\overline{\text{INT}}$ 输入低电压	3V	—	0	—	$0.3 V_{DD}$	V
		5V	—	0	—	$0.3 V_{DD}$	V
V_{IH1}	I/O 口、TMR、 $\overline{\text{INT}}$ 输入高电压	3V	—	$0.7 V_{DD}$	—	V_{DD}	V
		5V	—	$0.7 V_{DD}$	—	V_{DD}	V
V_{IL2}	输入低电压 ($\overline{\text{RES}}$)	3V	—	0	—	$0.4 V_{DD}$	V
		5V	—	0	—	$0.4 V_{DD}$	V
V_{IH2}	输入高电压 ($\overline{\text{RES}}$)	3V	—	$0.9 V_{DD}$	—	V_{DD}	V
		5V	—	$0.9 V_{DD}$	—	V_{DD}	V
V_{LVR}	低电压复位	—	—	2.7	3.0	3.3	V
I_{OL}	I/O 口灌电流	3V	$V_{OL}=0.1 V_{DD}$	4	8	—	mA
		5V	$V_{OL}=0.1 V_{DD}$	10	20	—	mA
I_{OH}	I/O 口源电流	3V	$V_{OH}=0.9 V_{DD}$	-2	-4	—	mA
		5V	$V_{OH}=0.9 V_{DD}$	-5	-10	—	mA
R_{PH}	上拉电阻	3V	—	40	60	80	$\text{K}\Omega$
		5V	—	10	30	50	$\text{K}\Omega$
E_{AD}	A/D 转换误差	5V	—	—	± 0.5	± 1	LSB

交流特性
 $T_a=25^{\circ}\text{C}$

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
f_{SYS1}	系统时钟 (石英晶体振荡器)	3V	—	400	—	4000	kHz
		5V	—	400	—	8000	kHz
f_{SYS2}	系统时钟 (RC 振荡器)	3V	—	400	—	4000	kHz
		5V	—	400	—	8000	kHz
f_{TIMER}	定时器 I/P 频率 (TMR)	3V	—	0	—	4000	kHz
		5V	—	0	—	8000	kHz
t_{AD}	A/D 时钟	5V	—	1	—	—	μs
t_{ADC}	A/D 转换时间	—	—	—	76	—	t_{AD}
t_{WDIOSC}	看门狗振荡器	3V	—	43	86	168	μs
		5V	—	36	72	144	μs

t _{WDT1}	看门狗溢出周期 (RC 振荡器)	3V	—	1.4	2.8~5.6	11	s
		5V		1.1	2.3~4.7	9.4	s

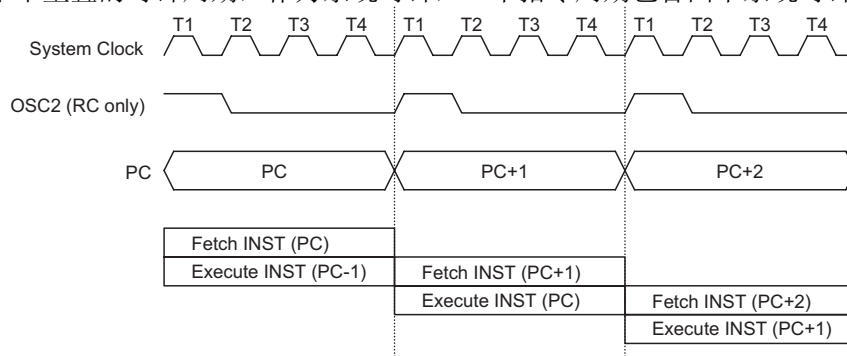
符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
t _{WDT2}	看门狗溢出周期 (系统时钟)	—	—	2 ¹⁷	—	2 ¹⁸	t _{sys}
t _{RES}	外部复位低电平的脉宽	—	—	1	—	—	μs
t _{SST}	系统启动延迟时间	—	上电或从 HALT 状态唤醒	—	1024	—	*t _{sys}
t _{INT}	中断脉冲宽度	—	—	1	—	—	μs

注：*t_{sys} = 1/f_{sys}

功能描述

指令执行时序

系统频率由石英晶体振荡器、RC 振荡器或实时时钟晶振产生振荡频率。系统内部对此频率进行四分频，有效地产生四个不重叠的时钟周期，作为系统时钟，一个指令周期包含四个系统时钟周期。



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。如果涉及到的指令要改变程序计数器的值，就需要花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 PC 控制了存储在程序存储器 PROM 里的指令的执行顺序，且 PC 内容可指定程序存储器的整个空间。

取得指令码以后，程序计数器会自动增 1，指向下一个指令码的地址。

但是如果执行跳转、条件跳转、子程序调用、中断、复位初始化或由中断返回等情况，程序计数器会载入相对应的地址，而非下一个地址。

举例而言，如果执行条件跳跃指令，当条件符合时，则在执行此条指令时抓取的指令码会被抛弃，且一个假指令周期 (dummy cycle) 会被插入，程序计数器才会指向正确的指令码地址，因此需要两个指令周期。反之，程序计数器指向下一条指令码的地址。

程序计数器的低字节单元 (PCL) 是一个可读写的暂存器 (06H)，如果写入一个值将会产生一个一个短程跳跃动作，这个短程跳跃范围是 256 个地址。同样在此情况下也会插入一个假指令周期。

模 式	程序计数器										
	*10	*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	0
外部中断	0	0	0	0	0	0	0	0	1	0	0
定时/计数器溢出	0	0	0	0	0	0	0	1	0	0	0
A/D 转换中断	0	0	0	0	0	0	0	1	1	0	0
短跳转	PC+2										
装载 PCL	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
跳转, 子程序调用	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
从子程序返回	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

程序计数器

注: *11 ~ *0 : 程序计数器位

S11 ~ S0 : 堆栈寄存器位

#11 ~ #0 : 指令代码位

@7 ~ @0 : PCL 位

程序存储器 (PROM)

程序存储器 (PROM) 用来存放要执行的指令代码。还包括数据, 表格和中断向量。程序存储器 (PROM) 为 2048×14 位。程序存储器 (PROM) 的空间都可以由程序计数器 (PC) 或表格指针 (Table Pointer) 来寻址。

下所列出的程序存储器的地址是专为特殊用途而保留的。

- 地址 000H

此地址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

此地址保留给外部中断服务使用。当中断允许, 且堆栈未满, 则一旦 INT 被电平触发, 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。

- 地址 008H

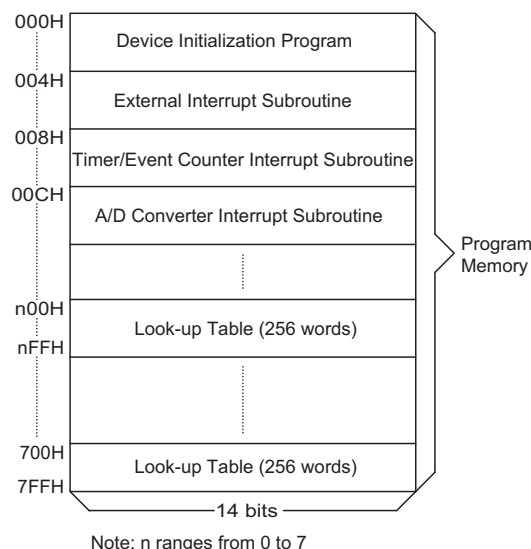
此地址保留给定时/计数器中断服务使用。当中断允许, 且堆栈未满, 则一旦定时/计数器发生溢出时, 就能产生中断, 程序会从 008H 地址开始执行中断服务程序。

- 地址 00CH

此地址保留给 A/D 转换中断使用。当中断允许, 且堆栈未满, 则一旦发生 A/D 转换中断, 程序会从 00CH 地址开始执行外部中断服务程序。

- 表格地址

PROM 内的任何地址都可用来作为查表地址使用。查表指令为 TABRDC [m] 与 TABRDL [m]。TABRDC [m] 是查当前页的内容 [1 页=256 个字 (word)]。TABRDL [m] 是查表最后一页的内容。[m] 为数据被存入的地址。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], 而这个字的高位字节传送到 TBLH (08H)。表格中的低位字节被送到目标地址中, 而表格中的其他位送到 TBLH 的低位, 剩余的一位读数为 0。TBLH 为只读寄存器, 而表格指针 (TBLP) 是可以读写的寄存器 (07H), 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。查表指令要花两个指令周期来完成这一条指令的操作。按照用户的需要, 表格地址这些位置可以作为正常的程序存储器来使用。高位字节寄存器 TBLH 只能读出, 不能写入。如果主程序和中断服务程序 (ISR) 同时使用查表指令, 那么主程序读取的高位字节 (即存放于高位字节寄存器 TBLH 之中) 可能会被中断服务程序的查表指令改写而产生错误。因此, 应该避免主程序和中断服务程序 (ISR) 同时使用查表指令。但是, 如果主程序和中断服务程序必须同时使用查表指令的话, 那么, 主程序在使用查表指令之前, 必须先关闭所有使用查表指令的相关中断, 直到高位字节寄存器 TBLH 的内容被备份好再开放这些中断。查表指令要花两个指令周期来完成这一条指令的操作。按照用户的需要, 表格地址这些位置可以作为正常的程序存储器来使用。



指令	表格地址										
	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC[m]	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL[m]	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

表格地址

注：*10 ~ *0 : 表格地址

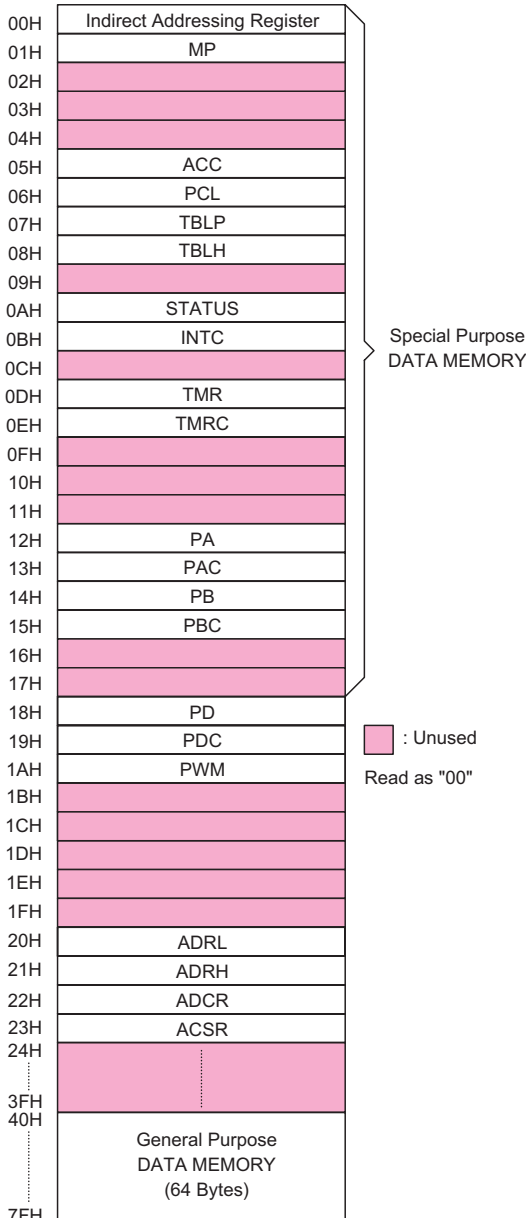
P10 ~ P8 : 当前程序计数器

@7 ~ @0 : 表格指针

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, HT46C47 有 6 级堆栈。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) 来索引, 堆栈指针也不能读出与写入。一旦发生子程序调用或中断被响应时, 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被响应结束时 (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丢失。只有最后的六个返回地址会被保留着。



数据存储器 (RAM)

数据存储器 RAM 由 85×8 位组成。它可分成两个功能组, 即特殊功能寄存器和通用数据存储器 (64×8)。这两个功能组的极大部单元可以读写, 数据存储器而某些单元只能读出。

这两个功能组中, 特殊功能寄存器包括间接寻址寄存器 0 (00H), 定时/计数器 (TMR; 0DH), 定时/计数器控制寄存器 (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), 表格指针寄存器 (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), PWM 数据寄存器 (PWM; 1AH), A/D 转换低位寄存器 (ADRL; 20H), A/D 转换高位寄存器 (ADRH; 21H), A/D 转换控制寄存器 (ADCR; 22H), A/D 时钟设置寄存器 (ACSR; 23H), I/O 寄存器 (PA; 12H, PB; 14H, PD; 18H) 和 I/O 控制寄存器 (PAC; 13H, PBC; 15H, PDC; 19H)。余下的 40H 之前的存储器空间为保留部分, 读取这些单元, 会得到 "00H" 的值。从 40H 到 7FH 的通用数据存储器作为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 指令来置位和复位。这些 RAM 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 作为间接寻址寄存器。没有实际的物理结构。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 单元, 将会返回 00H, 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 的宽度是 7 位, MP 的第 7 位没有定义, 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 (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，并能对立即数进行操作，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。它提供如下的功能：

- 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的值。

状态寄存器 (STATUS)

状态寄存器 (0AH) 的宽度为 8 位，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)、省电标志位 (PD) 和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操作会导致与预期不一样的结果。系统上电、看门狗定时器溢出、执行“HALT”指令或“CLR WDT”都会改变 TO 标志位；PD 标志位只有在系统上电、执行“HALT”指令或“CLR WDT”指令时才会改变。而 Z、OV、AC 和 C 标志位都反映了当前的操作状态。

进入中断程序或执行子程序调用时，状态寄存器不会自动压入堆栈。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。

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,那么被 C 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	如果运算结果次高位向最高位进位, 但最高位并不产生进位输出, 或情况相反, 则 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。WDT 定时溢出, TO 被置位。
—	6, 7	未定义, 读出为零

状态寄存器

中断

HT46C47 提供一个外部中断、一个内部定时/计数器中断和一个 A/D 转换中断。中断控制寄存器 (INTC; 0BH) 具有的中断控制位是用来设置中断允许/禁止及中断请求标志位。

一旦有中断子程序被服务，其它的中断都被禁止（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，只是把这个中断请求的标志记录下来。如果在一个中断服务程序中有另一个中断须要服务的话，程序员可以设置 EMI 位及 INTC0 或 INTC1 所对应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。

所有的中断都具有唤醒功能。当一个中断被服务时，通过将程序计数器 (PC) 压入堆栈来产生一个控制转移，转移到被指定的 PROM 地址中的子程序的入口。只有程序计数器 (PC) 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 $\overline{\text{INT}}$ 引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 04H 单元的子程序调用。中断请求标志（EIF）和 EMI 位将也会被清除来禁止另外的中断发生。

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 5 位），中断的请求是由定时器溢出产生的。当中断是允许的，堆栈又未滿，并且 TF 已被置位，就会产生 08H 单元的子程序调用。该中断请求标志位（TF）被复位并且 EMI 位将被清除，以便将其他中断禁止。

A/D 转换中断发生时，会设定 A/D 转换请求标志位（ADF；INTC 的第六位），中断的请求是完成一次 A/D 转换产生的。当中断是允许的，堆栈又未滿，并且 ADF 已被置位，就会产生 0CH 单元的子程序调用。该中断请求标志位（ADF）被复位并且 EMI 位将被清除，以便将其他中断禁止。

单片机在执行中断子程序期间，其他的中断响应会被屏蔽，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（当堆栈是未滿时）。若要从中断子程序返回时，只要执行 RET 或 RETI 指令。RETI 指令将会自动置位 EMI 为允许中断服务，而 RET 则不能自动置位 EMI。

如果中断在两个 T2 脉冲的上升沿之间发生，且中断响应被允许的话，那么在第二个 T2 周期，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。这种优先权地位也可以通过 EMI 位的复位来屏蔽。

No.	中 断 源	优 先 权	中断向量
A	外部中断	1	04H
B	定时/计数器中断	2	08H
C	A/D 转换中断	3	0CH

中断控制寄存器（INTC）的 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）、A/D 转换中断请求标志位（ADF）、定时/计数器中断允许位（ETI）、外部中断允许位（EEI）、A/D 转换中断允许位（EADI）和主中断控制允许位（EMI）组成。EMI、EEI、ETI、和 EADI 都是用来控制中断的允许/禁止状态的。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被置位（TF，EIF，ADF），它们将在 INTC 中被保留下来，直到相关的中断被服务或被软件指令清除。

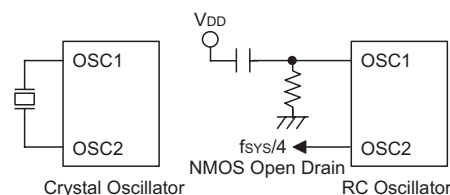
寄存器	位	符号	功 能
INTC(0BH)	0	EMI	总中断控制位 允许=1，禁止=0
	1	EEI	外部中断控制位 允许=1，禁止=0
	2	ETI	定时/计数器中断控制位 允许=1，禁止=0
	3	EADI	A/D 转换中断控制位 允许=1，禁止=0
	4	EIF	外部中断请求标志位 有效=1，无效=0
	5	TF	定时/计数器中断请求位 有效=1，无效=0
	6	ADF	A/D 转换中断请求位 有效=1，无效=0
	7	—	未使用位，读出为零

建议不要在中断子程序中使用“CALL”指令来调用子程序。因为中断经常随机发生在不可预见的情况中或某一个确定的应用程序可能要求立即服务的情况下，在这种时候，如果只剩下一个堆栈，若此时中断不能很好地被控制，在中断服务程序中调用了子程序调用的操作，可能使原先的控制序列被破坏。

振荡器

HT46C47 有两种方式提供系统时钟，RC 振荡器和石英晶体振荡器，可以通过掩膜选项设定。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以降低功耗。

如果采用 RC 振荡器，那么在 OSC1 与 Vss 之间要接一个外接电阻。其阻值范围为 30KΩ 到 750KΩ。在 OSC2 端接一个上拉电阻，可获得系统时钟四分频信号输出，它可以用于同步系统外部逻辑。RC 振荡器是一种低成本方案，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，它不能



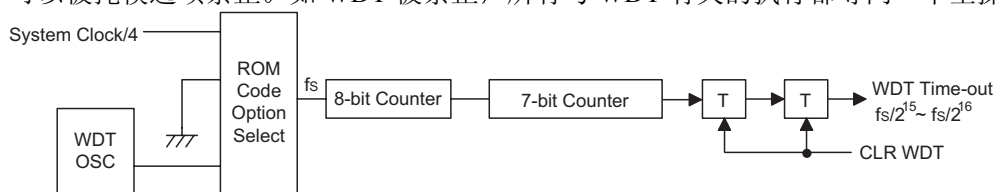
满足要求精确的振荡频率对时间敏感的应用。

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，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，可在 OSC1 与 OSC2 之间接一个陶瓷谐振器 (Resonator) 来取代石英振荡器用来得到频率产生，但是需要分别在 OSC1 和 OSC2 外接一个电容器 (如果振荡频率小于 1MHz)。

WDT 看门狗振荡器是一个芯片内部自由振荡的 RC 振荡器。不需连接外部元件。甚至当系统进入省电模式时，系统时钟会停止，但是 WDT 振荡器仍然以大约 $72 \mu s/5V$ 的周期工作着。由掩模选项，可以将 WDT 关闭来达到节省电源的目的。

看门狗定时器

WDT 的时钟源是由专用的 RC 振荡器 (WDT 振荡器) 或指令时钟 (系统时钟 4 分频) 来提供，可在 ROM 掩模时选择。WDT 是用来防止程序的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 可以被掩模选项禁止。如 WDT 被禁止，所有与 WDT 有关的执行都等同一个空操作。



如果 WDT 时钟源为内部 WDT 振荡器 (RC 振荡周期为 $72 \mu s/5V$)，则此频率经过 32768~65536 分频，得到 $2.3s@5V \sim 4.7s@5V$ 的正常溢出周期信号，其溢出时间会因为温度、 V_{DD} 以及芯片参数的变化而变化。如果 WDT 振荡器除能，WDT 时钟来源则为指令时钟，工作方式与 WDT 振荡器相似，只是在 HALT 状态时，WDT 会停止计数而失去保护功能。只能由外部逻辑来重新启动系统。因此若单片机工作在干扰很大的环境中，强烈建议使用片内的 RC 振荡器 (WDT OSC)，因为 HALT 模式会使系统时钟停止运作。

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 $\overline{RES}$ 端)、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“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 CLR WDT1 和 CLR WDT2 被选择的情况下 (即 CLR WDT 次数为 2)，那么要交替执行二条指令才可连续清除 WDT，否则，WDT 会由于溢出而使系统复位。

暂停模式—HALT

暂停模式是由 HALT 指令来实现的，有如下功能：

- 关闭系统振荡器，但 WDT 振荡器依然工作，(如果 WDT 振荡器或实时时钟被选择)。
- RAM 及寄存器的内容保持不变。
- WDT 被清除并重新计数 (如果 WDT 的时钟是来自 WDT 振荡器或实时时钟振荡器)。
- 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 溢出均可使系统脱离暂停状态。外部复位能使系统初始化，而 WDT 溢出能执行热复位。测试 TO 和 PD 状态后，系统复位的原因就可以确定了。PD 标志位是由系统上电复位和执行 CLR WDT 指令而被清除，而它的置位是由于执行了 HALT 指令。如 WDT 产生溢出，使 TO 标志位置位，还能产生唤醒，使得程序计数器 PC 和堆栈指针 SP 复位。其他都保持原状态。

PA 端口的唤醒和中断的方式作为正常运行的继续，PA 口的每一位都可以通过掩模选项来单独设定为对系统的唤醒。如果唤醒是来自于 I/O 口的信号变化，程序会继续执行下一条命令。如果唤醒是来自中断的话，则会产生二种情况：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。

当进入“HALT”状态以前某个中断请求位被置位，那么系统不能用这个中断来唤醒。当唤醒事件发生时，要花 $1024t_{sys}$ (系统时钟周期) 后，系统重新正常运行。这就是说，在唤醒后被插入了一个等待时

间。但如果系统时钟源来自 RC 振荡器，将继续运行。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。

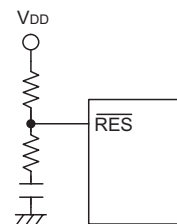
为了减小功耗，在进入 HALT 模式之前必须要小心处理 I/O 端口状态。

复位

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 期间由 $\overline{\text{RES}}$ 脚产生复位。
- 正常运行时，WDT 溢出复位。

在 HALT 期间 WDT 溢出与其它的复位操作情况不同，因为它可执行“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，极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。



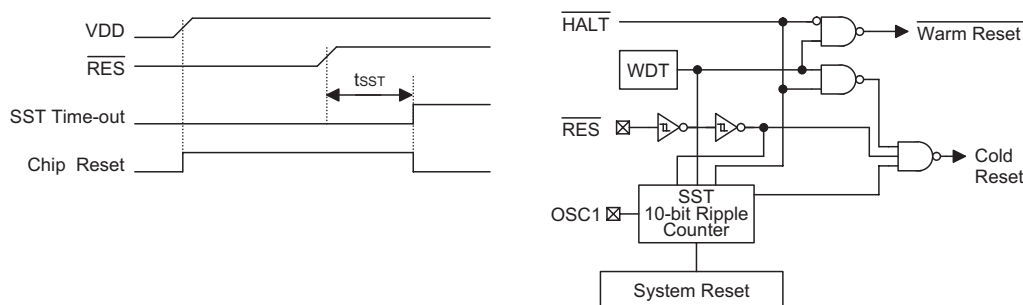
TO	PD	复位条件
0	0	上电时复位 $\overline{\text{RES}}$
u	u	正常动作时 $\overline{\text{RES}}$ 复位
0	1	$\overline{\text{RES}}$ 唤醒 HALT
1	u	正常动作时 WDT 定时溢出
1	1	WDT 唤醒 HALT

注：“u”表示未变。

为了保证系统振荡器起振并稳定运行，当系统上电或从 HALT 状态被唤醒时 SST (System Start-up Timer) 提供额外延迟了 1024 个系统时钟周期。当系统上电或从 HALT 状态被唤醒，SST 延迟被自动加上。在系统上电期间会加上一个额外的 SST 延迟。而从 HALT 状态唤醒只允许 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
WDT, RTC, 时基	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



特殊功能寄存状态概述表

寄存器	上电复位	正常运行期间		HALT 状态	
		WDT 溢出	RES 端复位	RES 端复位	WDT 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	-000 0000	-000 0000	-000 0000	-000 0000	-uuu 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	---- uuuu
PBC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PD	---- ---1	---- ---1	---- ---1	---- ---1	---- ---u
PDC	---- ---1	---- ---1	---- ---1	---- ---1	---- ---u
PWM	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ADRL ADRH	x--- ----	u--- ----	u--- ----	u--- ----	u--- ----
ADRH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ADCR	0100 0000	0100 0000	0100 0000	0100 0000	uuuu uuuu
ACSR	1--- --00	1--- --00	1--- --00	1--- --00	u--- --uu

注： 1. “*” 表示“热复位”。
2. “u” 表示不变化。
3. “x” 表示不确定。

定时/计数器

HT46C47 提供一个 8 位可编程的向上计数的计数器（TMR）。定时/计数器的时钟来源可以是系统时钟或是外部的时钟源。

如果用做内部定时器方式，则只有一个时钟来源，即系统时钟。外部时钟输入允许用户去计算外部事件，测量时间长度或脉宽、或产生一个精确的时基信号。

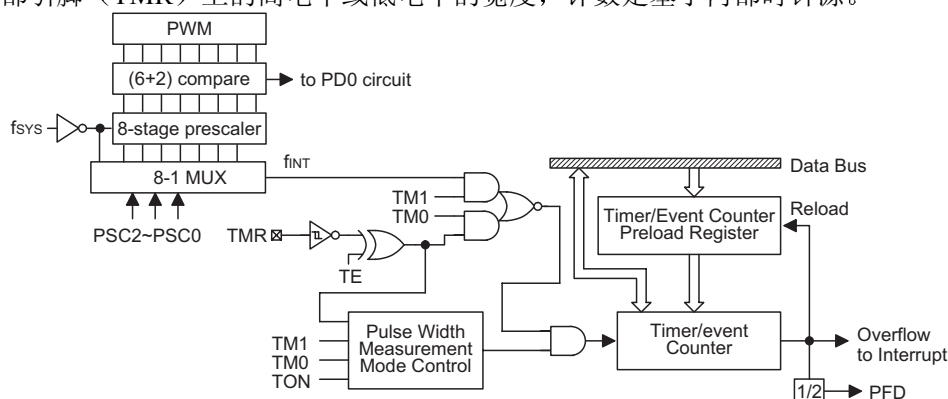
该定时/计数器产生 PFD 信号，时钟来源可使用外部时钟或内部时钟，PFD 频率可由等式 $f_{INT}/[2 \times (256-N)]$ 计算。

总共有两个与定时/计数器有关的寄存器，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。TMR0 是定时/计数器控制寄存器，其定义某些选项。

名称	位	功能
PSC0~PSC2	0~2	定义预分频器级数, PSC2, PSC1, PSC0= 000: $f_{INT}=f_{SYS}$ 001: $f_{INT}=f_{SYS}/2$ 010: $f_{INT}=f_{SYS}/4$ 011: $f_{INT}=f_{SYS}/8$ 100: $f_{INT}=f_{SYS}/16$ 101: $f_{INT}=f_{SYS}/32$ 110: $f_{INT}=f_{SYS}/64$ 111: $f_{INT}=f_{SYS}/128$
TE	3	定义定时/计数器 TMR 作用沿 (0=上升沿作用, 1=下降沿作用)
TON	4	允许/禁止定时器计数 (0=禁止, 1=允许)
—	5	未定义, 读取值为“0”
TM0 TM1	6 7	定义操作方式 (TN2,TN1,TN0) 01=计数器模式 (外部时钟) 10=定时器模式 (系统时钟) 11=脉冲宽度测量模式 (外部时钟) 00=未定义

TMRC 寄存器

TM0 和 TM1 位定义操作模式。外部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, 时钟来源为内部时钟源 f_{INT} 。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, 计数是基于内部时钟源。



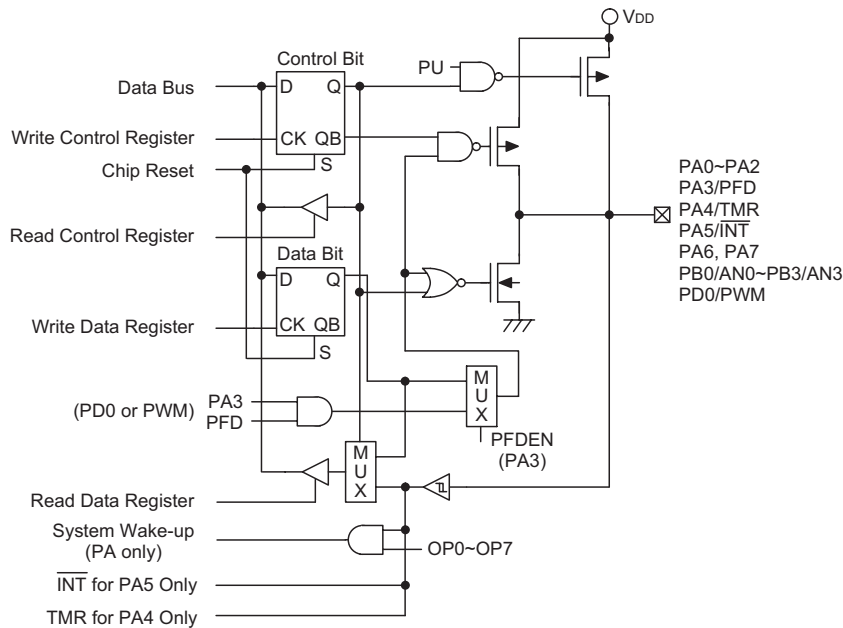
在事件计数或定时器模式中, 定时/计数器会从当前定时/计数器中的数值开始计数, 到 FFH 结束。如果产生溢出, 计数器会从定时/计数器预置寄存器重新装载计数值并且同时产生相应的中断请求标志 (TF; INTC 第 5 位)。

在脉冲宽度测量模式时, 其 TON 和 TE 位皆为 1 时, 如果引脚 TMR 接收到一个上升沿信号 (如 TE 值为 0, 则为下降沿信号) 时, 计数器会开始计数直至 TMR 引脚回到原来的电平为止, 并且会将 TON 清零。计数器停止计数, 测量的结果则仍然保留在定时/计数器之中。换句话说, 脉冲宽度测量模式只能测量一个脉冲。只要 TON 位又被置位, 则当引脚 TMR 接到跳变脉冲, 测量周期会再次执行下去。在这个脉冲宽度测量模式中, 定时/计数器并不会根据逻辑电平来计数, 其根据的标准为信号的跳变缘。一旦发生计数器溢出, 计数器会从定时/计数器加载寄存器重新装入, 同时还会发出中断请求, 这个情况和事件计数模式和定时器模式一样。要启动计数运作, 只要将定时器启动 ON 位 (TON; TMRC 的第 4 位) 被置成为 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/计数器的溢出是唤醒的信号之一。不管任何模式, 若写 0 到 ETI 位即可禁止定时/计数器中断服务。

在定时/计数器为关闭的状态下, 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。但是若定时/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, 定时/计数器继续原来的操作, 直到定时/计数器发生计数溢出为止。读取定时/计数器数据 (读 TMR) 时, 要停止计数以防出错。而停止计数会导致计数错误, 所以程序员必须加以仔细考虑。强烈建议首先装载一个指定的值到 TMR 寄存器, 然后启动定时/计数器作正常的运作。因为 TMR 的初始值是不确定的。

TMRC 的第 0 位~第 2 位用来对内部定时的时钟源进行预分频，其定义如表所示。定时/计数器的溢出信号可作为 PFD 输出。

输入/输出口



HT46C47 包含 13 位的双向输入/输出口，标记为 PA、PB 和 PD，其分别对应 RAM 的[12H]，[14H] 和[18H]。所有端口都可以用做输入或输出。作为输入时，所有端口不具有锁存功能，即输入数据必须在 MOV A, [m] (m=12H,14H 或 18H) 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，直到输出锁存器被改写。

每个输入/输出口都有一个控制寄存器（PAC，PBC，PDC），用来控制输入/输出的设置。使用控制寄存器，可对 CMOS 输出或斯密特触发输入在软件下动态地进行改变。要设置为输入功能，相应的控制寄存器必须写“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自这个引脚（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—修改—写”指令中发生。

对于输出功能，只能设置为 CMOS 输出。这些控制寄存器是对应于内存的 13H，15H 和 19H 地址。

芯片复位后，这些输入/输出口都会是高电平或浮空状态（可以由掩模选项确定）。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 18H）。

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），然后将结果写回这个锁存器或累加器。

PA 的每一个口都具有唤醒系统的能力。PB 端口的高 4 位和 PD 端口的高 7 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。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。

PA3 口和 PFD 共用一个引脚，如果选择 PFD 功能，则 PA3 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。一旦选择 PFD 功能，PFD 的输出信号只受 PA3 数据寄存器控制。向 PA3 数据寄存器写入“1”，则允许 PFD 输出；向 PA3 数据寄存器写入“0”，则 PA3 输出为“0”。PA3 的 I/O 功能如下所示：

I/O 模式	I/P (正常)	O/P (正常)	I/P (PFD)	O/P (PFD)
PA3	逻辑输入	逻辑输出	逻辑输入	PFD (定时/计数器开启)

注：PFD 的输出频率是定时/计数器溢出频率的 1/2

PA5、PA4 口分别和 INT、TMR 共用引脚。

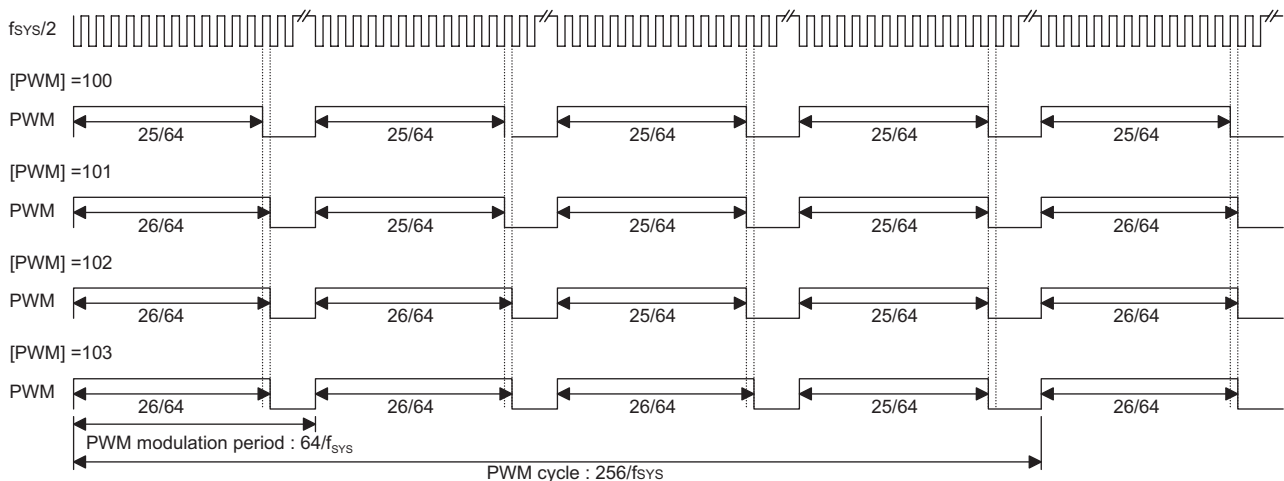
PB 口可以用做 A/D 转换输入，A/D 转换功能将在下面说明。PD0 和 PWM 共用引脚。如果选择 PWM 功能，则 PD0 口会有 PWM 信号输出（PD0 为输出模式）。PD0 的 I/O 功能如下所示：

I/O 模式	I/P (正常)	O/P (正常)	I/P (PWM)	O/P (PWM)
PD0	逻辑输入	逻辑输出	逻辑输入	PWM

为了避免在浮空状态下功耗太大，建议将未用的或没有连接到外部的 I/O 口由软件指令设置成输出引脚。

PWM

HT46C47 有 1 个通道 (6+2) 位的 PWM 输出，与 PD0 共用引脚。PWM 通道用一个数据寄存器来控制 PWM 输出 (1AH)。PWM 计数器的时钟来源为系统时钟 (f_{SYS})。PWM 寄存器是一个 8 位的寄存器。PWM 的输出波形如图所示。一旦 PD0 选择为 PWM 输出，并且 PD0 为输出模式 (PDC.0=“0”)，则向 PD0 寄存器写“1”能够产生 PWM 输出，向 PD0 寄存器写“0”会使 PD0 输出保持为“0”。



一个 PWM 周期被分为四个调制周期 (调制周期 0~调制周期 3)，每个调制周期有 64 个 PWM 输入时钟。在 (6+2) 位的 PWM 功能中，PWM 寄存器被分为 2 个部分。第一部分是 DC，由 PWM.7~PWM.2 控制；第二部分是 AC，由 PWM.1~PWM.0 控制。

在 PWM 周期中，每个调制周期的占空比见下表：

参数	AC (0~3)	占空比
调制周期 i (i=0~3)	i<AC	$\frac{DC+1}{64}$
	i≥AC	$\frac{DC}{64}$

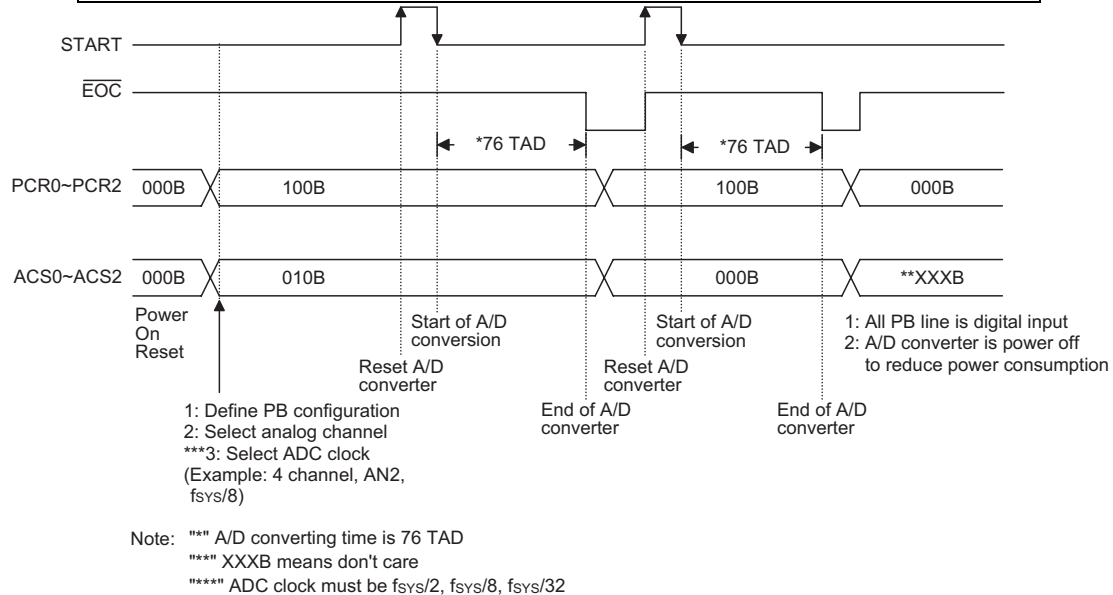
PWM 的调制频率、周期频率、占空比的关系总结如下：

PWM 调制频率	PWM 周期频率	PWM 占空比
$f_{SYS}/64$	$f_{SYS}/256$	[PWM]/256

A/D 转换

HT46C47 有 4 个通道、9 位解析度 (精确到 8 位) 的 A/D 转换器。其参考电压为 V_{DD} 。与 A/D 转换有关的寄存器有 4 个，ADRL (20H)、ADRH (21H)、ADCR (22H) 和 ACSR (23H)。ADRH 和 ADRL 是 A/D 转换的高位和低位寄存器，是只读寄存器。当完成一次 A/D 转换后，ADRH 和 ADRL 用来存放 A/D 转换的结果。ADCR 是 A/D 转换控制寄存器，用来定义 A/D 转换通道、模拟通道、A/D 转换开始控制和完成标志。如果要进行 A/D 转换，要先定义好 PB 口的设置，选择转换的模拟通道，然后给 START 控制位一个上升沿信号和一个下降沿信号 (0→1→0)。完成 A/D 转换后，EOC 位会被清除，并且产生 A/D 转换中断 (如果 A/D 转换允许)。ACSR 是 A/D 时钟控制寄存器，用来选择 A/D 的时钟来源。

符号 (ADCR)	位	功能
ACS0 ACS1 ACS2	0 1 2	ACS2, ACS1, ACS0: A/D 通道选择 0,0,0: AN0 0,0,1: AN1 0,1,0: AN2 0,1,1: AN3 1,x,x: 未定义, 不可用
PCR0 PCR1 PCR2	3 4 5	PCR2, PCR1, PCR0: PB3~PB0 定义 0,0,0: PB3,PB2,PB1,PB0 0,0,1: PB3,PB2,PB1,AN0 0,1,0: PB3,PB2,AN1,AN0 0,1,1: PB3,AN2,AN1,AN0 1,x,x: AN3,AN2,AN1,AN0
$\overline{EOC}$	6	A/D 转换结束标志 (0: A/D 转换结束)
START	7	A/D 转换起始控制位 0→1→0: 开始 1→0: A/D 转换复位, $\overline{EOC}$ 设置为 “1”



A/D 转换控制寄存器用来控制 A/D 转换。ADCR 的第 2 位~第 0 位用来选择模拟信号输入通道，总共有 4 个通道可以选择。ADCR 的第 5 位~第 3 位用来设置 PB 的工作模式，PB 可以作为模拟信号的输入通道，或是数字输入输出，由这 3 位来决定。如果 PB 选择为模拟输入，则其输入/输出功能和上拉电阻将无效。 $\overline{EOC}$ 位 (ADCR 的第 6 位) 是 A/D 转换结束标志位。通过检测这个标志位可以知道 A/D 转换是否结束。ADCR 的起始位 START 用来开启 A/D 转换，给 START 位一个上升沿信号和一个下降沿信号可以开始 A/D 转换。为了确保 A/D 转换顺利完成，START 位应保持为 “0”，直到 $\overline{EOC}$ 位变为 “0” (A/D 转换完成信号)。

	位	功能
ADCS0 ADCS1	0 1	ADCS1, ADCS0: 选择 A/D 时钟源 0,0: f _{sys} /2 0,1: f _{sys} /8 1,0: f _{sys} /32 1,1: 未定义, 不可用
—	2~6	未定义, 读取值为 “0”
TEST	7	只作为内部测试用

ACSR 的第 7 位是用来内部检测的，不提供给用户使用。ACSR 的第 1 位和第 0 位用来选择 A/D 转换

的时钟来源。

当 A/D 转换完成时，A/D 中断标志被置位。当 START 标志由“0”置为“1”时， $\overline{EOC}$ 也置为“1”。

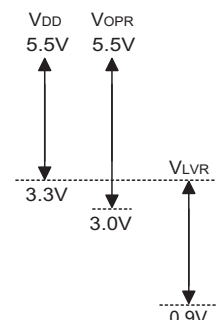
寄存器	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRL	D0	—	—	—	—	—	—	—
ADRH	D8	D7	D6	D5	D4	D3	D2	D1

低电压复位—LVR

为了监控器件的工作电压，单片机提供低电压复位电路。如果器件的工作电压在 0.9V~3.3V 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。

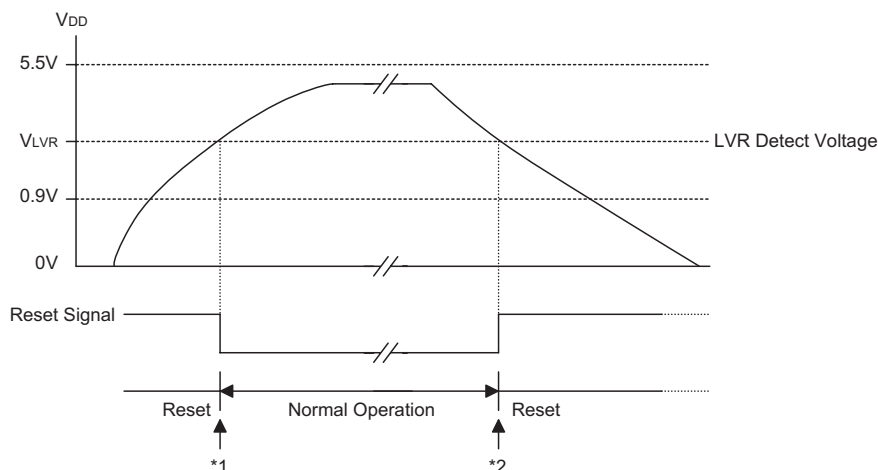
LVR 具有下列功能说明：

- 低电压（0.9V~3.3V）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 通过与信号的“或”的功能来执行系统复位。



附注：VOPR 是在系统时钟为 4MHz 时，使得芯片正常运行的电压域值

VDD 与 VLVR 之间的关系如下所示：



注意：*1： 要保证系统振荡器起振并稳定运行，在系统进入正常运行以前，SST 提供额外的 1024 个系统时钟周期的延迟。

*2： 因为低电压状态必须保持 1ms 以上，因此进入复位模式就要有 1ms 的延迟。

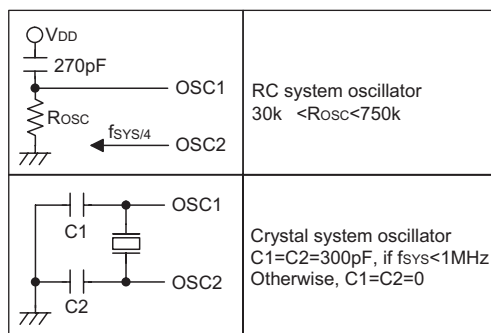
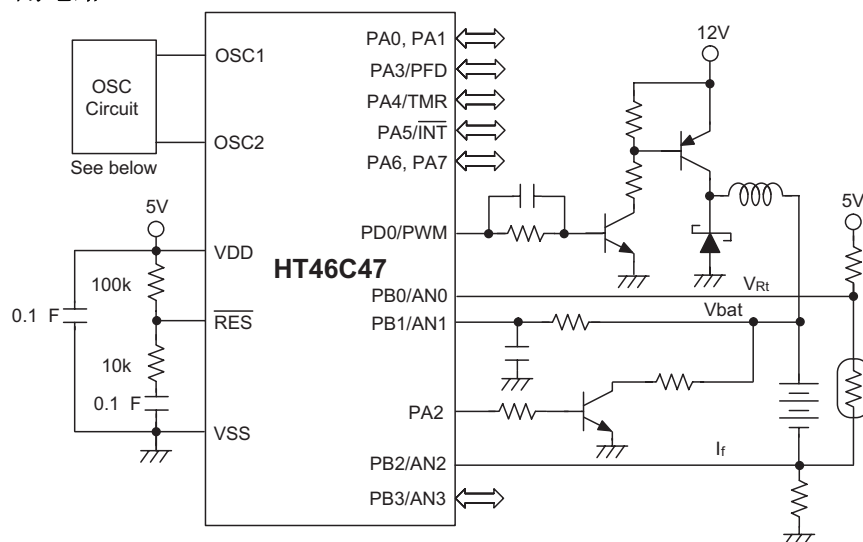
掩模选项

下表示出了这种单片机的各种类型的掩模选项。为了满足系统功能，所有的掩模选项必须正确定义。

编号	选项
1	WDT 时钟源: WDTOSC/ f_{TID}
2	WDT 允许/禁止: 允许/禁止
3	CLR WDT 指令: 1 或 2 条指令
4	系统时钟源: RC 或晶体振荡
5	上拉电阻 (PA、PB、PD): 有或无上拉电阻
6	PWM 允许/禁止
7	PA0~PA7 唤醒: 禁止/允许
8	PFD 允许/禁止
9	低电压复位选择: 允许或禁止 LVR 功能

应用电路

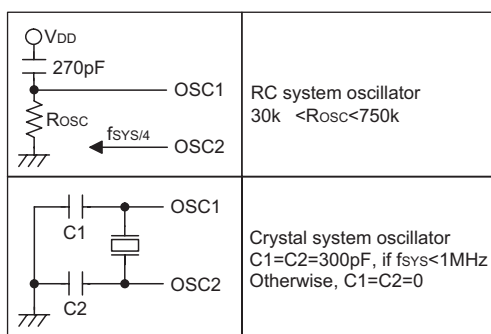
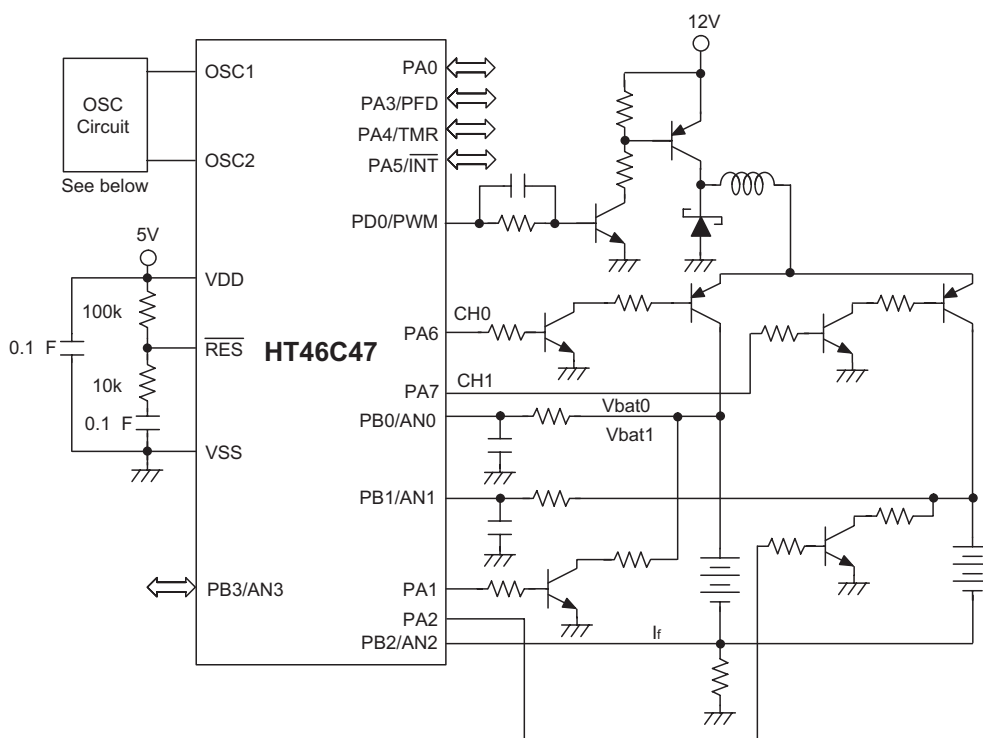
1 节电池的充电器应用电路



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 RES to high.

Vbat: 3.6V (NiCd, NiMH) or 4.1V (Li⁺)

2 节电池的充电器应用电路



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 RES to high.
 Vbat: 3.6V (NiCd, NiMH) or 4.1V (Li⁺)

注：本文翻译至 May 22, 2002 的 46C47.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>.