内置 EEPROM 增强 I/O 型 Flash 单片机

# HT68F20
# HT68F30/HT68FU30
# HT68F40/HT68FU40
# HT68F50/HT68FU50
# HT68F60/HT68FU60

版本：V2.30　日期：2024-08-16

www.holtek.com

# 目 录

**特性 ..... 8**
CPU 特性 ..... 8
周边特性 ..... 8
**概述 ..... 9**
**选型表 ..... 9**
**方框图 ..... 10**
**引脚图 ..... 11**
**引脚说明 ..... 15**
**极限参数 ..... 21**
**直流电气特性 ..... 21**
**交流电气特性 ..... 23**
**比较器电气特性 ..... 24**
**上电复位特性 ..... 24**
**系统结构 ..... 25**
时序和流水线结构 ..... 25
程序计数器 ..... 26
堆栈 ..... 26
算术逻辑单元 — ALU ..... 27
**Flash 程序存储器 ..... 28**
结构 ..... 28
特殊向量 ..... 28
查表 ..... 29
查表范例 ..... 29
在线编程 ..... 30
**数据存储器 ..... 31**
结构 ..... 31
**特殊功能寄存器 ..... 37**
间接寻址寄存器 – IAR0，IAR1 ..... 37
间接寻址指针 – MP0，MP1 ..... 37
存储区指针 – BP ..... 38
累加器 – ACC ..... 39
程序计数器低字节寄存器 – PCL ..... 39
表格寄存器 – TBLP，TBHP，TBLH ..... 39
状态寄存器 – STATUS ..... 39
**EEPROM 数据寄存器 ..... 41**
EEPROM 数据寄存器结构 ..... 41
从 EEPROM 中读取数据 ..... 44
写数据到 EEPROM ..... 44
写保护 ..... 44

EEPROM 中断 ..... 44
编程注意事项 ..... 44

**振荡器 ..... 46**
振荡器概述 ..... 46
系统时钟配置 ..... 46
外部晶体 / 陶瓷振荡器 – HXT ..... 47
外部 RC 振荡器 – ERC ..... 47
内部 RC 振荡器 – HIRC ..... 48
外部 32.768kHz 晶体振荡器 – LXT ..... 48
LXT 振荡器低功耗功能 ..... 49
内部 32kHz 振荡器 – LIRC ..... 49
辅助振荡器 ..... 49

**工作模式和系统时钟 ..... 49**
系统时钟 ..... 49
系统工作模式 ..... 51
控制寄存器 ..... 52
快速唤醒 ..... 53
工作模式切换和唤醒 ..... 54
正常模式切换到低速模式 ..... 54
低速模式切换到正常模式 ..... 55
进入休眠模式 0 ..... 55
进入休眠模式 1 ..... 55
进入空闲模式 0 ..... 55
进入空闲模式 1 ..... 56
静态电流的注意事项 ..... 57
唤醒 ..... 57
编程注意事项 ..... 58

**看门狗定时器 ..... 58**
看门狗定时器时钟源 ..... 58
看门狗定时器控制寄存器 ..... 59
看门狗定时器操作 ..... 59

**复位和初始化 ..... 60**
复位功能 ..... 61
复位初始状态 ..... 63

**输入 / 输出端口 ..... 73**
上拉电阻 ..... 75
PA 口唤醒 ..... 77
输入 / 输出端口控制寄存器 ..... 78
引脚重置功能 ..... 80
引脚重置寄存器 ..... 80
输入 / 输出引脚结构 ..... 85
编程注意事项 ..... 85

**定时器模块 – TM ..... 86**
简介 ..... 86
TM 操作 ..... 86
TM 时钟源 ..... 87
TM 中断 ..... 87
TM 外部引脚 ..... 87
TM 输入 / 输出引脚控制寄存器 ..... 87
编程注意事项 ..... 96
**简易型 TM ..... 97**
简易型 TM 操作 ..... 97
简易型 TM 寄存器介绍 ..... 98
简易型 TM 工作模式 ..... 101
比较匹配输出模式 ..... 101
定时 / 计数器模式 ..... 104
PWM 输出模式 ..... 104
**标准型 TM – STM ..... 107**
标准型 TM 操作 ..... 107
标准型 TM 寄存器介绍 ..... 107
标准型 TM 工作模式 ..... 114
比较匹配输出模式 ..... 114
定时 / 计数器模式 ..... 116
PWM 输出模式 ..... 116
单脉冲模式 ..... 119
捕捉输入模式 ..... 121
**增强型 TM – ETM ..... 122**
增强型 TM 操作 ..... 122
增强型 TM 寄存器介绍 ..... 123
增强型 TM 工作模式 ..... 128
比较匹配输出模式 ..... 128
定时 / 计数器模式 ..... 132
PWM 输出模式 ..... 132
单脉冲输出模式 ..... 138
捕捉输入模式 ..... 139
**比较器 ..... 141**
比较器操作 ..... 141
比较器寄存器 ..... 142
比较器中断 ..... 144
编程注意事项 ..... 144
**串行接口模块 – SIM ..... 144**
SPI 接口 ..... 144
SPI 接口操作 ..... 144
SPI 寄存器 ..... 146
SPI 通信 ..... 148

I²C 接口 ..... 150
I²C 接口操作 ..... 151
I²C 寄存器 ..... 152
I²C 总线通信 ..... 154
I²C 总线起始信号 ..... 155
从机地址 ..... 155
I²C 总线读 / 写信号 ..... 156
I²C 总线从机地址确认信号 ..... 156
I²C 总线数据和确认信号 ..... 156
**外围时钟输出 ..... 157**
外围时钟操作 ..... 157
**中断 ..... 158**
中断寄存器 ..... 158
中断操作 ..... 169
外部中断 ..... 173
比较器中断 ..... 173
多功能中断 ..... 173
时基中断 ..... 173
串行接口模块中断 ..... 174
外部设备中断 ..... 175
EEPROM 中断 ..... 175
LVD 中断 ..... 175
TM 中断 ..... 175
中断唤醒功能 ..... 176
编程注意事项 ..... 176
**暂停模式和唤醒 ..... 176**
进入空闲或休眠模式 ..... 176
静态电流的注意事项 ..... 177
唤醒 ..... 177
低电压检测 – LVD ..... 177
LVD 寄存器 ..... 177
LVD 操作 ..... 178
**带 SCOM 功能的 LCD ..... 179**
LCD 操作 ..... 179
LCD 偏压控制 ..... 179
**配置选项 ..... 181**
**应用电路 ..... 182**
**UART 模块串行接口 ..... 183**
UART 模块特性 ..... 183
UART 模块概述 ..... 183
UART 模块方框图 ..... 183
引脚图 ..... 184
UART 模块引脚说明 ..... 186

UART 模块直流电气特性 ..... 186
UART 模块交流电气特性 ..... 187
UART 模块功能描述 ..... 188
UART 模块内部信号 ..... 188
UART 模块 SPI 接口 ..... 188
SPI 时序 ..... 189
UART 模块外部引脚接口 ..... 189
UART 数据传输方案 ..... 190
UART 命令 ..... 190
UART 状态和控制寄存器 ..... 190
波特率发生器 ..... 195
UART 模块的设置与控制 ..... 197
UART 模块中断结构 ..... 201
UART 模块暂停和唤醒 ..... 202
UART 功能的使用 ..... 202
UART 模块应用电路 ..... 204

**指令集 ..... 205**
简介 ..... 205
指令周期 ..... 205
数据的传送 ..... 205
算术运算 ..... 205
逻辑和移位运算 ..... 205
分支和控制转换 ..... 206
位运算 ..... 206
查表运算 ..... 206
其它运算 ..... 206

**指令集概要 ..... 207**
惯例 ..... 207
指令定义 ..... 209

**封装信息 ..... 219**
16-pin DIP (300mil) 外形尺寸 ..... 220
16-pin NSOP (150mil) 外形尺寸 ..... 223
16-pin SSOP (150mil) 外形尺寸 ..... 224
20-pin DIP (300mil) 外形尺寸 ..... 225
20-pin SOP (300mil) 外形尺寸 ..... 227
20-pin SSOP (150mil) 外形尺寸 ..... 228
24-pin SKDIP (300mil) 外形尺寸 ..... 229
24-pin SOP (300mil) 外形尺寸 ..... 232
24-pin SSOP (150mil) 外形尺寸 ..... 233
28-pin SKDIP (300mil) 外形尺寸 ..... 234
28-pin SOP (300mil) 外形尺寸 ..... 235
28-pin SSOP (150mil) 外形尺寸 ..... 236
SAW Type 32-pin QFN (5mm×5mm×0.75mm) 外形尺寸 ..... 237
SAW Type 40-pin QFN (6mm×6mm×0.75mm) 外形尺寸 ..... 238

44-pin LQFP (10mm×10mm) (FP 2.0mm) 外形尺寸 ..... 239
48-pin SSOP (300mil) 外形尺寸 ..... 240
SAW Type 48-pin QFN (7mm×7mm×0.85mm) 外形尺寸 ..... 241
48-pin LQFP (7mm×7mm) 外形尺寸 ..... 242

**注意，规格中提到的 HT68F20/HT68F30/HT68F50/HT68F60/HT68FU30/HT68FU40/HT68FU50/HT68FU60 产品已终止，目前不再可用。**

# 特性

## CPU 特性

- 工作电压：
  – $f_{SYS}$=8MHz：2.2V ～ 5.5V
  – $f_{SYS}$=12MHz：2.7V ～ 5.5V
  – $f_{SYS}$=20MHz：4.5V ～ 5.5V
- $V_{DD}$=5V，系统时钟为 20MHz 时，指令周期为 0.2μs
- 提供暂停和唤醒功能，以降低功耗
- 五种振荡模式：
  – 外部晶振 -- HXT
  – 外部 32.768kHz 晶振 -- LXT
  – 外部 RC -- ERC
  – 内部 RC -- HIRC
  – 内部 32kHz RC -- LIRC
- 多种工作模式：正常、低速、空闲和休眠
- 内部集成 4MHz，8MHz 和 12MHz 振荡器，无需外接元件
- 所有指令都可在 1 或 2 个指令周期内完成
- 查表指令
- 63 条指令
- 多达 12 层堆栈
- 位操作指令

## 周边特性

- Flash 程序存储：1K×14 ~ 12K×16
- RAM 数据存储：64×8 ~ 576×8
- EEPROM 存储器：32×8 ~ 256×8
- 看门狗定时器功能
- 多达 50 个双向 I/O 口
- 4 个软件控制 SCOM 口 1/2 bias LCD 驱动
- 多个引脚与外部中断口共用
- 多个定时器模块用于时间测量、捕捉输入、比较匹配输出、PWM 输出及单脉冲输出
- 串行接口模块 -- SIM，用于 SPI 或 $I^2C$ 通信
- 双比较器功能
- 双时基功能，可提供固定时间的中断信号
- 低电压复位功能
- 低电压检测功能
- 可选外设 -- UART 模块，可用于全双工异步通信
- 多种封装类型

- Flash 程序存储器烧录可达 100,000 次
- Flash 程序存储器数据可保存 10 年以上
- EEPROM 数据存储器烧录可达 1,000,000 次
- EEPROM 数据存储器数据可保存 10 年以上

## 概述

HT68FXX 系列单片机是一款 I/O 型具有 8 位高性能精简指令集的 Flash 单片机。该系列单片机具有一系列功能和特性，其 Flash 存储器可多次编程的特性给用户提供了较大的方便。存储器方面，还包含了一个 RAM 数据存储器和一个可用于存储序号、校准数据等非易失性数据的 EEPROM 存储器。

这款单片机带有多个使用灵活的定时器模块，可提供定时功能、脉冲产生功能及 PWM 产生功能。在模拟特性方面，还包含两个比较器功能。内建完整的 SPI 和 $I^2C$ 功能，为设计者提供了一个易与外部硬件通信的接口。内部看门狗定时器、低电压复位和低电压检测等内部保护特性，外加优秀的抗干扰和 ESD 保护性能，确保单片机在恶劣的电磁干扰环境下可靠地运行。

这款单片机提供了丰富的 HXT、LXT、ERC、HIRC 和 LIRC 振荡器功能选项，且内建完整的系统振荡器，无需外围元器件。其在不同工作模式之间动态切换的能力，为用户提供了一个优化单片机操作和减少功耗的手段。

HT68FUx0 系列单片机内含 UART 模块，它可以支持诸如单片机之间的数据通信网络，低成本 PC 和外部设备间的数据连接，便携式和电池供电设备间的通信等。

外加时基功能、I/O 使用灵活等其它特性，使这款单片机可以广泛应用于各种产品中，例如电子测量仪器、环境监控、手持式测量工具、家庭应用、电子控制工具、马达控制等方面。

## 选型表

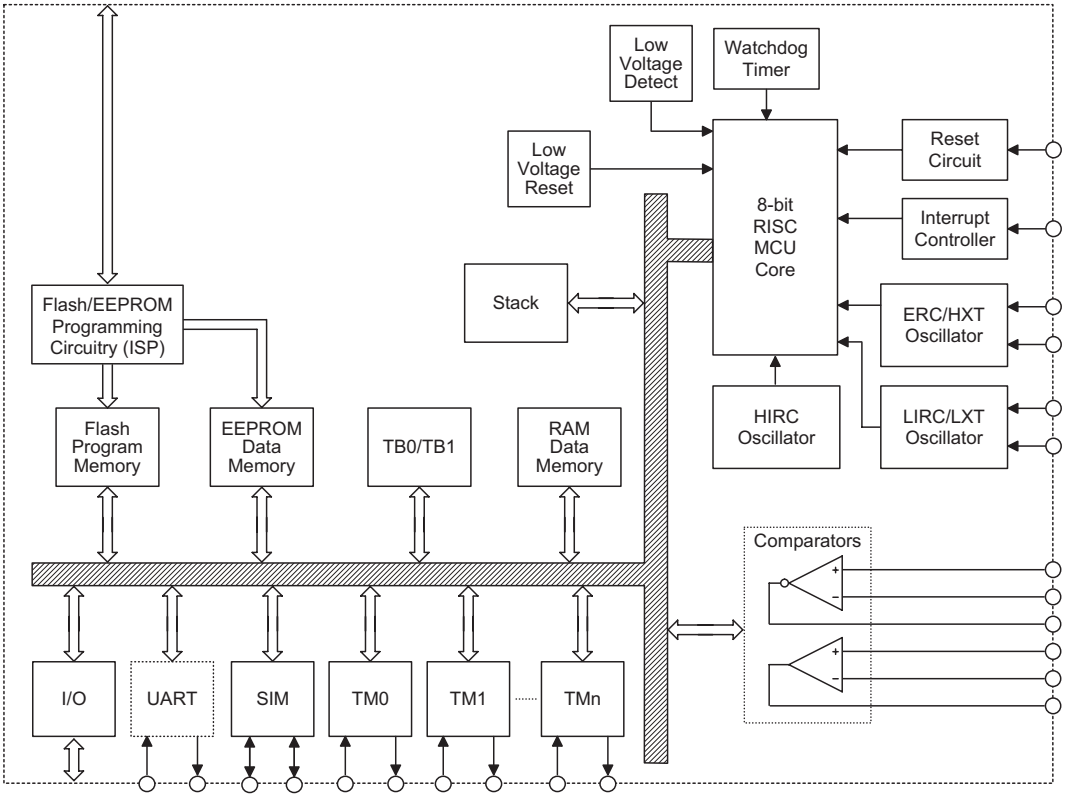
对此系列的芯片而言，大多数的特性参数都是一样的。主要差异在于程序存储器的容量，I/O 数量，TM 特性，堆栈层数和封装类型。下表列出了各单片机的主要特性。

| 型号 | $V_{DD}$ | ROM | RAM | EEPROM | I/O | 外部中断 | TM 模块 | 接口（SPI/$I^2C$） | UART | 堆栈 | 封装形式 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HT68F20* | 2.2V~5.5V | 1K×14 | 64×8 | 32×8 | 18 | 2 | 10-bit CTM×1<br>10-bit STM×1 | √ | — | 4 | 16DIP/NSOP/SSOP<br>20DIP/SOP/SSOP |
| HT68F30 | 2.2V~5.5V | 2K×14 | 96×8 | 64×8 | 22 | 2 | 10-bit CTM×1<br>10-bit ETM×1 | √ | — | 4 | 16DIP/NSOP/SSOP<br>20DIP/SOP/SSOP<br>24SKDIP/SOP/SSOP |
| HT68FU30 | | | | | 14 | | | | √ | | 24SKDIP/SOP |
| HT68F40 | 2.2V~5.5V | 4K×15 | 192×8 | 128×8 | 42 | 2 | 10-bit CTM×1<br>10-bit ETM×1<br>16-bit STM×1 | √ | — | 8 | 24/28SKDIP/SOP/SSOP<br>44LQFP, 32/40QFN<br>48SSOP/QFN |
| HT68FU40 | | | | | 34 | | | | √ | | 40QFN, 44LQFP,<br>48SSOP/QFN |

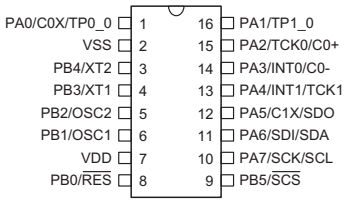
| 型号 | $V_{DD}$ | ROM | RAM | EEPROM | I/O | 外部中断 | TM 模块 | 接口（SPI/I²C） | UART | 堆栈 | 封装形式 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HT68F50 | 2.2V~5.5V | 8K×16 | 384×8 | 256×8 | 42 | 2 | 10-bit CTM×2<br>10-bit ETM×1<br>16-bit STM×1 | √ | — | 8 | 28SKDIP/SOP/SSOP<br>44LQFP, 40QFN<br>48SSOP/QFN |
| HT68FU50 | | | | | 34 | | | | √ | | 44LQFP, 48QFN |
| HT68F60 | 2.2V~5.5V | 12K×16 | 576×8 | 256×8 | 50 | 4 | 10-bit CTM×2<br>10-bit ETM×1<br>16-bit STM×1 | √ | — | 12 | 40QFN, 44LQFP<br>48SSOP/LQFP/QFN |
| HT68FU60 | | | | | 42 | | | | √ | | 40QFN, 44LQFP<br>48LQFP/QFN |

注：对于有不止一种封装形式的芯片，选型表针对较大的封装的情况。
HT68FU30、HT68FU40、HT68FU50 和 HT68FU60 中含有 UART 模块，所有相关 UART 模块的详细信息请参考其相关章节。

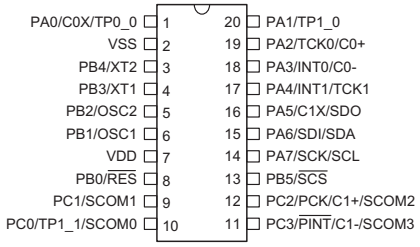
## 方框图



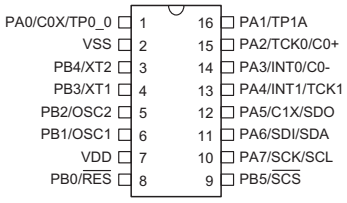
# 引脚图



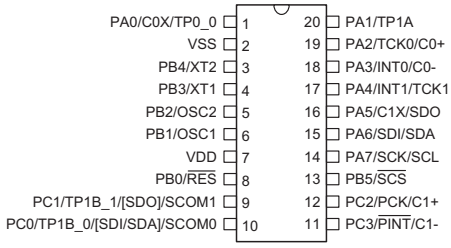

**HT68F20**
**16 DIP-A/NSOP-A/SSOP-A**



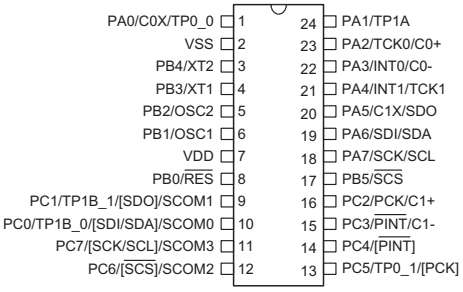

**HT68F20**
**20 DIP-A/SOP-A/SSOP-A**



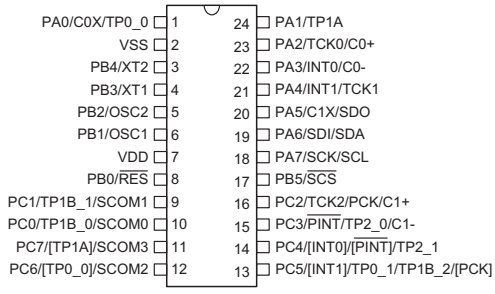

**HT68F30**
**16 DIP-A/NSOP-A/SSOP-A**



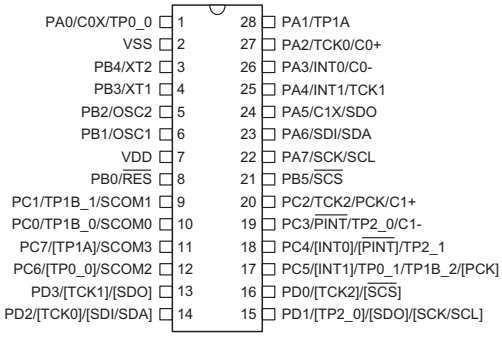

**HT68F30**
**20 DIP-A/SOP-A/SSOP-A**




**HT68F30**
**24 SKDIP-A/SOP-A/SSOP-A**

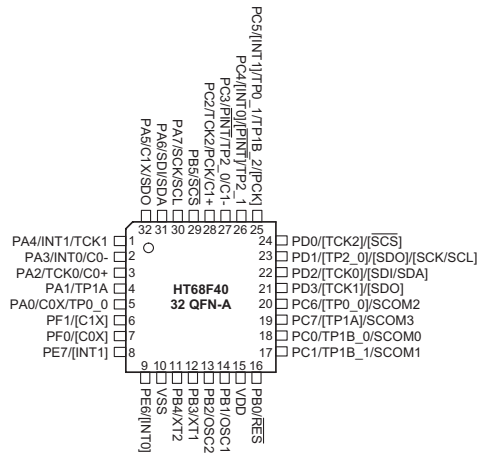


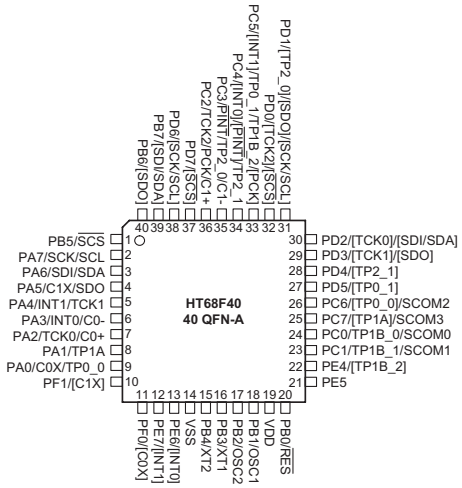

**HT68F40**
**24 SKDIP-A/SOP-A/SSOP-A**

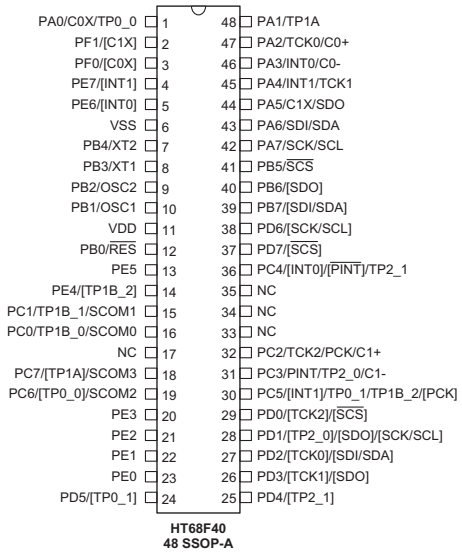


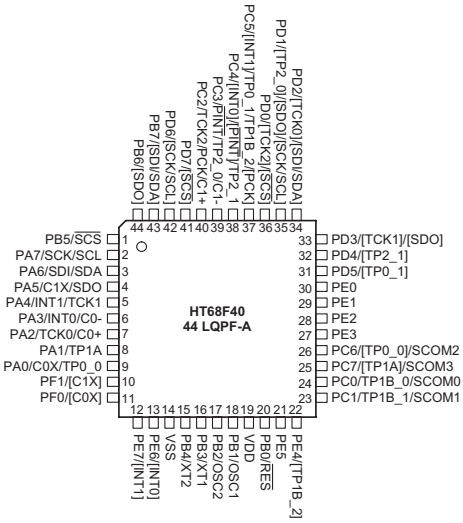

**HT68F40**
**28 SKDIP-A/SOP-A/SSOP-A**

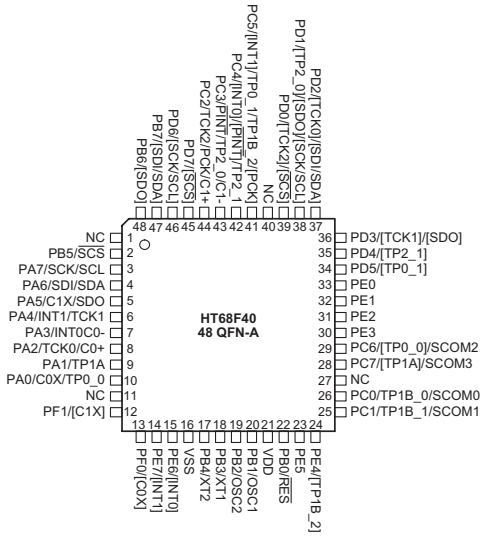
注：1、括号内的引脚为可编程改变位置的引脚。
2、若共用脚同时有多种输出，“/”号右侧的引脚名具有更高的优先级。



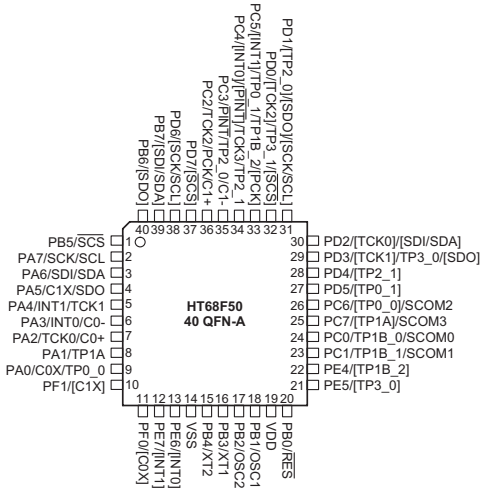



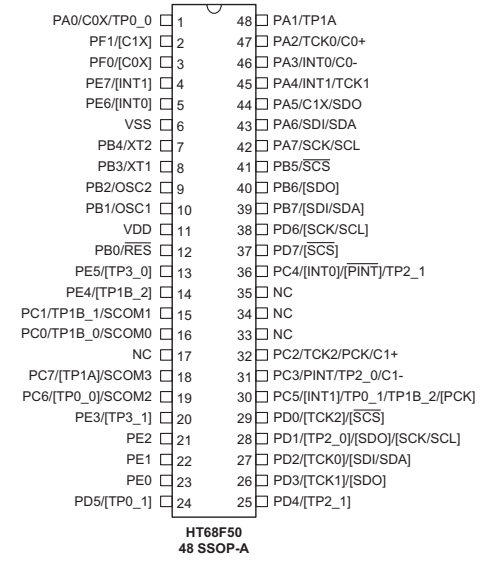



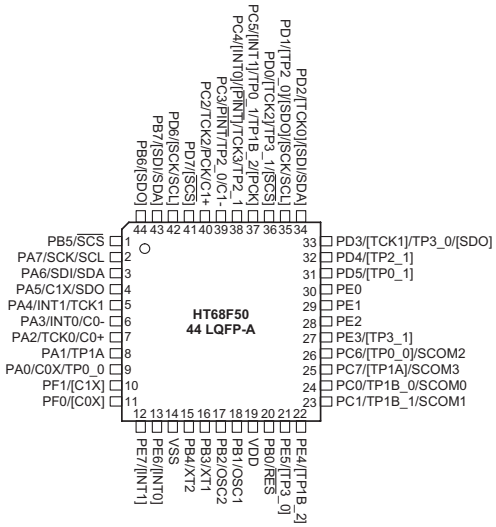

注：1、括号内的引脚为可编程改变位置的引脚。

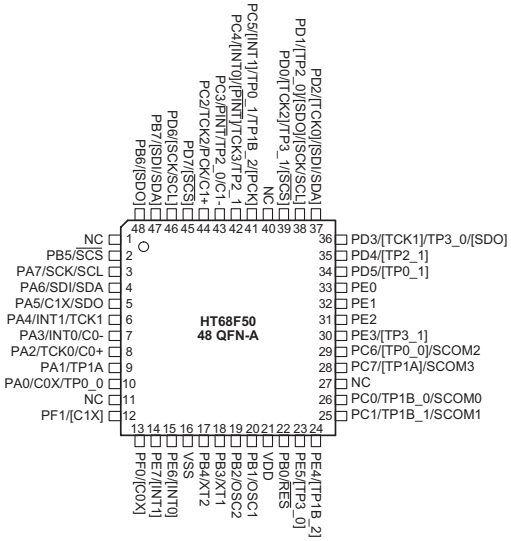
2、若共用脚同时有多种输出，“/”号右侧的引脚名具有更高的优先级。



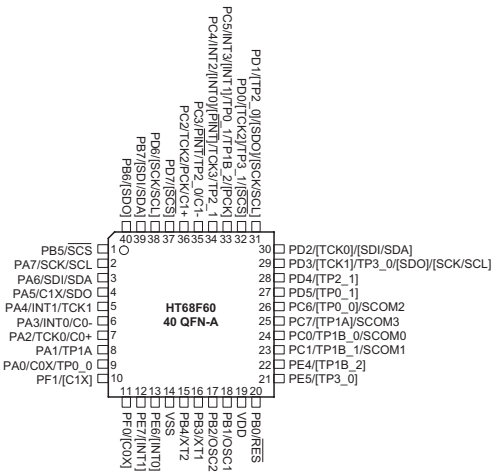



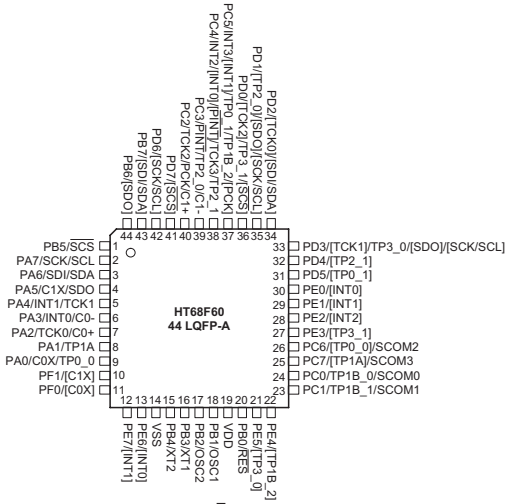



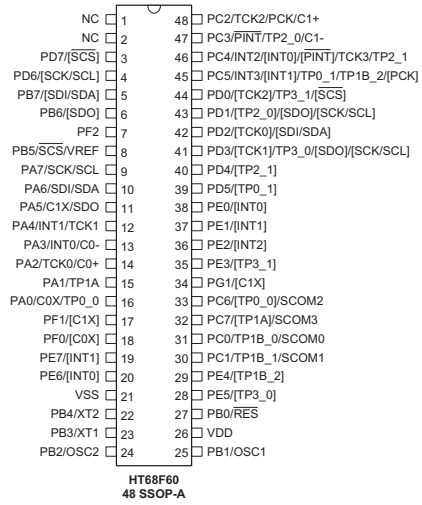



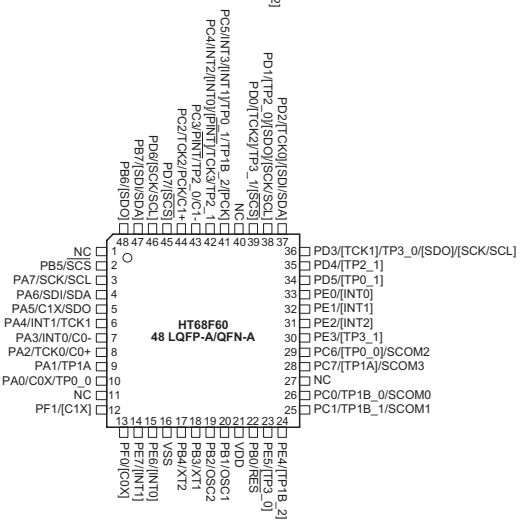

注：1、括号内的引脚为可编程改变位置的引脚。

2、若共用脚同时有多种输出，“/”号右侧的引脚名具有更高的优先级。




注：1、括号内的引脚为可编程改变位置的引脚。

2、若共用脚同时有多种输出，“/”号右侧的引脚名具有更高的优先级。

## 引脚说明

除了电源引脚外，该系列单片机的所有引脚都以它们的端口名称进行标注，例如 PA.0、PA.1 等，用于描述这些引脚的数字输入 / 输出功能。然而，这些引脚也与其它功能共用，如串行接口引脚等。每个引脚的功能如下表所述，而引脚配置的详细内容见规格书其它章节。

**HT68F20**

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| PA0~PA7 | 端口 A | PAWU<br>PAPU | ST | CMOS | — |
| PB0~PB5 | 端口 B | PBPU | ST | CMOS | — |
| PC0~PC3 | 端口 C | PCPU | ST | CMOS | — |
| C0-，C1- | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA3，PC3 |
| C0+，C1+ | 比较器 0，1 输入 | | AN | — | PA2，PC2 |
| C0X，C1X | 比较器 0，1 输出 | | — | CMOS | PA0，PA5 |
| TCK0，TCK1 | TM0，TM1 输入 | — | ST | — | PA2，PA4 |
| TP0_0 | TM0 输入 / 输出 | TMPC0 | ST | CMOS | PA0 |
| TP1_0，TP1_1 | TM1 输入 / 输出 | TMPC0 | ST | CMOS | PA1，PC0 |
| INT0，INT1 | 外部中断 0，1 | — | ST | — | PA3，PA4 |
| $\overline{\text{PINT}}$ | 外围中断 | — | ST | — | PC3 |
| PCK | 外围时钟输出 | — | — | CMOS | PC2 |
| SDI | SPI 数据输入 | — | ST | — | PA6 |
| SDO | SPI 数据输出 | — | — | CMOS | PA5 |
| $\overline{\text{SCS}}$ | SPI 从机选择 | — | ST | CMOS | PB5 |
| SCK | SPI 串行时钟 | — | ST | CMOS | PA7 |
| SCL | $I^2C$ 时钟 | — | ST | NMOS | PA7 |
| SDA | $I^2C$ 数据 | — | ST | NMOS | PA6 |
| SCOM0~SCOM3 | SCOM0~SCOM3 | SCOMC | — | SCOM | PC0，PC1，PC2，PC3 |
| OSC1 | HXT/ERC 脚 | CO | HXT | — | PB1 |
| OSC2 | HXT 脚 | CO | — | HXT | PB2 |
| XT1 | LXT 脚 | CO | LXT | — | PB3 |
| XT2 | LXT 脚 | CO | — | LXT | PB4 |
| $\overline{\text{RES}}$ | 复位输入 | CO | ST | — | PB0 |
| VDD | 电源电压 | — | PWR | — | — |
| VSS | 地 | — | PWR | — | — |

注：I/T：输入类型　　O/T：输出类型　　OP：通过配置选项（CO）或寄存器选项来设置
PWR：电源　　CO：配置选项　　ST：斯密特触发输入
CMOS：CMOS 输出　　NMOS：NMOS 输出　　SCOM：软件控制的 LCD COM；
AN：模拟输入脚　　HXT：高速晶体振荡器　　LXT：低速晶体振荡器
此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

**HT68F30**

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| PA0~PA7 | 端口 A | PAWU<br>PAPU | ST | CMOS | — |
| PB0~PB5 | 端口 B | PBPU | ST | CMOS | — |
| PC0~PC7 | 端口 C | PCPU | ST | CMOS | — |
| C0-，C1- | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA3，PC3 |
| C0+，C1+ | 比较器 0，1 输入 | | AN | — | PA2，PC2 |
| C0X，C1X | 比较器 0，1 输出 | | — | CMOS | PA0，PA5 |
| TCK0，TCK1 | TM0，TM1 输入 | — | ST | — | PA2，PA4 |
| TP0_0，TP0_1 | TM0 输入 / 输出 | TMPC0 | ST | CMOS | PA0，PC5 |
| TP1A | TM1 输入 / 输出 | TMPC0 | ST | CMOS | PA1 |
| TP1B_0，TP1B_1 | TM1 输入 / 输出 | TMPC0 | ST | CMOS | PC0，PC1 |
| INT0，INT1 | 外部中断 0，1 | — | ST | — | PA3，PA4 |
| $\overline{\text{PINT}}$ | 外围中断 | PRM0 | ST | — | PC3 或 PC4 |
| PCK | 外围时钟输出 | PRM0 | — | CMOS | PC2 或 PC5 |
| SDI | SPI 数据输入 | PRM0 | ST | — | PA6 或 PC0 |
| SDO | SPI 数据输出 | PRM0 | — | CMOS | PA5 或 PC1 |
| $\overline{\text{SCS}}$ | SPI 从机选择 | PRM0 | ST | CMOS | PB5 或 PC6 |
| SCK | SPI 串行时钟 | PRM0 | ST | CMOS | PA7 或 PC7 |
| SCL | $I^2C$ 时钟 | PRM0 | ST | NMOS | PA7 或 PC7 |
| SDA | $I^2C$ 数据 | PRM0 | ST | NMOS | PA6 或 PC0 |
| SCOM0~SCOM3 | SCOM0~SCOM3 | SCOMC | — | SCOM | PC0，PC1，PC6，PC7 |
| OSC1 | HXT/ERC 脚 | CO | HXT | — | PB1 |
| OSC2 | HXT 脚 | CO | — | HXT | PB2 |
| XT1 | LXT 脚 | CO | LXT | — | PB3 |
| XT2 | LXT 脚 | CO | — | LXT | PB4 |
| $\overline{\text{RES}}$ | 复位输入 | CO | ST | — | PB0 |
| VDD | 电源电压 | — | PWR | — | — |
| VSS | 地 | — | PWR | — | — |

注：I/T：输入类型　　O/T：输出类型　　OP：通过配置选项（CO）或寄存器选项来设置
PWR：电源　　CO：配置选项　　ST：斯密特触发输入
CMOS：CMOS 输出　　NMOS：NMOS 输出　　SCOM：软件控制的 LCD COM
AN：模拟输入脚　　HXT：高速晶体振荡器　　LXT：低速晶体振荡器
此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

**HT68F40**

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| PA0~PA7 | 端口 A | PAWU<br>PAPU | ST | CMOS | — |
| PB0~PB7 | 端口 B | PBPU | ST | CMOS | — |
| PC0~PC7 | 端口 C | PCPU | ST | CMOS | — |
| PD0~PD7 | 端口 D | PDPU | ST | CMOS | — |
| PE0~PE7 | 端口 E | PEPU | ST | CMOS | — |
| PF0~PF1 | 端口 F | PFPU | ST | CMOS | — |
| C0-，C1- | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA3，PC3 |
| C0+，C1+ | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA2，PC2 |
| C0X,C1X | 比较器 0，1 输出 | CP0C<br>CP1C<br>PRM0 | — | CMOS | PA0，PA5 或 PF0，PF1 |
| TCK0~TCK2 | TM0~TM2 输入 | PRM1 | ST | — | PA2，PA4，PC2 或<br>PD2，PD3，PD0 |
| TP0_0，TP0_1 | TM0 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA0，PC5 或 PC6，PD5 |
| TP1A | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA1 或 PC7 |
| TP1B_0~TP1B_2 | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PC0，PC1，<br>PC5 或 -，-，PE4 |
| TP2_0，TP2_1 | TM2 输入 / 输出 | TMPC1<br>PRM2 | ST | CMOS | PC3，PC4 或 PD1，PD4 |
| INT0,INT1 | 外部中断 0，1 | PRM1 | ST | — | PA3，PA4 或 PC4，<br>PC5 或 PE6，PE7 |
| $\overline{\text{PINT}}$ | 外围中断 | PRM0 | ST | — | PC3 或 PC4 |
| PCK | 外围时钟输出 | PRM0 | — | CMOS | PC2 或 PC5 |
| SDI | SPI 数据输入 | PRM0 | ST | — | PA6 或 PD2 或 PB7 |
| SDO | SPI 数据输出 | PRM0 | — | CMOS | PA5 或 PD3 或 PB6 |
| $\overline{\text{SCS}}$ | SPI 从机选择 | PRM0 | ST | CMOS | PB5 或 PD0 或 PD7 |
| SCK | SPI 串行时钟 | PRM0 | ST | CMOS | PA7 或 PD1 或 PD6 |
| SCL | $I^2C$ 时钟 | PRM0 | ST | NMOS | PA7 或 PD1 或 PD6 |
| SDA | $I^2C$ 数据 | PRM0 | ST | NMOS | PA6 或 PD2 或 PB7 |
| SCOM0~SCOM3 | SCOM0~SCOM3 | SCOMC | — | SCOM | PC0，PC1，PC6，PC7 |
| OSC1 | HXT/ERC 脚 | CO | HXT | — | PB1 |
| OSC2 | HXT 脚 | CO | — | HXT | PB2 |
| XT1 | LXT 脚 | CO | LXT | — | PB3 |

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| XT2 | LXT 脚 | CO | — | LXT | PB4 |
| $\overline{RES}$ | 复位输入 | CO | ST | — | PB0 |
| VDD | 电源电压 | — | PWR | — | — |
| VSS | 地 | — | PWR | — | — |

注：I/T：输入类型　　O/T：输出类型　　OP：通过配置选项（CO）或寄存器选项来设置
PWR：电源　　CO：配置选项　　ST：斯密特触发输入
CMOS：CMOS 输出　　NMOS：NMOS 输出　　SCOM：软件控制的 LCD COM
AN：模拟输入脚　　HXT：高速晶体振荡器　　LXT：低速晶体振荡器
此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

**HT68F50**

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| PA0~PA7 | 端口 A | PAWU<br>PAPU | ST | CMOS | — |
| PB0~PB7 | 端口 B | PBPU | ST | CMOS | — |
| PC0~PC7 | 端口 C | PCPU | ST | CMOS | — |
| PD0~PD7 | 端口 D | PDPU | ST | CMOS | — |
| PE0~PE7 | 端口 E | PEPU | ST | CMOS | — |
| PF0~PF1 | 端口 F | PFPU | ST | CMOS | — |
| C0-，C1- | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA3，PC3 |
| C0+，C1+ | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA2，PC2 |
| C0X，C1X | 比较器 0，1 输出 | CP0C<br>CP1C<br>PRM0 | — | CMOS | PA0，PA5 或 PF0，PF1 |
| TCK0~TCK3 | TM0~TM3 输入 | PRM1 | ST | — | PA2，PA4，PC2，PC4 或 PD2，PD3，PD0，- |
| TP0_0，TP0_1 | TM0 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA0，PC5 或 PC6，PD5 |
| TP1A | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA1 或 PC7 |
| TP1B_0~TP1B_2 | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PC0，PC1，PC5 或 -，-，PE4 |
| TP2_0，TP2_1 | TM2 输入 / 输出 | TMPC1<br>PRM2 | ST | CMOS | PC3，PC4 或 PD1，PD4 |
| TP3_0，TP3_1 | TM3 输入 / 输出 | TMPC1<br>PRM2 | ST | CMOS | PD3，PD0 或 PE5，PE3 |
| INT0，INT1 | 外部中断 0，1 | PRM1 | ST | — | PA3，PA4 或 PC4，PC5 或 PE6，PE7 |
| $\overline{PINT}$ | 外围中断 | PRM0 | ST | — | PC3 或 PC4 |
| PCK | 外围时钟输出 | PRM0 | — | CMOS | PC2 或 PC5 |

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| SDI | SPI 数据输入 | PRM0 | ST | — | PA6 或 PD2 或 PB7 |
| SDO | SPI 数据输出 | PRM0 | — | CMOS | PA5 或 PD3 或 PB6 |
| $\overline{SCS}$ | SPI 从机选择 | PRM0 | ST | CMOS | PB5 或 PD0 或 PD7 |
| SCK | SPI 串行时钟 | PRM0 | ST | CMOS | PA7 或 PD1 或 PD6 |
| SCL | $I^2C$ 时钟 | PRM0 | ST | NMOS | PA7 或 PD1 或 PD6 |
| SDA | $I^2C$ 数据 | PRM0 | ST | NMOS | PA6 或 PD2 或 PB7 |
| SCOM0~SCOM3 | SCOM0~SCOM3 | SCOMC | — | SCOM | PC0，PC1，PC6，PC7 |
| OSC1 | HXT/ERC 脚 | CO | HXT | — | PB1 |
| OSC2 | HXT 脚 | CO | — | HXT | PB2 |
| XT1 | LXT 脚 | CO | LXT | — | PB3 |
| XT2 | LXT 脚 | CO | — | LXT | PB4 |
| $\overline{RES}$ | 复位输入 | CO | ST | — | PB0 |
| VDD | 电源电压 | — | PWR | — | — |
| VSS | 地 | — | PWR | — | — |

注：I/T：输入类型　　O/T：输出类型　　OP：通过配置选项（CO）或寄存器选项来设置
PWR：电源　　CO：配置选项　　ST：斯密特触发输入
CMOS：CMOS 输出　　NMOS：NMOS 输出　　SCOM：软件控制的 LCD COM
AN：模拟输入脚　　HXT：高速晶体振荡器　　LXT：低速晶体振荡器
此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

**HT68F60**

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| PA0~PA7 | 端口 A | PAWU<br>PAPU | ST | CMOS | — |
| PB0~PB7 | 端口 B | PBPU | ST | CMOS | — |
| PC0~PC7 | 端口 C | PCPU | ST | CMOS | — |
| PD0~PD7 | 端口 D | PDPU | ST | CMOS | — |
| PE0~PE7 | 端口 E | PEPU | ST | CMOS | — |
| PF0~PF7 | 端口 F | PFPU | ST | CMOS | — |
| PG0~PG1 | 端口 G | PGPU | ST | CMOS | — |
| C0-，C1- | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA3，PC3 |
| C0+，C1+ | 比较器 0，1 输入 | CP0C<br>CP1C | AN | — | PA2，PC2 |
| C0X，C1X | 比较器 0，1 输出 | CP0C<br>CP1C<br>PRM0 | — | CMOS | PA0，PA5 或 PF0，PF1 或 PG0，PG1 |

| 引脚名称 | 功能 | OP | I/T | O/T | 共用引脚映射 |
|---|---|---|---|---|---|
| TCK0~TCK3 | TM0~TM3 输入 | PRM1 | ST | — | PA2，PA4，PC2，PC4 或 PD2，PD3，PD0，- |
| TP0_0，TP0_1 | TM0 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA0，PC5 或 PC6，PD5 |
| TP1A | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PA1 或 PC7 |
| TP1B_0~TP1B_2 | TM1 输入 / 输出 | TMPC0<br>PRM2 | ST | CMOS | PC0，PC1，PC5 或 -，-，PE4 |
| TP2_0，TP2_1 | TM2 输入 / 输出 | TMPC1<br>PRM2 | ST | CMOS | PC3，PC4 或 PD1，PD4 |
| TP3_0，TP3_1 | TM3 输入 / 输出 | TMPC1<br>PRM2 | ST | CMOS | PD3，PD0 或 PE5，PE3 |
| INT0~INT3 | 外部中断 0~3 | PRM1 | ST | — | PA3，PA4，PC4，PC5 或 PC4，PC5，PE2，- 或 PE0，PE1，-，- 或 PE6，PE7，-，- |
| $\overline{\text{PINT}}$ | 外围中断 | PRM0 | ST | — | PC3 或 PC4 |
| PCK | 外围时钟输出 | PRM0 | — | CMOS | PC2 或 PC5 |
| SDI | SPI 数据输入 | PRM0 | ST | — | PA6 或 PD2 或 PB7 |
| SDO | SPI 数据输出 | PRM0 | — | CMOS | PA5 或 PD3 或 PB6 或 PD1 |
| $\overline{\text{SCS}}$ | SPI 从机选择 | PRM0 | ST | CMOS | PB5 或 PD0 或 PD7 |
| SCK | SPI 串行时钟 | PRM0 | ST | CMOS | PA7 或 PD1 或 PD6 或 PD3 |
| SCL | $I^2C$ 时钟 | PRM0 | ST | NMOS | PA7 或 PD1 或 PD6 或 PD3 |
| SDA | $I^2C$ 数据 | PRM0 | ST | NMOS | PA6 或 PD2 或 PB7 |
| SCOM0~SCOM3 | SCOM0~SCOM3 | SCOMC | — | SCOM | PC0，PC1，PC6，PC7 |
| OSC1 | HXT/ERC 脚 | CO | HXT | — | PB1 |
| OSC2 | HXT 脚 | CO | — | HXT | PB2 |
| XT1 | LXT 脚 | CO | LXT | — | PB3 |
| XT2 | LXT 脚 | CO | — | LXT | PB4 |
| $\overline{\text{RES}}$ | 复位输入 | CO | ST | — | PB0 |
| VDD | 电源电压 | — | PWR | — | — |
| VSS | 地 | — | PWR | — | — |

注：I/T：输入类型　　O/T：输出类型　　OP：通过配置选项（CO）或寄存器选项来设置
PWR：电源　　CO：配置选项　　ST：斯密特触发输入
CMOS：CMOS 输出　　NMOS：NMOS 输出　　SCOM：软件控制的 LCD COM
AN：模拟输入脚　　HXT：高速晶体振荡器　　LXT：低速晶体振荡器
此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

## 极限参数

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

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

## 直流电气特性

Ta= 25℃

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $V_{DD}$ | 工作电压 (HXT,ERC,HIRC) | — | $f_{SYS}$=8MHz | 2.2 | — | 5.5 | V |
| | | | $f_{SYS}$=12MHz | 2.7 | — | 5.5 | V |
| | | | $f_{SYS}$=20MHz | 4.5 | — | 5.5 | V |
| $I_{DD1}$ | 工作电流，正常模式，$f_{SYS}=f_H$ (HXT,ERC,HIRC) | 3V | 无负载，$f_{SYS}=f_H$=4MHz, WDT 使能 | — | 0.7 | 1.1 | mA |
| | | 5V | | — | 1.8 | 2.7 | mA |
| | | 3V | 无负载，$f_{SYS}=f_H$=8MHz, WDT 使能 | — | 1.6 | 2.4 | mA |
| | | 5V | | — | 3.3 | 5.0 | mA |
| | | 3V | 无负载，$f_{SYS}=f_H$=12MHz, WDT 使能 | — | 2.2 | 3.3 | mA |
| | | 5V | | — | 5.0 | 7.5 | mA |
| $I_{DD2}$ | 工作电流，正常模式，$f_{SYS}=f_H$ (HXT) | 5V | 无负载，$f_{SYS}=f_H$ =20MHz, WDT 使能 | — | 6.0 | 9.0 | mA |
| $I_{DD3}$ | 工作电流，低速模式，$f_{SYS}=f_L$ (LXT,LIRC) | 3V | 无负载，$f_{SYS}=f_L$，WDT 使能 | — | 10 | 20 | μA |
| | | 5V | | — | 30 | 50 | μA |
| $I_{IDLE0}$ | IDLE0 模式静态电流 (LXT 或 LIRC on) | 3V | 无负载，WDT 使能 | — | 1.5 | 3.0 | μA |
| | | 5V | | — | 3.0 | 6.0 | μA |
| $I_{IDLE1}$ | IDLE1 模式静态电流 (HXT,ERC,HIRC) | 3V | 无负载，WDT 使能 $f_{SYS}$= 12MHz on | — | 0.55 | 0.83 | mA |
| | | 5V | | — | 1.30 | 2.00 | mA |
| $I_{SLEEP0}$ | SLEEP0 模式静态电流 (LXT 或 LIRC off) | 3V | 无负载，WDT 除能 | — | — | 1 | μA |
| | | 5V | | — | — | 2 | μA |

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $I_{SLEEP1}$ | SLEEP1 模式静态电流 (LXT 或 LIRC on) | 3V | 无负载，WDT 使能 | — | 1.5 | 3.0 | μA |
| | | 5V | | — | 2.5 | 5.0 | μA |
| $V_{IL1}$ | 输入 / 输出口或除 $\overline{RES}$ 脚以外的低电平输入电压 | — | — | 0 | — | $0.3V_{DD}$ | V |
| $V_{IH1}$ | 输入 / 输出口或除 $\overline{RES}$ 脚以外的高电平输入电压 | — | — | $0.7V_{DD}$ | — | $V_{DD}$ | V |
| $V_{IL2}$ | 低电平输入电压 ($\overline{RES}$) | — | — | 0 | — | $0.4V_{DD}$ | V |
| $V_{IH2}$ | 高电平输入电压 ($\overline{RES}$) | — | — | $0.9V_{DD}$ | — | $V_{DD}$ | V |
| $V_{LVR}$ | 低电压复位电压 | — | LVR 使能，选择 2.1V | -5% | 2.1 | +5% | V |
| | | | LVR 使能，选择 2.55V | -5% | 2.55 | +5% | V |
| | | | LVR 使能，选择 3.15V | -5% | 3.15 | +5% | V |
| | | | LVR 使能，选择 4.2V | -5% | 4.20 | +5% | V |
| $V_{LVD}$ | 低电压检测电压 | — | LVDEN=1，$V_{LVD}$=2.0V | -5% | 2.00 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=2.2V | -5% | 2.20 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=2.4V | -5% | 2.40 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=2.7V | -5% | 2.70 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=3.0V | -5% | 3.00 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=3.3V | -5% | 3.30 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=3.6V | -5% | 3.60 | +5% | V |
| | | | LVDEN=1，$V_{LVD}$=4.4V | -5% | 4.4 | +5% | V |
| $I_{LV}$ | 使用 LVR 和 LVD 的额外功耗 | — | LVR 使能，LVDEN=0 | — | 60 | 90 | μA |
| | | | LVR 除能，LVDEN=1 | — | 75 | 115 | μA |
| | | | LVR 使能，LVDEN=1 | — | 90 | 135 | μA |
| $V_{OL}$ | 输入 / 输出口输出低电平 | 3V | $I_{OL}$=9mA | — | — | 0.3 | V |
| | | 5V | $I_{OL}$=20mA | — | — | 0.5 | V |
| $V_{OH}$ | 输入 / 输出口输出高电平 | 3V | $I_{OH}$=-3.2mA | 2.7 | — | — | V |
| | | 5V | $I_{OH}$=-7.4mA | 4.5 | — | — | V |
| $R_{PH}$ | 输入 / 输出口上拉电阻 | 3V | — | 20 | 60 | 100 | kW |
| | | 5V | | 10 | 30 | 50 | kW |
| $I_{SCOM}$ | SCOM 工作电流 | 5V | SCOMC,ISEL[1:0]=00 | 17.5 | 25.0 | 32.5 | μA |
| | | | SCOMC,ISEL[1:0]=01 | 35 | 50 | 65 | μA |
| | | | SCOMC,ISEL[1:0]=10 | 70 | 100 | 130 | μA |
| | | | SCOMC,ISEL[1:0]=11 | 140 | 200 | 260 | μA |
| $V_{SCOM}$ | 用于 LCD COM 的 $V_{DD}$/2 电压 | 5V | 无负载 | 0.475 | 0.500 | 0.525 | $V_{DD}$ |

## 交流电气特性

Ta= 25℃

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $f_{CPU}$ | 工作时钟 | — | 2.2V~5.5V | DC | — | 8 | MHz |
| | | | 2.7V~5.5V | DC | — | 12 | MHz |
| | | | 4.5V~5.5V | DC | — | 20 | MHz |
| $f_{SYS}$ | 系统时钟 (HXT) | — | 2.2V~5.5V | 0.4 | — | 8 | MHz |
| | | | 2.7V~5.5V | 0.4 | — | 12 | MHz |
| | | | 4.5V~5.5V | 0.4 | — | 20 | MHz |
| $f_{HIRC}$ | 系统时钟 (HIRC) | 3V/5V | Ta=25℃ | -2% | 4 | +2% | MHz |
| | | 3V/5V | Ta=25℃ | -2% | 8 | +2% | MHz |
| | | 5V | Ta=25℃ | -2% | 12 | +2% | MHz |
| | | 3V/5V | Ta=0~70℃ | -5% | 4 | +5% | MHz |
| | | 3V/5V | Ta=0~70℃ | -4% | 8 | +4% | MHz |
| | | 5V | Ta=0~70℃ | -5% | 12 | +3% | MHz |
| | | 2.2V~3.6V | Ta=0~70℃ | -7% | 4 | +7% | MHz |
| | | 3.0V~5.5V | Ta=0~70℃ | -5% | 4 | +9% | MHz |
| | | 2.2V~3.6V | Ta=0~70℃ | -6% | 8 | +4% | MHz |
| | | 3.0V~5.5V | Ta=0~70℃ | -4% | 8 | +9% | MHz |
| | | 3.0V~5.5V | Ta=0~70℃ | -6% | 12 | +7% | MHz |
| | | 2.2V~3.6V | Ta=-40℃ ~ 85℃ | -12% | 4 | +8% | MHz |
| | | 3.0V~5.5V | Ta=-40℃ ~ 85℃ | -10% | 4 | +9% | MHz |
| | | 2.2V~3.6V | Ta=-40℃ ~ 85℃ | -15% | 8 | +4% | MHz |
| | | 3.0V~5.5V | Ta=-40℃ ~ 85℃ | -8% | 8 | +9% | MHz |
| | | 3.0V~5.5V | Ta=-40℃ ~ 85℃ | -12% | 12 | +7% | MHz |
| $f_{ERC}$ | 系统时钟 (ERC) | 5V | Ta=25℃，R=120kΩ* | -2% | 8 | +2% | MHz |
| | | 5V | Ta=0℃ ~70℃，R=120kΩ* | -5% | 8 | +6% | MHz |
| | | 5V | Ta=-40℃ ~85℃，R=120kΩ* | -7% | 8 | +9% | MHz |
| | | 3.0V~5.5V | Ta=-40℃ ~85℃，R=120kΩ* | -9% | 8 | +10% | MHz |
| | | 2.2V~5.5V | Ta=-40℃ ~85℃，R=120kΩ* | -15% | 8 | +10% | MHz |
| $f_{LXT}$ | 系统时钟 (LXT) | — | — | — | 32.768 | — | kHz |
| $f_{LIRC}$ | 系统时钟 (LIRC) | 5V | Ta=25℃ | -10% | 32 | +10% | kHz |
| | | 2.2V~5.5V | Ta=-40℃ ~ 85℃ | -50% | 32 | +60% | kHz |
| $f_{TIMER}$ | 定时器输入频率 | — | — | — | — | 1 | $f_{SYS}$ |
| $t_{RES}$ | 外部复位低电平脉宽 | — | — | 1 | — | — | μs |
| $t_{INT}$ | 中断脉宽 | — | — | 1 | — | — | $t_{SYS}$ |
| $t_{LVR}$ | 低压复位时间 | — | — | 120 | 240 | 480 | μs |

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $t_{LVD}$ | 低压中断时间 | — | — | 20 | 45 | 90 | μs |
| $t_{LVDS}$ | LVDO 稳定的时间 | — | — | 15 | — | — | μs |
| $t_{BGS}$ | VBG 打开稳定时间 | — | — | 200 | — | — | μs |
| $t_{EERD}$ | EEPROM 读周期 | — | — | — | 45 | 90 | μs |
| $t_{EEWR}$ | EEPROM 写周期 | — | — | — | 2 | 4 | ms |
| $t_{SST}$ | 系统启动时间（从 HALT 中唤醒） | — | $f_{SYS}$=HXT 或 LXT | — | 1024 | — | $t_{SYS}$ |
| | | | $f_{SYS}$=ERC 或 HIRC | — | 15~16 | — | |
| | | | $f_{SYS}$=LIRC | — | 1~2 | — | |

注：1、$t_{SYS}$ =1/ $f_{SYS}$
2、*：表示电阻的公差会影响外部 RC 的频率，建议使用精密度较高的电阻。
3、为了保证 HIRC 振荡器的频率精度，VDD 与 VSS 间连接一个 0.1μF 的去耦电容，并尽可能接近芯片。

## 比较器电气特性

Ta= 25℃

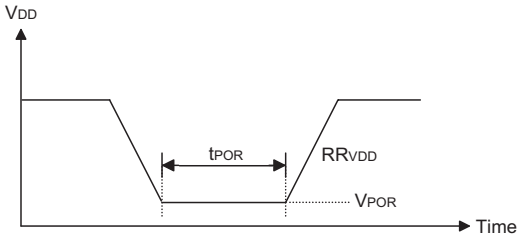
| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $V_{CMP}$ | 比较器工作电压 | — | — | 2.2 | — | 5.5 | V |
| $I_{CMP}$ | 比较器工作电流 | 3V | — | — | 37 | 56 | μA |
| | | 5V | — | — | 130 | 200 | μA |
| $V_{CMPOS}$ | 比较器输入失调电压 | — | — | -10 | — | 10 | mV |
| $V_{HYS}$ | 迟滞宽度 | — | — | 20 | 40 | 60 | mV |
| $V_{CM}$ | 比较器共模电压范围 | — | — | $V_{SS}$ | — | $V_{DD}$-1.4V | V |
| $A_{OL}$ | 比较器开环增益 | — | — | 60 | 80 | — | dB |
| $t_{PD}$ | 比较器响应时间 | — | 100mV 偏置（注） | — | 370 | 560 | ns |

注：测量方式为：当一只输入脚的输入电压为 $V_{CM}$=($V_{DD}$-1.4)/2 时，另一只输入脚的输入电压从 $V_{SS}$ 到 ($V_{CM}$+100mV) 或从 $V_{DD}$ 到 ($V_{CM}$-100mV) 转变。

## 上电复位特性

Ta= 25℃

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

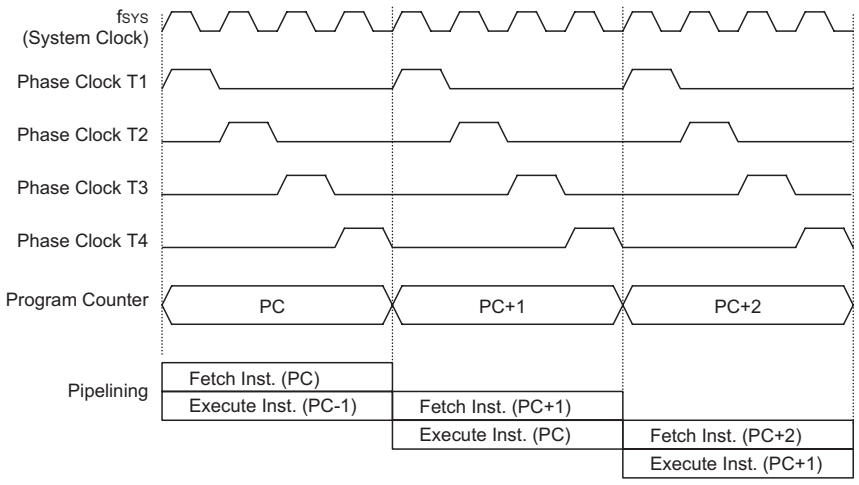



# 系统结构

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

## 时序和流水线结构

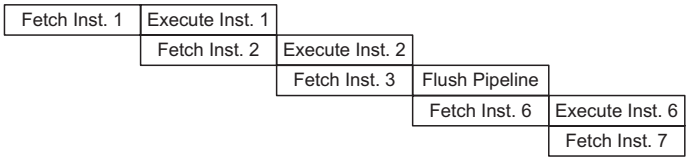
主系统时钟由 HXT，LXT，HIRC，LIRC 或 ERC 振荡器提供，它被细分为 T1~T4 四个内部产生的非重叠时序。在 T1 时间，程序计数器自动加一并抓取一条新的指令。剩下的时间 T2~T4 完成译码和执行功能，因此，一个 T1~T4 时钟周期构成一个指令周期。虽然指令的抓取和执行发生在连续的指令周期，但单片机流水线结构会保证指令在一个指令周期内被有效执行。除非程序计数器的内容被改变，如子程序的调用或跳转，在这种情况下指令将需要多一个指令周期的时间去执行。




系统时序和流水线

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

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




指令捕捉

## 程序计数器

在程序执行期间，程序计数器用来指向下一个要执行的指令地址。除了“JMP”和“CALL”指令需要跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完成以后自动加一。选择不同型号的单片机，程序寄存器的宽度会因程序存储器的容量的不同而不同。只有较低的 8 位，即所谓的程序计数器低字节寄存器 PCL，可以被用户直接读写。

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

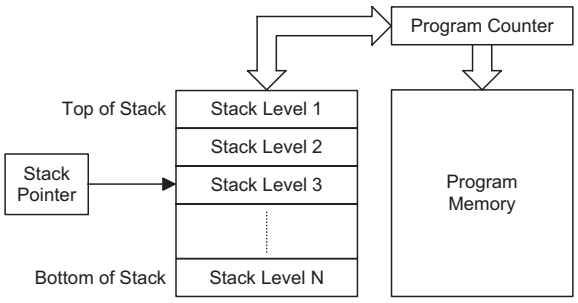
| 单片机型号 | 程序计数器 | |
|---|---|---|
| | 程序计数器高字节 | PCL 寄存器 |
| HT68F20 | PC9，PC8 | PCL7~PCL0 |
| HT68F30 | PC10~PC8 | |
| HT68F40 | PC11~PC8 | |
| HT68F50 | PC12~PC8 | |
| HT68F60 | PC13~PC8 | |

程序计数器

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

## 堆栈

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




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

若堆栈溢出，则首个存入堆栈的程序计数器数据将会丢失。

| 单片机型号 | 堆栈层数 |
|---|---|
| HT68F20/ HT68F30 | 4 |
| HT68F40/ HT68F50 | 8 |
| HT68F60 | 12 |

## 算术逻辑单元 — ALU

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

- 算术运算：ADD，ADDM，ADC，ADCM，SUB，SUBM，SBC，SBCM，DAA
- 逻辑运算：AND，OR，XOR，ANDM，ORM，XORM，CPL，CPLA
- 移位运算：RRA，RR，RRCA，RRC，RLA，RL，RLCA，RLC
- 递增和递减：INCA，INC，DECA，DEC
- 分支判断：JMP，SZ，SZA，SNZ，SIZ，SDZ，SIZA，SDZA，CALL，RET，RETI

# Flash 程序存储器

程序存储器用来存放用户代码即储存程序。程序存储器为 FLASH 类型意味着可以多次重复编程，方便用户使用同一芯片进行程序的修改。使用适当的单片机编程工具，此系列所有单片机提供用户灵活便利的调试方法和项目开发规划及更新。

## 结构

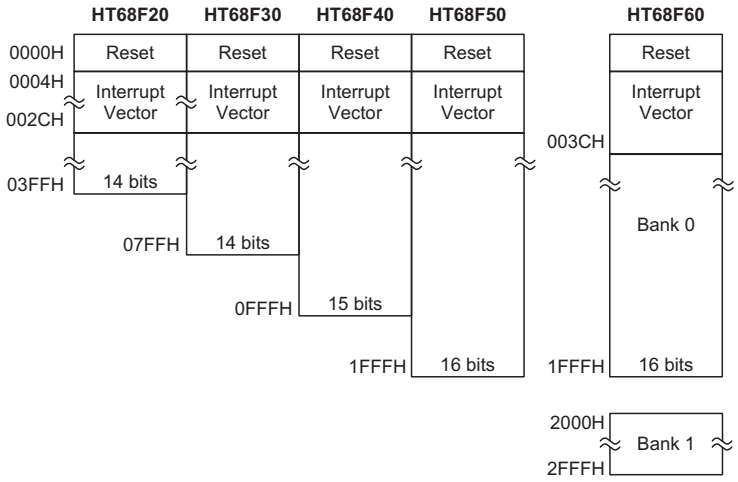
程序存储器的容量为 1K×14 位到 12K×16 位，程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口。数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。

| 单片机型号 | 容量 | Banks |
|---|---|---|
| HT68F20 | 1K×14 | 0 |
| HT68F30 | 2K×14 | 0 |
| HT68F40 | 4K×15 | 0 |
| HT68F50 | 8K×16 | 0 |
| HT68F60 | 12K×16 | 0，1 |

HT68F60 程序存储器分为两个 Bank，Bank 0 和 Bank 1。通过 BP 寄存器的第 5 位选择所需要的 Bank。

## 特殊向量

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



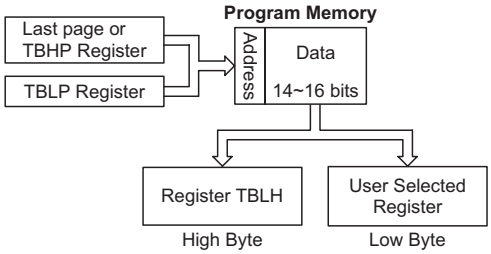

程序存储器结构

## 查表

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

在设定完表格指针后，表格数据可以使用“TABRD [m]”或“TABRDL [m]”指令分别从程序存储器中查表读取。当这些指令执行时，程序存储器中表格数据低字节，将被传送到使用者所指定的数据存储器 [m]，程序存储器中表格数据的高字节，则被传送到 TBLH 特殊寄存器，而高字节中未使用的位将被读取为“0”。

下图是查表中寻址 / 数据流程：




## 查表范例

以下范例说明表格指针和表格数据如何被定义和执行。这个例子使用的表格数据用 ORG 伪指令储存在存储器中。HT68F30 中 ORG 指令的值“700H”指向的地址是 2K 程序存储器中最后一页的起始地址。表格指针的初始值设为 06H，这可保证从数据表格读取的第一笔数据位于程序存储器地址 706H，即最后一页起始地址后的第六个地址。值得注意的是，假如“TABRD [m]”指令被使用，则表格指针指向当前页。在这个例子中，表格数据的高字节等于零，而当“TABRD [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

### 表格读取程序举例

```
tempreg1 db ?       ; temporary register #1
tempreg2 db ?       ; temporary register #2
:
:
mov a,06h           ; initialise low table pointer - note that this address
mov tblp,a          ; is referenced
mov a,07h           ; initialise high table pointer
tbhp,a
:
:
tabrd tempreg1      ; transfers value in table referenced by table pointer
                    ; data at program memory address "706H" transferred to
                    ; tempreg1 and TBLH
dec tblp            ; reduce value of table pointer by one
tabrd tempreg2      ; transfers value in table referenced by table pointer
                    ; data at program memory address "705H" transferred to
                    ; tempreg2 and TBLH in this example the data "1AH" is
                    ; transferred to tempreg1 and data "0FH" to register tempreg2
:
:
org 700h            ; sets initial address of program memory
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
:
:
```

TBLH 寄存器为只读寄存器，不能重新储存，若主程序和中断服务程序都使用表格读取指令，应该注意它的保护。使用表格读取指令，中断服务程序可能会改变 TBLH 的值，若随后在主程序中再次使用这个值，则会发生错误，因此建议避免同时使用表格读取指令。然而在某些情况下，如果同时使用表格读取指令是不可避免的，则在执行任何主程序的表格读取指令前，中断应该先除能，另外要注意的是所有与表格相关的指令，都需要两个指令周期去完成操作。

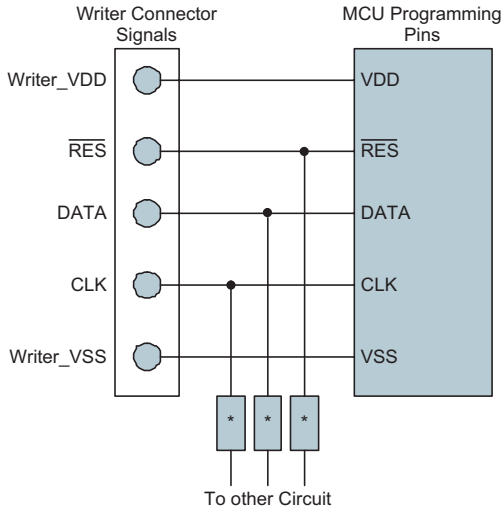
## 在线编程

Flash 型程序存储器提供用户便利地对同一芯片进行程序的更新和修改。另外，HOLTEK 单片机提供 5 线接口的在线编程方式。用户可将进行过编程或未经过编程的单片机芯片连同电路板一起制成，最后阶段进行程序的更新和程序的烧写，在无需去除或重新插入芯片的情况下方便地保持程序为最新版。

| MCU 在线编程引脚名称 | 功能 |
|---|---|
| PA0 | 串行数据输入 / 输出 |
| PA2 | 串行时钟 |
| $\overline{\text{RES}}$ | 复位 |
| VDD | 电源 |
| VSS | 地 |

芯片内部程序存储器和 EEPROM 存储器都可以通过 5 线的接口在线进行编程。其中 PA0 用于数据串行下载或上传、PA2 用于串行时钟、两条用于提供电源，另外一条用于复位。芯片在线烧写的详细使用说明超出此文档的描述范围，将由专门的参考文献提供。

在编程过程中，编程器会将 $\overline{\text{RES}}$ 脚一直拉低以除能单片机工作，并控制 PA0 和 PA2 脚进行数据和时钟编程，用户必须确保这两个引脚没有连接至其它输出脚。




注：* 可能为电阻或电容。若为电阻则其值必须大于 1kΩ，若为电容则其必须小于 1nF。

| 在线编程引脚 | MCU 引脚 |
|---|---|
| $\overline{\text{RES}}$ | PB0 |
| DATA | PA0 |
| CLK | PA2 |

在线编程和 MCU 引脚

# 数据存储器

数据存储器是内容可更改的 8 位 RAM 内部存储器，用来储存临时数据。

## 结构

数据存储器分为两个区，第一部分是特殊功能数据存储器。这些寄存器有固定的地址且与单片机的正确操作密切相关。大多特殊功能寄存器都可在程序控制下直接读取和写入，但有些被加以保护而不对用户开放。

第二部分数据存储器是做一般用途使用，都可在程序控制下进行读取和写入。

总的数据存储器被分为几个区，由所选的单片机决定的。大部分特殊功能数据寄存器均可在所有 Bank 被访问，处于“40H”地址的 EEC 寄存器却只能在 Bank1 中被访问到。切换不同区域可通过设置区域指针 (BP) 实现。所有单片机的数据存储器的起始地址都是“00H”。

| 单片机型号 | 容量 | Bank |
|---|---|---|
| HT68F20 | 64×8 | 0: 60H~7FH<br>1: 60H~7FH |
| HT68F30 | 96×8 | 0: 60H~7FH<br>1: 60H~7FH<br>2: 60H~7FH |
| HT68F40 | 192×8 | 0: 80H~FFH<br>1: 80H~BFH |
| HT68F50 | 384×8 | 0: 80H~FFH<br>1: 80H~FFH<br>2: 80H~FFH |
| HT68F60 | 576×8 | 0: 80H~FFH<br>1: 80H~FFH<br>2: 80H~FFH<br>3: 80H~FFH<br>4: 80H~BFH |

| | Bank 0, 1 |
|---|---|
| 00H | IAR0 |
| 01H | MP0 |
| 02H | IAR1 |
| 03H | MP1 |
| 04H | BP |
| 05H | ACC |
| 06H | PCL |
| 07H | TBLP |
| 08H | TBLH |
| 09H | TBHP |
| 0AH | STATUS |
| 0BH | SMOD |
| 0CH | LVDC |
| 0DH | INTEG |
| 0EH | WDTC |
| 0FH | TBC |
| 10H | INTC0 |
| 11H | INTC1 |
| 12H | INTC2 |
| 13H | Unused |
| 14H | MFI0 |
| 15H | MFI1 |
| 16H | MFI2 |
| 17H | Unused |
| 18H | PAWU |
| 19H | PAPU |
| 1AH | PA |
| 1BH | PAC |
| 1CH | PBPU |
| 1DH | PB |
| 1EH | PBC |
| 1FH | PCPU |
| 20H | PC |
| 21H | PCC |
| 22H | Unused |
| 23H | Unused |
| 24H | Unused |
| 25H | Unused |
| 26H | Unused |
| 27H | Unused |
| 28H | Unused |
| 29H | Unused |
| 2AH | Unused |
| 2BH | Unused |
| 2CH | Unused |
| 2DH | Unused |
| 2EH | Unused |
| 2FH | Unused |

| | Bank 0 | Bank 1 |
|---|---|---|
| 30H | Unused | |
| 31H | Unused | |
| 32H | Unused | |
| 33H | Unused | |
| 34H | CP0C | |
| 35H | CP1C | |
| 36H | SIMC0 | |
| 37H | SIMC1 | |
| 38H | SIMD | |
| 39H | SIMA/SIMC2 | |
| 3AH | TM0C0 | |
| 3BH | TM0C1 | |
| 3CH | TM0DL | |
| 3DH | TM0DH | |
| 3EH | TM0AL | |
| 3FH | TM0AH | |
| 40H | Unused | EEC |
| 41H | EEA | |
| 42H | EED | |
| 43H | TMPC0 | |
| 44H | Unused | |
| 45H | Unused | |
| 46H | Unused | |
| 47H | Unused | |
| 48H | TM1C0 | |
| 49H | TM1C1 | |
| 4AH | Unused | |
| 4BH | TM1DL | |
| 4CH | TM1DH | |
| 4DH | TM1AL | |
| 4EH | TM1AH | |
| 4FH | Unused | |
| 50H | Unused | |
| 51H | Unused | |
| 52H | Unused | |
| 53H | Unused | |
| 54H | Unused | |
| 55H | Unused | |
| 56H | Unused | |
| 57H | Unused | |
| 58H | Unused | |
| 59H | Unused | |
| 5AH | Unused | |
| 5BH | Unused | |
| 5CH | Unused | |
| 5DH | Unused | |
| 5EH | SCOMC | |
| 5FH | Unused | |

**HT68F20 特殊功能数据存储器结构**

| | Bank 0, 1, 2 |
|---|---|
| 00H | IAR0 |
| 01H | MP0 |
| 02H | IAR1 |
| 03H | MP1 |
| 04H | BP |
| 05H | ACC |
| 06H | PCL |
| 07H | TBLP |
| 08H | TBLH |
| 09H | TBHP |
| 0AH | STATUS |
| 0BH | SMOD |
| 0CH | LVDC |
| 0DH | INTEG |
| 0EH | WDTC |
| 0FH | TBC |
| 10H | INTC0 |
| 11H | INTC1 |
| 12H | INTC2 |
| 13H | Unused |
| 14H | MFI0 |
| 15H | MFI1 |
| 16H | MFI2 |
| 17H | Unused |
| 18H | PAWU |
| 19H | PAPU |
| 1AH | PA |
| 1BH | PAC |
| 1CH | PBPU |
| 1DH | PB |
| 1EH | PBC |
| 1FH | PCPU |
| 20H | PC |
| 21H | PCC |
| 22H | Unused |
| 23H | Unused |
| 24H | Unused |
| 25H | Unused |
| 26H | Unused |
| 27H | Unused |
| 28H | Unused |
| 29H | Unused |
| 2AH | Unused |
| 2BH | Unused |
| 2CH | Unused |
| 2DH | Unused |
| 2EH | Unused |
| 2FH | Unused |

| | Bank 0, 2 | Bank 1 |
|---|---|---|
| 30H | Unused | |
| 31H | Unused | |
| 32H | Unused | |
| 33H | Unused | |
| 34H | CP0C | |
| 35H | CP1C | |
| 36H | SIMC0 | |
| 37H | SIMC1 | |
| 38H | SIMD | |
| 39H | SIMA/SIMC2 | |
| 3AH | TM0C0 | |
| 3BH | TM0C1 | |
| 3CH | TM0DL | |
| 3DH | TM0DH | |
| 3EH | TM0AL | |
| 3FH | TM0AH | |
| 40H | Unused | EEC |
| 41H | EEA | |
| 42H | EED | |
| 43H | TMPC0 | |
| 44H | Unused | |
| 45H | PRM0 | |
| 46H | Unused | |
| 47H | Unused | |
| 48H | TM1C0 | |
| 49H | TM1C1 | |
| 4AH | TM1C2 | |
| 4BH | TM1DL | |
| 4CH | TM1DH | |
| 4DH | TM1AL | |
| 4EH | TM1AH | |
| 4FH | TM1BL | |
| 50H | TM1BH | |
| 51H | Unused | |
| 52H | Unused | |
| 53H | Unused | |
| 54H | Unused | |
| 55H | Unused | |
| 56H | Unused | |
| 57H | Unused | |
| 58H | Unused | |
| 59H | Unused | |
| 5AH | Unused | |
| 5BH | Unused | |
| 5CH | Unused | |
| 5DH | Unused | |
| 5EH | SCOMC | |
| 5FH | Unused | |

**HT68F30 特殊功能数据存储器结构**

| | Bank 0, 1 |
|---|---|
| 00H | IAR0 |
| 01H | MP0 |
| 02H | IAR1 |
| 03H | MP1 |
| 04H | BP |
| 05H | ACC |
| 06H | PCL |
| 07H | TBLP |
| 08H | TBLH |
| 09H | TBHP |
| 0AH | STATUS |
| 0BH | SMOD |
| 0CH | LVDC |
| 0DH | INTEG |
| 0EH | WDTC |
| 0FH | TBC |
| 10H | INTC0 |
| 11H | INTC1 |
| 12H | INTC2 |
| 13H | Unused |
| 14H | MFI0 |
| 15H | MFI1 |
| 16H | MFI2 |
| 17H | Unused |
| 18H | PAWU |
| 19H | PAPU |
| 1AH | PA |
| 1BH | PAC |
| 1CH | PBPU |
| 1DH | PB |
| 1EH | PBC |
| 1FH | PCPU |
| 20H | PC |
| 21H | PCC |
| 22H | PDPU |
| 23H | PD |
| 24H | PDC |
| 25H | PEPU |
| 26H | PE |
| 27H | PEC |
| 28H | PFPU |
| 29H | PF |
| 2AH | PFC |
| 2BH | Unused |
| 2CH | Unused |
| 2DH | Unused |
| 2EH | Unused |
| 2FH | Unused |
| 30H | Unused |
| 31H | Unused |
| 32H | Unused |
| 33H | Unused |
| 34H | CP0C |
| 35H | CP1C |
| 36H | SIMC0 |
| 37H | SIMC1 |
| 38H | SIMD |
| 39H | SIMA/SIMC2 |
| 3AH | TM0C0 |
| 3BH | TM0C1 |
| 3CH | TM0DL |
| 3DH | TM0DH |
| 3EH | TM0AL |
| 3FH | TM0AH |

| | Bank 0 | Bank 1 |
|---|---|---|
| 40H | Unused | EEC |
| 41H | EEA | |
| 42H | EED | |
| 43H | TMPC0 | |
| 44H | TMPC1 | |
| 45H | PRM0 | |
| 46H | PRM1 | |
| 47H | PRM2 | |
| 48H | TM1C0 | |
| 49H | TM1C1 | |
| 4AH | TM1C2 | |
| 4BH | TM1DL | |
| 4CH | TM1DH | |
| 4DH | TM1AL | |
| 4EH | TM1AH | |
| 4FH | TM1BL | |
| 50H | TM1BH | |
| 51H | TM2C0 | |
| 52H | TM2C1 | |
| 53H | TM2DL | |
| 54H | TM2DH | |
| 55H | TM2AL | |
| 56H | TM2AH | |
| 57H | TM2RP | |
| 58H | Unused | |
| 59H | Unused | |
| 5AH | Unused | |
| 5BH | Unused | |
| 5CH | Unused | |
| 5DH | Unused | |
| 5EH | SCOMC | |
| 5FH | Unused | |
| 60H | Unused | |
| 61H | Unused | |
| 62H | Unused | |
| 63H | Unused | |
| 64H | Unused | |
| 65H | Unused | |
| 66H | Unused | |
| 67H | Unused | |
| 68H | Unused | |
| 69H | Unused | |
| 6AH | Unused | |
| 6BH | Unused | |
| 6CH | Unused | |
| 6DH | Unused | |
| 6EH | Unused | |
| 6FH | Unused | |
| 70H | Unused | |
| 71H | Unused | |
| 72H | Unused | |
| 73H | Unused | |
| 74H | Unused | |
| 75H | Unused | |
| 76H | Unused | |
| 77H | Unused | |
| 78H | Unused | |
| 79H | Unused | |
| 7AH | Unused | |
| 7BH | Unused | |
| 7CH | Unused | |
| 7DH | Unused | |
| 7EH | Unused | |
| 7FH | Unused | |

**HT68F40** 特殊功能数据存储器结构

| | Bank 0, 1, 2 |
|---|---|
| 00H | IAR0 |
| 01H | MP0 |
| 02H | IAR1 |
| 03H | MP1 |
| 04H | BP |
| 05H | ACC |
| 06H | PCL |
| 07H | TBLP |
| 08H | TBLH |
| 09H | TBHP |
| 0AH | STATUS |
| 0BH | SMOD |
| 0CH | LVDC |
| 0DH | INTEG |
| 0EH | WDTC |
| 0FH | TBC |
| 10H | INTC0 |
| 11H | INTC1 |
| 12H | INTC2 |
| 13H | Unused |
| 14H | MFI0 |
| 15H | MFI1 |
| 16H | MFI2 |
| 17H | MFI3 |
| 18H | PAWU |
| 19H | PAPU |
| 1AH | PA |
| 1BH | PAC |
| 1CH | PBPU |
| 1DH | PB |
| 1EH | PBC |
| 1FH | PCPU |
| 20H | PC |
| 21H | PCC |
| 22H | PDPU |
| 23H | PD |
| 24H | PDC |
| 25H | PEPU |
| 26H | PE |
| 27H | PEC |
| 28H | PFPU |
| 29H | PF |
| 2AH | PFC |
| 2BH | Unused |
| 2CH | Unused |
| 2DH | Unused |
| 2EH | Unused |
| 2FH | Unused |
| 30H | Unused |
| 31H | Unused |
| 32H | Unused |
| 33H | Unused |
| 34H | CP0C |
| 35H | CP1C |
| 36H | SIMC0 |
| 37H | SIMC1 |
| 38H | SIMD |
| 39H | SIMA/SIMC2 |
| 3AH | TM0C0 |
| 3BH | TM0C1 |
| 3CH | TM0DL |
| 3DH | TM0DH |
| 3EH | TM0AL |
| 3FH | TM0AH |

| | Bank 0, 2 | Bank 1 |
|---|---|---|
| 40H | Unused | EEC |
| 41H | EEA | |
| 42H | EED | |
| 43H | TMPC0 | |
| 44H | TMPC1 | |
| 45H | PRM0 | |
| 46H | PRM1 | |
| 47H | PRM2 | |
| 48H | TM1C0 | |
| 49H | TM1C1 | |
| 4AH | TM1C2 | |
| 4BH | TM1DL | |
| 4CH | TM1DH | |
| 4DH | TM1AL | |
| 4EH | TM1AH | |
| 4FH | TM1BL | |
| 50H | TM1BH | |
| 51H | TM2C0 | |
| 52H | TM2C1 | |
| 53H | TM2DL | |
| 54H | TM2DH | |
| 55H | TM2AL | |
| 56H | TM2AH | |
| 57H | TM2RP | |
| 58H | TM3C0 | |
| 59H | TM3C1 | |
| 5AH | TM3DL | |
| 5BH | TM3DH | |
| 5CH | TM3AL | |
| 5DH | TM3AH | |
| 5EH | SCOMC | |
| 5FH | Unused | |
| 60H | Unused | |
| 61H | Unused | |
| 62H | Unused | |
| 63H | Unused | |
| 64H | Unused | |
| 65H | Unused | |
| 66H | Unused | |
| 67H | Unused | |
| 68H | Unused | |
| 69H | Unused | |
| 6AH | Unused | |
| 6BH | Unused | |
| 6CH | Unused | |
| 6DH | Unused | |
| 6EH | Unused | |
| 6FH | Unused | |
| 70H | Unused | |
| 71H | Unused | |
| 72H | Unused | |
| 73H | Unused | |
| 74H | Unused | |
| 75H | Unused | |
| 76H | Unused | |
| 77H | Unused | |
| 78H | Unused | |
| 79H | Unused | |
| 7AH | Unused | |
| 7BH | Unused | |
| 7CH | Unused | |
| 7DH | Unused | |
| 7EH | Unused | |
| 7FH | Unused | |

**HT68F50 特殊功能数据存储器结构**

| | Bank 0, 1, 2, 3, 4 |
|---|---|
| 00H | IAR0 |
| 01H | MP0 |
| 02H | IAR1 |
| 03H | MP1 |
| 04H | BP |
| 05H | ACC |
| 06H | PCL |
| 07H | TBLP |
| 08H | TBLH |
| 09H | TBHP |
| 0AH | STATUS |
| 0BH | SMOD |
| 0CH | LVDC |
| 0DH | INTEG |
| 0EH | WDTC |
| 0FH | TBC |
| 10H | INTC0 |
| 11H | INTC1 |
| 12H | INTC2 |
| 13H | INTC3 |
| 14H | MFI0 |
| 15H | MFI1 |
| 16H | MFI2 |
| 17H | MFI3 |
| 18H | PAWU |
| 19H | PAPU |
| 1AH | PA |
| 1BH | PAC |
| 1CH | PBPU |
| 1DH | PB |
| 1EH | PBC |
| 1FH | PCPU |
| 20H | PC |
| 21H | PCC |
| 22H | PDPU |
| 23H | PD |
| 24H | PDC |
| 25H | PEPU |
| 26H | PE |
| 27H | PEC |
| 28H | PFPU |
| 29H | PF |
| 2AH | PFC |
| 2BH | PGPU |
| 2CH | PG |
| 2DH | PGC |
| 2EH | Unused |
| 2FH | Unused |
| 30H | Unused |
| 31H | Unused |
| 32H | Unused |
| 33H | Unused |
| 34H | CP0C |
| 35H | CP1C |
| 36H | SIMC0 |
| 37H | SIMC1 |
| 38H | SIMD |
| 39H | SIMA/SIMC2 |
| 3AH | TM0C0 |
| 3BH | TM0C1 |
| 3CH | TM0DL |
| 3DH | TM0DH |
| 3EH | TM0AL |
| 3FH | TM0AH |

| | Bank 0, 2, 3, 4 | Bank 1 |
|---|---|---|
| 40H | Unused | EEC |
| 41H | EEA | |
| 42H | EED | |
| 43H | TMPC0 | |
| 44H | TMPC1 | |
| 45H | PRM0 | |
| 46H | PRM1 | |
| 47H | PRM2 | |
| 48H | TM1C0 | |
| 49H | TM1C1 | |
| 4AH | TM1C2 | |
| 4BH | TM1DL | |
| 4CH | TM1DH | |
| 4DH | TM1AL | |
| 4EH | TM1AH | |
| 4FH | TM1BL | |
| 50H | TM1BH | |
| 51H | TM2C0 | |
| 52H | TM2C1 | |
| 53H | TM2DL | |
| 54H | TM2DH | |
| 55H | TM2AL | |
| 56H | TM2AH | |
| 57H | TM2RP | |
| 58H | TM3C0 | |
| 59H | TM3C1 | |
| 5AH | TM3DL | |
| 5BH | TM3DH | |
| 5CH | TM3AL | |
| 5DH | TM3AH | |
| 5EH | SCOMC | |
| 5FH | Unused | |
| 60H | Unused | |
| 61H | Unused | |
| 62H | Unused | |
| 63H | Unused | |
| 64H | Unused | |
| 65H | Unused | |
| 66H | Unused | |
| 67H | Unused | |
| 68H | Unused | |
| 69H | Unused | |
| 6AH | Unused | |
| 6BH | Unused | |
| 6CH | Unused | |
| 6DH | Unused | |
| 6EH | Unused | |
| 6FH | Unused | |
| 70H | Unused | |
| 71H | Unused | |
| 72H | Unused | |
| 73H | Unused | |
| 74H | Unused | |
| 75H | Unused | |
| 76H | Unused | |
| 77H | Unused | |
| 78H | Unused | |
| 79H | Unused | |
| 7AH | Unused | |
| 7BH | Unused | |
| 7CH | Unused | |
| 7DH | Unused | |
| 7EH | Unused | |
| 7FH | Unused | |

**HT68F60** 特殊功能数据存储器结构

# 特殊功能寄存器

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

## 间接寻址寄存器 – IAR0，IAR1

间接寻址寄存器 IAR0 和 IAR1 的地址虽位于数据存储区，但其并没有实际的物理地址。间接寻址的方法准许使用间接寻址指针做数据操作，以取代定义实际存储器地址的直接存储器寻址方法。在间接寻址寄存器（IAR0 和 IAR1）上的任何动作，将对间接寻址指针 (MP0 和 MP1) 所指定的存储器地址产生对应的读 / 写操作。它们总是成对出现，IAR0 和 MP0 可以访问 Bank 0，而 IAR1 和 MP1 可以访问 Bank 0 和 Bank 1。因为这些间接寻址寄存器不是实际存在的，直接读取将返回“00H”的结果，而直接写入此寄存器则不做任何操作。

## 间接寻址指针 – MP0，MP1

该系列单片机提供两个间接寻址指针，即 MP0 和 MP1。由于这些指针在数据存储器中能像普通的寄存器一般被操作，因此提供了一个寻址和数据追踪的有效方法。当对间接寻址寄存器进行任何操作时，单片机指向的实际地址是由间接寻址指针所指定的地址。MP0，IAR0 用于访问 Bank 0，而 MP1 和 IAR1 可通过 BP 寄存器访问所有的 Bank。直接寻址仅可用于 Bank 0，其它所有 Bank 都可使用 MP1 和 IAR1 进行间接寻址。注意，对于 HT68F20 和 HT68F30，间接寻址指针的第 7 位没有作用，必须注意当读取间接寻址指针时，第 7 位将返回“1”。

以下例子说明如何清除一个具有 4 RAM 地址的区块，它们已事先定义成地址 adres1 到 adres4。

### 间接寻址程序举例

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

在上面的例子中有一点值得注意，即并没有确定 RAM 地址。

## 存储区指针 – BP

数据存储器被分为几个部分，具体数目由所选择的单片机型号决定。可以通过设置 Bank 指针（Bank Pointer）值来访问不同的程序和数据存储区。BP 指针的第 5 位用于选择程序存储区 0 或 1，第 0~2 位用于选择数据存储区 0~4。

复位后，数据存储器会初始化到存储区 0，但是在暂停模式下的 WDT 溢出复位，不会改变通用数据存储器的存储区号。应该注意的是特殊功能数据存储器不受存储区的影响，也就是说，不论是在哪一个存储区，都能对特殊功能寄存器进行读写操作。数据存储器的直接寻址总是访问 Bank 0，不影响存储区指针的值。要访问 Bank 0 之外的存储区，则必须要使用间接寻址方式。

由于程序存储器和数据存储器共享同一个 BP 寄存器，编程时务必注意。

| 单片机型号 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HT68F20<br>HT68F40 | — | — | — | — | — | — | — | DMBP0 |
| HT68F30<br>HT68F50 | — | — | — | — | — | — | DMBP1 | DMBP0 |
| HT68F60 | — | — | PMBP0 | — | — | DMBP2 | DMBP1 | DMBP0 |

存储区指针

### BP 寄存器

- **HT68F20/HT68F40**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | — | DMBP0 |
| R/W | — | — | — | — | — | — | — | R/W |
| POR | — | — | — | — | — | — | — | 0 |

Bit 7~1　未使用，读为“0”

Bit 0　**DMBP0**：数据存储区选择位
0：Bank 0
1：Bank 1

- **HT68F30/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | DMBP1 | DMBP0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 0 |

Bit 7~2　未使用，读为“0”

Bit 1~0　**DMBP1, DMBP0**：数据存储区选择位
00：Bank 0
01：Bank 1
10：Bank 2
11：未定义

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | PMBP0 | — | — | DMBP2 | DMBP1 | DMBP0 |
| R/W | — | — | R/W | — | — | R/W | R/W | R/W |
| POR | — | — | 0 | — | — | 0 | 0 | 0 |

Bit 7~6　未使用，读为“0”

Bit 5　**PMBP0**：程序存储区选择位
0：Bank 0，程序存储器地址 0000H~1FFFH
1：Bank 1，程序存储器地址 2000H~2FFFH

Bit 4~3　未使用，读为“0”

Bit 2~0　**DMBP2~DMBP0**：数据存储区选择位
000：Bank 0
001：Bank 1
010：Bank 2
011：Bank 3
100：Bank 4
101~111：未定义

## 累加器 – ACC

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

## 程序计数器低字节寄存器 – PCL

为了提供额外的程序控制功能，程序计数器低字节设置在数据存储器的特殊功能区域内，程序员可对此寄存器进行操作，很容易的直接跳转到其它程序地址。直接给 PCL 寄存器赋值将导致程序直接跳转到程序存储器的某一地址，然而由于寄存器只有 8 位长度，因此只允许在本页的程序存储器范围内进行跳转，而当使用这种运算时，要注意会插入一个空指令周期。

## 表格寄存器 – TBLP，TBHP，TBLH

这三个特殊功能寄存器对存储在程序存储器中的表格进行操作。TBLP 和 TBHP 为表格指针，指向表格数据存储的地址。它们的值必须在任何表格读取指令执行前加以设定，由于它们的值可以被如“INC”或“DEC”的指令所改变，这就提供了一种简单的方法对表格数据进行读取。表格读取数据指令执行之后，表格数据高字节存储在 TBLH 中。其中要注意的是，表格数据低字节会被传送到使用者指定的地址。

## 状态寄存器 – STATUS

这 8 位的状态寄存器由零标志位 (Z)、进位标志位 (C)、辅助进位标志位 (AC)、溢出标志位 (OV)、暂停标志位 (PDF) 和看门狗定时器溢出标志位 (TO) 组成。这些算术 / 逻辑操作和系统运行标志位是用来记录单片机的运行状态。

除了 PDF 和 TO 标志外，状态寄存器中的位像其它大部分寄存器一样可以被改变。任何数据写入到状态寄存器将不会改变 TO 或 PDF 标志位。另外，执行不

同的指令后，与状态寄存器有关的运算可能会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出或执行“CLR WDT”或“HALT”指令影响。PDF 标志位只会受执行“HALT”或“CLR WDT”指令或系统上电影响。

Z、OV、AC 和 C 标志位通常反映最近运算的状态。

- C：当加法运算的结果产生进位，或减法运算的结果没有产生借位时，则 C 被置位，否则 C 被清零，同时 C 也会被带进位的移位指令所影响
- AC：当低半字节加法运算的结果产生进位，或低半字节减法运算的结果没有产生借位时，AC 被置位，否则 AC 被清零
- Z：当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零
- OV：当运算结果高两位的进位状态异或结果为 1 时，OV 被置位，否则 OV 被清零
- PDF：系统上电或执行“CLR WDT”指令会清零 PDF，而执行“HALT”指令则会置位 PDF
- TO：系统上电或执行“CLR WDT”或“HALT”指令会清零 TO，而当 WDT 溢出则会置位 TO

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

**STATUS 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | TO | PDF | OV | Z | AC | C |
| R/W | — | — | R | R | R/W | R/W | R/W | R/W |
| POR | — | — | 0 | 0 | × | × | × | × |

“×”为未知

Bit 7~6　未使用，读为“0”

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

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

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

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

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

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

# EEPROM 数据寄存器

此系列所有单片机的一个特性是内建 EEPROM 数据存储器。“Electrically Erasable Programmable Read Only Memory”为电可擦可编程只读存储器，由于其非易失的存储结构，即使在电源掉电的情况下存储器内的数据仍然保存完好。这种存储区扩展了 ROM 空间，对设计者来说增加了许多新的应用机会，EEPROM 可以用来存储产品编号、校准值、用户特定数据、系统配置参数或其它产品信息等。EEPROM 的数据读取和写入过程也会变得更简单。

## EEPROM 数据寄存器结构

EEPROM 数据寄存器容量为 32×8~256×8，根据所选单片机型号而定。由于映射方式与程序存储器和数据存储器不同，因此不能像其它类型的存储器一样寻址。使用 Bank 0 中的一个地址和数据寄存器以及 Bank 1 中的一个控制寄存器，可以实现对 EEPROM 的单字节读写操作。

| 单片机型号 | 容量 | 地址 |
|---|---|---|
| HT68F20 | 32×8 | 00H~1FH |
| HT68F30 | 64×8 | 00H~3FH |
| HT68F40 | 128×8 | 00H~7FH |
| HT68F50/ HT68F60 | 256×8 | 00H~FFH |

**EEPROM 寄存器**

有三个寄存器控制内部 EEPROM 数据存储器总的操作。地址寄存器 EEA、数据寄存器 EED 及控制寄存器 EEC。EEA 和 EED 位于 Bank0 中，它们能像其它特殊功能寄存器一样直接被访问。EEC 位于 Bank1 中，不能被直接访问，仅能通过 MP1 和 IAR1 进行间接读取或写入。由于 EEC 控制寄存器位于 Bank1 中的“40H”，在 EEC 寄存器上的任何操作被执行前，MP1 必须先设为“40H”，BP 被设为“01H”。

### EEPROM 寄存器列表

- **HT68F20**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EEA | — | — | — | D4 | D3 | D2 | D1 | D0 |
| EED | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EEC | — | — | — | — | WREN | WR | RDEN | RD |

- **HT68F30**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EEA | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| EED | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EEC | — | — | — | — | WREN | WR | RDEN | RD |

- **HT68F40**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EEA | — | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EED | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EEC | — | — | — | — | WREN | WR | RDEN | RD |

- **HT68F50/HT68F60**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EEA | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EED | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| EEC | — | — | — | — | WREN | WR | RDEN | RD |

## EEA 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | D4 | D3 | D2 | D1 | D0 |
| R/W | — | — | — | R/W | R/W | R/W | R/W | R/W |
| POR | — | — | — | × | × | × | × | × |

“×”为未知

Bit 7~5　未定义，读为“0”

Bit 4~0　数据 EEPROM 地址
数据 EEPROM 地址 Bit 4~Bit 0

- **HT68F30**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| R/W | — | — | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | — | — | × | × | × | × | × | × |

“×”为未知

Bit 7~6　未定义，读为“0”

Bit 5~0　数据 EEPROM 地址
数据 EEPROM 地址 Bit 5~Bit 0

- **HT68F40**

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

“×”为未知

Bit 7　未定义，读为“0”

Bit 6~0　数据 EEPROM 地址
数据 EEPROM 地址 Bit 6~Bit 0

- **HT68F50/HT68F60**

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

“×”为未知

Bit 7~0 数据 EEPROM 地址
数据 EEPROM 地址 Bit 7~Bit 0

## EEC 寄存器

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

Bit 7~4 未定义，读为“0”

Bit 3 **WREN**：数据 EEPROM 写使能位
0：除能
1：使能
此位为数据 EEPROM 写使能位，向数据 EEPROM 写操作之前需将此位置高。将此位清零时，则禁止向数据 EEPROM 写操作。

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

Bit 1 **RDEN**：数据 EEPROM 读使能位
0：除能
1：使能
此位为数据 EEPROM 读使能位，向数据 EEPROM 读操作之前需将此位置高。将此位清零时，则禁止 向数据 EEPROM 读操作。

Bit 0 **RD**：EEPROM 读控制位
0：读周期结束
1：读周期有效
此位为数据 EEPROM 读控制位，由应用程序将此位置高将激活读周期。读周期结束后，硬件自动将此位清零。当 RDEN 未首先置高时，此位置高无效。

注：在同一条指令中 WREN、WR、RDEN 和 RD 不能同时置为“1”。WR 和 RD 不能同时置为“1”。

### 从 EEPROM 中读取数据

从 EEPROM 中读取数据，EEC 寄存器中的读使能位 RDEN 先置为高以使能读功能，EEPROM 中读取数据的地址要先放入 EEA 寄存器中。若 EEC 寄存器中的 RD 位被置高，一个读周期将开始。若 RD 位已置为高而 RDEN 位还未被设置则不能开始读操作。若读周期结束，RD 位将自动清除为“0”，数据可以从 EED 寄存器中读取。数据在其它读或写操作执行前将一直保留在 EED 寄存器中。应用程序将轮询 RD 位以确定数据可以有效地被读取。

### 写数据到 EEPROM

写数据至 EEPROM，EEC 寄存器中的写使能位 WREN 先置为高以使能写功能。EEPROM 中写入数据的地址要先放入 EEA 寄存器中，写入的数据需存入 EED 寄存器中。若 EEC 寄存器中 WR 位被置为高，一个内部写周期将开始。若 WR 位已置为高而 WREN 位还未被设置则不能开始写操作。由于控制 EEPROM 写周期是一个内部时钟，与单片机的系统时钟异步，所以数据写入 EEPROM 的时间将有所延迟。可通过轮询 EEC 寄存器中的 WR 位或判断 EEPROM 中断以侦测写周期是否完成。若写周期完成，WR 位将自动清除为“0”，通知用户数据已写入 EEPROM。因此，应用程序将轮询 WR 位以确定写周期是否结束。

### 写保护

防止误写入的写保护有以下几种。单片机上电后控制寄存器中的写使能位将被清除以杜绝任何写入操作。上电后 BP 将重置为“0”，这意味着数据存储区 Bank 0 被选中。由于 EEPROM 控制寄存器位于 Bank 1 中，这增加了对写操作的保护措施。在正常程序操作中确保控制寄存器中的写使能位被清除将能防止不正确的写操作。

### EEPROM 中断

EEPROM 写或读周期结束后将产生 EEPROM 写或读中断，需先通过设置相关中断寄存器的 DEE 位使能 EEPROM 中断。由于 EEPROM 中断包含在多功能中断中，相应的多功能中断使能位需被设置。当 EEPROM 写周期结束，DEF 请求标志位及其相关多功能中断请求标志位将被置位。若 EEPROM 和多功能中断使能且堆栈未满的情况下将跳转到相应的多功能中断向量中执行。当中断被响应，只有多功能中断标志位将自动复位，而 EEPROM 中断标志将通过应用程序手动复位。更多细节将在中断章节讲述。

### 编程注意事项

必须注意的是数据不会无意写入 EEPROM。在没有写动作时写使能位被正常清零可以增强保护功能。BP 指针也可以正常清零以阻止进入 EEPROM 控制寄存器存在的 Bank1。尽管没有必要，写一个简单的读回程序以检查新写入的数据是否正确还是应该考虑的。

## 程序举例

- 从 EEPROM 中读取数据—轮询法

```
MOV A, EEPROM_ADRES               ; user defined address
MOV EEA, A
MOV A, 040H                       ; setup memory pointer MP1
MOV MP1, A                        ; MP1 points to EEC register
MOV A, 01H                        ; setup Bank Pointer
MOV BP, A
SET IAR1.1                        ; set RDEN bit, enable read operations
SET IAR1.0                        ; start Read Cycle - set RD bit
BACK:
SZ IAR1.0                         ; check for read cycle end
JMP BACK
CLR IAR1                          ; disable EEPROM read/write
CLR BP
MOV A, EEDATA                     ; move read data to register
MOV READ_DATA, A
```

- 写数据到 EEPROM—轮询法

```
MOV A, EEPROM_ADRES               ; user defined address
MOV EEA, A
MOV A, EEPROM_DATA                ; user defined data
MOV EED, A
MOV A, 040H                       ; setup memory pointer MP1
MOV MP1, A                        ; MP1 points to EEC register
MOV A, 01H                        ; setup Bank Pointer
MOV BP, A
CLR EMI
SET IAR1.3                        ; set WREN bit, enable write operations
SET IAR1.2                        ; start Write Cycle - set WR bit
SET EMI
BACK:
SZ IAR1.2                         ; check for write cycle end
JMP BACK
CLR IAR1                          ; disable EEPROM read/write
CLR BP
```

# 振荡器

不同的振荡器选择可以让使用者在不同的应用需求中实现更大范围的功能。振荡器的灵活性使得在速度和功耗方面可以达到较佳的优化。振荡器选择是通过配置选项和寄存器共同完成的。

## 振荡器概述

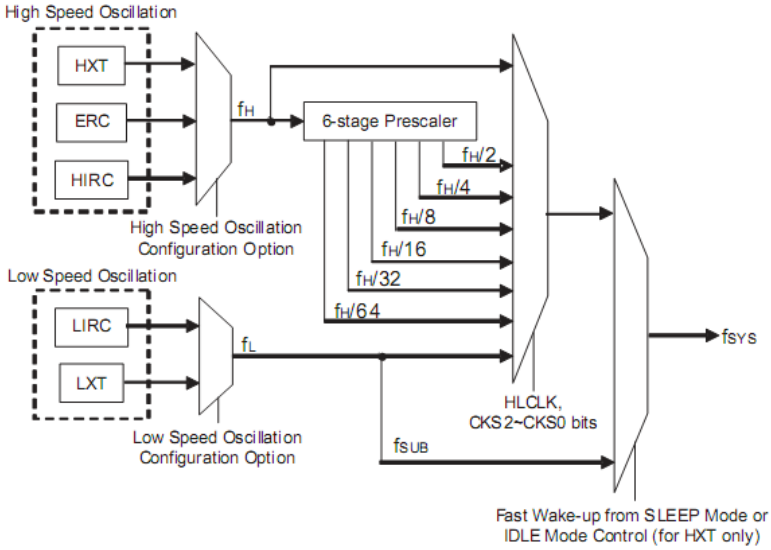
振荡器除了作为系统时钟源，还作为看门狗定时器和时基功能的时钟源。外部振荡器需要一些外围器件，而集成的两个内部振荡器不需要任何外围器件。它们提供的高速和低速系统振荡器具有较宽的频率范围。所有振荡器选择通过配置选项选择。较高频率的振荡器提供更高的性能，但要求有更高的功率，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活而优化的性能 / 功耗比，此特性对功耗敏感的应用领域尤为重要。

| 类型 | 名称 | 频率 | 引脚 |
|---|---|---|---|
| 外部晶振 | HXT | 400kHz~20MHz | OSC1/OSC2 |
| 外部 RC | ERC | 8MHz | OSC1 |
| 内部高速 RC | HIRC | 4，8 或 12MHz | — |
| 外部低速晶振 | LXT | 32.768kHz | XT1/ XT2 |
| 内部低速 RC | LIRC | 32kHz | — |

振荡器类型

## 系统时钟配置

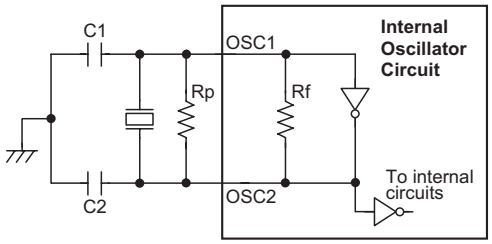
此系列的单片机有五个系统振荡器，包括三个高速振荡器和两个低速振荡器。高速振荡器有外部晶体 / 陶瓷振荡器，外部 RC 振荡器和内部 4MHz，8MHz 或 12MHz RC 振荡器。两个低速振荡器包括外部 32.768kHz 振荡器和内部 32kHz 振荡器。使用高速或低速振荡器作为系统时钟的选择是通过设置 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位决定的，系统时钟可动态选择。高速或低速振荡器的实际时钟源经由配置选项选择。低速或高速系统时钟频率由 SMOD 寄存器的 HLCLK 位及 CKS2~CKS0 位决定的。请注意，两个振荡器必须做出选择，即一个高速和一个低速振荡器。




系统时钟配置

## 外部晶体 / 陶瓷振荡器 – HXT

外部高频晶体 / 陶瓷振荡器可通过配置选项选择。对于晶体振荡器，只要简单地将晶体连接至 OSC1 和 OSC2，则会产生振荡所需的相移及反馈，而不需其它外部器件。为保证某些低频率的晶体振荡和陶瓷谐振器的振荡频率更精准，建议连接两个小容量电容 C1 和 C2 到 VSS，具体数值与客户选择的晶体 / 陶瓷晶振有关。




Note: 1. Rp is normally not required. C1 and C2 are required.
2. Although not shown OSC1/OSC2 pins have a parasitic capacitance of around 7pF.

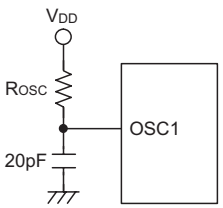
**晶体 / 陶瓷振荡器 – HXT**

| 晶体振荡器 C1 和 C2 值 | | |
|---|---|---|
| 晶体频率 | C1 | C2 |
| 12MHz | 0pF | 0pF |
| 8 MHz | 0pF | 0pF |
| 4 MHz | 0pF | 0pF |
| 1 MHz | 100pF | 100pF |
| 注：C1 和 C2 数值仅作参考用 | | |

**晶体振荡器电容推荐值**

## 外部 RC 振荡器 – ERC

ERC 振荡器只需要在 OSC1 和 VDD 之间连接一个阻值约在 56kΩ 到 2.4MΩ 之间的电阻，OSC1 与 VSS 之间连接一个电容。系统频率由外部所接电阻的大小决定，外部电容并不会影响振荡器的频率值，在这里只起到稳定的作用。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因 VDD、温度以及芯片制成工艺不同的影响较大程度地降低。这里，提供一个电阻 / 频率的参考：使用外部 120K 电阻连接到 5V 电源电压，在 25℃下，振荡器的频率为 8MHz，容差 2%。外部 RC 振荡器仅使用 OSC1 引脚，OSC1 与 PB1 引脚共用，此时 PB2 引脚可以作为普通的 I/O 口使用。




**外部 RC 振荡器 -- ERC**

## 内部 RC 振荡器 – HIRC

内部 RC 振荡器是一个集成的系统振荡器，不需其它外部器件。内部 RC 振荡器具有三种固定的频率：4MHz，8MHz，12MHz。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因 VDD、温度以及芯片制成工艺不同的影响较大程度地降低。在电源电压为 3.3V 或 5V 及温度为 25℃ 的条件下，4MHz，8MHz，12MHz 这三个固定频率的容差为 2%。如果选择了该内部时钟，无需额外的引脚；PB1 和 PB2 可以作为通用 I/O 口使用。

## 外部 32.768kHz 晶体振荡器 – LXT

外部 32.768kHz 晶体振荡器是一个低频振荡器，经由配置选项选择。时钟频率固定为 32.768kHz，此时 XT1 和 XT2 间引脚必须连接 32.768kHz 的晶体振荡器。需要外部电阻和电容连接到 32768Hz 晶振以帮助起振。对于那些要求精确频率的场合中，可能需要这些元件来对由制程产生的误差提供频率补偿。在系统上电期间，LXT 振荡器启动需要一定的延时。

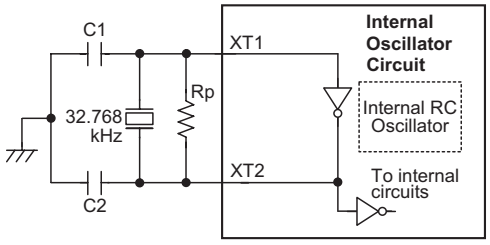
当系统进入空闲 / 休眠模式，系统时钟关闭以降低功耗。然而在某些应用，比如空闲 / 休眠模式

下要保持内部定时器功能，必须提供额外的时钟，且与系统时钟无关。

然而，对于一些晶体，为了保证系统频率的启动与精度要求，需要外接两个小容量电容 C1 和 C2，具体数值与客户选择的晶体规格有关。外部并联的反馈电阻 $R_P$，是必需的。

一些配置选项决定是否 XT1/XT2 脚是用于 LXT 还是作为普通 I/O 口使用。

- 若 LXT 振荡器未被用于任何时钟源，XT1/XT2 脚能被用作一般 I/O 口使用。
- 若 LXT 振荡器被用于一些时钟源，32.768kHz 晶体应被连接至 XT1/XT2 脚。




Note: 1. Rp, C1 and C2 are required.
2. Although not shown pins have a parasitic capacitance of around 7pF.

外部 LXT 振荡器

| LXT 振荡器 C1 和 C2 值 | | |
|---|---|---|
| 晶体频率 | C1 | C2 |
| 32.768kHz | 10pF | 10pF |
| 注：1、C1 和 C2 数值仅作参考用<br>2、$R_P$ 的建议值为 5M~10MΩ | | |

32.768kHz 振荡器电容推荐值

### LXT 振荡器低功耗功能

LXT 振荡器可以工作在快速启动模式或低功耗模式，可通过设置 TBC 寄存器中的 LXTLP 位进行模式选择。

| LXTLP 位 | LXT 模式 |
|---|---|
| 0 | 快速启动 |
| 1 | 低功耗 |

系统上电时会清零 LXTLP 位来快速启动 LXT 振荡器。在快速启动模式，LXT 振荡器将起振并快速稳定下来。LXT 振荡器完全起振后，可以通过设置 LXTLP 位为高进入低功耗模式。振荡器可以继续运行，其间耗电将少于快速启动模式。在功耗敏感的应用领域如电池应用方面，功耗必须限制为一个最小值。为了降低功耗，建议系统上电 2 秒后，在应用程序中将 LXTLP 位设为“1”。应注意的是，无论 LXTLP 位是什么值，LXT 振荡器会一直运作，不同的只是在低功耗模式时启动时间更长。

### 内部 32kHz 振荡器 – LIRC

内部 32kHz 系统振荡器也是一个低频振荡器，经由配置选项选择。这种单片机有一个完全集成 RC 振荡器，它在 5V 电压下运行的典型频率值为 32kHz 且无需外部元件。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡器因电源电压、温度及芯片制成工艺不同的影响较大程度地降低。因此，内部 32kHz 振荡器频率在 25℃温度 5V 电压下的精度保持在 10% 以内。

### 辅助振荡器

低速振荡器除了提供一个系统时钟源外，也用来为看门狗定时器和时基中断提供时钟来源。

## 工作模式和系统时钟

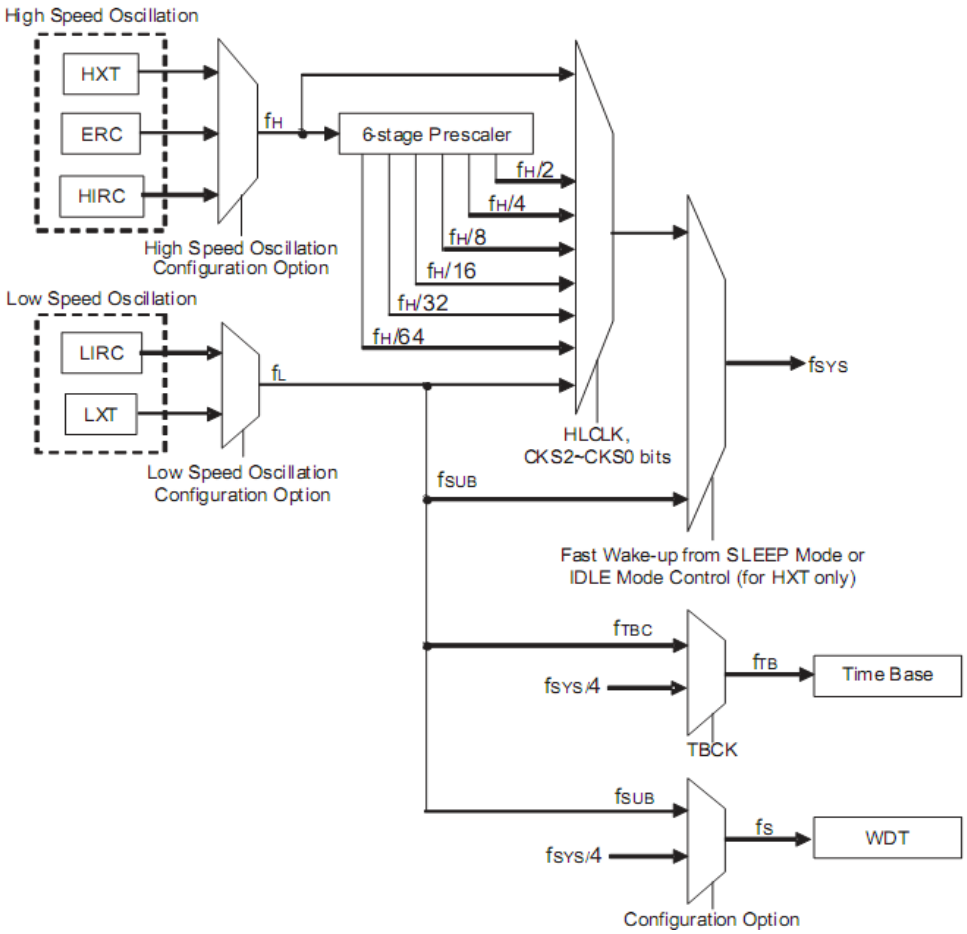
现今的应用要求单片机具有较高的性能及尽可能低的功耗，这种矛盾的要求在便携式电池供电的应用领域尤为明显。高性能所需要的高速时钟将增加功耗，反之亦然。此单片机提供高、低速两种时钟源，它们之间可以动态切换，用户可通过优化单片机操作来获得较佳性能 / 功耗比。

### 系统时钟

单片机为 CPU 和外围功能操作提供了多种不同的时钟源。用户使用配置选项和寄存器编程可获取多种时钟，进而使系统时钟获取较大的应用性能。

主系统时钟可来自高频时钟源 $f_H$ 或低频时钟源 $f_L$，通过 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位进行选择。高频时钟来自 HXT、ERC 或 HIRC 振荡器，可通过配置选项选择，低频系统时钟源来自内部时钟 $f_L$，若 $f_L$ 被选择，可通过配置选项设定为 LXT 或 LIRC 振荡器。其它系统时钟还有高速系统振荡器的分频 $f_H/2$~$f_H/64$。

另外两个内部时钟用于外围电路，次时钟源 $f_{SUB}$ 和时基时钟 $f_{TBC}$。这两个时钟源来自 LXT 或 LIRC 振荡器，通过配置选项选择。快速唤醒发生后，$f_{SUB}$ 为单片机提供一个次时钟。$f_{SUB}$ 和 $f_{SYS}/4$ 用于看门狗定时器的的时钟源。$f_{TBC}$ 用于时基中断功能和 TMs 的时钟源。




系统时钟选项

注：当系统时钟源 $f_{SYS}$ 由 $f_H$ 到 $f_L$ 转换时，高速振荡器将停止以节省耗电。因此，没有为外围电路提供 $f_H/2$~$f_H/64$ 的频率。

## 系统工作模式

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

| 工作模式 | 说明 | | | | |
|---|---|---|---|---|---|
| | CPU | $f_{SYS}$ | $f_{SUB}$ | $f_S$ | $f_{TBC}$ |
| 正常模式 | On | $f_H$~$f_H$/64 | On | On | On |
| 低速模式 | On | $f_L$ | On | On | On |
| 空闲模式 0 | Off | Off | On | On/Off | On |
| 空闲模式 1 | Off | On | On | On | On |
| 休眠模式 0 | Off | Off | Off | Off | Off |
| 休眠模式 1 | Off | Off | On | On | Off |

### 正常模式

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

### 低速模式

此模式的系统时钟虽为较低速时钟源，但单片机仍能正常工作。该低速时钟源可来自 LXT 或 LIRC 振荡器。单片机在此模式中运行所耗工作电流较低。在低速模式下，$f_H$ 关闭。

### 休眠模式 0

在 HALT 指令执行后且 SMOD 寄存器中 IDLEN 位为低时，系统进入休眠模式。在休眠模式 0 中，CPU、$f_{SUB}$ 及 $f_S$ 停止运行，看门狗定时器功能除能。在该模式中 LVDEN 位需置为“0”，否则将不能进入休眠模式 0 中。

### 休眠模式 1

在 HALT 指令执行后且 SMOD 寄存器中 IDLEN 位为低时，系统进入休眠模式。在休眠模式 1 中，CPU 停止运行。然而当其时钟源经配置选项选择为 $f_{SUB}$ 时，若 LVDEN 位为“1”或看门狗定时器功能使能，$f_{SUB}$ 及 $f_S$ 继续运行。

### 空闲模式 0

执行 HALT 指令后且 SMOD 寄存器中 IDLEN 位为高，WDTC 寄存器中 FSYSON 位为低时，系统进入空闲模式 0。在空闲模式 0 中，CPU 停止，但一些外围功能如看门狗定时器、TMs 和 SIM 将继续工作。在空闲模式 0 中，系统振荡器停止，看门狗定时器时钟 $f_S$ 开启或关闭由 $f_S$ 所选时钟源决定。若时钟源为 $f_{SYS}$/4，$f_S$ 关闭；若时钟源为 $f_{SUB}$，$f_S$ 开启。

### 空闲模式 1

执行 HALT 指令后且 SMOD 寄存器中 IDLEN 位为高，WDTC 寄存器中 FSYSON 位为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但会提供一个时钟源给一些外围功能如看门狗定时器、TMs 和 SIM。在空闲模式 1 中，系统振荡器继续运行，该系统振荡器可以为高速或低速系统振荡器。在该模式中看门狗定时器时钟 $f_S$ 开启。若时钟源为 $f_{SYS}$/4 或 $f_{SUB}$，$f_S$ 开启。

## 控制寄存器

寄存器 SMOD 用于控制单片机内部时钟。

**SMOD 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CKS2 | CKS1 | CKS0 | FSTEN | LTO | HTO | IDLEN | HLCLK |
| R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Bit 7~5 **CKS2~CKS0**：当 HLCLK 为“0”时系统时钟选择位
000：$f_L$ ($f_{LXT}$ 或 $f_{LIRC}$)
001：$f_L$ ($f_{LXT}$ 或 $f_{LIRC}$)
010：$f_H/64$
011：$f_H/32$
100：$f_H/16$
101：$f_H/8$
110：$f_H/4$
111：$f_H/2$
这三位用于选择系统时钟源。除了 LXT 或 LIRC 振荡器提供的系统时钟源外，也可使用高频振荡器的分频作为系统时钟。

Bit 4 **FSTEN**：快速唤醒控制位（仅用于 HXT）
0：除能
1：使能
此位为快速唤醒控制位，用于决定单片机被唤醒后 $f_{SUB}$ 是否开始工作。

Bit 3 **LTO**：低速振荡器就绪标志位
0：未就绪
1：就绪
此位为低速系统振荡器就绪标志位，用于表明低速系统振荡器在系统上电复位或经唤醒后何时稳定下来。当系统处于 SLEEP0 模式时，该标志为低。若系统时钟来自 LXT 振荡器，系统唤醒后该位转换为高需 1024 个时钟周期；若系统时钟来自 LIRC 振荡器，该位转换为高需 1~2 个时钟周期。

Bit 2 **HTO**：高速振荡器就绪标志位
0：未就绪
1：就绪
此位为高速系统振荡器就绪标志位，用于表明高速系统振荡器何时稳定下来。此标志在系统上电后经硬件清零，高速系统振荡器稳定后变为高电平。因此，此位在单片机上电后由应用程序读取的总为“1”。该标志由休眠模式或空闲模式 0 中唤醒后会处于低电平状态，若使用 HXT 振荡器，该位将在 1024 个时钟周期后变为高电平状态，若使用 ERC 或 HIRC 振荡器则只需 15~16 个时钟周期即可。

Bit 1 **IDLEN**：空闲模式控制位
0：除能
1：使能
此位为空闲模式控制位，用于决定 HALT 指令执行后发生的动作。若此位为高，当指令 HALT 执行后，单片机进入空闲模式。若 FSYSON 位为高，在空闲模式 1 中 CPU 停止运行，系统时钟将继续工作以保持外围功能继续工作；若 FSYSON 为低，在空闲模式 0 中 CPU 和系统时钟都将停止运行。若此位为低，单片机将在 HALT 指令执行后进入休眠模式。

Bit 0 **HLCLK**：系统时钟选择位
0：$f_H/2$~ $f_H/64$ 或 $f_L$
1：$f_H$
此位用于选择 $f_H$ 或 $f_H/2$~ $f_H/64$ 还是 $f_L$ 作为系统时钟。该位为高时选择 $f_H$ 作为系统时钟，为低时则选择 $f_H/2$~ $f_H/64$ 或 $f_L$ 作为系统时钟。当系统时钟由 $f_H$ 时钟向 $f_L$ 时钟转换时，$f_H$ 将自动关闭以降低功耗。

## 快速唤醒

单片机进入休眠模式或空闲模式 0 后，系统时钟将停止以降低功耗。然而单片机再次唤醒，原来的系统时钟重新起振、稳定且恢复正常工作需要一定的时间。为确保单片机能够尽快的开始工作，系统提供了一个快速唤醒功能。需提供一个临时时钟源 $f_{SUB}$ 先驱动系统直至原系统振荡器稳定，这个临时时钟可来自 LXT 或 LIRC 振荡器。快速启动功能的时钟源为 $f_{SUB}$，该功能仅在休眠模式 1 和空闲模式 0 中有效。当单片机由休眠模式 0 唤醒时，因 $f_{SUB}$ 已停止，故快速唤醒功能无效。快速唤醒功能使能 / 除能由 SMOD 寄存器中 FSTEN 位控制的。

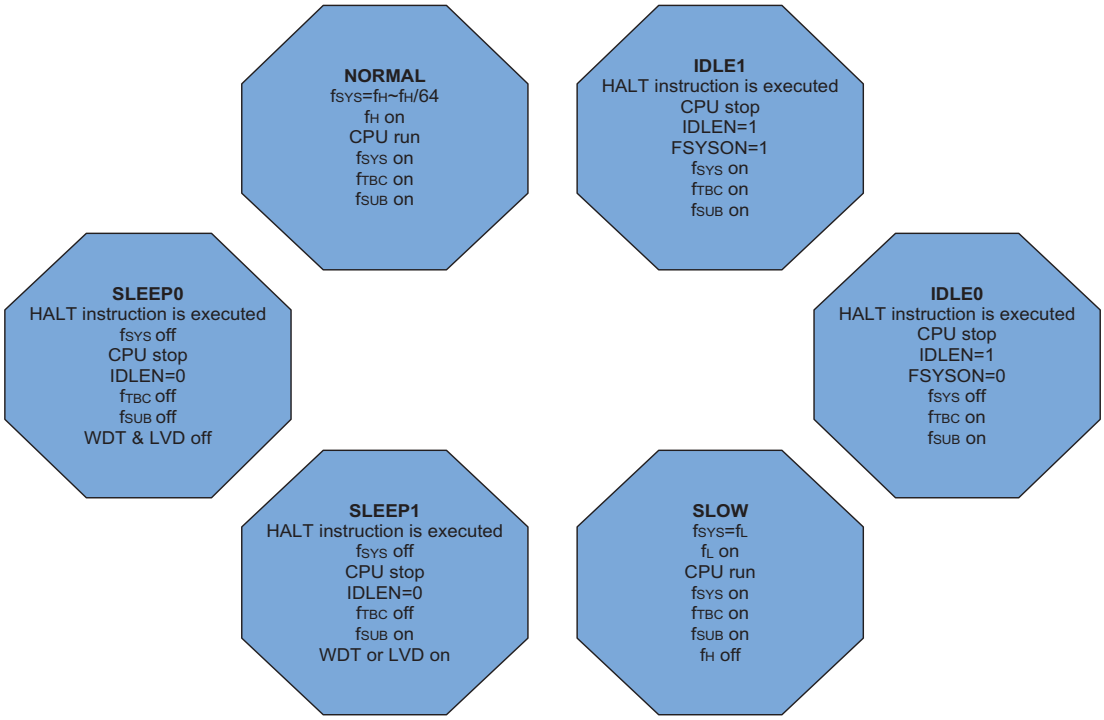
若 HXT 振荡器作为正常模式的系统时钟，且快速唤醒功能使能，系统唤醒将需 1~2 个 $t_{SUB}$ 时钟周期。系统开始在 $f_{SUB}$ 时钟源下运行直至 1024 个 HXT 时钟周期后 HTO 标志转换为高，系统将切换到 HXT 振荡器运行。

若系统振荡器选用 ERC 或 HIRC，将系统从休眠模式或空闲模式 0 中唤醒需 15~16 个时钟周期；若选用 LIRC，则需 1~2 个周期。快速唤醒位 FSTEN 在这些情况下不受影响。

| 系统振荡器 | FSTEN 位 | 唤醒时间（休眠模式 0） | 唤醒时间（休眠模式 1） | 唤醒时间（空闲模式 0） | 唤醒时间（空闲模式 1） |
|---|---|---|---|---|---|
| HXT | 0 | 1024 个 HXT 周期 | 1024 个 HXT 周期 | | 1~2 个 HXT 周期 |
| | 1 | 1024 个 HXT 周期 | 1~2 个 $f_{SUB}$ 周期（系统在 $f_{SUB}$ 下运行 1024 个 HXT 周期后切换到 HXT 振荡器运行） | | 1~2 个 HXT 周期 |
| ERC | × | 15~16 个 ERC 周期 | 15~16 个 ERC 周期 | | 1~2 个 ERC 周期 |
| HIRC | × | 15~16 个 HIRC 周期 | 15~16 个 HIRC 周期 | | 1~2 个 HIRC 周期 |
| LIRC | × | 1~2 个 LIRC 周期 | 1~2 个 LIRC 周期 | | 1~2 个 LIRC 周期 |
| LXT | × | 1024 个 LXT 周期 | 1024 个 LXT 周期 | | 1~2 个 LXT 周期 |

唤醒时间

注：若看门狗定时器除能，意味着 LXT 和 LIRC 都关闭，当单片机由休眠模式 0 中唤醒时快速唤醒功能不可用。




## 工作模式切换和唤醒

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

简单来说，正常模式和低速模式间的切换仅需设置 SMOD 中的 HLCLK 位及 CKS2~CKS0 位即可实现，而正常模式 / 低速模式与休眠模式 / 空闲模式间的切换经由 HALT 指令实现。当 HALT 指令执行后，单片机是否进入空闲模式或休眠模式由 SMOD 寄存器中的 IDLEN 位和 WDTC 寄存器中的 FSYSON 位决定的。

当 HLCLK 位变为低电平时，时钟源将由高速时钟源 $f_H$ 转换成时钟源 $f_H/2$~$f_H/64$ 或 $f_L$。若时钟源来自 $f_L$，高速时钟源将停止运行以节省耗电。此时须注意，$f_H/16$ 和 $f_H/64$ 内部时钟源也将停止运行，由此会影响到如 TMs 和 SIM 等内部功能的工作。所附流程图显示了单片机在不同工作模式间切换时的变化。

## 正常模式切换到低速模式

系统运行在正常模式时使用高速系统振荡器，因此较为耗电。可通过设置 SMOD 寄存器中的 HLCLK 位为“0”及 CKS2~CKS0 位为“000”或“001”使系统时钟切换至运行在低速模式下。此时将使用低速系统振荡器以节省耗电。用户可在对性能要求不高的操作中使用此方法以减少耗电。

低速模式的时钟源来自 LXT 或 LIRC 振荡器，因此要求这些振荡器在所有模式切换动作发生前稳定下来。该动作由 SMOD 寄存器中 LTO 位控制。

### 低速模式切换到正常模式

在低速模式系统使用 LXT 或 LIRC 低速振荡器。切换到使用高速系统时钟振荡器的正常模式需设置 HLCLK 位为“1”，也可设置 HLCLK 位为“0”但 CKS2~CKS0 需设为“010”、“011”、“100”、“101”、“110”或“111”。高频时钟需要一定的稳定时间，通过检测 HTO 位的状态可进行判断。高速振荡器的稳定时间由所使用高速系统振荡器的类型决定。

### 进入休眠模式 0

进入休眠模式 0 的方法仅有一种 – 应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 和 LVD 功能除能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟、WDT 时钟和时基时钟停止运行，应用程序停止在“HALT”指令处
- 数据存储器中的内容和寄存器将保持当前值
- 无论 WDT 时钟源来自 $f_{SUB}$ 时钟或系统时钟，WDT 都将被清除并停止运行
- 输入 / 输出口将保持当前值
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除

### 进入休眠模式 1

进入休眠模式 1 的方法仅有一种 – 应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 或 LVD 功能使能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟和时基时钟停止运行，应用程序停止在“HALT”指令处。WDT 或 LVD 继续运行，其时钟源来自 $f_{SUB}$
- 数据存储器中的内容和寄存器将保持当前值
- 若 WDT 使能且其时钟源来自 $f_{SUB}$，则 WDT 将被清零并重新开始计数
- 输入 / 输出口将保持当前值
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除

### 进入空闲模式 0

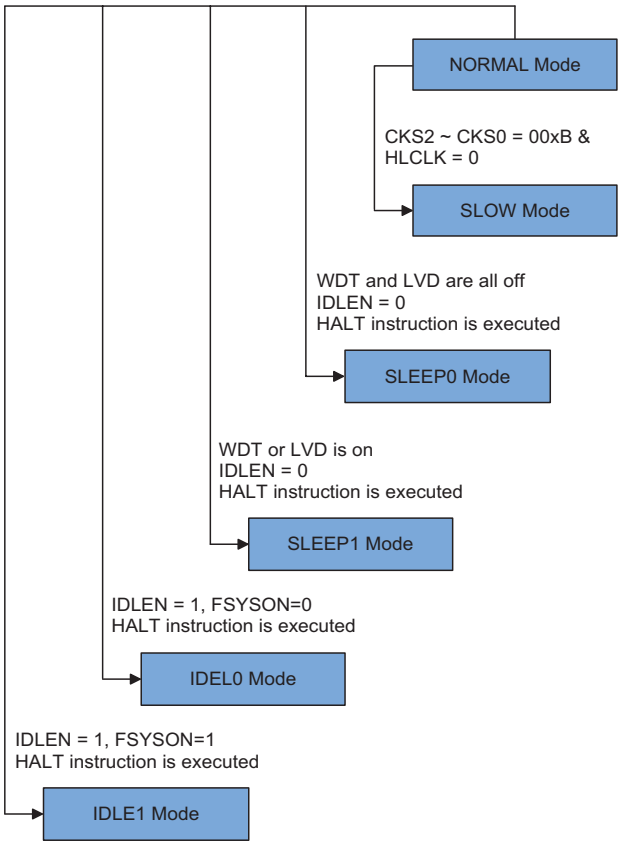
进入空闲模式 0 的方法仅有一种 – 应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“1”且 WDTC 寄存器中的 FSYSON 位为“0”。在上述条件下执行该指令后，将发生的情况如下：

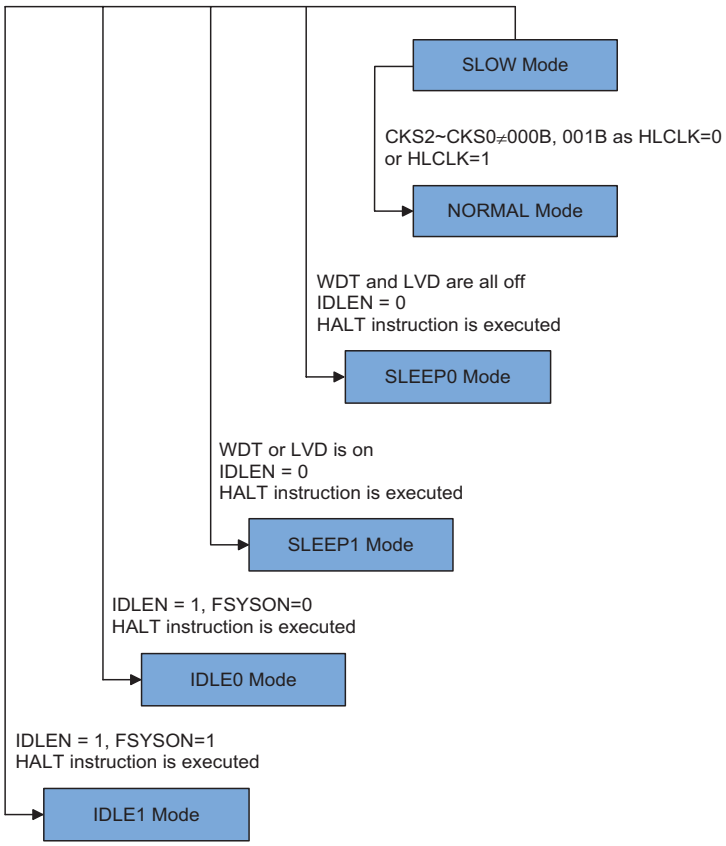
- 系统时钟停止运行，应用程序停止在“HALT”指令处，时基时钟和 $f_{SUB}$ 时钟将继续运行
- 数据存储器中的内容和寄存器将保持当前值
- 若 WDT 使能且其时钟源来自 $f_{SUB}$，则 WDT 将被清零并重新开始计数；若其时钟源来自系统时钟，则 WDT 将停止运行
- 输入 / 输出口将保持当前值
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除

## 进入空闲模式 1

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

- 系统时钟、时基时钟和 $f_{SUB}$ 开启，应用程序停止在“HALT”指令处
- 数据存储器中的内容和寄存器将保持当前值
- 若 WDT 使能，无论 WDT 时钟源来自 $f_{SUB}$ 或是系统时钟，则 WDT 将被清零并重新开始计数
- 输入 / 输出口将保持当前值
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除






## 静态电流的注意事项

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

另外还需注意单片机设为输出的 I/O 引脚上的负载。应将它们设置在有最小拉电流的状态或将它们和其它的 CMOS 输入一样接到没有拉电流的外部电路上。还应注意的是，如果使能配置选项中的 LXT 或 LIRC 振荡器，会导致耗电增加。在空闲模式 1 中，系统时钟开启。若系统时钟来自高速系统振荡器，额外的静态电流也可能会有几百微安。

## 唤醒

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- 外部复位
- PA 口下降沿
- 系统中断
- WDT 溢出

若由外部 RES 引脚唤醒，系统会经过完全复位的过程；若由 WDT 溢出唤醒，则会发生看门狗定时器复位。这两种唤醒方式都会使系统复位，可以通过状态寄存器中 TO 和 PDF 位来判断它的唤醒源。系统上电或执行清除看门狗的指令，会清零 PDF；执行 HALT 指令，PDF 将被置位。看门狗计数器溢出将会置位 TO 标志并唤醒系统，这种复位会重置程序计数器和堆栈指针，其它标志保持原有状态。

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

### 编程注意事项

HXT 和 LXT 振荡器使用相同的 SST 计数器。例如，若系统从休眠模式 0 中唤醒，HXT 和 LXT 振荡器都需从关闭状态快速启动。HXT 振荡器结束其 SST 周期后，LXT 振荡器才开始使用 SST 计数器。

- 若单片机从休眠模式 0 唤醒后进入正常模式，高速系统振荡器需要一个 SST 周期。在 HTO 为“1”后，单片机开始执行首条指令。此时，若 $f_{SUB}$ 时钟来源于 LXT 振荡器，LXT 振荡器可能不是稳定的，上电状态可能会发生类似情况，首条指令执行时 LXT 振荡器还未就绪。
- 若单片机从休眠模式 1 唤醒后进入正常模式，系统时钟源来自 HXT 振荡器且 FSTEN 为“1”，唤醒后，系统时钟可切换至 LXT 或 LIRC 振荡器。
- 一些外围功能，如 WDT，TMs 和 SIM，采用系统时钟 $f_{SYS}$ 时，在系统时钟源由 $f_H$ 切换至 $f_L$ 时，以上这些功能的时钟源也要随之改变。
- 当 WDT 时钟源选择为 $f_{SUB}$ 时，$f_{SUB}$ 和 $f_S$ 的开启或关闭由 WDT 是否使能决定的。

## 看门狗定时器

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

### 看门狗定时器时钟源

WDT 定时器时钟源来自于内部时钟 $f_S$，而 $f_S$ 的时钟源又是通过配置选项从 $f_{SUB}$ 和 $f_{SYS}/4$ 中选择。$f_{SUB}$ 时钟由 LXT 或 LIRC 振荡器提供，可通过配置选项设置。看门狗定时器的时钟源可分频为 $2^8$~$2^{15}$ 以提供更大的溢出周期，分频比由 WDTC 寄存器中的 WS2~WS0 位来决定。电压为 5V 时内部振荡器 LIRC 的周期大约为 32kHz。需要注意的是，这个特殊的内部时钟周期随 VDD、温度和制成的不同而变化。LXT 振荡器由一个外部 32.768kHz 晶振提供。另一个看门狗定时器时钟源选项为 $f_{SYS}/4$。看门狗定时器时钟源可来自内部 LIRC 振荡器、LXT 振荡器或 $f_{SYS}/4$。

## 看门狗定时器控制寄存器

WDTC 寄存器用于控制 WDT 功能的使能 / 除能及选择溢出周期。寄存器结合配置选项控制看门狗定时器的工作。

**WDTC 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | FSYSON | WS2 | WS1 | WS0 | WDTEN3 | WDTEN2 | WDTEN1 | WDTEN0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Bit 7 **FSYSON**：$f_{SYS}$ 在空闲模式下的控制位
0：除能
1：使能

Bit 6~4 **WS2，WS1，WS0**：WDT 溢出周期选择位
000：$256/f_S$
001：$512/f_S$
010：$1024/f_S$
011：$2048/f_S$
100：$4096/f_S$
101：$8192/f_S$
110：$16384/f_S$
111：$32768/f_S$
这三位控制 WDT 时钟源的分频比，从而实现对 WDT 溢出周期的控制。

Bit 3~0 **WDTEN3，WDTEN2，WDTEN1，WDTEN0**：WDT 软件控制位
1010：除能
其它：使能

## 看门狗定时器操作

当 WDT 溢出时，它产生一个芯片复位的动作。这也就意味着正常工作期间，用户需在应用程序中看门狗溢出前有策略地清看门狗定时器以防止其产生复位，可使用清除看门狗指令实现。无论什么原因，程序失常跳转到一个未知的地址或进入一个死循环，这些清除指令都不能被正确执行，此种情况下，看门狗将溢出以使单片机复位。通过配置选项选择看门狗定时器的一些选项，如使能 / 除能、时钟源选择及清除指令类型。除了配置选项使能 / 除能看门狗定时器外，WDTC 寄存器中的 WDTEN3~WDTEN0 位也可用来除能看门狗定时器，此时需设置 WDTEN3~WDTEN0 为“1010”。若使用看门狗定时器功能，推荐设置这四位为“0101”，提供最大可能的防干扰能力。注意，若看门狗定时器被除能，相关操作的任何指令都不会工作。

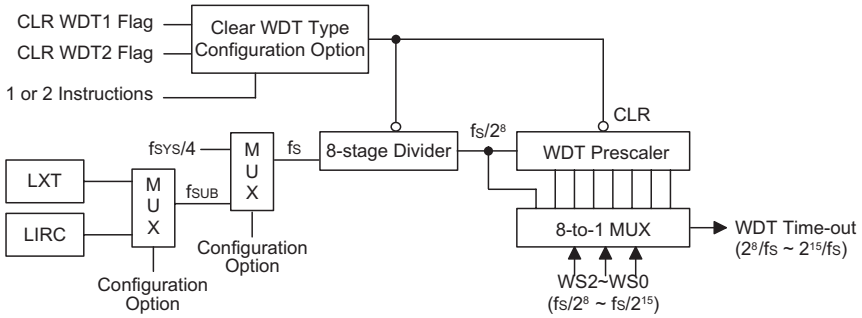
| WDT 配置选项 | WDTEN3~WDTEN0 位 | WDT 功能 |
|---|---|---|
| WDT 使能 | ×××× | 使能 |
| WDT 除能 | 除 1010 外其它值 | 使能 |
| WDT 除能 | 1010 | 除能 |

**看门狗定时器使能 / 除能控制**

程序正常运行时，WDT 溢出将导致芯片复位，并置位状态标志位 TO。若系统处于休眠或空闲模式，当 WDT 发生溢出时，状态寄存器中的 TO 应置位，仅 PC 和堆栈指针复位。有三种方法可以用来清除 WDT 的内容。第一种是外部硬件复位 ($\overline{RES}$ 引脚低电平 )，第二种是通过软件清除指令，而第三种是通过“HALT”指令。

软件指令有两种用于清除看门狗寄存器，需通过配置选项选择。第一种选择是使用一条“CLR WDT”指令，而第二种是使用“CLR WDT1”和“CLR WDT2”两个指令。对于第一种选择，只要执行“CLR WDT”便清除 WDT。而第二种选择，需要交替执行“CLR WDT1”和“CLR WDT2”两者才能成功的清除 WDT。关于第二种选择，如果“CLR WDT1”正被使用来清除 WDT，接着再执行这条指令将是无效的，只有执行“CLR WDT2”指令才能清除 WDT。同样的“CLR WDT2”指令已经执行后，只有接着执行“CLR WDT1”指令才可以清除看门狗定时器。

当设置分频比为 $2^{15}$ 时，溢出周期最大。例如，时钟源为 32.768kHz LXT 振荡器，分频比为 $2^{15}$ 时最大溢出周期约 1s，分频比为 $2^8$ 时最小溢出周期约 7.8ms。如果 $f_{SYS}/4$ 作为看门狗定时器时钟源，需要注意，当系统工作在休眠或空闲模式时，系统时钟停止工作，看门狗失去保护作用。如果系统工作在干扰大的环境中，强烈建议使用 $f_{SUB}$ 作为时钟源。




看门狗定时器

## 复位和初始化

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

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

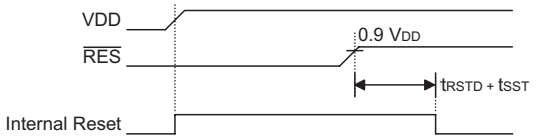
另一种复位为看门狗溢出单片机复位。不同方式的复位操作会对寄存器产生不同的影响。另一种复位为低电压复位即 LVR 复位，在电源供应电压低于 LVR 设定值时，系统会产生 LVR 复位，这种复位与与 $\overline{RES}$ 脚拉低复位方式相似。

## 复位功能

包括内部和外部事件触发复位，单片机共有五种复位方式：

### 上电复位

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




注：$t_{RSTD}$ 为上电延迟时间，典型值为 100ms

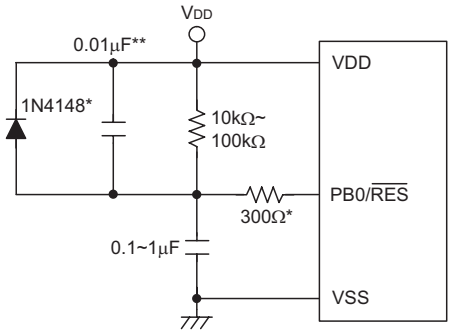
上电复位时序图

### RES 引脚复位

由于复位引脚与 PB.0 共用，复位功能必须使用配置选项选择。虽然单片机有一个内部 RC 复位功能，如果电源上升缓慢或上电时电源不稳定，内部 RC 振荡可能导致芯片复位不良，所以推荐使用和 RES 引脚连接的外部 RC 电路，由 RC 电路所造成的时间延迟使得 RES 引脚在电源供应稳定前的一段延长周期内保持在低电平。在这段时间内，单片机的正常操作是被禁止的。RES 引脚达到一定电压值后，再经过延迟时间 $t_{RSTD}$ 单片机可以开始进行正常操作。下图中 SST 是系统延迟周期 System Start-up Timer 的缩写。

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

当系统在较强干扰的场合工作时，建议使用增强型的复位电路，如下图所示。

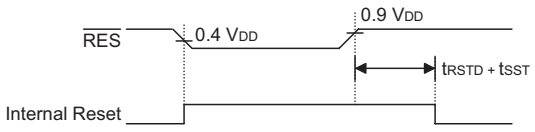



注：“*”表示建议加上此元件以加强静电保护。
“**”表示建议在电源有较强干扰场合加上此元件。

外部 RES 电路

欲知有关外部复位电路的更多信息可参考 HOLTEK 网站上的应用范例 HA0075S。

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




注：$t_{RSTD}$ 为上电延迟时间，典型值为 100ms。

**$\overline{\text{RES}}$ 复位时序图**

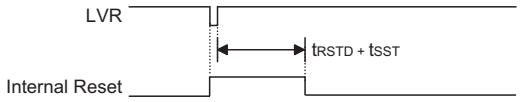
当复位脚由外部硬件强制拉低时，单片机大部分引脚也将强制设为一个高阻抗的状态。但需特别注意的是，当复位脚处于低电平状态时，PA5/C1X/SDO/AN5 和 PB2/OSC2 两引脚将强制设为逻辑输出低的状态。因此，在应用电路中遇到此情况时，建议不要将这两个引脚连接到低阻抗源，防止低阻抗源连接在一起。此种情况仅针对复位脚由外部硬件强制拉低的情况，其它复位情况时不予考虑。

| 引脚名称 | 引脚状态 |
|---|---|
| PA5/C1X/SDO/AN5 | 输出低 |
| PB2/OSC2 | 输出低 |
| 其它脚 | 高阻抗 |

**复位脚强制拉低时的引脚状态**

### 低电压复位 – LVR

单片机具有低电压复位电路，用来监测它的电源电压，可通过配置选项进行选择。例如在更换电池的情况下，单片机供应的电压可能会落在 0.9V~$V_{LVR}$ 的范围内，这时 LVR 将会自动复位单片机。LVR 包含以下的规格：有效的 LVR 信号，即在 0.9V~$V_{LVR}$ 的低电压状态的时间，必须超过交流电气特性中 $t_{LVR}$ 参数的值。如果低电压存在不超过 $t_{LVR}$ 参数的值，则 LVR 将会忽略它且不会执行复位功能。$V_{LVR}$ 参数值可通过配置选项进行设定。

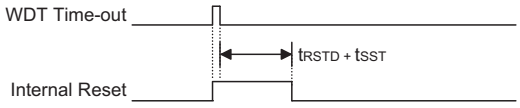



注：$t_{RSTD}$ 为上电延迟时间，典型值为 100ms。

**低电压复位时序图**

### 正常运行时看门狗溢出复位

除了看门狗溢出标志位 TO 将被设为“1”之外，正常运行时看门狗溢出复位和 $\overline{\text{RES}}$ 复位相同。

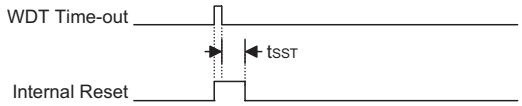



注：$t_{RSTD}$ 为上电延迟时间，典型值为 100ms。

**正常运行时看门狗溢出时序图**

### 休眠或空闲时看门狗溢出复位

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




注：如果系统时钟源为 ERC 或 HIRC 时，$t_{SST}$ 为 15~16 个时钟周期。
如果系统时钟源为 HXT 或 LXT，则 $t_{SST}$ 为 1024 个时钟周期。
如果系统时钟源为 LIRC，则 $t_{SST}$ 为 1~2 个时钟周期。

休眠或空闲时看门狗溢出复位时序图

## 复位初始状态

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

| T0 | PDF | 复位条件 |
|---|---|---|
| 0 | 0 | 上电时的 $\overline{RES}$ 复位 |
| u | u | 正常模式或低速模式时的 $\overline{RES}$ 复位或 LVR 复位 |
| 1 | u | 正常模式或低速模式时的 WDT 溢出复位 |
| 1 | 1 | 空闲或休眠模式时的 WDT 溢出复位 |

注：“u”代表不改变

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

| 项目 | 复位后情况 |
|---|---|
| 程序计数器 | 清除为零 |
| 中断 | 所有中断被除能 |
| 看门狗定时器 | WDT 清除并重新计数 |
| 定时 / 计数器 | 所有定时 / 计数器停止 |
| 输入 / 输出口 | 所有 I/O 设为输入模式 |
| 堆栈指针 | 堆栈指针指向堆栈顶端 |

不同的复位形式对单片机内部寄存器的影响是不同的。为保证复位后程序能正常执行，了解寄存器在特定条件复位后的设置是非常重要的。下表即为不同方式复位后内部寄存器的状况。若芯片有多种封装类型，表格反应较大的封装的情况。

**HT68F20**

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| MP0 | - xxx xxxx | - xxx xxxx | - xxx xxxx | - uuu uuuu |
| MP1 | - xxx xxxx | - xxx xxxx | - xxx xxxx | - uuu uuuu |
| BP | - - - - - - -0 | - - - - - - -0 | - - - - - - -0 | - - - - - - -u |
| ACC | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| PCL | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| TBLP | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBLH | - - xx xxxx | - - uu uuuu | - -uu uuuu | - - uu uuuu |
| TBHP | - - - - - -xx | - - - - - -uu | - - - - - -uu | - - - - - - uu |
| STATUS | - -00 xxxx | - -uu uuuu | - - 1u uuuu | - - 11 uuuu |
| SMOD | 0000 0011 | 0000 0011 | 0000 0011 | uuuu uuuu |
| LVDC | - - 00 - 000 | - - 00 - 000 | - - 00 - 000 | - - uu - uuu |
| INTEG | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| WDTC | 0111 1010 | 0111 1010 | 0111 1010 | uuuu uuuu |
| TBC | 0011 0111 | 0011 0111 | 0011 0111 | uuuu uuuu |
| INTC0 | - 000 0000 | - 000 0000 | - 000 0000 | - uuu uuuu |
| INTC1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI0 | - -00 - -00 | - -00 - -00 | - -00 - -00 | - -uu - -uu |
| MFI1 | - -00 - -00 | - -00 - -00 | - -00 - -00 | - -uu - -uu |
| MFI2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAWU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PA | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PAC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBPU | - - 00 0000 | - - 00 0000 | - - 00 0000 | - - uu uuuu |
| PB | - - 11 1111 | - - 11 1111 | - - 11 1111 | - - uu uuuu |
| PBC | - - 11 1111 | - - 11 1111 | - - 11 1111 | - - uu uuuu |
| PCPU | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| PC | - - - - 1111 | - - - - 1111 | - - - - 1111 | - - - - uuuu |
| PCC | - - - - 1111 | - - - - 1111 | - - - - 1111 | - - - - uuuu |
| CP0C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| CP1C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| SIMC0 | 1110 000 - | 1110 000 - | 1110 000 - | uuuu uuu - |
| SIMC1 | 1000 0001 | 1000 0001 | 1000 0001 | uuuu uuuu |
| SIMD | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SIMA/SIMC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| TM0AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| EEA | - - - x xxxx | - - - x xxxx | - - - x xxxx | - - - 0 0000 |
| EED | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EEC | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| TMPC0 | - - 01 - - - 1 | - - 01 - - - 1 | - - 01 - - - 1 | - - uu - - -u |
| TM1C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| SCOMC | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

注：“u”表示不改变；“x”表示未知；“-”表示未定义

**HT68F30**

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| MP0 | - xxx xxxx | - xxx xxxx | - xxx xxxx | - uuu uuuu |
| MP1 | - xxx xxxx | - xxx xxxx | - xxx xxxx | - uuu uuuu |
| BP | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| ACC | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| PCL | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| TBLP | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBLH | - - xx xxxx | - - uu uuuu | - -uu uuuu | - - uu uuuu |
| TBHP | - - - - - xxx | - - - - - uuu | - - - - - uuu | - - - - - uuu |
| STATUS | - -00 xxxx | - -uu uuuu | - - 1u uuuu | - - 11 uuuu |
| SMOD | 0000 0011 | 0000 0011 | 0000 0011 | uuuu uuuu |
| LVDC | - - 00 - 000 | - - 00 - 000 | - - 00 - 000 | - - uu - uuu |
| INTEG | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| WDTC | 0111 1010 | 0111 1010 | 0111 1010 | uuuu uuuu |
| TBC | 0011 0111 | 0011 0111 | 0011 0111 | uuuu uuuu |
| INTC0 | - 000 0000 | - 000 0000 | - 000 0000 | - uuu uuuu |
| INTC1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI0 | - -00 - -00 | - -00 - -00 | - -00 - -00 | - -uu - -uu |
| MFI1 | - 000 - 000 | - 000 - 000 | - 000 -000 | - uuu - uuu |
| MFI2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAWU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| PAPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PA | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PAC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBPU | - - 00 0000 | - - 00 0000 | - - 00 0000 | - - uu uuuu |
| PB | - - 11 1111 | - - 11 1111 | - - 11 1111 | - - uu uuuu |
| PBC | - - 11 1111 | - - 11 1111 | - - 11 1111 | - - uu uuuu |
| PCPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| CP0C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| CP1C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| SIMC0 | 1110 000 - | 1110 000 - | 1110 000 - | uuuu uuu - |
| SIMC1 | 1000 0001 | 1000 0001 | 1000 0001 | uuuu uuuu |
| SIMD | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SIMA/SIMC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM0AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| EEA | - - xx xxxx | - - xx xxxx | - - xx xxxx | - - uu uuuu |
| EED | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EEC | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| TMPC0 | 1 - 01 - - 01 | 1 - 01 - - 01 | 1 - 01 - - 01 | u - uu - - uu |
| PRM0 | - - - - - 000 | - - - - - 000 | - - - - - 000 | - - - - - uuu |
| TM1C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1BL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1BH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| SCOMC | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

注：“u”表示不改变；“x”表示未知；“-”表示未定义

**HT68F40**

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| MP0 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| MP1 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| BP | - - - - - - -0 | - - - - - - -0 | - - - - - - -0 | - - - - - - -u |
| ACC | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| PCL | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| TBLP | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBLH | - xxx xxxx | - uuu uuuu | - uuu uuuu | - uuu uuuu |
| TBHP | - - - - xxxx | - - - - uuuu | - - - - uuuu | - - - - uuuu |
| STATUS | - -00 xxxx | - -uu uuuu | - - 1u uuuu | - - 11 uuuu |
| SMOD | 0000 0011 | 0000 0011 | 0000 0011 | uuuu uuuu |
| LVDC | - - 00 - 000 | - - 00 - 000 | - - 00 - 000 | - - uu - uuu |
| INTEG | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| WDTC | 0111 1010 | 0111 1010 | 0111 1010 | uuuu uuuu |
| TBC | 0011 0111 | 0011 0111 | 0011 0111 | uuuu uuuu |
| INTC0 | - 000 0000 | - 000 0000 | - 000 0000 | - uuu uuuu |
| INTC1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI1 | - 000 - 000 | - 000 - 000 | - 000 - 000 | - uuu - uuu |
| MFI2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAWU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PA | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PAC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PB | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PD | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PE | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PFPU | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| PF | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |
| PFC | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| CP0C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| CP1C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| SIMC0 | 1110 000 - | 1110 000 - | 1110 000 - | uuuu uuu - |
| SIMC1 | 1000 0001 | 1000 0001 | 1000 0001 | uuuu uuuu |
| SIMD | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SIMA/SIMC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM0AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| EEA | - xxx xxxx | - xxx xxxx | - xxx xxxx | - uuu uuuu |
| EED | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EEC | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| TMPC0 | 1001 - - 01 | 1001 - - 01 | 1001 - - 01 | uuuu - - uu |
| TMPC1 | - - - - - - 01 | - - - - - - 01 | - - - - - - 01 | - - - - - - uu |
| PRM0 | - 0- 0 0000 | - 0- 0 0000 | - 0- 0 0000 | - u- u uuuu |
| PRM1 | 000 - 0000 | 000 - 0000 | 000 - 0000 | uuu - uuuu |
| PRM2 | - -00 0000 | - -00 0000 | - -00 0000 | - -uu uuuu |
| TM1C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1BL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1BH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM2C0 | 0000 0 - - - | 0000 0 - - - | 0000 0 - - - | uuuu u - - - |
| TM2C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2RP | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| SCOMC | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

注：“u”表示不改变；“x”表示未知；“-”表示未定义

**HT68F50**

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| MP0 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| MP1 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| BP | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| ACC | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| PCL | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| TBLP | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBLH | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBHP | - - -x xxxx | - - -u uuuu | - - -u uuuu | - - -u uuuu |
| STATUS | - -00 xxxx | - -uu uuuu | - - 1u uuuu | - - 11 uuuu |
| SMOD | 0000 0011 | 0000 0011 | 0000 0011 | uuuu uuuu |
| LVDC | - - 00 - 000 | - - 00 - 000 | - - 00 - 000 | - - uu - uuu |
| INTEG | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| WDTC | 0111 1010 | 0111 1010 | 0111 1010 | uuuu uuuu |
| TBC | 0011 0111 | 0011 0111 | 0011 0111 | uuuu uuuu |
| INTC0 | - 000 0000 | - 000 0000 | - 000 0000 | - uuu uuuu |
| INTC1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI1 | - 000 - 000 | - 000 - 000 | - 000 - 000 | - uuu - uuu |
| MFI2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI3 | - -00 - -00 | - -00 - -00 | - -00 - -00 | - -uu - -uu |
| PAWU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PA | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PAC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PB | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PD | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PE | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PFPU | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| PF | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |

| 寄存器 | 上电复位 | RES 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| PFC | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |
| CP0C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| CP1C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u - -u |
| SIMC0 | 1110 000 - | 1110 000 - | 1110 000 - | uuuu uuu - |
| SIMC1 | 1000 0001 | 1000 0001 | 1000 0001 | uuuu uuuu |
| SIMD | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SIMA/SIMC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM0AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| EEA | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EED | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EEC | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| TMPC0 | 1001 - - 01 | 1001 - - 01 | 1001 - - 01 | uuuu - - uu |
| TMPC1 | - - 01 - - 01 | - - 01 - - 01 | - - 01 - - 01 | - - uu - - uu |
| PRM0 | - 0- 0 0000 | - 0- 0 0000 | - 0- 0 0000 | - u- u uuuu |
| PRM1 | 000 - 0000 | 000 - 0000 | 000 - 0000 | uuu - uuuu |
| PRM2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1BL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1BH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM2C0 | 0000 0- - - | 0000 0- - - | 0000 0- - - | uuuu u- - - |
| TM2C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2RP | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3DH | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| TM3AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3AH | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| SCOMC | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

注： “u”表示不改变； “x”表示未知； “-”表示未定义

**HT68F60**

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| MP0 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| MP1 | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| BP | - - 0- - 000 | - - 0- - 000 | - - 0- - 000 | - - u- - uuu |
| ACC | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| PCL | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| TBLP | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBLH | xxxx xxxx | uuuu uuuu | uuuu uuuu | uuuu uuuu |
| TBHP | - - xx xxxx | - - uu uuuu | - - uu uuuu | - - uu uuuu |
| STATUS | - -00 xxxx | - -uu uuuu | - - 1u uuuu | - - 11 uuuu |
| SMOD | 0000 0011 | 0000 0011 | 0000 0011 | uuuu uuuu |
| LVDC | - - 00 - 000 | - - 00 - 000 | - - 00 - 000 | - - uu - uuu |
| INTEG | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| WDTC | 0111 1010 | 0111 1010 | 0111 1010 | uuuu uuuu |
| TBC | 0011 0111 | 0011 0111 | 0011 0111 | uuuu uuuu |
| INTC0 | - 000 0000 | - 000 0000 | - 000 0000 | - uuu uuuu |
| INTC1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| INTC3 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI1 | - 000 - 000 | - 000 - 000 | - 000 - 000 | - uuu - uuu |
| MFI2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| MFI3 | - -00 - -00 | - -00 - -00 | - -00 - -00 | - -uu - -uu |
| PAWU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PAPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PA | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PAC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PB | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PBC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PCC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| PD | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PDC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PE | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PEC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PFPU | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PF | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PFC | 1111 1111 | 1111 1111 | 1111 1111 | uuuu uuuu |
| PGPU | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| PG | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |
| PGC | - - - - - -11 | - - - - - -11 | - - - - - -11 | - - - - - -uu |
| CP0C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u- -u |
| CP1C | 1000 0 - -1 | 1000 0 - -1 | 1000 0 - -1 | uuuu u- -u |
| SIMC0 | 1110 000 - | 1110 000 - | 1110 000 - | uuuu uuu - |
| SIMC1 | 1000 0001 | 1000 0001 | 1000 0001 | uuuu uuuu |
| SIMD | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| SIMA/SIMC2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM0AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM0AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| EEA | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EED | xxxx xxxx | xxxx xxxx | xxxx xxxx | uuuu uuuu |
| EEC | - - - - 0000 | - - - - 0000 | - - - - 0000 | - - - - uuuu |
| TMPC0 | 1001 - - 01 | 1001 - - 01 | 1001 - - 01 | uuuu - - uu |
| TMPC1 | - - 01 - - 01 | - - 01 - - 01 | - - 01 - - 01 | - - uu - - uu |
| PRM0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PRM1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PRM2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1C2 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1DH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1AH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM1BL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM1BH | - - - - - - 00 | - - - - - - 00 | - - - - - - 00 | - - - - - - uu |
| TM2C0 | 0000 0- - - | 0000 0- - - | 0000 0- - - | uuuu u- - - |

| 寄存器 | 上电复位 | $\overline{\text{RES}}$ 或 LVR 复位 | WDT 溢出 ( 正常模式 ) | WDT 溢出 ( 空闲模式 ) |
|---|---|---|---|---|
| TM2C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2DH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2AH | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM2RP | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3C0 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3C1 | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3DL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3DH | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| TM3AL | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TM3AH | - - - - - -00 | - - - - - -00 | - - - - - -00 | - - - - - -uu |
| SCOMC | 0000 0000 | 0000 0000 | 0000 0000 | uuuu uuuu |

注：“u”表示不改变；“x”表示未知；“-”表示未定义

## 输入 / 输出端口

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

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

### 输入 / 输出寄存器列表

- **HT68F20**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAWU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PA | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBPU | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PB | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PBC | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PCPU | — | — | — | — | D3 | D2 | D1 | D0 |
| PC | — | — | — | — | D3 | D2 | D1 | D0 |
| PCC | — | — | — | — | D3 | D2 | D1 | D0 |

- **HT68F30**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAWU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PA | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBPU | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PB | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PBC | — | — | D5 | D4 | D3 | D2 | D1 | D0 |
| PCPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PCC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

- **HT68F40/HT68F50**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAWU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PA | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PB | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PCPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PCC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PDPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PD | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PDC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PEPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PE | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PEC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PFPU | — | — | — | — | — | — | D1 | D0 |
| PF | — | — | — | — | — | — | D1 | D0 |
| PFC | — | — | — | — | — | — | D1 | D0 |

- **HT68F60**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAWU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PA | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PB | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PBC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PCPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PCC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PDPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PD | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PDC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PEPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PE | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PEC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PFPU | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PF | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PFC | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PGPU | — | — | — | — | — | — | D1 | D0 |
| PG | — | — | — | — | — | — | D1 | D0 |
| PGC | — | — | — | — | — | — | D1 | D0 |

## 上拉电阻

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

**PAPU 寄存器**

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

**PBPU 寄存器**

- **HT68F40/HT68F50/HT68F60**

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

### PCPU 寄存器

- **HT68F30/HT68F40/HT68F50/HT68F60**

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

### PDPU 寄存器

- **HT68F40/HT68F50/HT68F60**

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

### PEPU 寄存器

- **HT68F40/HT68F50/HT68F60**

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

### PFPU 寄存器

- **HT68F60**

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

Bit 7~0 **PFPU**: I/O 口 bit 7~bit 0 上拉电阻控制位
0：除能
1：使能

### PBPU 寄存器

- **HT68F20/HT68F30**

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

Bit 7~6 未定义，读为“0”

Bit 5~0 **PBPU**：PB 口 bit 5~bit 0 上拉电阻控制位
0：除能
1：使能

### PCPU 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | D3 | D2 | D1 | D0 |
| R/W | — | — | — | — | R/W | R/W | R/W | R/W |
| POR | — | — | — | — | 0 | 0 | 0 | 0 |

Bit 7~4　未定义，读为“0”

Bit 3~0　PCPU：PC 口 bit 3~bit 0 上拉电阻控制位
0：除能
1：使能

### PFPU 寄存器

- **HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | D1 | D0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 0 |

Bit 7~2　未定义，读为“0”

Bit 1~0　**PFPU**：PF 口 bit 1~bit 0 上拉电阻控制位
0：除能
1：使能

### PGPU 寄存器

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | D1 | D0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 0 |

Bit 7~2　未定义，读为“0”

Bit 1~0　**PGPU**：PG 口 bit 1~bit 0 上拉电阻控制位
0：除能
1：使能

## PA 口唤醒

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

### PAWU 寄存器

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

Bit 7~0　**PAWU**：PA 口 bit 7~bit 0 唤醒功能控制位
0：除能
1：使能

## 输入 / 输出端口控制寄存器

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

**PAC 寄存器**

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

**PBC 寄存器**

- **HT68F40/HT68F50/HT68F60**

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

**PCC 寄存器**

- **HT68F30/HT68F40/HT68F50/HT68F60**

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

**PDC 寄存器**

- **HT68F40/HT68F50/HT68F60**

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

**PEC 寄存器**

- **HT68F40/HT68F50/HT68F60**

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

**PFC 寄存器**

- **HT68F60**

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

Bit 7~0 **PFC**：I/O 口 bit 7~bit 0 输入 / 输出控制位
0：输出
1：输入

## PBC 寄存器

- **HT68F20/HT68F30**

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

Bit 7~6 未定义，读为“0”

Bit 5~0 PBC：PB 口 bit 5~bit 0 输入 / 输出控制位
0：输出
1：输入

## PCC 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | D3 | D2 | D1 | D0 |
| R/W | — | — | — | — | R/W | R/W | R/W | R/W |
| POR | — | — | — | — | 0 | 0 | 0 | 0 |

Bit 7~4 未定义，读为“0”

Bit 3~0 PCC：PC 口 bit 3~bit 0 输入 / 输出控制位
0：输出
1：输入

### PFC 寄存器

- **HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | D1 | D0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 0 |

Bit 7~2　未定义，读为“0”

Bit 1~0　**PFC**：PF 口 bit 1~bit 0 输入 / 输出控制位
0：输出
1：输入

### PGC 寄存器

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | D1 | D0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 0 |

Bit 7~2　未定义，读为“0”

Bit 1~0　**PGC**：PG 口 bit 1~bit 0 输入 / 输出控制位
0：输出
1：输入

## 引脚重置功能

引脚的多功能可以增加单片机应用的灵活性。有限的引脚个数将会限制设计者，而引脚的多功能将会解决很多此类问题。每个功能可单独选择所在的引脚，以及一个确定的优先级，使得引脚上多种功能可以同时使用。此外，一些引脚功能可以通过寄存器 PRM0，PRM1 和 PRM2 进行设定。

## 引脚重置寄存器

封装中有限的引脚个数会对某些单片机功能造成影响。然而，引脚功能重置和引脚功能选择，使得小封装单片机具有更多不同的功能。一些单片机的特定引脚功能可以通过寄存器 PRM0，PRM1 或 PRM2 进行设定。

### 引脚重置寄存器列表

- **HT68F30**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRM0 | — | — | — | — | — | PCPRM | SIMPS0 | PCKPS |

- **HT68F40**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRM0 | — | C1XPS0 | — | C0XPS0 | PDPRM | SIMPS1 | SIMPS0 | PCKPS |
| PRM1 | TCK2PS | TCK1PS | TCK0PS | — | INT1PS1 | INT1PS0 | INT0PS1 | INT0PS0 |
| PRM2 | — | — | TP21PS | TP20PS | TP1B2PS | TP1APS | TP01PS | TP00PS |

- **HT68F50**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRM0 | — | C1XPS0 | — | C0XPS0 | PDPRM | SIMPS1 | SIMPS0 | PCKPS |
| PRM1 | TCK2PS | TCK1PS | TCK0PS | — | INT1PS1 | INT1PS0 | INT0PS1 | INT0PS0 |
| PRM2 | TP31PS | TP30PS | TP21PS | TP20PS | TP1B2PS | TP1APS | TP01PS | TP00PS |

- **HT68F60**

| 寄存器名称 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRM0 | C1XPS1 | C1XPS0 | C0XPS1 | C0XPS0 | PDPRM | SIMPS1 | SIMPS0 | PCKPS |
| PRM1 | TCK2PS | TCK1PS | TCK0PS | INT2PS1 | INT1PS1 | INT1PS0 | INT0PS1 | INT0PS0 |
| PRM2 | TP31PS | TP30PS | TP21PS | TP20PS | TP1B2PS | TP1APS | TP01PS | TP00PS |

**PRM0 寄存器**

- **HT68F30**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | PCPRM | SIMPS0 | PCKPS |
| R/W | — | — | — | — | — | R/W | R/W | R/W |
| POR | — | — | — | — | — | 0 | 0 | 0 |

Bit 7~3　未定义，读为“0”

Bit 2　**PCPRM**：PC1~PC0 引脚功能重置控制位
0：不变
1：如果 SIMPS0=1 时，TP1B_0 由 PC0 更改为 PA6，TP1B_1 由 PC1 更改为 PA7

Bit 1　**SIMPS0**：SIM 引脚重置控制位
0：SDO on PA5；SDI/SDA on PA6；SCK/SCL on PA7；$\overline{\text{SCS}}$ on PB5
1：SDO on PC1；SDI/SDA on PC0；SCK/SCL on PC7；SCS on PC6

Bit 0　**PCKPS**：PCK 和 $\overline{\text{PINT}}$ 引脚重置控制位
0：PCK on PC2；$\overline{\text{PINT}}$ on PC3
1：PCK on PC5；PINT on PC4

**PRM0 寄存器**

- **HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | C1XPS0 | — | C0XPS0 | PDPRM | SIMPS1 | SIMPS0 | PCKPS |
| R/W | — | R/W | — | R/W | R/W | R/W | R/W | R/W |
| POR | — | 0 | — | 0 | 0 | 0 | 0 | 0 |

Bit 7　未定义，读为“0”

Bit 6　**C1XPS0**：C1X 引脚重置控制位
0：C1X on PA5
1：C1X on PF1

Bit 5　未定义，读为“0”

Bit 4　**C0XPS0**：C0X 引脚重置控制位
0：C0X on PA0
1：C0X on PF0

Bit 3 **PDPRM**：PD3~PD0 引脚功能重置控制位
0：不变
1：如果 SIMPS1，SIMPS0=01 时，TCK2 由 PD0 更改为 PB6，TP2_0 由 PD1 更改为 PB7，TCK0 由 PD2 更改为 PD6，TCK1 由 PD3 更改为 PD7

Bit 2~1 **SIMPS1**，SIMPS0：SIM 引脚重置控制位
00：SDO on PA5；SDI/SDA on PA6；SCK/SCL on PA7；$\overline{\text{SCS}}$ on PB5
01：SDO on PD3；SDI/SDA on PD2；SCK/SCL on PD1；$\overline{\text{SCS}}$ on PD0
10：SDO on PB6；SDI/SDA on PB7；SCK/SCL on PD6；$\overline{\text{SCS}}$ on PD7
11：未定义

Bit 0 **PCKPS**：PCK 和 $\overline{\text{PINT}}$ 引脚重置控制位
0：PCK on PC2；$\overline{\text{PINT}}$ on PC3
1：PCK on PC5；$\overline{\text{PINT}}$ on PC4

## PRM0 寄存器

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | C1XPS1 | C1XPS0 | C0XPS1 | C0XPS0 | PDPRM | SIMPS1 | SIMPS0 | PCKPS |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **C1XPS1,C1XPS0**：C1X 引脚重置控制位
00：C1X on PA5
01：C1X on PF1
10：C1X on PG1
11：未定义

Bit 5~4 **C0XPS1,C0XPS0**：C0X 引脚重置控制位
00：C0X on PA0
01：C0X on PF0
10：C0X on PG0
11：未定义

Bit 3 **PDPRM**：PD3~PD0 引脚功能重置控制位
0：不变
1：如果 SIMPS1，SIMPS0=01 或 11 时，TCK2 由 PD0 更改为 PB6，TP2_0 由 PD1 更改为 PB7，TCK0 由 PD2 更改为 PD6，TCK1 由 PD3 更改为 PD7

Bit 2~1 **SIMPS1,SIMPS0**：SIM 引脚重置控制位
00：SDO on PA5；SDI/SDA on PA6；SCK/SCL on PA7；$\overline{\text{SCS}}$ on PB5
01：SDO on PD3；SDI/SDA on PD2；SCK/SCL on PD1；$\overline{\text{SCS}}$ on PD0
10：SDO on PB6；SDI/SDA on PB7；SCK/SCL on PD6；$\overline{\text{SCS}}$ on PD7
11：SDO on PD1；SDI/SDA on PD2；SCK/SCL on PD3；SCS on PD0

Bit 0 **PCKPS**：PCK 和 $\overline{\text{PINT}}$ 引脚重置控制位
0：PCK on PC2；$\overline{\text{PINT}}$ on PC3
1：PCK on PC5；$\overline{\text{PINT}}$ on PC4

## PRM1 寄存器

- **HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TCK2PS | TCK1PS | TCK0PS | — | INT1PS1 | INT1PS0 | INT0PS1 | INT0PS0 |
| R/W | R/W | R/W | R/W | — | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | — | 0 | 0 | 0 | 0 |

Bit 7 **TCK2PS**：TCK2 引脚重置控制位
0：TCK2 on PC2
1：TCK2 on PD0

Bit 6 **TCK1PS**：TCK1 引脚重置控制位
0：TCK1 on PA4
1：TCK1 on PD3

Bit 5 **TCK0PS**：TCK0 引脚重置控制位
0：TCK0 on PA2
1：TCK0 on PD2

Bit 4 未定义，读为“0”

Bit 3~2 **INT1PS1,INT1PS0**：INT1 引脚重置控制位
00：INT1 on PA4
01：INT1 on PC5
10：未定义
11：INT1 on PE7

Bit 1~0 **INT0PS1,INT0PS0**：INT0 引脚功能重置控制位
00：INT0 on PA3
01：INT0 on PC4
10：未定义
11：INT0 on PE6

### PRM1 寄存器

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TCK2PS | TCK1PS | TCK0PS | INT2PS | INT1PS1 | INT1PS0 | INT0PS1 | INT0PS0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **TCK2PS**：TCK2 引脚重置控制位
0：TCK2 on PC2
1：TCK2 on PD0

Bit 6 **TCK1PS**：TCK1 引脚重置控制位
0：TCK1 on PA4
1：TCK1 on PD3

Bit 5 **TCK0PS**：TCK0 引脚重置控制位
0：TCK0 on PA2
1：TCK0 on PD2

Bit 4 **INT2PS**：INT2 引脚重置控制位
0：INT2 on PC4
1：INT2 on PE2

Bit 3~2 **INT1PS1,INT1PS0**：INT1 引脚重置控制位
00：INT1 on PA4
01：INT1 on PC5
10：INT1 on PE1
11：INT1 on PE7

Bit 1~0 **INT0PS1,INT0PS0**：INT0 引脚功能重置控制位
00：INT0 on PA3
01：INT0 on PC4
10：INT0 on PE0
11：INT0 on PE6

### PRM2 寄存器

- **HT68F40**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | TP21PS | TP20PS | TP1B2PS | TP1APS | TP01PS | TP00PS |
| R/W | — | — | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | — | — | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 未定义，读为“0”

Bit 5 **TP21PS**：TP2_1 引脚重置控制位
0：TP2_1 on PC4
1：TP2_1 on PD4

Bit 4 **TP20PS**：TP2_0 引脚重置控制位
0：TP2_0 on PC3
1：TP2_0 on PD1

Bit 3 **TP1B2PS**：TP1B_2 引脚重置控制位
0：TP1B_2 on PC5
1：TP1B_2 on PE4

Bit 2 **TP1APS**：TP1A 引脚重置控制位
0：TP1A on PA1
1：TP1A on PC7

Bit 1 **TP01PS**：TP0_1 引脚重置控制位
0：TP0_1 on PC5
1：TP0_1 on PD5

Bit 0 **TP00PS**：TP0_0 引脚重置控制位
0：TP0_0 on PA0
1：TP0_0 on PC6

### PRM2 寄存器

- **HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TP31PS | TP30PS | TP21PS | TP20PS | TP1B2PS | TP1APS | TP01PS | TP00PS |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **TP31PS**：TP3_1 引脚重置控制位
0：TP3_1 on PD0
1：TP3_1 on PE3

Bit 6 **TP30PS**：TP3_0 引脚重置控制位
0：TP3_0 on PD3
1：TP3_0 on PE5

Bit 5 **TP21PS**：TP2_1 引脚重置控制位
0：TP2_1 on PC4
1：TP2_1 on PD4

Bit 4 **TP20PS**：TP2_0 引脚重置控制位
0：TP2_0 on PC3
1：TP2_0 on PD1

Bit 3 **TP1B2PS**：TP1B_2 引脚重置控制位
0：TP1B_2 on PC5
1：TP1B_2 on PE4

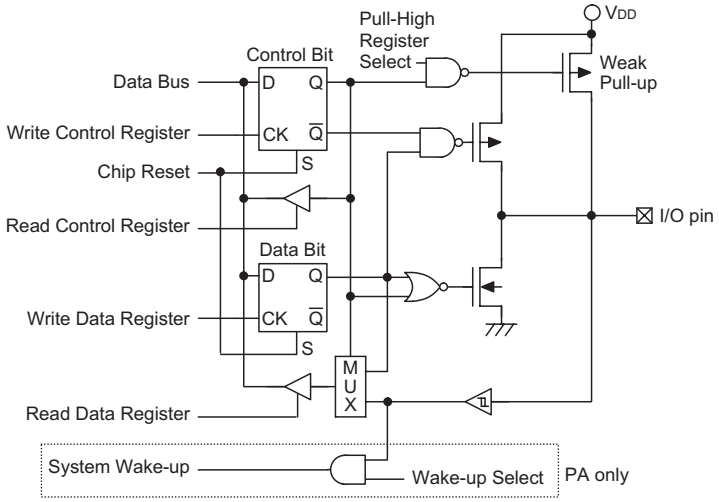
Bit 2 **TP1APS**：TP1A 引脚重置控制位
0：TP1A on PA1
1：TP1A on PC7

Bit 1 **TP01PS**：TP0_1 引脚重置控制位
0：TP0_1 on PC5
1：TP0_1 on PD5

Bit 0 **TP00PS**：TP0_0 引脚重置控制位
0：TP0_0 on PA0
1：TP0_0 on PC6

## 输入 / 输出引脚结构

下图为输入 / 输出引脚的内部结构图。输入 / 输出引脚的准确逻辑结构图可能与此图不同，这里只是为了方便对 I/O 引脚功能的理解提供的一个参考。图中的引脚共用结构并非针对所有单片机。




通用输入 / 输出端口

## 编程注意事项

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

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

# 定时器模块 – TM

控制和测量时间在任何单片机中都是一个很重要的部分。每个单片机提供几个定时器模块 ( 简称 TM)，来实现和时间有关的功能。定时器模块是包括多种操作的定时单元，提供的操作有：定时 / 事件计数器，捕捉输入，比较匹配输出，单脉冲输出以及 PWM 输出等功能。每个定时器模块有两个或三个独立中断。每个 TM 外加的输入输出引脚，扩大了定时器的灵活性，便于用户使用。

这里只介绍各种 TM 的共性，更多详细资料请参考简易型，标准型和增强型定时器章节。

## 简介

该系列单片机包含 2 到 4 个 TM，取决于所选单片机的型号，分别命名为 TM0，TM1，TM2 和 TM3。每个 TM 可被划分为一个特定的类型，即简易型 TM，标准型 TM 或增强型 TM。虽然性质相似，但不同 TM 特性复杂度不同。本章介绍简易型，标准型和增强型 TM 的共性，更多详细资料分别见后面各章。三种类型 TM 的特性和区别见下表。

| 功能 | CTM | STM | ETM |
|---|---|---|---|
| 定时 / 计数器 | √ | √ | √ |
| 捕捉输入 | — | √ | √ |
| 比较匹配输出 | √ | √ | √ |
| PWM 通道数 | 1 | 1 | 2 |
| 单脉冲输出 | — | 1 | 1 |
| PWM 对齐方式 | 边沿对齐 | 边沿对齐 | 边沿 & 中心对齐 |
| PWM 调节周期 & 占空比 | 占空比或周期 | 占空比或周期 | 占空比或周期 |

**TM 功能概要**

该系列每款单片机包括一定数目的定时器单元，其中有简易型，标准型和增强型 TM，依次命名为 TM0~TM3 并见下表。

| 单片机 | TM0 | TM1 | TM2 | TM3 |
|---|---|---|---|---|
| HT68F20 | 10-bit CTM | 10-bit STM | — | — |
| HT68F30 | 10-bit CTM | 10-bit ETM | — | — |
| HT68F40 | 10-bit CTM | 10-bit ETM | 16-bit STM | — |
| HT68F50 | 10-bit CTM | 10-bit ETM | 16-bit STM | 10-bit CTM |
| HT68F60 | 10-bit CTM | 10-bit ETM | 16-bit STM | 10-bit CTM |

**TM 名称 / 类型参考**

## TM 操作

三种不同类型的 TM 提供从简单的定时操作到 PWM 信号产生等多种功能。理解 TM 操作的关键是比较 TM 内独立运行的计数器的值与内部比较器的预置值。当计数器的值与比较器的预置值相同时，则比较匹配，TM 中断信号产生，清零计数器并改变 TM 输出引脚的状态。用户选择内部时钟或外部时钟来驱动内部 TM 计数器。

## TM 时钟源

驱动 TM 计数器的时钟源很多。通过设置 TM 控制寄存器的 TnCK2~TnCK0 位，选择所需的时钟源。该时钟源来自系统时钟 $f_{SYS}$ 或内部高速时钟 $f_H$ 或 $f_{TBC}$ 时钟源或外部 TCKn 引脚时钟的分频比。注意：设置 TnCK2~TnCK0 为 101，将选择 TM 预设时钟输入，有效切断 TM 时钟源。TCKn 引脚时钟源用于允许外部信号作为 TM 时钟源或用于事件计数。

## TM 中断

简易型和标准型 TM 都拥有两个内部中断，分别是内部比较器 A 或比较器 P，当比较匹配发生时产生 TM 中断。增强型 TM 有三个内部比较器，即比较器 A 或比较器 B 或比较器 P，相应的有三个内部中断。当 TM 中断产生时，计数器清零并改变 TM 输出引脚的状态。

## TM 外部引脚

无论哪种类型的 TM，都有一个 TM 输入引脚 TCKn。通过设置 TMnC0 寄存器中的 TnCK2~TnCK0 位，选择 TM 功能并将该引脚作为 TM 时钟源输入脚。外部时钟源可通过该引脚来驱动内部 TM。外部 TM 输入脚也与其它功能共用，但是，如果设置适当值给 TnCK2~TnCK0，该引脚会连接到内部 TM。TM 引脚可选择上升沿有效或下降沿有效。

每个 TM 有一个或多个输出引脚 TPn。当 TM 工作在比较匹配输出模式且比较匹配发生时，这些引脚会由 TM 控制切换到高电平或低电平或翻转。外部 TPn 输出引脚也被 TM 用来产生 PWM 输出波形。当 TM 输出引脚与其它功能共用时，TM 输出功能需要通过寄存器先被设置。寄存器中的一个单独位用于决定其相关引脚用于外部 TM 输出还是用于其它功能。每个单片机和不同类型 TM 中输出引脚的个数是不同的，详见下表。

| 单片机 | CTM | STM | ETM | 寄存器 |
|---|---|---|---|---|
| HT68F20 | TP0_0 | TP1_0，TP1_1 | — | TMPC0 |
| HT68F30 | TP0_0，TP0_1 | — | TP1A，TP1B_0，TP1B_1 | TMPC0 |
| HT68F40 | TP0_0，TP0_1 | TP2_0，TP2_1 | TP1A，TP1B_0，TP1B_1，TP1B_2 | TMPC0，TMPC1 |
| HT68F50 | TP0_0，TP0_1<br>TP3_0，TP3_1 | TP2_0，TP2_1 | TP1A，TP1B_0，TP1B_1，TP1B_2 | TMPC0，TMPC1 |
| HT68F60 | TP0_0，TP0_1<br>TP3_0，TP3_1 | TP2_0，TP2_1 | TP1A，TP1B_0，TP1B_1，TP1B_2 | TMPC0，TMPC1 |

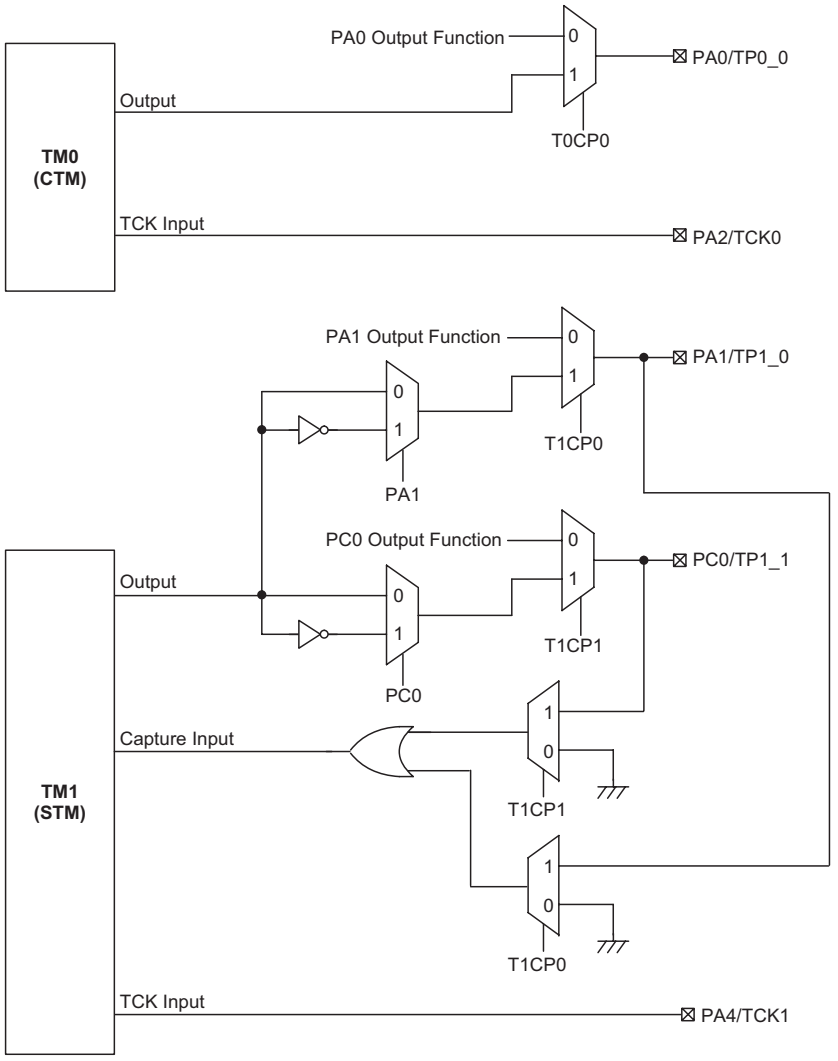
**TM 输出引脚**

## TM 输入 / 输出引脚控制寄存器

通过设置一或两个与 TM 输入 / 输出引脚相关的寄存器的一位，选择作为 TM 输入 / 输出功能或其它共用功能。设定为高时，相关引脚用作 TM 输入 / 输出，清零时将保持原来的功能。

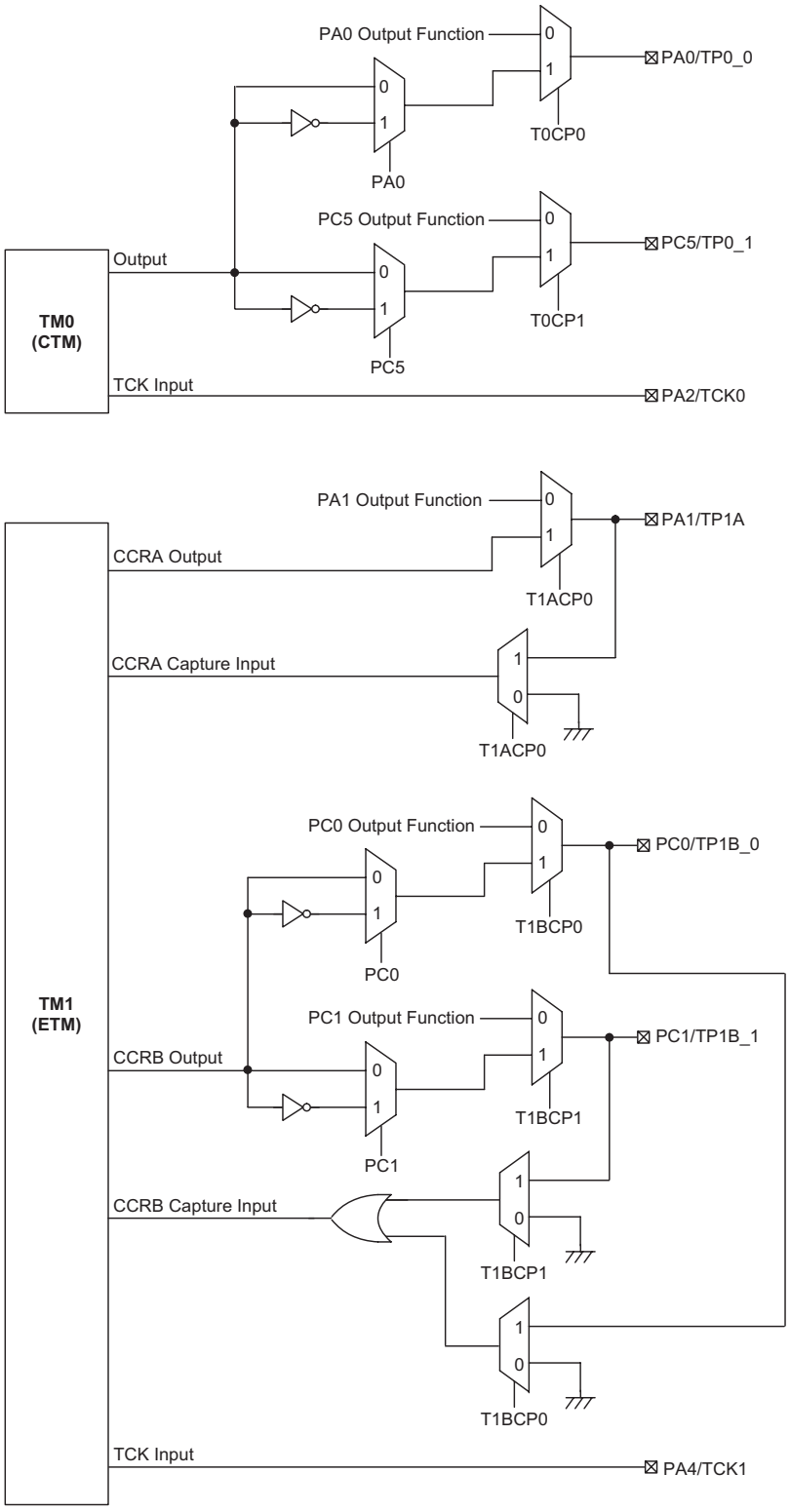
| 寄存器 | 单片机 | 位 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TMPC0 | HT68F20 | — | — | T1CP1 | T1CP0 | — | — | — | T0CP0 |
| TMPC0 | HT68F30 | T1ACP0 | — | T1BCP1 | T1BCP0 | — | — | T0CP1 | T0CP0 |
| TMPC0 | HT68F40<br>HT68F50<br>HT68F60 | T1ACP0 | T1BCP2 | T1BCP1 | T1BCP0 | — | — | T0CP1 | T0CP0 |
| TMPC1 | HT68F40 | — | — | — | — | — | — | T2CP1 | T2CP0 |
| TMPC1 | HT68F50<br>HT68F60 | — | — | T3CP1 | T3CP0 | — | — | T2CP1 | T2CP0 |

**TM 输入 / 输出引脚控制寄存器列表**



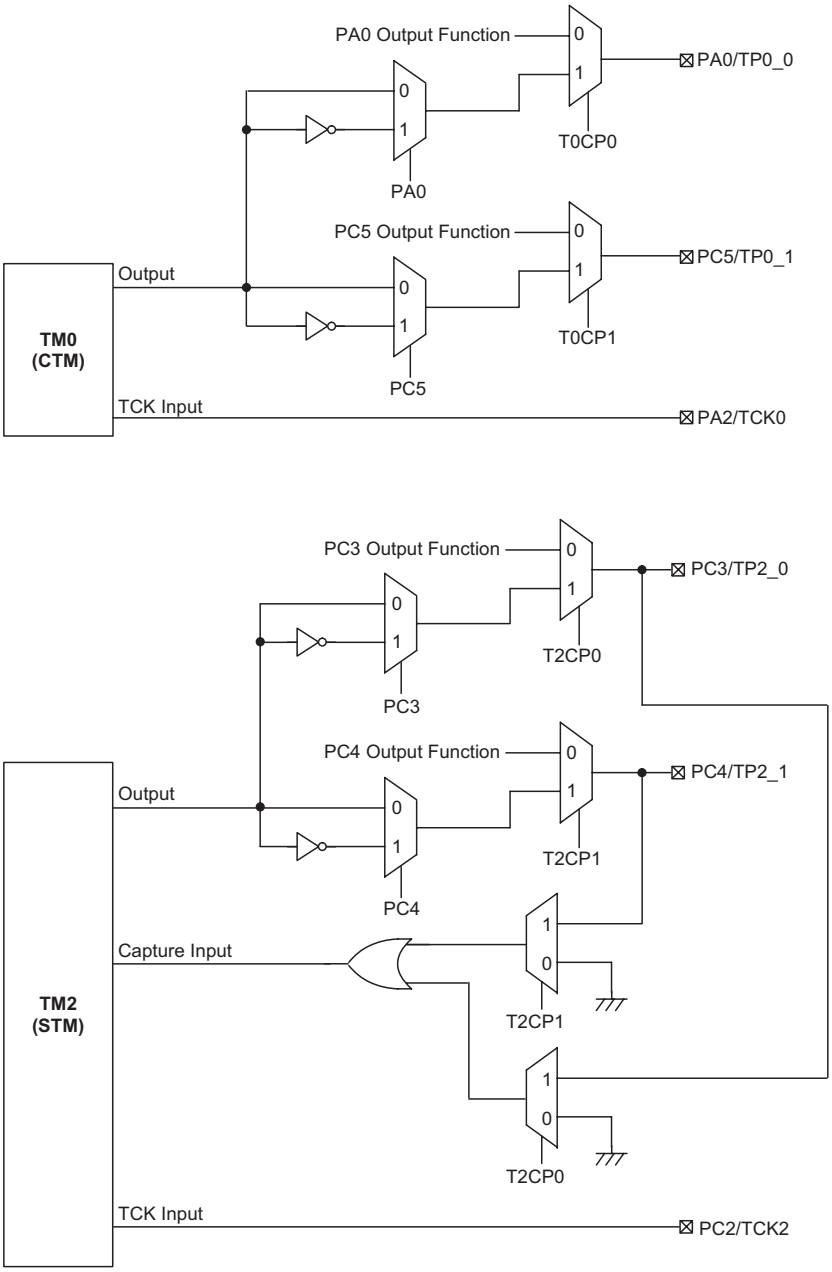

**HT68F20 TM 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。



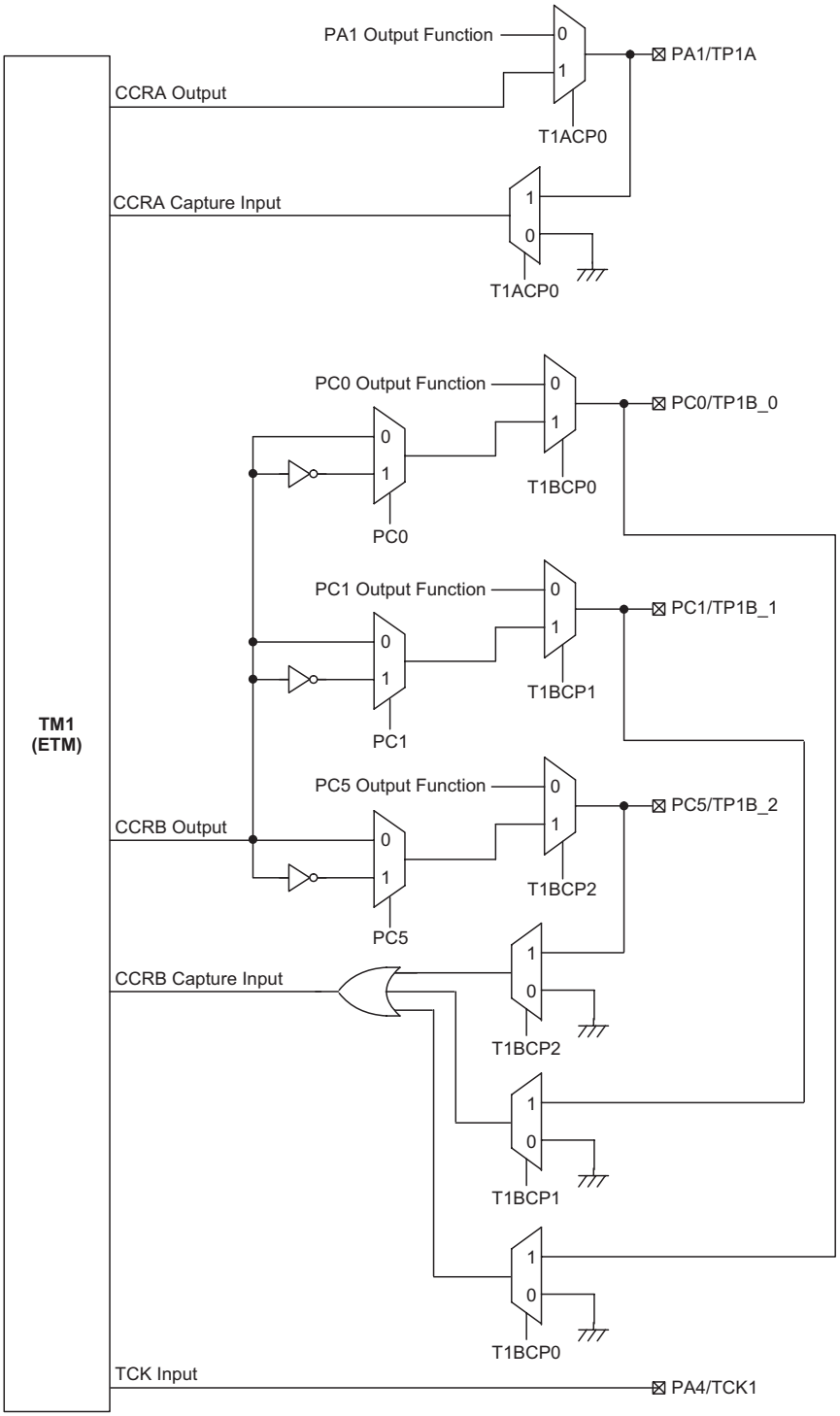

**HT68F30 TM 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。



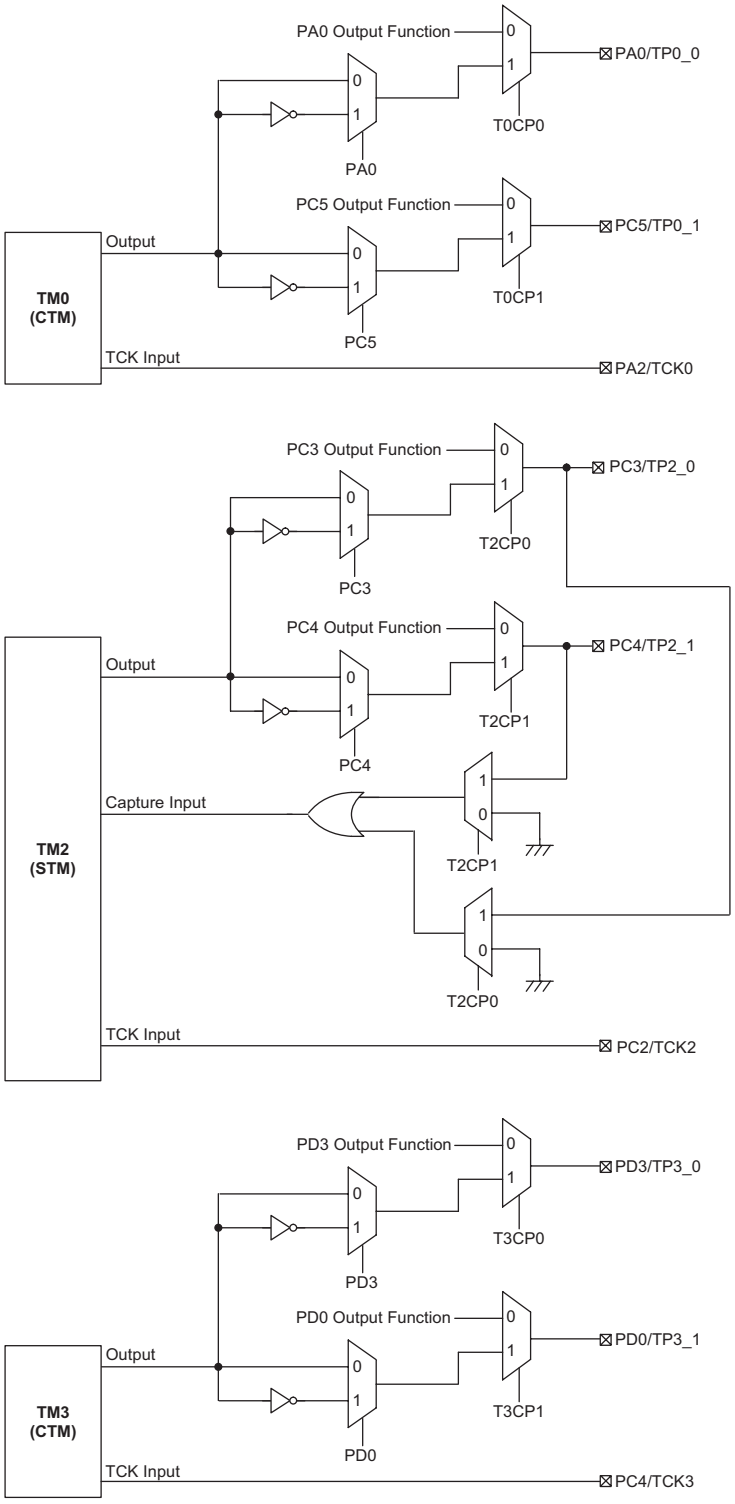

**HT68F40 TM0&TM2 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。



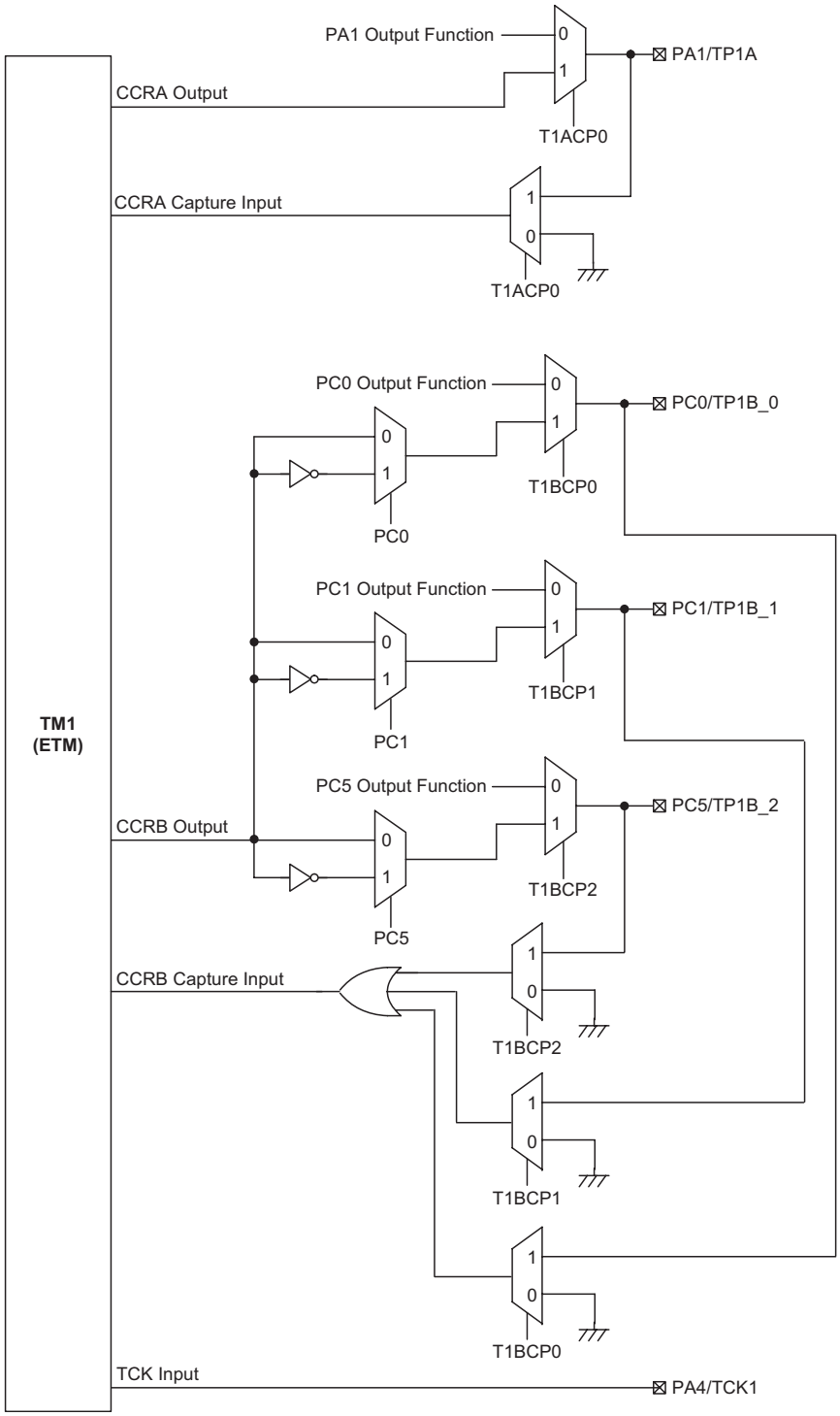

**HT68F40 TM1 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。




**HT68F50 和 HT68F60 TM0，TM2，TM3 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。




**HT68F50 和 HT68F60 TM1 功能引脚控制方框图**

注：1. 上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。
2. 捕捉输入模式中，TM 引脚控制寄存器最多只能使能一个 TM 引脚输入。

### TMPC0 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | T1CP1 | T1CP0 | — | — | — | T0CP0 |
| R/W | — | — | R/W | R/W | — | — | — | R/W |
| POR | — | — | 0 | 1 | — | — | — | 1 |

Bit 7~6 未定义，读为“0”

Bit 5 **T1CP1**：TP1_1 引脚控制位
0：除能
1：使能

Bit 4 **T1CP0**：TP1_0 引脚控制位
0：除能
1：使能

Bit 3~1 未定义，读为“0”

Bit 0 **T0CP0**：TP0_0 引脚控制位
0：除能
1：使能

- **HT68F30**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1ACP0 | — | T1BCP1 | T1BCP0 | — | — | T0CP1 | T0CP0 |
| R/W | R/W | — | R/W | R/W | — | — | R/W | R/W |
| POR | 1 | — | 0 | 1 | — | — | 0 | 1 |

Bit 7 **T1ACP0**：TP1A 引脚控制位
0：除能
1：使能

Bit 6 未定义，读为“0”

Bit 5 **T1BCP1**：TP1B_1 引脚控制位
0：除能
1：使能

Bit 4 **T1BCP0**：TP1B_0 引脚控制位
0：除能
1：使能

Bit 3~2 未定义，读为“0”

Bit 1 **T0CP1**：TP0_1 引脚控制位
0：除能
1：使能

Bit 0 **T0CP0**：TP0_0 引脚控制位
0：除能
1：使能

- **HT68F40/HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1ACP0 | T1BCP2 | T1BCP1 | T1BCP0 | — | — | T0CP1 | T0CP0 |
| R/W | R/W | R/W | R/W | R/W | — | — | R/W | R/W |
| POR | 1 | 0 | 0 | 1 | — | — | 0 | 1 |

Bit 7 **T1ACP0**：TP1A 引脚控制位
0：除能
1：使能

Bit 6 **T1BCP2**：TP1B_2 引脚控制位
0：除能
1：使能

Bit 5 **T1BCP1**：TP1B_1 引脚控制位
0：除能
1：使能

Bit 4 **T1BCP0**：TP1B_0 引脚控制位
0：除能
1：使能

Bit 3~2 未定义，读为“0”

Bit 1 **T0CP1**：TP0_1 引脚控制位
0：除能
1：使能

Bit 0 **T0CP0**：TP0_0 引脚控制位
0：除能
1：使能

### TMPC1 寄存器

- **HT68F40**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | — | — | — | — | T2CP1 | T2CP0 |
| R/W | — | — | — | — | — | — | R/W | R/W |
| POR | — | — | — | — | — | — | 0 | 1 |

Bit 7~2 未定义，读为“0”

Bit 1 **T2CP1**：TP2_1 引脚控制位
0：除能
1：使能

Bit 0 **T2CP0**：TP2_0 引脚控制位
0：除能
1：使能

- **HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | T3CP1 | T3CP0 | — | — | T2CP1 | T2CP0 |
| R/W | — | — | R/W | R/W | — | — | R/W | R/W |
| POR | — | — | 0 | 1 | — | — | 0 | 1 |

Bit 7~6 未定义，读为“0”

Bit 5 **T3CP1**：TP3_1 引脚控制位
0：除能
1：使能

Bit 4 **T3CP0**：TP3_0 引脚控制位
0：除能
1：使能

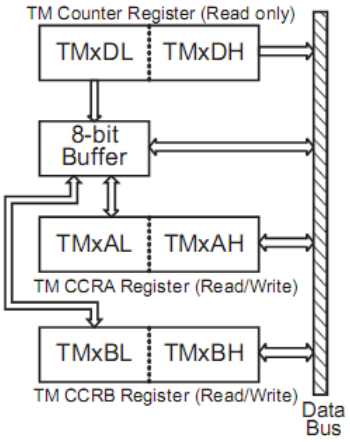
Bit 3~2 未定义，读为“0”

Bit 1 **T2CP1**：TP2_1 引脚控制位
0：除能
1：使能

Bit 0 **T2CP0**：TP2_0 引脚控制位
0：除能
1：使能

## 编程注意事项

TM 计数寄存器和捕捉 / 比较寄存器 CCRA、CCRB 为 10-bit 或 16-bit 的寄存器，含有低字节和高字节结构。高字节可直接访问，低字节则仅能通过一个内部 8-bit 的缓存器进行访问。读写这些成对的寄存器需通过特殊的方式。值得注意的是 8-bit 缓存器的存取数据及相关低字节的读写操作仅在其相应的高字节读取操作执行时发生。



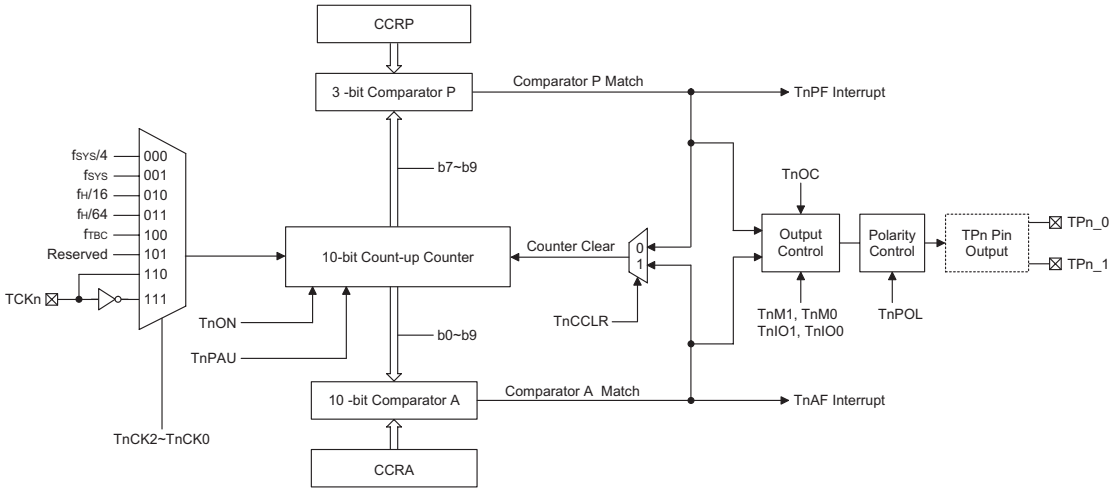

读写流程如下步骤所示：

- 写数据至 CCRB 或 CCRA
  - ♦ 步骤 1. 写数据至低字节寄存器 TMxAL 或 TMxBL
    - – 注意，此时数据仅写入 8-bit 缓存器。
  - ♦ 步骤 2. 写数据至高字节寄存器 TMxAH 或 TMxBH
    - – 注意，此时数据直接写入高字节寄存器，同时锁存在 8-bit 缓存器中的数据写入低字节寄存器。
- 由计数器寄存器和 CCRB 或 CCRA 中读取数据
  - ♦ 步骤 1. 从高字节寄存器 TMxDH，TMxAH 或 TMxBH 中读取数据
    - – 注意，此时高字节寄存器中的数据直接读取，同时由低字节寄存器读取的数据锁存至 8-bit 缓存器中。
  - ♦ 步骤 2. 由低字节寄存器 TMxDL、TMxAL 或 TMxBL 读取数据
    - – 注意，此时读取 8-bit 缓存器中的数据。

# 简易型 TM

虽然简易型 TM 是三种 TM 类型中最简单的形式，但仍然包括三种工作模式，即比较匹配输出，定时 / 事件计数器和 PWM 输出模式。简易型 TM 也由外部输入脚控制并驱动一或两个外部输出脚。两个外部输出脚信号可以相同也可以相反。

| CTM | 名称 | TM 编号 | TM 输入引脚 | TM 输出引脚 |
|---|---|---|---|---|
| HT68F20 | 10-bit CTM | 0 | TCK0 | TP0_0 |
| HT68F30 | 10-bit CTM | 0 | TCK0 | TP0_0，TP0_1 |
| HT68F40 | 10-bit CTM | 0 | TCK0 | TP0_0，TP0_1 |
| HT68F50 | 10-bit CTM | 0，3 | TCK0，TCK3 | TP0_0，TP0_1；TP3_0，TP3_1 |
| HT68F60 | 10-bit CTM | 0，3 | TCK0，TCK3 | TP0_0，TP0_1；TP3_0，TP3_1 |




简易型 TM 方框图

## 简易型 TM 操作

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

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

## 简易型 TM 寄存器介绍

简易型 TM 的所有操作由一组（六个）或两组（十二个）寄存器控制。每组中包含一对只读寄存器用来存放 10 位计数器的值，一对读 / 写寄存器存放 10 位 CCRA 的值，剩下两个控制寄存器设置不同的操作和控制模式以及 CCRP 的 3 个位。

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| TMnC0 | TnPAU | TnCK2 | TnCK1 | TnCK0 | TnON | TnRP2 | TnRP1 | TnRP0 |
| TMnC1 | TnM1 | TnM0 | TnIO1 | TnIO0 | TnOC | TnPOL | TnDPX | TnCCLR |
| TMnDL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TMnDH | — | — | — | — | — | — | D9 | D8 |
| TMnAL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TMnAH | — | — | — | — | — | — | D9 | D8 |

**简易型 TM 寄存器列表 (n=0 或 3)**

### TMnDL 寄存器

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

Bit 7~0 **TMnDL**：TMn 计数器低字节寄存器 bit 7~bit 0
TMn 10-bit 计数器 bit 7~bit 0

### TMnDH 寄存器

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

Bit 7~2 未定义，读为“0”

Bit 1~0 **TMnDH**：TMn 计数器高字节寄存器 bit 1~bit 0
TMn 10-bit 计数器 bit 9~bit 8

### TMnAL 寄存器

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

Bit 7~0 **TMnAL**：TMn CCRA 低字节寄存器 bit 7~bit 0
TMn 10-bit CCRA bit 7~bit 0

### TMnAH 寄存器

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

Bit 7~2 未定义，读为“0”

Bit 1~0 **TMnAH**：TMn CCRA 高字节寄存器 bit 1~bit 0
TMn 10-bit CCRA bit 9~bit 8

### TMnC0 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TnPAU | TnCK2 | TnCK1 | TnCK0 | TnON | TnRP2 | TnRP1 | TnRP0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

Bit 6~4 **TnCK2~TnCK0**：选择 TMn 计数时钟位
000：$f_{SYS}/4$
001：$f_{SYS}$
010：$f_H/16$
011：$f_H/64$
100：$f_{TBC}$
101：保留位
110：TCKn 上升沿时钟
111：TCKn 下降沿时钟
此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。$f_{SYS}$ 是系统时钟，$f_H$ 和 $f_{TBC}$ 是其它的内部时钟源，细节方面请参考振荡器章节。

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

Bit 2~0 **TnRP2~TnRP0**：TMn CCRP 3-bit 寄存器，对应于 TMn 计数器 bit 9~bit 7 比较器 P 匹配周期
000：1024 个 TMn 时钟周期
001：128 个 TMn 时钟周期
010：256 个 TMn 时钟周期
011：384 个 TMn 时钟周期
100：512 个 TMn 时钟周期
101：640 个 TMn 时钟周期
110：768 个 TMn 时钟周期
111：896 个 TMn 时钟周期
此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 TnCCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。TnCCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

### TMnC1 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TnM1 | TnM0 | TnIO1 | TnIO0 | TnOC | TnPOL | TnDPX | TnCCLR |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **TnM1~TnM0**：选择 TMn 工作模式位
00：比较匹配输出模式
01：未定义模式
10：PWM 模式
11：定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 TnM1 和 TnM0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **TnIO1~TnIO0**：选择 TPn_0，TPn_1 输出功能位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 模式
00：强制无效状态
01：强制有效状态
10：PWM 输出
11：未定义
定时 / 计数器模式
未使用
此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。
在比较匹配输出模式下，TnIO1 和 TnIO0 位决定当比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TMnC1 寄存器的 TnOC 位设置取得。注意，由 TnIO1 和 TnIO0 位得到的输出电平必须与通过 TnOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 TnON 位由低到高电平的转换复位至初始值。
在 PWM 模式，TnIO1 和 TnIO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TMn 关闭时改变 TnIO1 和 TnIO0 位的值是很有必要的。若在 TM 运行时改变 TnIO1 和 TnIO0 的值，PWM 输出的值是无法预料的。

Bit 3 **TnOC**：TPn_0，TPn_1 输出控制位
比较匹配输出模式
0：初始低
1：初始高
PWM 模式
0：低有效
1：高有效
这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式。若 TM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **TnPOL**：TPn_0，TPn_1 输出极性控制位
0：同相
1：反相
此位控制 TPn_0 或 TPn_1 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

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

Bit 0 **TnCCLR**：选择 TMn 计数器清零条件位
0：TM0 比较器 P 匹配
1：TM0 比较器 A 匹配
此位用于选择清除计数器的方法。简易型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。TnCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。TnCCLR 位在 PWM 模式时未使用。

## 简易型 TM 工作模式

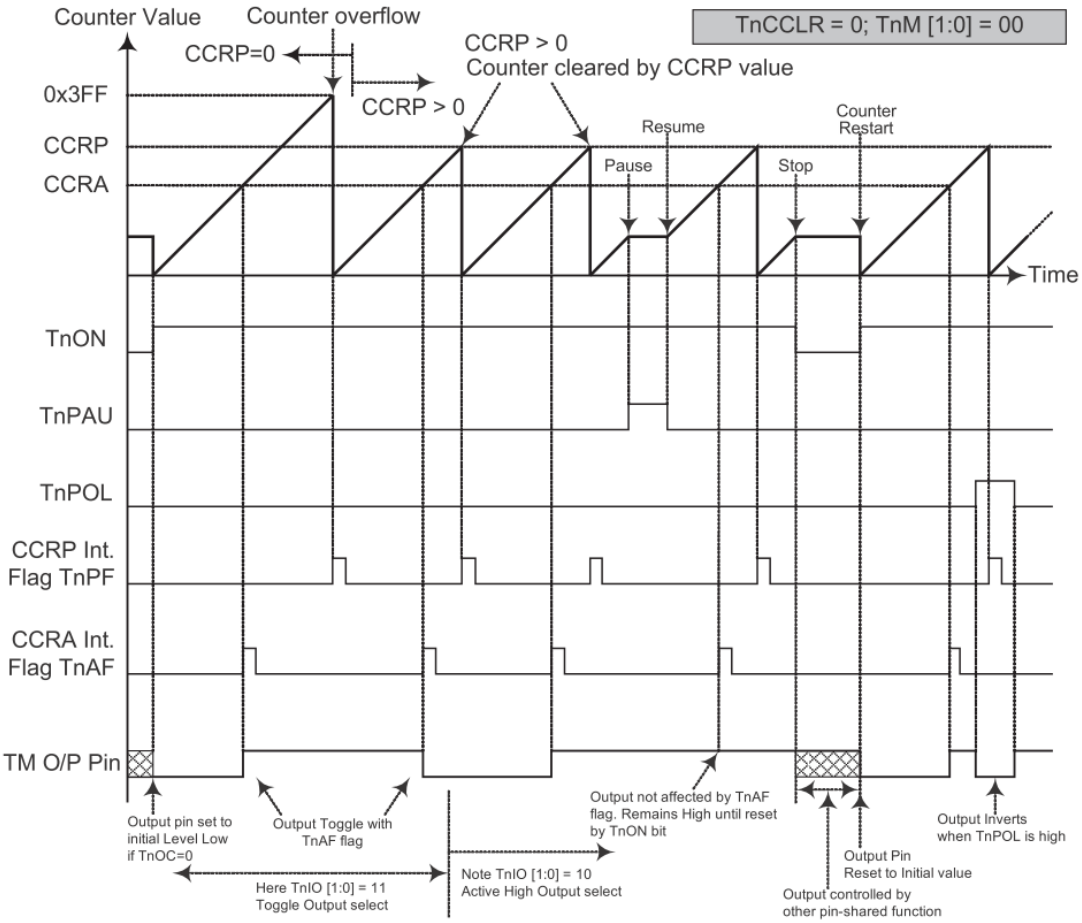
简易型 TM 有三种工作模式，即比较匹配输出模式，PWM 模式或定时 / 计数器模式。通过设置 TMnC1 寄存器的 TnM1 和 TnM0 位选择任意工作模式。

## 比较匹配输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 TnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 TnAF 和 TnPF 将分别置起。

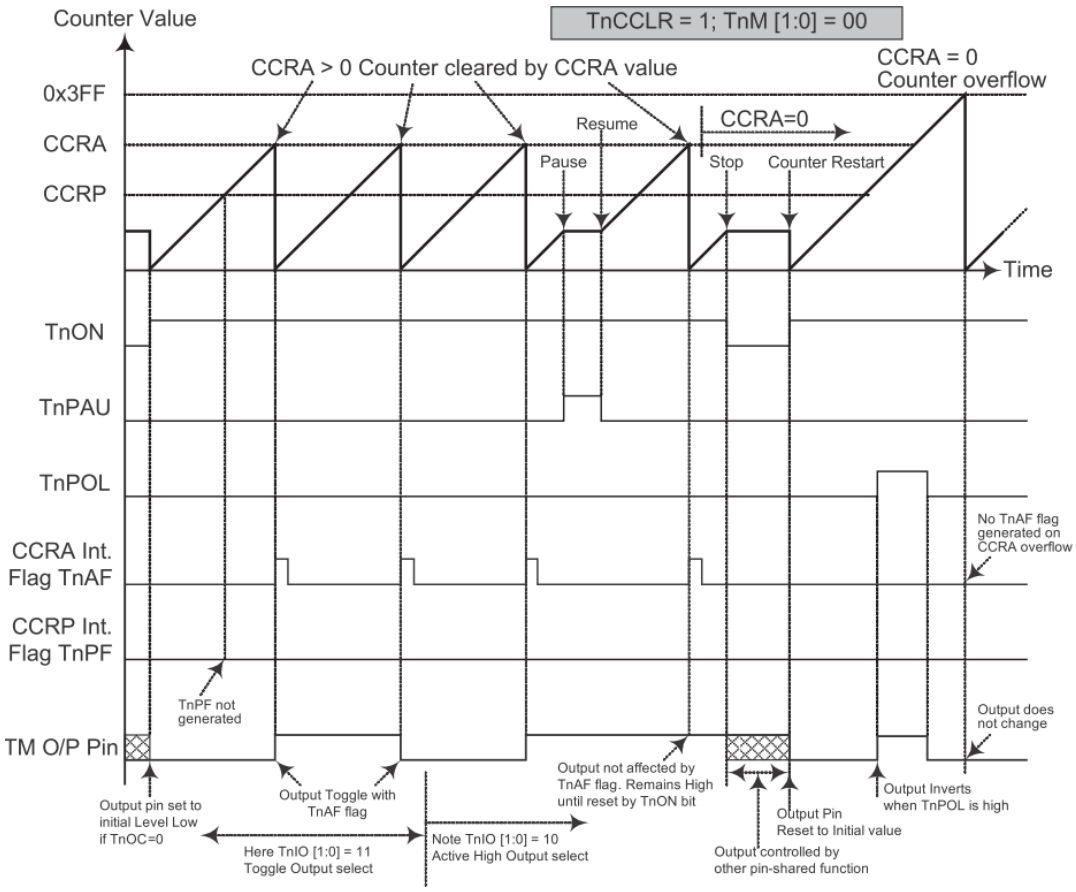
如果 TMnC1 寄存器的 TnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 TnAF 中断请求标志产生。所以当 TnCCLR 为高时，不产生 TnPF 中断请求标志。如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 TnAF 请求标志。

正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 比较匹配发生后 TnAF 标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 TMnC1 寄存器中 TnIO1 和 TnIO0 位决定。当比较器 A 比较匹配发生时，TnIO1 和 TnIO0 位决定 TM 输出脚输出高，低或翻转当前状态。TM 输出脚初始值，既可以通过 TnON 位由低到高电平的变化设置，也可以由 TnOC 位设置。注意，若 TnIO1 和 TnIO0 位同时为 0 时，引脚输出不变。




**比较匹配输出模式 -- TnCCLR=0**

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




**比较匹配输出模式 – TnCCLR=1**

注： 1.TnCCLR=1，比较器 A 匹配将清除计数器
2.TM 输出脚仅由 TnAF 标志位控制
3. 在 TnON 上升沿 TM 输出脚复位至初始值
4. 当 TnCCLR=1 时，TnPF 标志位不会产生

## 定时 / 计数器模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“11”。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 TM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

## PWM 输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“10”。TM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 TM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 模式中，TnCCLR 位不影响 PWM 操作。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 TMnC1 寄存器的 TnDPX 位。所以 PWM 波形频率和占空比由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。TMnC1 寄存器中的 TnOC 位决定 PWM 波形的极性，TnIO1 和 TnIO0 位使能 PWM 输出或将 TM 输出脚置为逻辑高或逻辑低。TnPOL 位对 PWM 输出波形的极性取反。

**CTM，PWM 模式，边沿对齐模式，TnDPX=0**

| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | 128 | 256 | 384 | 512 | 640 | 768 | 896 | 1024 |
| Duty | CCRA | | | | | | | |

若 $f_{SYS}$=16MHz，TM 时钟源选择 $f_{SYS}$/4，CCRP=100b，CCRA=128，

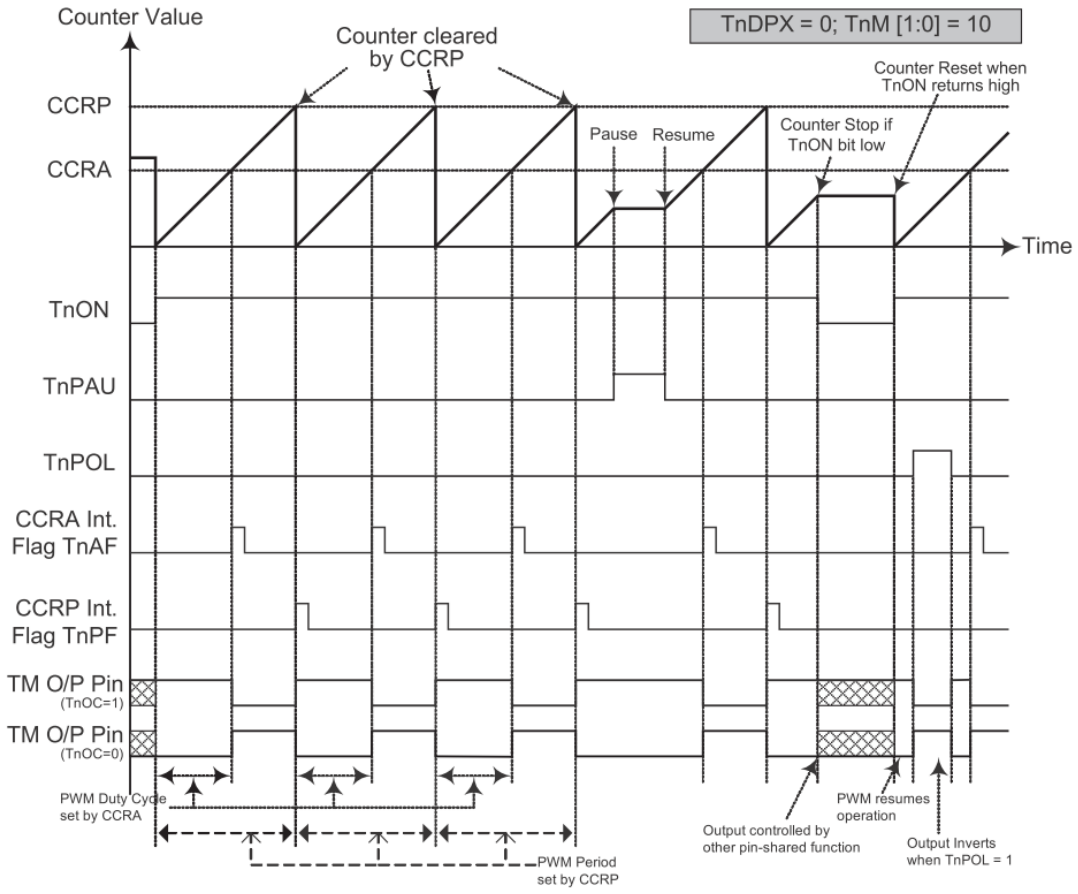
CTM PWM 输出频率 =($f_{SYS}$/4)/512= $f_{SYS}$/2048=7.8125kHz，duty=128/512=25%

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

**CTM，PWM 模式，边沿对齐模式，TnDPX=1**

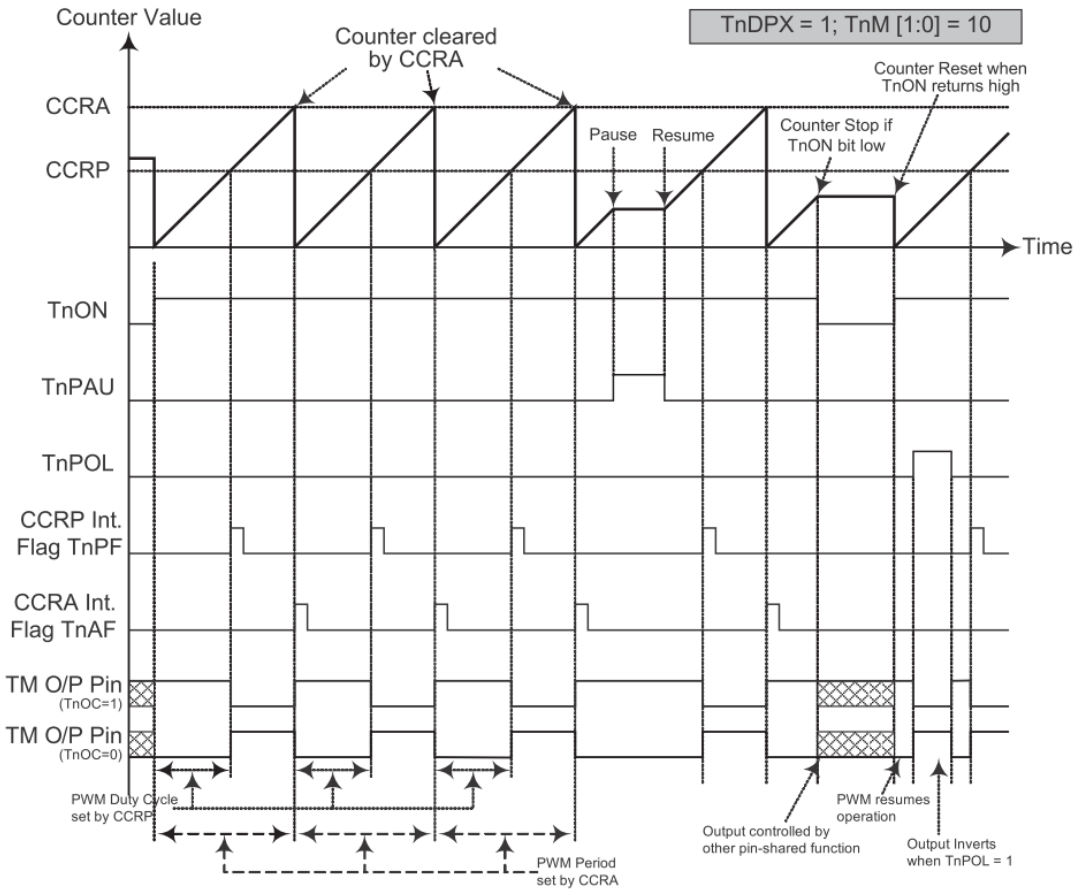
| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | CCRA | | | | | | | |
| Duty | 128 | 256 | 384 | 512 | 640 | 768 | 896 | 1024 |

PWM 的输出周期由 CCRA 寄存器的值与 TM 的时钟共同决定，PWM 的占空比由 CCRP 寄存器的值决定。




**PWM 模式 -- TnDPX=0**

注： 1.TnDPX=0，CCRP 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 TnIO1，TnIO0=00 或 01，PWM 功能不变
4.TnCCLR 位不影响 PWM 操作




**PWM 模式 -- TnDPX=1**

注： 1.TnDPX=1，CCRA 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 TnIO1，TnIO0=00 或 01，PWM 功能不变
4.TnCCLR 位不影响 PWM 操作

# 标准型 TM – STM

标准型 TM 包括 5 种工作模式，即比较匹配输出，定时 / 事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。标准型 TM 也由外部输入脚控制并驱动一个或两个外部输出脚。

| CTM | 名称 | TM 编号 | TM 输入引脚 | TM 输出引脚 |
|---|---|---|---|---|
| HT68F20 | 10-bit STM | 1 | TCK1 | TP1_0，TP1_1 |
| HT68F30 | — | — | — | — |
| HT68F40 | 16-bit STM | 2 | TCK2 | TP2_0，TP2_1 |
| HT68F50 | 16-bit STM | 2 | TCK2 | TP2_0，TP2_1 |
| HT68F60 | 16-bit STM | 2 | TCK2 | TP2_0，TP2_1 |

## 标准型 TM 操作

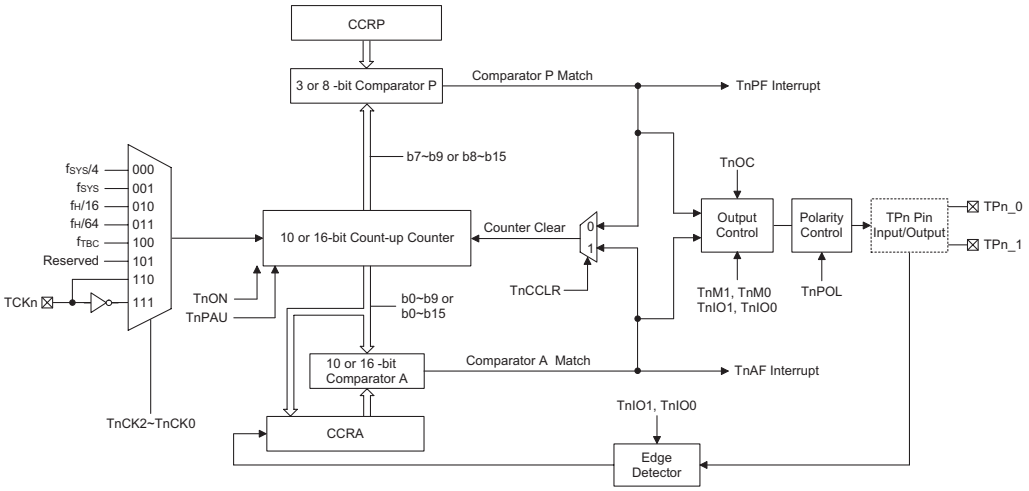
标准型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式和定时 / 计数器模式。通过 TMnC1 寄存器的 TnM1 和 TnM0 位选择工作模式。

有两个标准型 TM，一种是 10 位宽度，另一种是 16 位宽度。核心是一个由用户选择的内部或外部时钟源驱动的 10 位向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 是 3 位或 8 位宽度，与计数器的高 3 位或 8 位比较；而 CCRA 是 10 位或 16 位的，与计数器的所有位比较。

通过应用程序改变 10 或 16 位计数器值的唯一方法是使 TnON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 TM 中断信号。标准型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。

## 标准型 TM 寄存器介绍

标准型 TM 的所有工作模式由一系列寄存器控制。一对只读寄存器用来存放 10 或 16 位计数器的值，一对读 / 写寄存器存放 10 或 16 位 CCRA 的值。剩下两个控制寄存器设置工作模式，以及 CCRP 的 3 或 8 个位。




标准型 TM 框图

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| TM1C0 | T1PAU | T1CK2 | T1CK1 | T1CK0 | T1ON | T1RP2 | T1RP1 | T1RP0 |
| TM1C1 | T1M1 | T1M0 | T1IO1 | T1IO0 | T1OC | T1POL | T1DPX | T1CCLR |
| TM1DL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM1DH | — | — | — | — | — | — | D9 | D8 |
| TM1AL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM1AH | — | — | — | — | — | — | D9 | D8 |

**10-bit 标准型 TM 寄存器列表 (HT68F20)**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| TM2C0 | T2PAU | T2CK2 | T2CK1 | T2CK0 | T2ON | — | — | — |
| TM2C1 | T2M1 | T2M0 | T2IO1 | T2IO0 | T2OC | T2POL | T2DPX | T2CCLR |
| TM2DL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM2DH | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| TM2AL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM2AH | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| TM2RP | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

**16-bit 标准型 TM 寄存器列表 (HT68F40/HT68F50/HT68F60)**

## 10-bit 标准型 TM 寄存器列表 -HT68F20

- **TM1C0 寄存器 -- 10-bit STM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1PAU | T1CK2 | T1CK1 | T1CK0 | T1ON | T1RP2 | T1RP1 | T1RP0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **T1PAU**：TM1 计数器暂停控制位
0：运行
1：暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转变时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **T1CK2~T1CK0**：选择 TM1 计数器时钟位
000：$f_{SYS}/4$
001：$f_{SYS}$
010：$f_H/16$
011：$f_H/64$
100：$f_{TBC}$
101：保留位
110：TCK1 上升沿时钟
111：TCK1 下降沿时钟
此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。$f_{SYS}$ 是系统时钟，$f_H$ 和 $f_{TBC}$ 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **T1ON**：TM1 计数器 On/Off 控制位
0：Off
1：On
此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转换时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。
若 TM 处于比较匹配输出模式时，当 T1ON 位经由低到高的转换时，TM 输出脚将复位至 T1OC 位指定的初始值。

Bit 2~0 **T1RP2~T1RP0**：TM1 CCRP 3-bit 寄存器，对应于 TM1 计数器 bit 9~bit 7 比较器 P 匹配周期
000：1024 个 TM1 时钟周期
001：128 个 TM1 时钟周期
010：256 个 TM1 时钟周期
011：384 个 TM1 时钟周期
100：512 个 TM1 时钟周期
101：640 个 TM1 时钟周期
110：768 个 TM1 时钟周期
111：896 个 TM1 时钟周期
此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 T1CCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。T1CCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

- **TM1C1 寄存器 -- 10-bit STM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1M1 | T1M0 | T1IO1 | T1IO0 | T1OC | T1POL | T1DPX | T1CCLR |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **T1M1~T1M0**：选择 TM1 工作模式位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1M1 和 T1M0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T1IO1~T1IO0**：选择 TP1_0，TP1_1 输出功能位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 模式 / 单脉冲输出模式
00：强制无效状态
01：强制有效状态
10：PWM 输出
11：单脉冲输出
捕捉输入模式
00：在 TP1_0，TP1_1 上升沿输入捕捉
01：在 TP1_0，TP1_1 下降沿输入捕捉
10：在 TP1_0，TP1_1 双沿输入捕捉
11：输入捕捉除能
定时 / 计数器模式
未使用

此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。

在比较匹配输出模式下，T1IO1 和 T1IO0 位决定当从比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM1C1 寄存器的 T1OC 位设置取得。注意，由 T1IO1 和 T1IO0 位得到的输出电平必须与通过 T1OC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ON 位由低到高电平的转换复位至初始值。

在 PWM 模式，T1IO1 和 T1IO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TM1 关闭时改变 T1IO1 和 T1IO0 位的值是很有必要的。若在 TM 运行时改变 T1IO1 和 T1IO0 的值，PWM 输出的值是无法预料的。

Bit 3 **T1OC**：TP1_0，TP1_1 输出控制位
比较匹配输出模式
0：初始低
1：初始高
PWM 模式 / 单脉冲输出模式
0：低有效
1：高有效
这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式 / 单脉冲输出模式。若 TM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **T1POL**：TP1_0，TP1_1 输出极性控制位
0：同相
1：反相
此位控制 TP1_0 或 TP1_1 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1 **T1DPX**：TM1 PWM 周期 / 占空比控制位
0：CCRP - 周期；CCRA - 占空比
1：CCRP - 占空比；CCRA - 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **T1CCLR**：选择 TM1 计数器清零条件位
0：TM1 比较器 P 匹配
1：TM1 比较器 A 匹配
此位用于选择清除计数器的方法。标准型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。T1CCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。T1CCLR 位在 PWM，单脉冲或输入捕捉模式时未使用。

- **TM1DL 寄存器 -- 10-bit STM**

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

Bit 7~0 **TM1DL**：TM1 计数器低字节寄存器 bit 7~bit 0
TM1 10-bit 计数器 bit 7~bit 0

● **TM1DH 寄存器 -- 10-bit STM**

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

Bit 7~2　未定义，读为“0”

Bit 1~0　**TM1DH**：TM1 计数器高字节寄存器 bit 1~bit 0
TM1 10-bit 计数器 bit 9~bit 8

● **TM1AL 寄存器 -- 10-bit STM**

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

Bit 7~0　**TM1AL**：TM1 CCRA 低字节寄存器 bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

● **TM1AH 寄存器 -- 10-bit STM**

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

Bit 7~2　未定义，读为“0”

Bit 1~0　**TM1AH**：TM1 CCRA 高字节寄存器 bit 1~bit 0
TM1 10-bit CCRA bit 9~bit 8

## 16-bit 标准型 TM 寄存器列表 -HT68F40/HT68F50/HT68F60

● **TM2C0 寄存器 -- 16-bit STM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T2PAU | T2CK2 | T2CK1 | T2CK0 | T2ON | — | — | — |
| R/W | R/W | R/W | R/W | R/W | R/W | — | — | — |
| POR | 0 | 0 | 0 | 0 | 0 | — | — | — |

Bit 7　**T2PAU**：TM2 计数器暂停控制位
0：运行
1：暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4　**T2CK2,T2CK1,T2CK0**：选择 TM2 计数时钟位
000：$f_{SYS}/4$
001：$f_{SYS}$
010：$f_H/16$
011：$f_H/64$
100：$f_{TBC}$
101：保留位
110：TCK2 上升沿时钟
111：TCK2 下降沿时钟

此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。$f_{SYS}$ 是系统时钟，$f_H$ 和 $f_{TBC}$ 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **T2ON**：TM2 计数器 On/Off 控制位
0：Off
1：On
此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。
若 TM 处于比较匹配输出模式时 ( 通过 T2OC 位指定 )，当 T2ON 位经由低到高的转换时，TM 输出脚将重置其初始值。

Bit 2~0 未定义，读为“0”

● **TM2C1 寄存器 -- 16-bit STM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T2M1 | T2M0 | T2IO1 | T2IO0 | T2OC | T2POL | T2DPX | T2CCLR |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **T2M1~T2M0**：选择 TM2 工作模式位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T2M1 和 T2M0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T2IO1~T2IO0**：选择 TP2_0，TP2_1 输出功能位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 模式 / 单脉冲输出模式
00：强制无效状态
01：强制有效状态
10：PWM 输出
11：单脉冲输出
捕捉输入模式
00：在 TP2_0，TP2_1 上升沿输入捕捉
01：在 TP2_0，TP2_1 下降沿输入捕捉
10：在 TP2_0，TP2_1 双沿输入捕捉
11：输入捕捉除能
定时 / 计数器模式
未使用
此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。
在比较匹配输出模式下，T2IO1 和 T2IO0 位决定当从比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM2C1 寄存器的 T2OC 位设置取得。注意，由 T2IO1 和 T2IO0 位得到的输出电平必须与通过 T2OC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T2ON 位由低到高电平的转换复位至初始值。

Bit 3 **T2OC**：TP2_0，TP2_1 输出控制位
比较匹配输出模式
0：初始低
1：初始高
PWM 模式 / 单脉冲输出模式
0：低有效
1：高有效
这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式 / 单脉冲输出模式。若 TM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **T2POL**：TP2_0，TP2_1 输出极性控制位
0：同相
1：反相
此位控制 TP2_0 或 TP2_1 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1 **T2DPX**：TM2 PWM 周期 / 占空比控制位
0：CCRP - 周期；CCRA - 占空比
1：CCRP - 占空比；CCRA - 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **T2CCLR**：选择 TM2 计数器清零条件位
0：TM2 比较器 P 匹配
1：TM2 比较器 A 匹配
此位用于选择清除计数器的方法。标准型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。T2CCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。T2CCLR 位在 PWM，单脉冲或输入捕捉模式时未使用。

- **TM2DL 寄存器 -- 16-bit STM**

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

Bit 7~0 **TM2DL**：TM2 计数器低字节寄存器 bit 7~bit 0
TM2 16-bit 计数器 bit 7~bit 0

- **TM2DH 寄存器 -- 16-bit STM**

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

Bit 7~0 **TM2DH**：TM2 计数器高字节寄存器 bit 7~bit 0
TM2 16-bit 计数器 bit 15~bit 8

● **TM2AL 寄存器 -- 16-bit STM**

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

Bit 7~0 **TM2AL**：TM2 CCRA 低字节寄存器 bit 7~bit 0
TM2 16-bit CCRA bit 7~bit 0

● **TM2AH 寄存器 -- 16-bit STM**

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

Bit 7~0 **TM2AH**：TM2 CCRA 高字节寄存器 bit 7~bit 0
TM2 16-bit CCRA bit 15~bit 8

● **TM2RP 寄存器 -- 16-bit STM**

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

Bit 7~0 **TM2RP**：TM2 CCRP 高字节寄存器 bit 7~bit 0
TM2 CCRP 8 位寄存器，与 TM2 计数器 bit 15~bit 8 比较。比较器 P 匹配周期
0：65536 个 TM2 时钟周期
1~255：256×(1~255) 个 TM2 时钟周期
此八位设定内部 CCRP 8-bit 寄存器的值，然后与内部计数器的高八位进行比较。如果 T2CCLR 位设为 0 时，比较结果为 0 并清除内部计数器。T2CCLR 位设为低，CCRP 比较匹配结果将重置内部计数器。由于 CCRP 只与计数器高八位比较，比较结果是 256 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

## 标准型 TM 工作模式

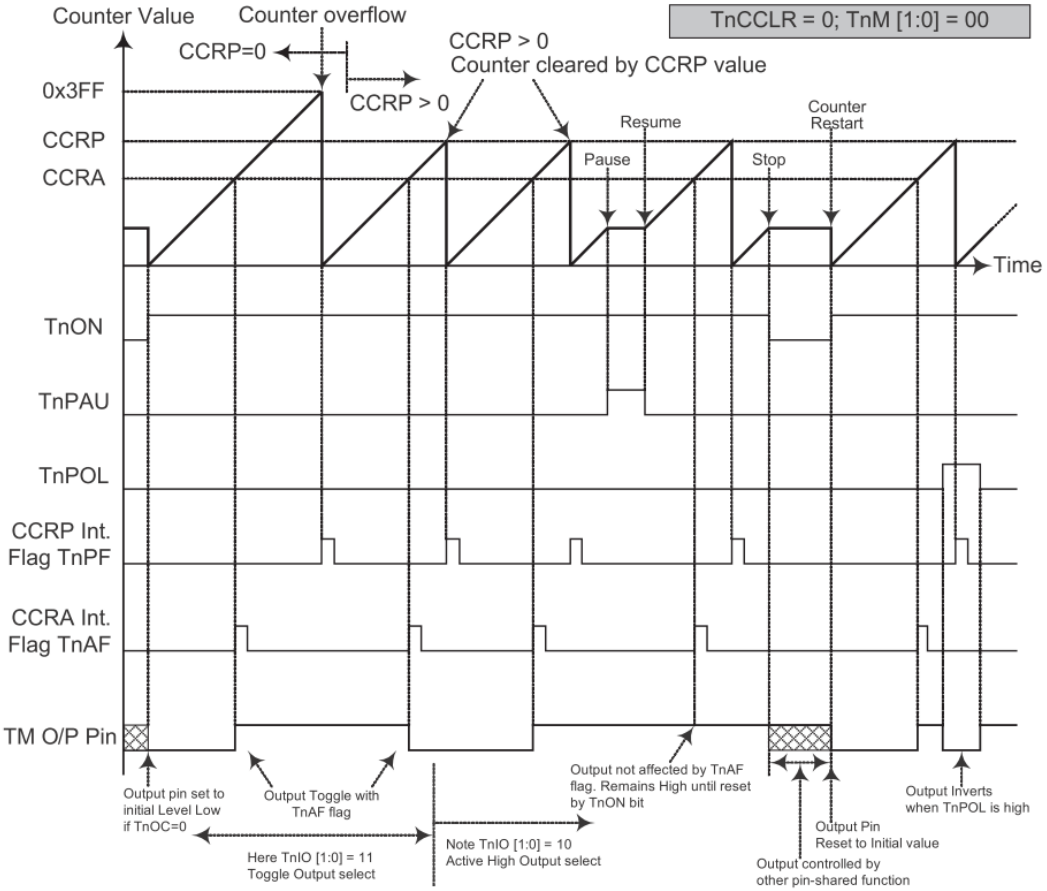
标准型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时 / 计数器模式。通过设置 TMnC1 寄存器的 TnM1 和 TnM0 位选择任意模式。

## 比较匹配输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 TnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 TnAF 和 TnPF 将分别置位。

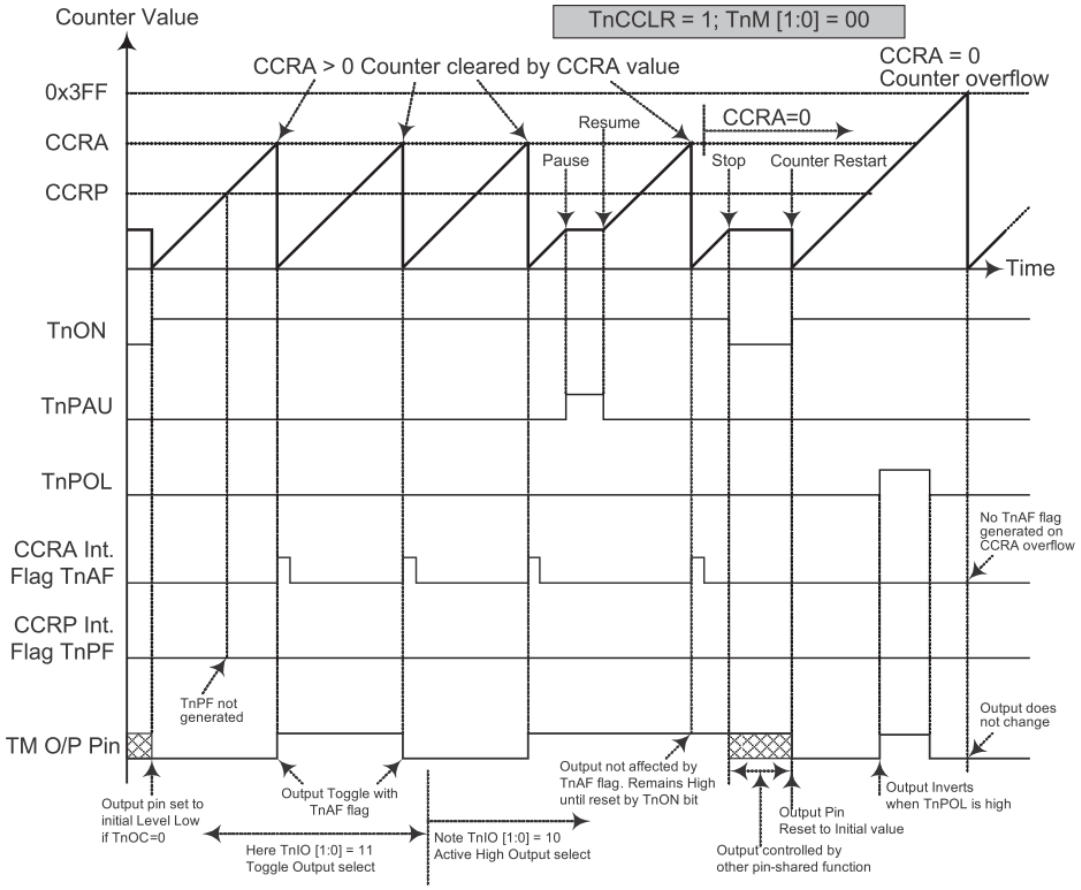
如果 TMnC1 寄存器的 TnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅产生 TnAF 中断请求标志。所以当 TnCCLR 为高时，不会产生 TnPF 中断请求标志。在比较匹配输出模式下，CCRA 不能设为“0”。

正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 比较匹配发生后 TnAF 标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 TMnC1 寄存器中 TnIO1 和 TnIO0 位决定。当比较器 A 比较匹配发生时，TnIO1 和 TnIO0 位决定 TM 输出脚输出高，低或翻转当前状态。TM 输出脚初始值，既可以通过 TnON 位由低到高电平的变化设置，也可以由 TnOC 位设置。注意，若 TnIO1 和 TnIO0 位同时为 0 时，引脚输出不变。




**比较匹配输出模式 -- TnCCLR=0**

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




**比较匹配输出模式 -- TnCCLR=1**

注： 上图注意要点：

1.TnCCLR=1，比较器 A 匹配将清除计数器

2.TM 输出脚仅由 TnAF 标志位控制

3. 在 TnON 上升沿 TM 输出脚复位至初始值

4. 当 TnCCLR=1 时，不会产生 TnPF 标志

## 定时 / 计数器模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“11”。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 TM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

## PWM 输出模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“10”，且 TnIO1 和 TnIO0 位也需要设置为“10”。TM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 TM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 模

式中，TnCCLR 位不影响 PWM 周期。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 TMnC1 寄存器的 TnDPX 位。所以 PWM 波形由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。TMnC1 寄存器中的 TnOC 位决定 PWM 波形的极性，TnIO1 和 TnIO0 位使能 PWM 输出或将 TM 输出脚置为逻辑高或逻辑低。TnPOL 位对 PWM 输出波形的极性取反。

**10-bit STM，PWM 模式，边沿对齐模式，TnDPX=0**

| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | 128 | 256 | 384 | 512 | 640 | 768 | 896 | 1024 |
| Duty | CCRA | | | | | | | |

若 $f_{SYS}$=16MHz，TM 时钟源选择 $f_{SYS}$/4，CCRP=100b，CCRA=128，

STM PWM 输出频率 =($f_{SYS}$/4)/512= $f_{SYS}$/2048=7.8125kHz，duty=128/512=25%

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

**10-bit STM，PWM 模式，边沿对齐模式，TnDPX=1**

| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | CCRA | | | | | | | |
| Duty | 128 | 256 | 384 | 512 | 640 | 768 | 896 | 1024 |

PWM 的输出周期由 CCRA 寄存器的值与 TM 的时钟共同决定，PWM 的占空比由 CCRP 寄存器的值决定。

**16-bit STM，PWM 模式，边沿对齐模式，TnDPX=0**

| CCRP | 1~255 | 0 |
|---|---|---|
| Period | CCRP×256 | 65536 |
| Duty | CCRA | |

若 $f_{SYS}$=16MHz，TM 时钟源选择 $f_{SYS}$/4，CCRP=2，CCRA=128，

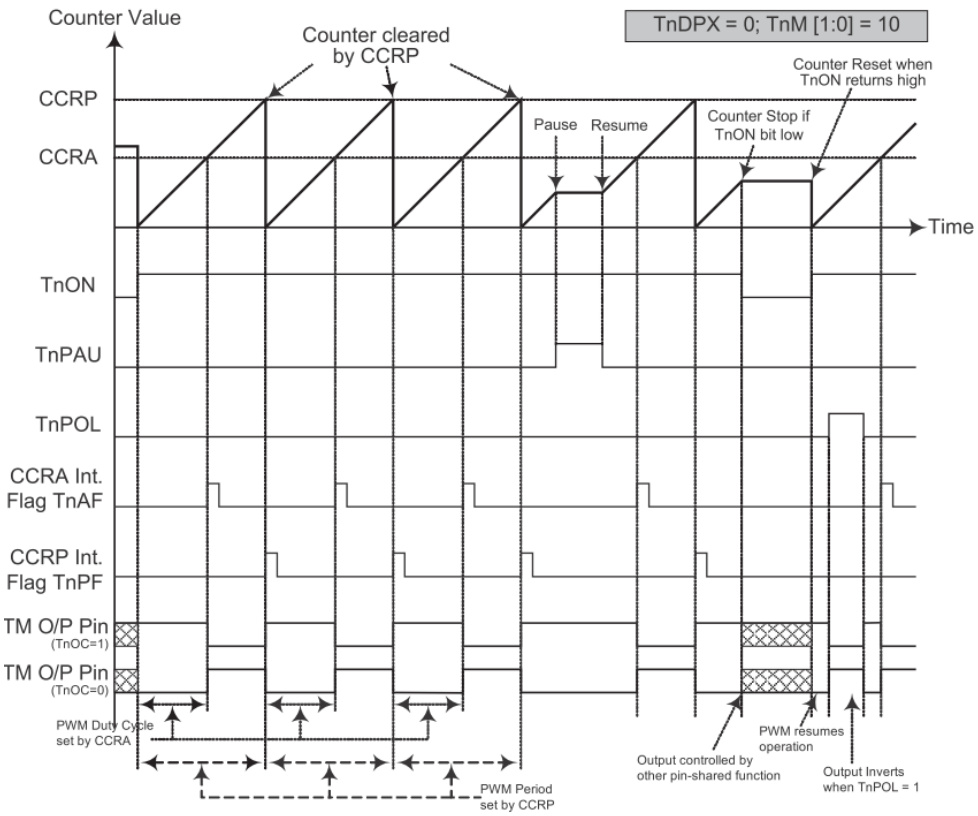
STM PWM 输出频率 =($f_{SYS}$/4)/(2×256)= $f_{SYS}$/2048=7.8125kHz，duty=128/(2×256)=25%

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

**16-bit STM，PWM 模式，边沿对齐模式，TnDPX=1**

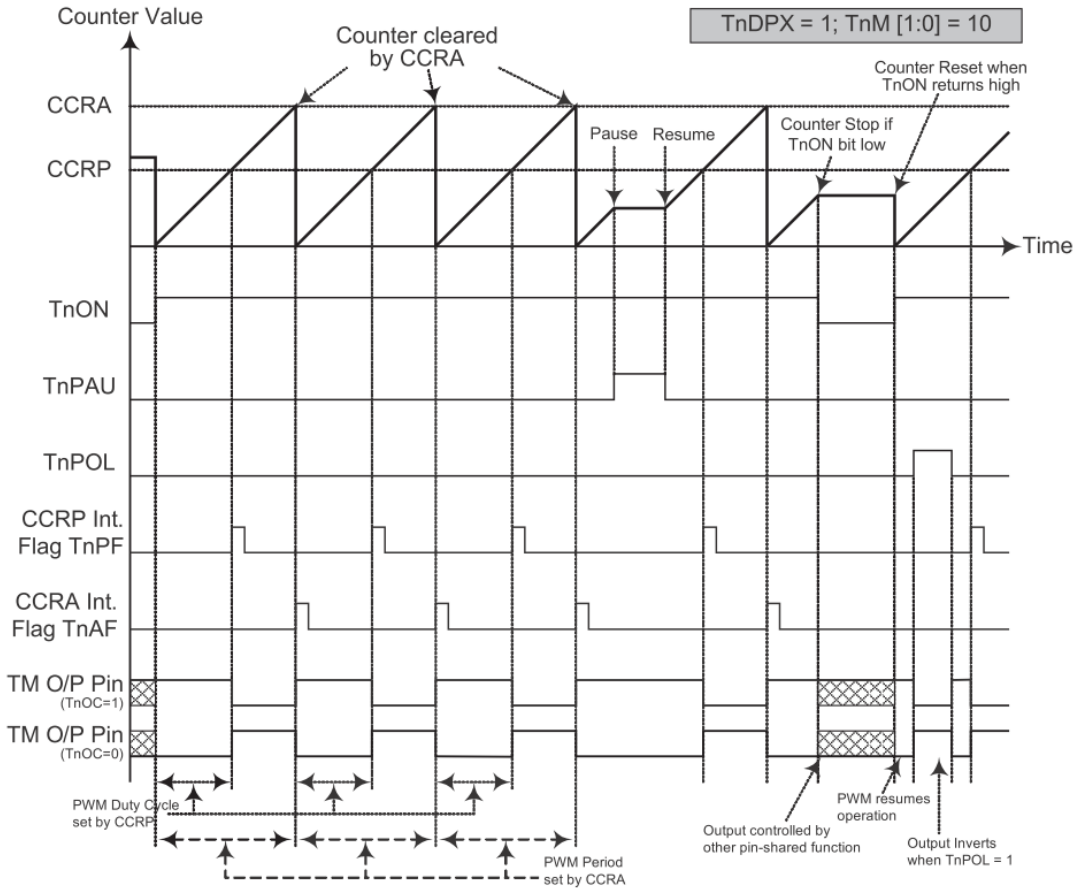
| CCRP | 1~255 | 0 |
|---|---|---|
| Period | CCRA | |
| Duty | CCRP×256 | 65536 |

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




**PWM 模式 -- TnDPX=0**

注： 1.TnDPX=0，CCRP 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 TnIO1，TnIO0=00 或 01，PWM 功能不变
4.TnCCLR 位不影响 PWM 操作



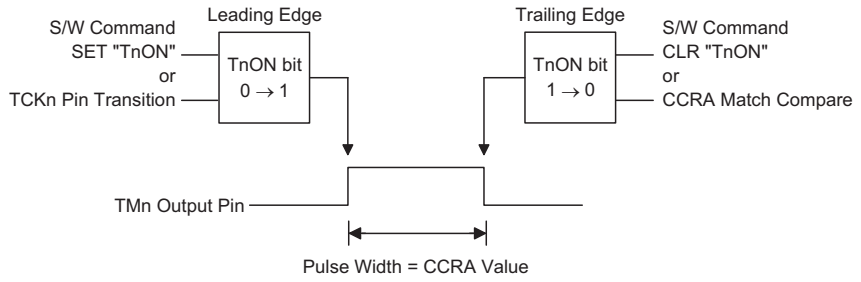

**PWM 模式 -- TnDPX=1**

注： 1.TnDPX=1，CCRA 清除计数器
2. 计数器清零并设置 PWM 周期
3. 当 TnIO1，TnIO0=00 或 01，PWM 功能不变
4.TnCCLR 位不影响 PWM 操作

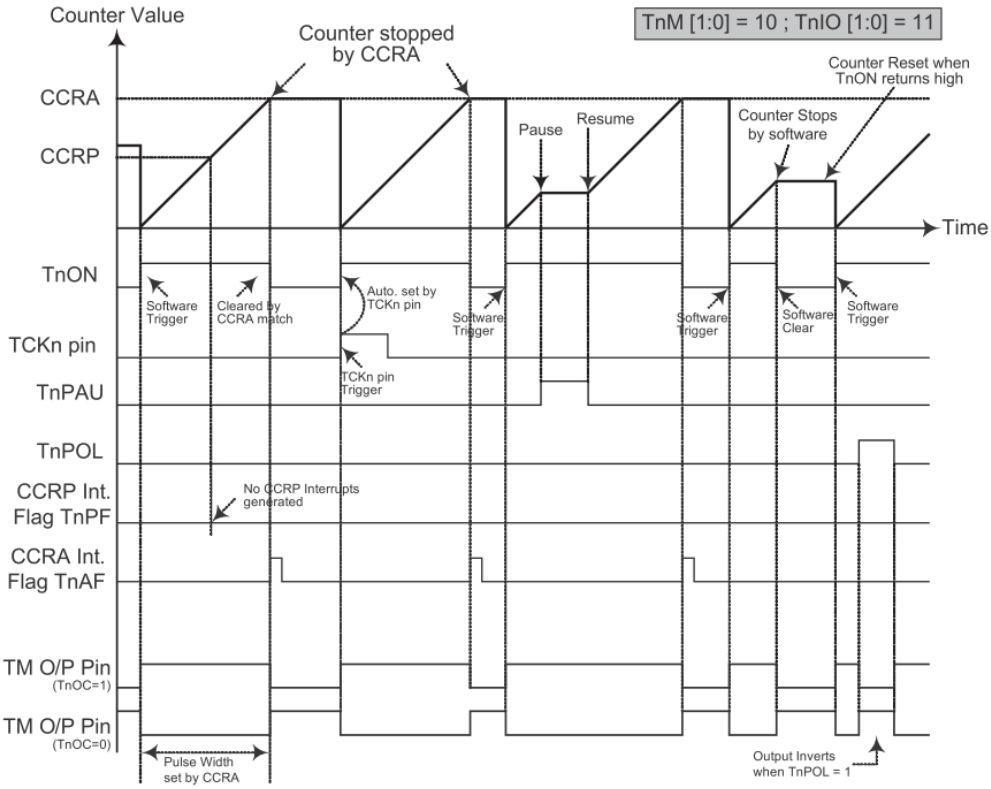
## 单脉冲模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“10”，同时 TnIO1 和 TnIO0 位需要设置为“11”。正如模式名所言，单脉冲输出模式，在 TM 输出脚将产生一个脉冲输出。

脉冲输出可以通过应用程序控制 TnON 位由低到高的转变来触发。而处于单脉冲模式时，TnON 位在 TCKn 脚自动由低转变为高，进而初始化单脉冲输出状态。当 TnON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 TnON 位保持高电平。通过应用程序使 TnON 位清零或比较器 A 比较匹配发生时，产生脉冲下降沿。




单脉冲产生示意图




单脉冲模式

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

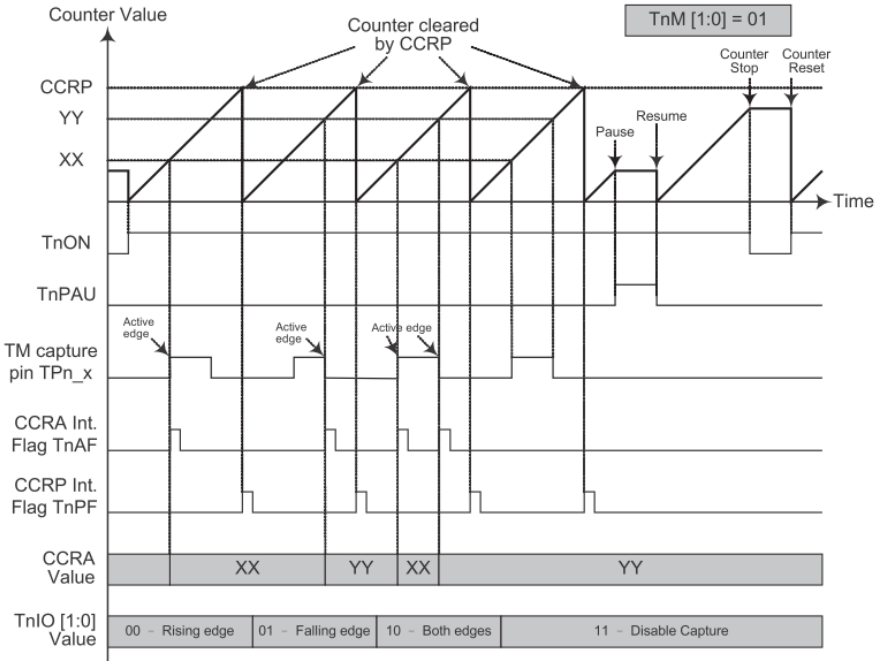
然而，比较器 A 比较匹配发生时，会自动清除 TnON 位并产生单脉冲输出下降沿。CCRA 的值通过这种方式控制脉冲宽度。比较器 A 比较匹配发生时，也会产生 TM 中断。TnON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲模式中，CCRP 寄存器，TnCCLR 和 TnDPX 位未使用。

## 捕捉输入模式

为使 TM 工作在此模式，TMnC1 寄存器中的 TnM1 和 TnM0 位需要设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。TPn_0 或 TPn_1 脚上的外部信号，通过设置 TMnC1 寄存器的 TnIO1 和 TnIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。计数器在 TnON 位由低到高转变时启动并通过应用程序初始化。

当 TPn_0 或 TPn_1 脚出现有效边沿转换时，计数器当前值被锁存到 CCRA 寄存器，并产生 TM 中断。不考虑 TPn_0 或 TPn_1 引脚事件，计数器继续工作直到 TnON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 TM 中断。记录 CCRP 溢出中断信号的值可以测量脉宽。通过设置 TnIO1 和 TnIO0 位选择 TPn_0 或 TPn_1 引脚为上升沿，下降沿或双沿有效。不考虑 TPn_0 或 TPn_1 引脚事件，如果 TnIO1 和 TnIO0 位设置为高，不会产生捕捉操作，但计数器继续运行。

当 TPn_0 或 TPn_1 引脚与其它功能共用，TM 工作在输入捕捉模式时需多加注意。这是因为如果引脚被设为输出，那么该引脚上的任何电平转变都可能执行输入捕捉操作。TnCCLR 和 TnDPX 位在此模式中未使用。




捕捉输入模式

注： 1.TnM1，TnM0=01 并通过 TnIO1 和 TnIO0 位设置有效边沿
2.TM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 中
3.TnCCLR 位未使用
4. 无输出功能 -- TnOC 和 TnPOL 位未使用
5. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大

# 增强型 TM – ETM

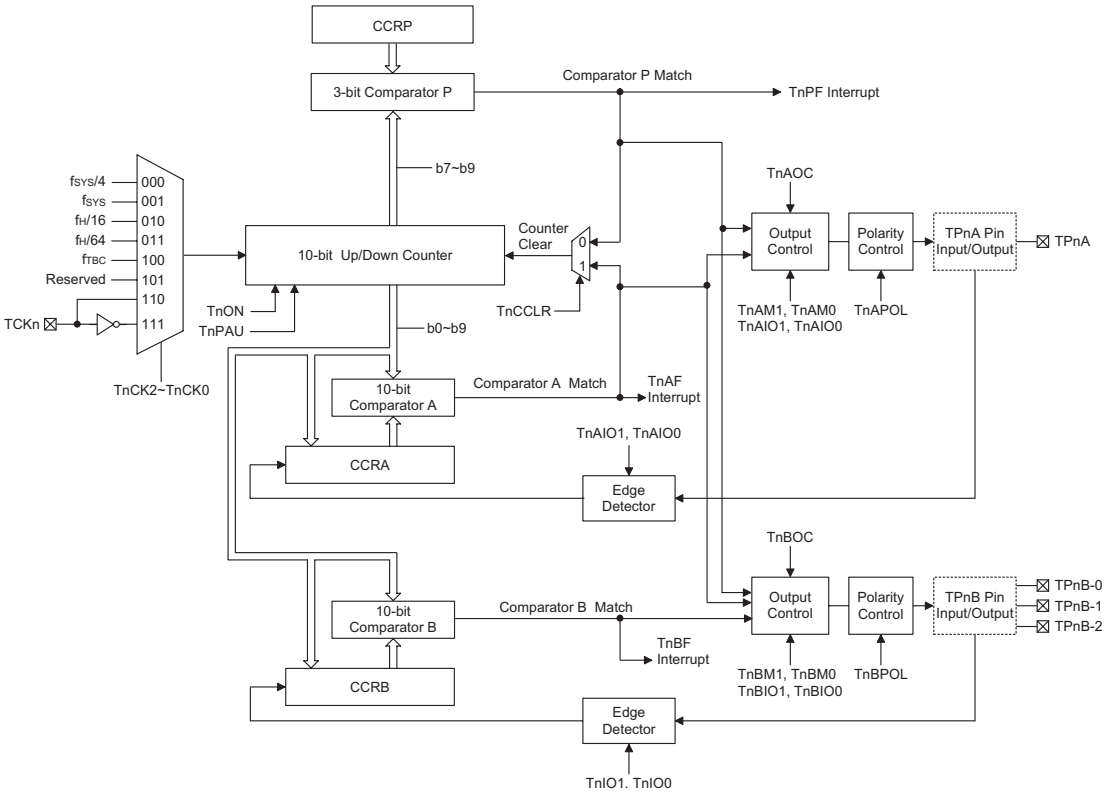
增强型 TM 包括 5 种工作模式，即比较匹配输出，定时 / 事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。增强型 TM 也由外部输入脚控制并驱动三个或四个外部输出脚。

| CTM | 名称 | TM 编号 | TM 输入引脚 | TM 输出引脚 |
|---|---|---|---|---|
| HT68F20 | — | — | — | — |
| HT68F30 | 10-bit ETM | 1 | TCK1 | TP1A，TP1B_0，TP1B_1 |
| HT68F40 | 10-bit ETM | 1 | TCK1 | TP1A，TP1B_0，TP1B_1，TP1B_2 |
| HT68F50 | 10-bit ETM | 1 | TCK1 | TP1A，TP1B_0，TP1B_1，TP1B_2 |
| HT68F60 | 10-bit ETM | 1 | TCK1 | TP1A，TP1B_0，TP1B_1，TP1B_2 |

## 增强型 TM 操作

增强型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 10 位向上 / 向下计数器，它还包括三个内部比较器即比较器 A，比较器 B 和比较器 P。这三个比较器将计数器的值与 CCRA，CCRB 和 CCRP 寄存器中的值进行比较。CCRP 是 3 位的，与计数器的高 3 位比较；而 CCRA 和 CCRB 是 10 位的，与计数器的所有位比较。

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




增强型 TM 方框图

## 增强型 TM 寄存器介绍

增强型 TM 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 10 位计数器的值，两对读 / 写寄存器存放 10 位 CCRA 和 CCRB 的值。剩下三个控制寄存器用来设置不同的操作和控制模式，以及 CCRP 的 3 个位。

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| TM1C0 | T1PAU | T1CK2 | T1CK1 | T1CK0 | T1ON | T1RP2 | T1RP1 | T1RP0 |
| TM1C1 | T1AM1 | T1AM0 | T1AIO1 | T1AIO0 | T1AOC | T1APOL | T1CDN | T1CCLR |
| TM1C2 | T1BM1 | T1BM0 | T1BIO1 | T1BIO0 | T1BOC | T1BPOL | T1PWM1 | T1PWM0 |
| TM1DL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM1DH | — | — | — | — | — | — | D9 | D8 |
| TM1AL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM1AH | — | — | — | — | — | — | D9 | D8 |
| TM1BL | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| TM1BH | — | — | — | — | — | — | D9 | D8 |

**10-bit 增强型 TM 寄存器列表 (ETM 为 TM1)**

**10-bit 增强型 TM 寄存器列表 -- HT68F30/HT68F40/HT68F50/HT68F60**

- **TM1C0 寄存器 -- 10-bit ETM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1PAU | T1CK2 | T1CK1 | T1CK0 | T1ON | T1RP2 | T1RP1 | T1RP0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **T1PAU**：TM1 计数器暂停控制位
0：运行
1：暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转变时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **T1CK2~T1CK0**：选择 TM1 计数时钟位
000：$f_{SYS}/4$
001：$f_{SYS}$
010：$f_H/16$
011：$f_H/64$
100：$f_{TBC}$
101：保留位
110：TCK1 上升沿时钟
111：TCK1 下降沿时钟
此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。$f_{SYS}$ 是系统时钟，$f_H$ 和 $f_{TBC}$ 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **T1ON**：TM1 计数器 On/Off 控制位
0：Off
1：On
此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转换时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。
若 TM 处于比较匹配输出模式时，当 T1ON 位经由低到高的转换时，TM 输出脚将复位至 T1OC 位指定的初始值。

Bit 2~0 **T1RP2~T1RP0**：TM1 CCRP 3-bit 寄存器，对应于 TM1 计数器 bit 9~bit 7 比较器 P 匹配周期
000：1024 个 TM1 时钟周期
001：128 个 TM1 时钟周期
010：256 个 TM1 时钟周期
011：384 个 TM1 时钟周期
100：512 个 TM1 时钟周期
101：640 个 TM1 时钟周期
110：768 个 TM1 时钟周期
111：896 个 TM1 时钟周期
此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 T1CCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。T1CCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

- **TM1C1 寄存器 -- 10-bit ETM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1AM1 | T1AM0 | T1AIO1 | T1AIO0 | T1AOC | T1APOL | T1CDN | T1CCLR |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **T1AM1~T1AM0**：选择 TM1 CCRA 工作模式位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1AM1 和 T1AM0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T1AIO1~T1AIO0**：选择 TP1A 输出功能位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 模式 / 单脉冲输出模式
00：强制无效状态
01：强制有效状态
10：PWM 输出
11：单脉冲输出
捕捉输入模式
00：在 TP1A 上升沿输入捕捉
01：在 TP1A 下降沿输入捕捉
10：在 TP1A 双沿输入捕捉
11：输入捕捉除能
定时 / 计数器模式
未使用
此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。
在比较匹配输出模式下，T1AIO1 和 T1AIO0 位决定当从比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM1C1 寄存器的 T1AOC 位设置取得。注意，由 T1AIO1 和 T1AIO0 位得到的输出电平必须与通过 T1AOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ON 位由低到高电平的转换复位至初始值。

在 PWM 模式，T1AIO1 和 T1AIO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TM1 关闭时改变 T1AIO1 和 T1AIO0 位的值是很有必要的。若在 TM 运行时改变 T1AIO1 和 T1AIO0 的值，PWM 输出的值是无法预料的。

Bit 3 **T1AOC**：TP1A 输出控制位
比较匹配输出模式
0：初始低
1：初始高
PWM 模式 / 单脉冲输出模式
0：低有效
1：高有效
这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式 / 单脉冲输出模式。若 TM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **T1APOL**：TP1A 输出极性控制位
0：同相
1：反相
此位控制 TP1A 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1 **T1CDN**：TM1 计数器向上 / 向下计数标志位
0：向上计数
1：向下计数

Bit 0 **T1CCLR**：选择 TM1 计数器清零条件位
0：TM1 比较器 P 匹配
1：TM1 比较器 A 匹配
此位用于选择清除计数器的方法。增强型 TM 包括三个比较器 -- 比较器 P、比较器 A 和比较器 B，其中比较器 P 和比较器 A 都可以用作清除内部计数器。T1CCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。T1CCLR 位在单脉冲或输入捕捉模式时未使用。

- **TM1C2 寄存器 -- 10-bit ETM**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T1BM1 | T1BM0 | T1BIO1 | T1BIO0 | T1BOC | T1BPOL | T1PWM1 | T1PWM0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **T1BM1~T1BM0**：选择 TM1 CCRB 工作模式位
00：比较匹配输出模式
01：捕捉输入模式
10：PWM 模式或单脉冲输出模式
11：定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1BM1 和 T1BM0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。

Bit 5~4 **T1BIO1~T1BIO0**：选择 TP1B_0，TP1B_1，TP1B_2 输出功能位
比较匹配输出模式
00：无变化
01：输出低
10：输出高
11：输出翻转
PWM 模式 / 单脉冲输出模式
00：强制无效状态
01：强制有效状态

10：PWM 输出
11：单脉冲输出
捕捉输入模式
00：在 TP1B_0，TP1B_1，TP1B_2 上升沿输入捕捉
01：在 TP1B_0，TP1B_1，TP1B_2 下降沿输入捕捉
10：在 TP1B_0，TP1B_1，TP1B_2 双沿输入捕捉
11：输入捕捉除能
定时 / 计数器模式
未使用

此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。

在比较匹配输出模式下，T1BIO1 和 T1BIO0 位决定当比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TM1C2 寄存器的 T1BOC 位设置取得。注意，由 T1BIO1 和 T1BIO0 位得到的输出电平必须与通过 T1BOC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ON 位由低到高电平的转换复位至初始值。

在 PWM 模式，T1BIO1 和 T1BIO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TM1 关闭时改变 T1BIO1 和 T1BIO0 位的值是很有必要的。若在 TM 运行时改变 T1BIO1 和 T1BIO0 的值，PWM 输出的值是无法预料的。

Bit 3 **T1BOC**：TP1B_0，TP1B_1，TP1B_2 输出控制位
比较匹配输出模式
0：初始低
1：初始高
PWM 模式 / 单脉冲输出模式
0：低有效
1：高有效

这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式 / 单脉冲输出模式。若 TM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是高有效还是低有效。

Bit 2 **T1BPOL**：TP1B_0，TP1B_1，TP1B_2 输出极性控制位
0：同相
1：反相

此位控制 TP1B_0，TP1B_1，TP1B_2 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1~0 **T1PWM1~T1PWM0**：选择 PWM 模式位
00：边沿对齐
01：中心对齐，向上计数比较匹配
10：中心对齐，向下计数比较匹配
11：中心对齐，向上 / 下计数比较匹配

- **TM1DL 寄存器 -- 10-bit ETM**

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

Bit 7~0 **TM1DL**：TM1 计数器低字节寄存器 bit 7~bit 0
TM1 10-bit 计数器 bit 7~bit 0

- **TM1DH 寄存器 -- 10-bit ETM**

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

Bit 7~2 未定义，读为“0”

Bit 1~0 **TM1DH**：TM1 计数器高字节寄存器 bit 1~bit 0
TM1 10-bit 计数器 bit 9~bit 8

- **TM1AL 寄存器 -- 10-bit ETM**

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

Bit 7~0 **TM1AL**：TM1 CCRA 低字节寄存器 bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

- **TM1AH 寄存器 -- 10-bit ETM**

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

Bit 7~2 未定义，读为“0”

Bit 1~0 **TM1AH**：TM1 CCRA 高字节寄存器 bit 1~bit 0
TM1 10-bit CCRA bit 9~bit 8

- **TM1BL 寄存器 -- 10-bit ETM**

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

Bit 7~0 **TM1BL**：TM1 CCRB 低字节寄存器 bit 7~bit 0
TM1 10-bit CCRB bit 7~bit 0

- **TM1BH 寄存器 -- 10-bit ETM**

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

Bit 7~2 未定义，读为“0”

Bit 1~0 **TM1BH**：TM1 CCRB 高字节寄存器 bit 1~bit 0
TM1 10-bit CCRB bit 9~bit 8

## 增强型 TM 工作模式

增强型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时 / 计数器模式。通过设置 TMnC1 寄存器的 TnAM1 和 TnAM0 位和 TMnC2 寄存器的 TnBM1 和 TnBM0 位选择任意模式。

| ETM 工作模式 | CCRA 比较匹配输出模式 | CCRA 定时 / 计数器模式 | CCRA PWM 输出模式 | CCRA 单脉冲输出模式 | CCRA 输入捕捉模式 |
|---|---|---|---|---|---|
| CCRB 比较匹配输出模式 | √ | — | — | — | — |
| CCRB 定时 / 计数器模式 | — | √ | — | — | — |
| CCRB PWM 输出模式 | — | — | √ | — | — |
| CCRB 单脉冲输出模式 | — | — | — | √ | — |
| CCRB 输入捕捉模式 | — | — | — | — | √ |

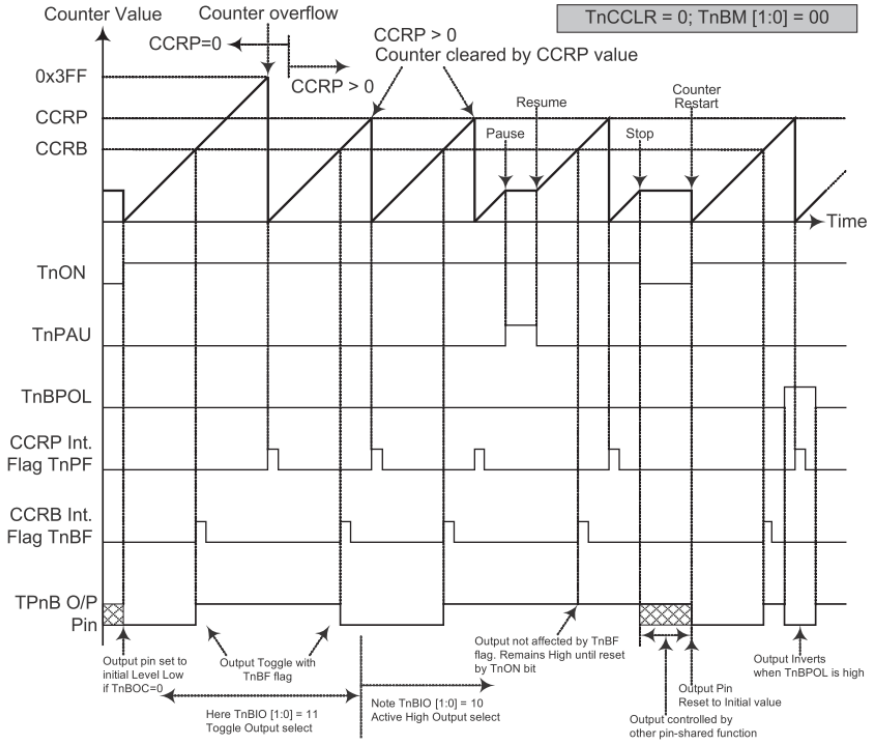
“√”：允许，“—”：不允许

## 比较匹配输出模式

为使 TM 工作在此模式，TMnC1 寄存器的 TnAM1，TnAM0 位和 TMnC2 寄存器的 TnBM1，TnBM0 位需要全部清零。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 TnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 TnAF 和 TnPF 将分别置起。

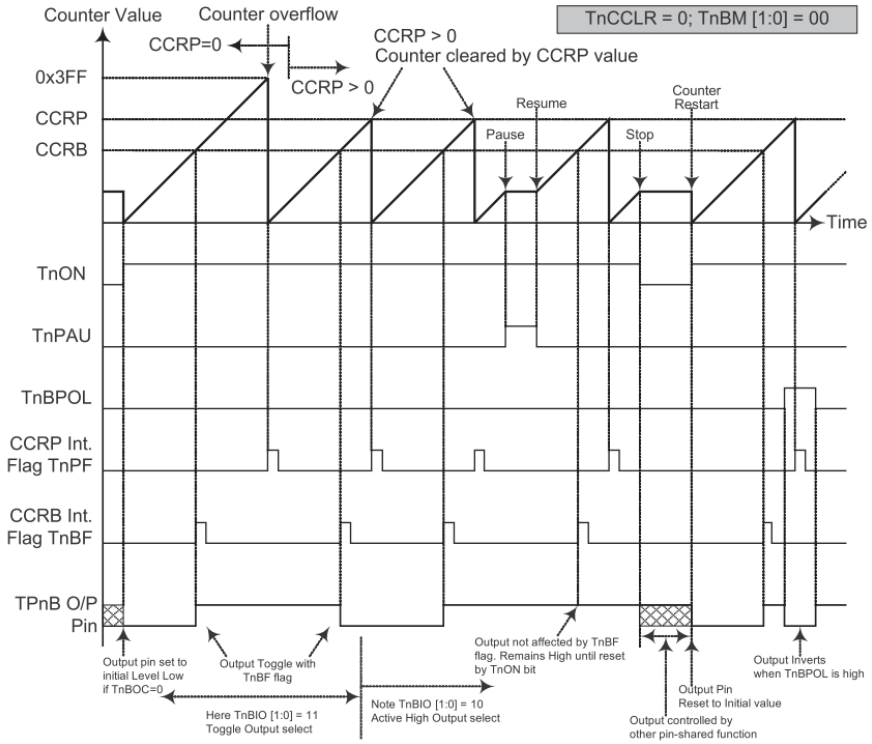
如果 TMnC1 寄存器的 TnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 TnAF 中断请求标志产生。所以当 TnCCLR 为高时，不会产生 TnPF 中断请求标志。

正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 或比较器 B 比较匹配发生后 TnAF 或 TnBF 中断请求标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 ETM CCRA 的 TMnC1 寄存器中 TnAIO1 和 TnAIO0 位，ETM CCRB 的 TMnC2 寄存器中的 TnBIO1 和 TnBIO0 位决定。当比较器 A 或比较器 B 比较匹配发生时，TnAIO1，TnAIO0 位 ( 对于 TPnA 引脚 ) 和 TnBIO1，TnBIO0 位 ( 对于 TPnB_0，TPnB_1，TPnB_2 引脚 ) 决定 TM 输出脚输出高，低或翻转当前状态。TM 输出脚初始值，既可以通过 TnON 位由低到高电平的变化设置，也可以由 TnAOC 或 TnBOC 位设置。注意，若 TnAIO1，TnAIO0 和 TnBIO1，TnBIO0 位同时为 0 时，引脚输出不变。



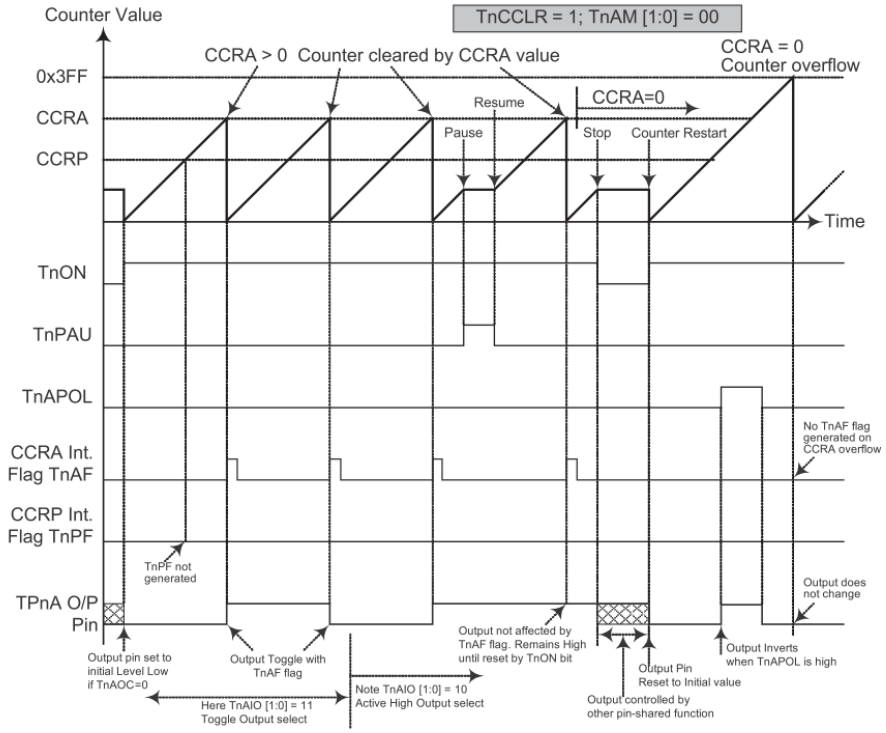

**ETM CCRA 比较匹配输出模式 -- TnCCLR=0**

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



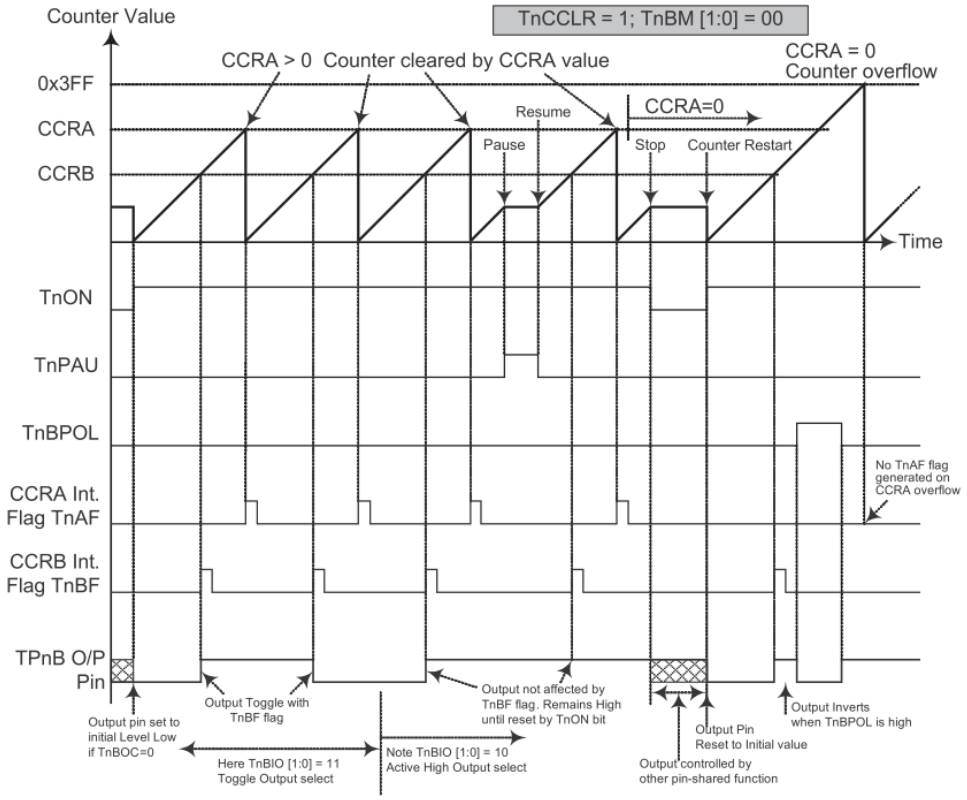

**ETM CCRB 比较匹配输出模式 -- TnCCLR=0**

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




**ETM CCRA 比较匹配输出模式 -- TnCCRL=1**

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




**ETM CCRB 比较匹配输出模式 -- TnCCRL=1**

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

## 定时 / 计数器模式

为使 TM 工作在此模式，TMnC1 寄存器的 TnAM1，TnAM0 位和 TMnC2 寄存器的 TnBM1，TnBM0 位需要全部设为高。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 TM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

## PWM 输出模式

为使 TM 工作在此模式，TnAM1，TnAM0 和 TnBM1，TnBM0 位需要分别设置为“10”，且 TnAIO1，TnAIO0 和 TnBIO1，TnBIO0 位也需要分别设置为“10”。TM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 TM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 模式中，TnCCLR 位决定 PWM 周期控制方式。当 TnCCLR 设为高，CCRA 寄存器控制 PWM 周期。在这种情况下，CCRB 寄存器设置 PWM 的占空比 ( 针对

TPnB 输出脚 )。CCRP 寄存器和 TPnA 输出脚未使用。PWM 输出只在 TPnB 输出脚产生。当 TnCCLR 清零时，PWM 周期通过 CCRP 三位中八个值之一设置，并且是 128 的倍数。此时，CCRA 和 CCRB 寄存器设置不同占空比，在 TPnA 和 TPnB 引脚输出两个 PWM 波形。

TnPWM1 和 TnPWM0 位决定 PWM 的对齐方式，即边沿或中心对齐方式。在边沿对齐方式中，当计数器清零时，产生 PWM 前沿信号。与此同时电流发生跳变，这在高功耗应用中会出现问题。在中心对齐方式中，PWM 波形中心持续产生有效信号，因此可以减少电流跳变引起的功耗问题。

当比较器 A，比较器 B 或比较器 P 比较匹配发生时，CCRA，CCRB 和 CCRP 中断标志位分别产生。TMnC1 寄存器的 TnAOC 位和 TMnC2 寄存器的 TnBOC 位选择 PWM 波形的极性，TnAIO1，TnAIO0 和 TnBIO1，TnBIO0 位使能 PWM 输出或迫使 TM 输出脚为高电平或低电平。TnAPOL 和 TnBPOL 位用来取反 PWM 输出波形的极性。

**ETM，PWM 模式，边沿对齐模式，TnCCLR=0**

| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | 128 | 256 | 384 | 512 | 640 | 768 | 896 | 1024 |
| A Duty | CCRA | | | | | | | |
| B Duty | CCRB | | | | | | | |

若 $f_{SYS}$=16MHz，TM 时钟源选择 $f_{SYS}$/4，CCRP=100b，CCRA=128，CCRB=256，

TP1A PWM 输出频率 =($f_{SYS}$/4)/512= $f_{SYS}$/2048=7.8125kHz，duty=128/512=25%

TP1B_n PWM 输出频率 =($f_{SYS}$/4)/512= $f_{SYS}$/2048=7.8125kHz，duty=256/512=50%

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

**ETM，PWM 模式，边沿对齐模式，TnCCLR=1**

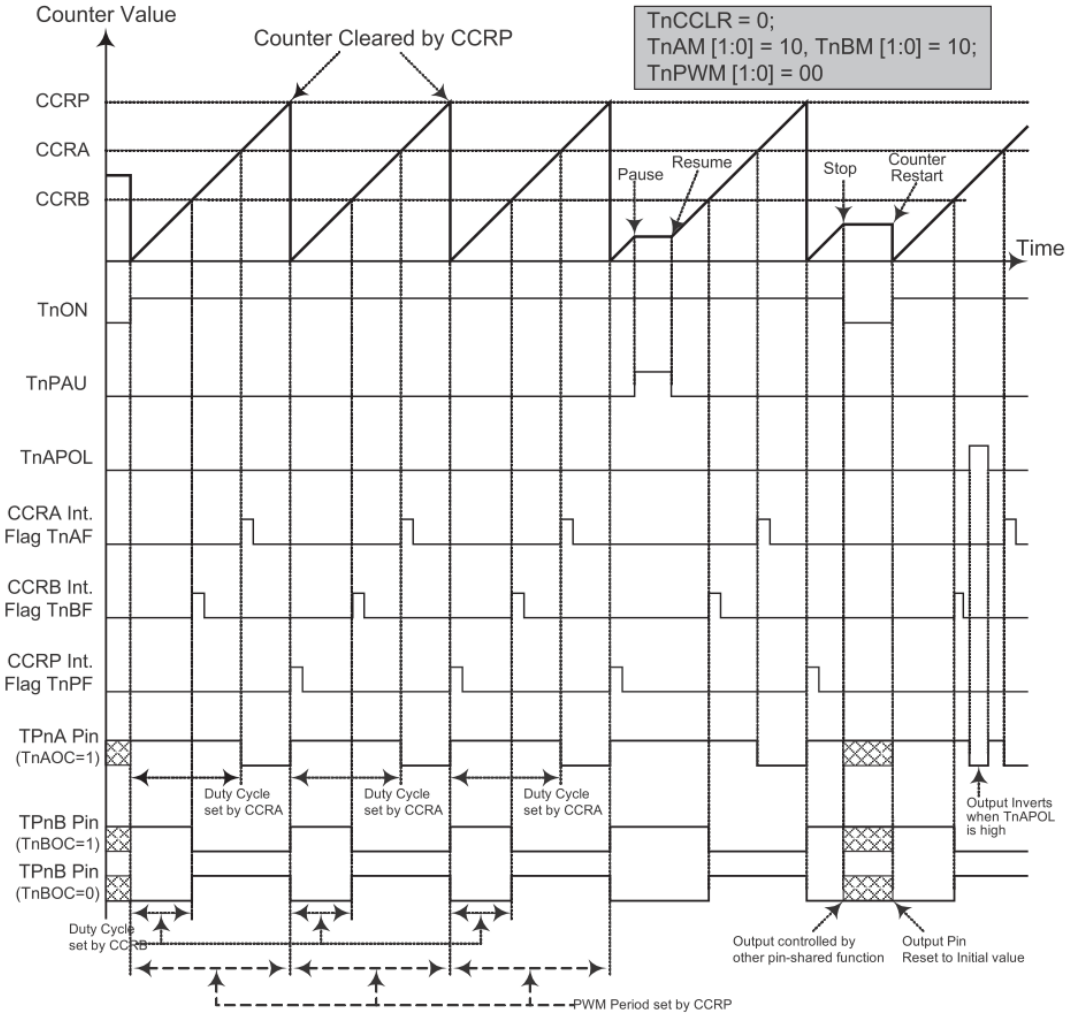
| CCRA | 1 | 2 | 3 | ····· | 511 | 512 | ····· | 1021 | 1022 | 1023 |
|---|---|---|---|---|---|---|---|---|---|---|
| Period | 1 | 2 | 3 | ····· | 511 | 512 | ····· | 1021 | 1022 | 1023 |
| B Duty | CCRB | | | | | | | | | |

**ETM，PWM 模式，中心对齐模式，TnCCLR=0**

| CCRP | 001b | 010b | 011b | 100b | 101b | 110b | 111b | 000b |
|---|---|---|---|---|---|---|---|---|
| Period | 256 | 512 | 768 | 1024 | 1280 | 1536 | 1792 | 2046 |
| A Duty | (CCRA×2) － 1 | | | | | | | |
| B Duty | (CCRB×2) － 1 | | | | | | | |

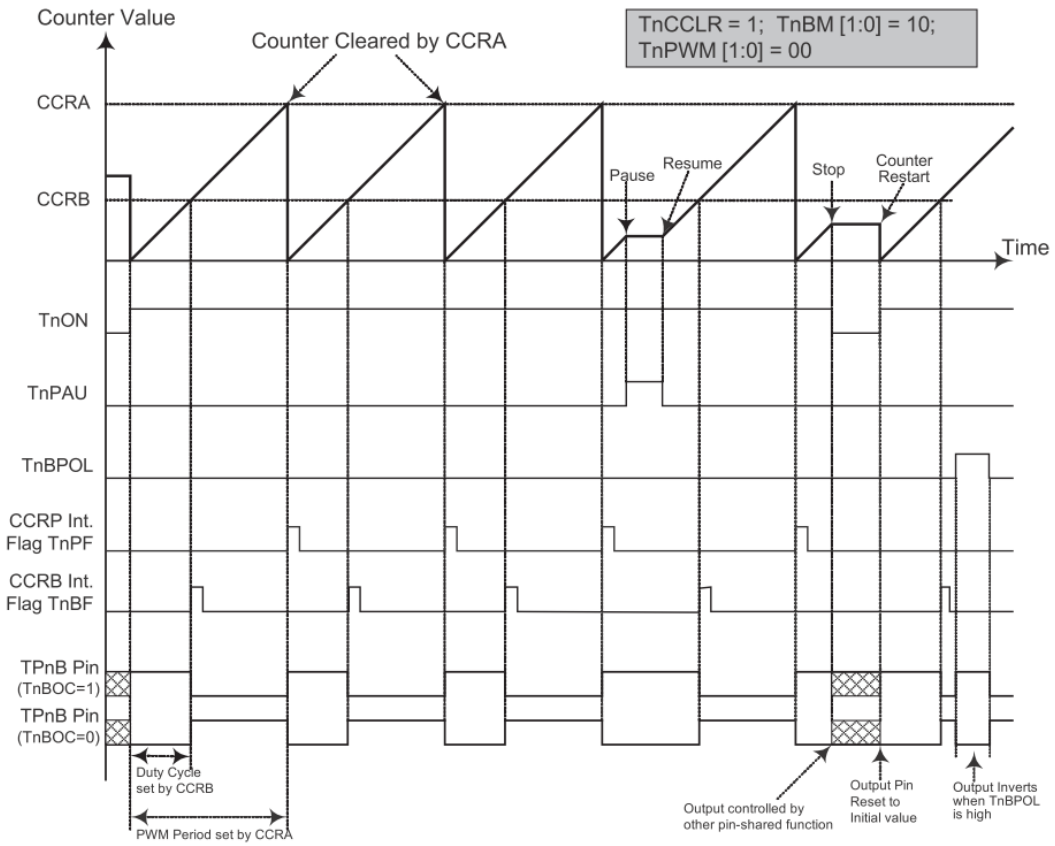
**ETM，PWM 模式，中心对齐模式，TnCCLR=1**

| CCRA | 1 | 2 | 3 | 511 | 512 | 1021 | 1022 | 1023 |
|---|---|---|---|---|---|---|---|---|
| Period | 2 | 4 | 6 | 1022 | 1024 | 2042 | 2044 | 2046 |
| B Duty | (CCRB×2) － 1 | | | | | | | |



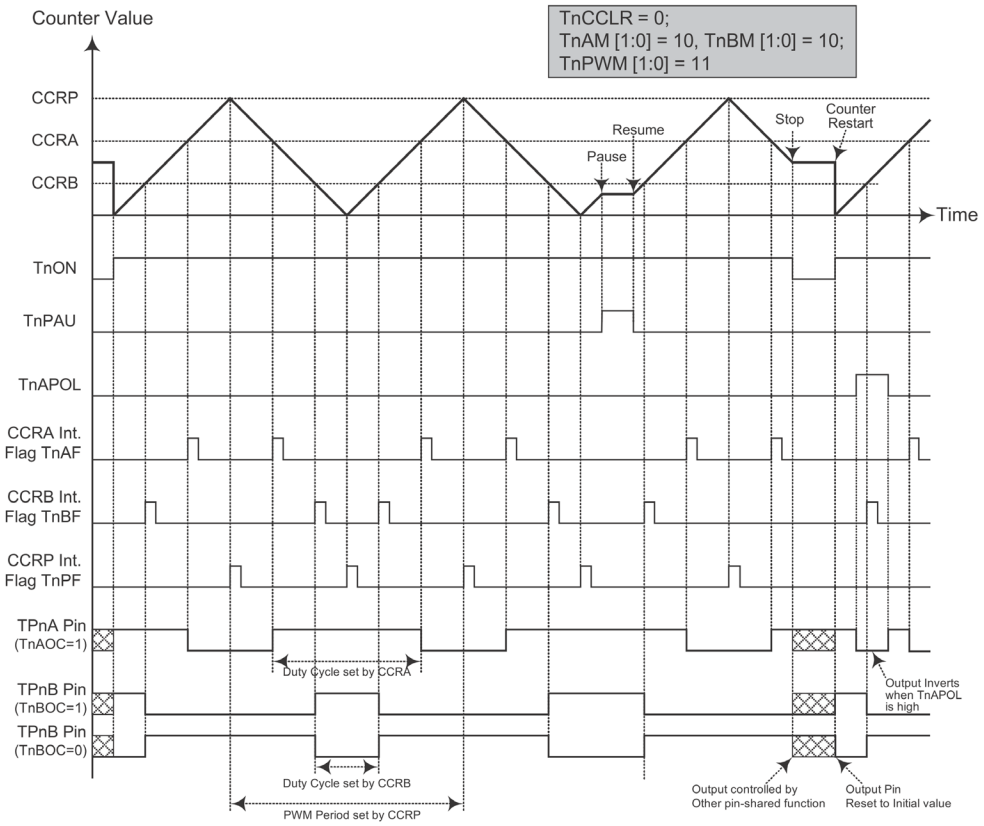

**PWM 模式 -- 边沿对齐**

注： 1. TnCCLR=0，CCRP 清除计数器并决定 PWM 周期
2. 当 TnAIO1，TnAIO0( 或 TnBIO1，TnBIO0)=00 或 01，PWM 功能不变
3. CCRA 控制 TPnA PWM 占空比，CCRB 控制 TPnB PWM 占空比



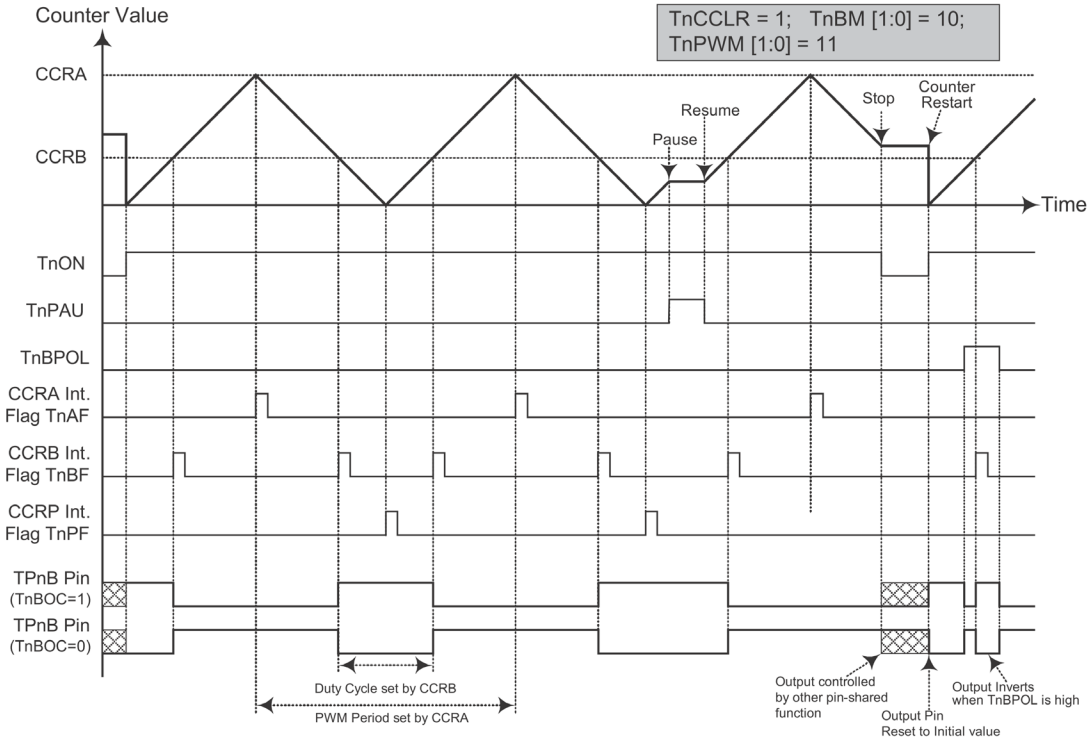

**PWM 模式 -- 边沿对齐**

注： 1. TnCCLR=1，CCRA 清除计数器并决定 PWM 周期
2. 当 TnBIO1，TnBIO0=00 或 01，PWM 功能不变
3. CCRA 控制 TPnB PWM 周期，CCRB 控制 TPnB PWM 占空比
4. 此时，TM 引脚控制寄存器不能使能 TPnA 作为 TM 输出引脚




**PWM 模式 -- 中心对齐**

注： 1. TnCCLR=0，CCRP 清除计数器并决定 PWM 周期
2. TnPWM1/TnPWM0=11，PWM 为中心对齐
3. 当 TnAIO1，TnAIO0( 或 TnBIO1，TnBIO0)=00 或 01，PWM 功能不变
4. CCRA 控制 TPnA PWM 占空比，CCRB 控制 TPnB PWM 占空比
5. 计数器值递减至“0”时 CCRP 将产生中断请求。




**PWM 模式 -- 中心对齐**

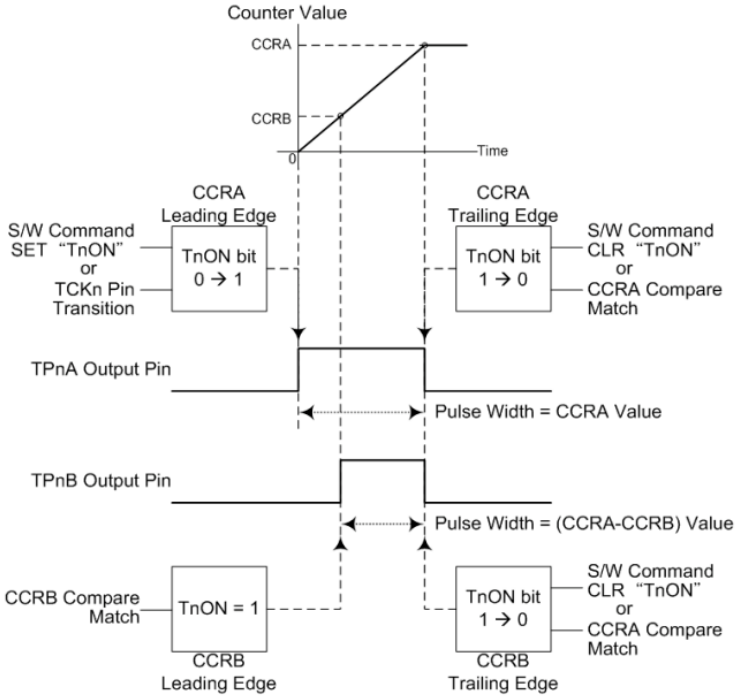
注： 1. TnCCLR=1，CCRA 清除计数器并决定 PWM 周期
2. TnPWM1/TnPWM0=11，PWM 为中心对齐
3. 当 TnBIO1，TnBIO0=00 或 01 时，PWM 功能不变
4. CCRA 控制 TPnB PWM 周期，CCRB 控制 TPnB PWM 占空比
5. 计数器值递减至“0”时 CCRP 将产生中断请求。

## 单脉冲输出模式

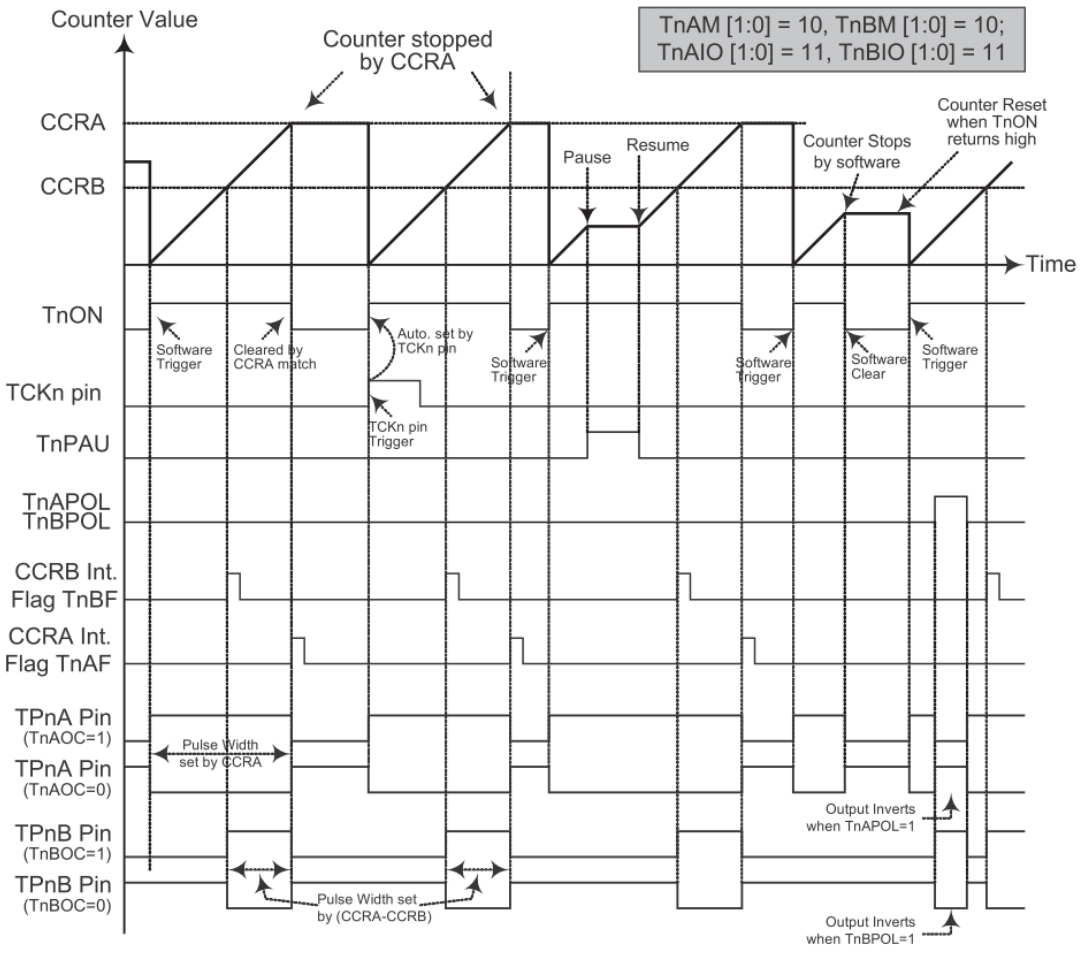
为使 TM 工作在此模式，TnAM1，TnAM0 和 TnBM1，TnBM0 位需要分别设置为“10”，并且相应的 TnAIO1，TnAIO0 和 TnBIO1，TnBIO0 需要分别设置为“11”。正如模式名所言，单脉冲输出模式，在 TM 输出脚将产生一个脉冲输出。

通过应用程序控制 TnON 位由低到高的转变来触发 TPnA 脉冲前沿输出。通过应用程序产生比较器 B 的比较匹配来触发 TPnB 脉冲前沿输出。而处于单脉冲模式时，TnON 位可由 TCKn 脚自动由低转变为高，进而依次初始化 TPnA 的单脉冲输出。当 TnON 位转变为高电平时，计数器将开始运行，并产生 TPnA 脉冲前沿。当脉冲有效时 TnON 位保持高电平。通过应用程序使 TnON 位清零或比较器 A 比较匹配发生时，产生 TPnA 和 TPnB 的脉冲下降沿。

而比较器 A 比较匹配发生时，会自动清除 TnON 位并产生 TPnA 和 TPnB 单脉冲输出下降沿。CCRA 的值通过这种方式控制 TPnA 的脉冲宽度，CCRA-CCRB 的值控制 TPnB 的脉冲宽度。比较器 A 和比较器 B 比较匹配发生时，也会产生 TM 中断。TnON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲模式中，CCRP 寄存器和 TnCCLR 位未使用。




单脉冲产生示意图




**ETM -- 单脉冲模式**

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

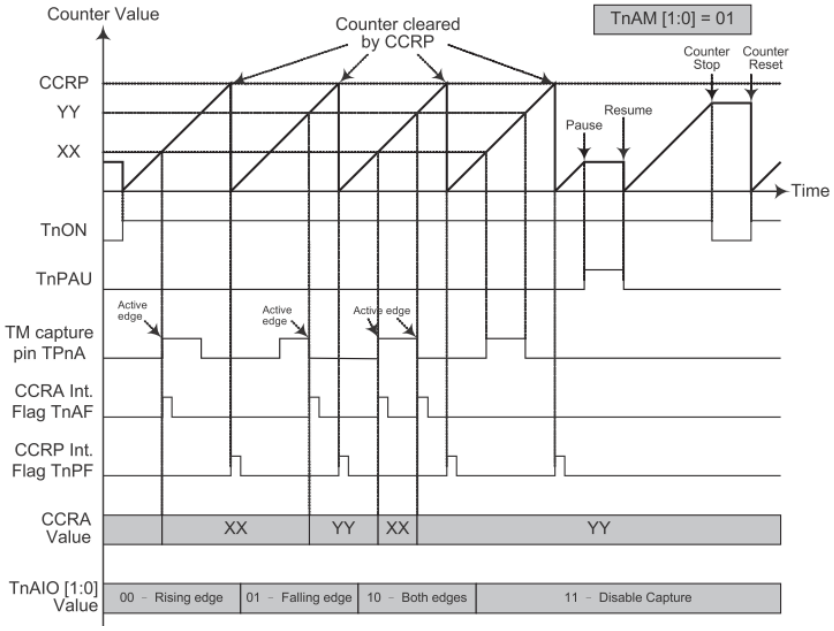
## 捕捉输入模式

为使 TM 工作在此模式，TMnC1 寄存器的 TnAM1，TnAM0 位和 TMnC2 寄存器的 TnBM1，TnBM0 位需要分别设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚上的外部信号，通过设置 TMnC1 寄存器的 TnAIO1，TnAIO0 位和 TMnC2 寄存器的 TnBIO1，TnBIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。计数器在 TnON 位由低到高转变时启动并通过应用程序初始化。

当 TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚出现有效边沿转换时，计数器当前值被锁存到 CCRA 和 CCRB 寄存器，并产生 TM 中断。不考虑 TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚事件，计数器继续工作直到 TnON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的值通过这种方式

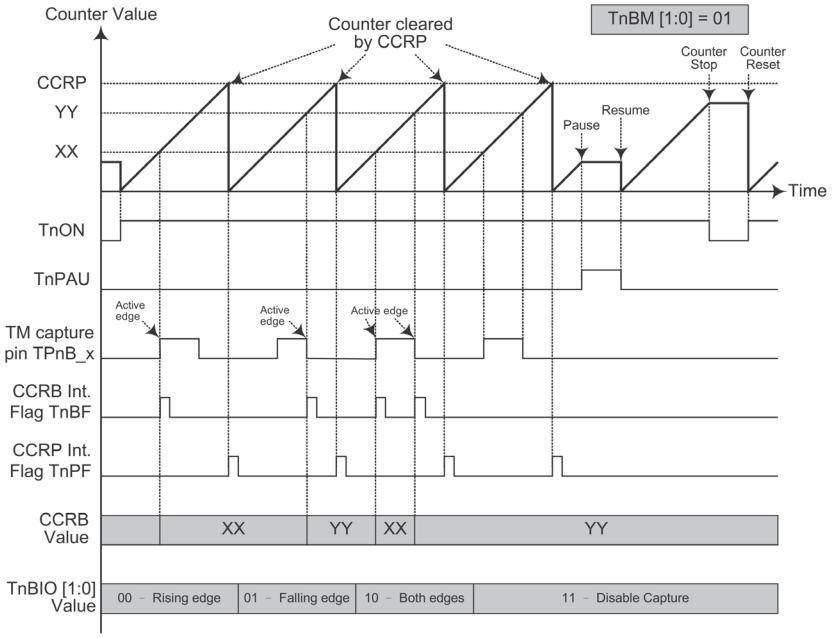
控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 TM 中断。记录 CCRP 溢出中断信号的值可以测量长脉宽。通过设置 TnAIO1，TnAIO0 位和 TnBIO1，TnBIO0 位选择 TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚为上升沿，下降沿或双沿有效。不考虑 TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚事件，如果 TnAIO1，TnAIO0 位和 TnBIO1，TnBIO0 位都设为高，不会产生捕捉操作，但计数器继续运行。

当 TPnA 和 TPnB_0，TPnB_1，TPnB_2 引脚与其它功能共用，TM 工作在输入捕捉模式时需多加注意。这是因为如果引脚被设为输出，那么该引脚上的任何电平转变都可能执行输入捕捉操作。TnCCLR，TnAOC，TnBOC，TnAPOL 和 TnBPOL 位在此模式中未使用。




**ETM CCRA 捕捉输入模式**

注： 1. TnAM1，TnAM0=01 并通过 TnAIO1 和 TnAIO0 位设置有效边沿
2. TM 捕捉输入脚的有效边沿将计数器的值转移到 CCRA 中
3. TnCCLR 位未使用
4. 无输出功能 -TnAOC 和 TnAPOL 位未使用
5. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大

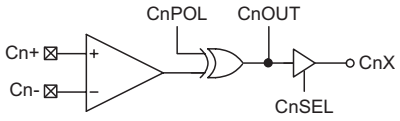



**ETM CCRB 捕捉输入模式**

注： 1. TnBM1，TnBM0=01 并通过 TnBIO1 和 TnBIO0 位设置有效边沿
2. TM 捕捉输入脚的有效边沿将计数器的值转移到 CCRB 中
3. TnCCLR 位未使用
4. 无输出功能 -TnBOC 和 TnBPOL 位未使用
5. 计数器值由 CCRP 决定，在 CCRP 为“0”时，计数器计数值可达最大

# 比较器

该系列芯片中含有两个独立的模拟比较器。它们具有暂停、极性选择、迟滞等功能，可通过寄存器进行灵活配置。比较器的引脚与普通 I/0 引脚共享，当比较器功能未使用时，此引脚可做普通引脚使用而不浪费 I/O 资源。




**比较器**

## 比较器操作

此系列单片机包含两个比较器功能，用于比较两个模拟电压，基于它们的差值上提供一个输出。控制寄存器 CP0C 和 CP1C 可分别控制相应的内部比较器。比较器的输出可由各自寄存器的一位记录，并且在共用的 I/O 口上输出。此外，比较器功能有输出极性，迟滞功能和暂停控制。

当比较器使能时，连接到与比较器共用的输入引脚的上拉电阻将自动失效。当比较器输入接近其切换电压时，由于输入信号上升或下降速度较慢，比较器输出端可能会产生一些伪输出信号。通过选择迟滞功能提供少量正反馈给比较器可使此种情况的发生率较大程度地降低。理想情况下正负输入信号在同一个电压点时比较器将发生开关动作，但是不可避免的输入失调电压会导致情况不确定。若迟滞功能使能，也可增加切换偏差值。

## 比较器寄存器

比较器工作相关的寄存器共有两个，分别对应两个比较器。两个比较器中的应答位有特殊的功能，两个寄存器列表如下。

| 寄存器名称 | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CP0C | C0SEL | C0EN | C0POL | C0OUT | C0OS | — | — | C0HYEN |
| CP1C | C1SEL | C1EN | C1POL | C1OUT | C1OS | — | — | C1HYEN |

比较器寄存器列表

### CP0C 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | C0SEL | C0EN | C0POL | C0OUT | C0OS | — | — | C0HYEN |
| R/W | R/W | R/W | R/W | R | R/W | — | — | R/W |
| POR | 1 | 0 | 0 | 0 | 0 | — | — | 1 |

Bit 7 **C0SEL**：比较器引脚或 I/O 引脚选择位
0：输入 / 输出引脚
1：比较器引脚
此位为比较器引脚或输入 / 输出引脚选择位。为高时，两个比较器输入引脚使能。此时，这两个引脚的输入 / 输出功能失效，任何与比较器共用引脚的上拉电阻配置选项将自动失效。

Bit 6 **C0EN**：比较器开 / 关控制位
0：关闭
1：开启
此位为比较器开 / 关控制位。为“0”时，比较器关闭，即使其引脚上加有模拟电压也不会产生功耗。对功耗要求严格的应用中，当比较器未使用或单片机进入休眠或空闲模式之前，此位应清零。

Bit 5 **C0POL**：比较器输出极性位
0：输出同相
1：输出反相
此位决定比较器极性。为“0”时，C0OUT 位与比较器输出条件同相；为“1”时，C0OUT 位与比较器输出条件反相。

Bit 4 **C0OUT**：比较器输出位
C0POL=0
0: C0+ < C0-
1: C0+ > C0-
C0POL=1
0: C0+ > C0-
1: C0+ < C0-
此位为比较器输出位。此位的极性由比较器输入电压和 C0POL 位的状态决定。

Bit 3 **C0OS**：输出路径选择位
0：C0X 引脚
1：内部使用
此位为比较器输出路径选择控制位。当此位为“0”且 C0SEL 位为“1”时，比较器输出连接到外部 C0X 引脚。当此位为“1”或 C0SEL 位为“0”时，比较器输出信号仅供单片机内部使用，且共用输出引脚作为普通输入 / 输出引脚使用。

Bit 2~1 未使用，读为“0”

Bit 0 **C0HYEN**：迟滞控制位
0：关闭
1：开启
此位为迟滞控制位。为“1”时，比较器有一定量迟滞，具体见比较器电气特性表。滞后产生的正反馈将减少比较器门槛附近的伪开关效应的影响。

### CP1C 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | C1SEL | C1EN | C1POL | C1OUT | C1OS | — | — | C1HYEN |
| R/W | R/W | R/W | R/W | R | R/W | — | — | R/W |
| POR | 1 | 0 | 0 | 0 | 0 | — | — | 1 |

Bit 7 **C1SEL**：比较器引脚或输入 / 输出引脚选择位
0：输入 / 输出引脚
1：比较器引脚
此位为比较器引脚或输入 / 输出引脚选择位。为高时，两个比较器输入引脚使能。此时，这两个引脚的输入 / 输出功能失效，任何与比较器共用引脚的上拉电阻配置选项将自动失效。

Bit 6 **C1EN**：比较器开 / 关控制位
0：关闭
1：开启
此位为比较器开 / 关控制位。为“0”时，比较器关闭，即使其引脚上加有模拟电压也不会产生功耗。对功耗要求严格的应用中，当比较器未使用或单片机进入休眠或空闲模式之前，此位应清零。

Bit 5 **C1POL**：比较器输出极性位
0：输出同相
1：输出反相
此位决定比较器极性。为“0”时，C1OUT 位与比较器输出条件同相；为“1”时，C1OUT 位与比较器输出条件反相。

Bit 4 **C1OUT**：比较器输出位
C1POL=0
0: C1+ < C1-
1: C1+ > C1-
C1POL=1
0: C1+ > C1-
1: C1+ < C1-
此位为比较器输出位。此位的极性由比较器输入电压和 C1POL 位的状态决定。

Bit 3 **C1OS**：输出路径选择位
0：C1X 引脚
1：内部使用
此位为比较器输出路径选择控制位。当此位为“0”且 C1SEL 位为“1”时，比较器输出连接到外部 C1X 引脚。当此位为“1”或 C1SEL 位为“0”时，比较器输出信号仅供单片机内部使用，且共用输出引脚作为普通输入 / 输出引脚使用。

Bit 2~1 未使用，读为“0”

Bit 0 **C1HYEN**：迟滞控制位
0：关闭
1：开启
此位为迟滞控制位。为“1”时，比较器有一定量迟滞，具体见比较器电气特性表。滞后产生的正反馈将减少比较器门槛附近的伪开关效应的影响。

### 比较器中断

每个比较器都有自己的中断功能。当任何一个状态改变时，其对应的中断标志将会置位，若应答中断使能位被置位，系统将跳转至相应的中断向量中执行。注意，触发比较器产生中断的条件是 C0OUT 或 C1OUT 位状态的改变而非比较器输出引脚状态的改变。单片机处于休眠或空闲模式且比较器使能时，若为外部输入线导致比较器输出状态发生的改变，则由此产生的中断标志也可产生一个唤醒动作。若除能唤醒功能，进入休眠或空闲模式前中断标志应先置为高。

### 编程注意事项

若比较器使能，当单片机进入休眠或空闲模式时其仍保持有效并会有一定的耗电，用户可考虑在进入休眠或空闲模式前先关闭比较器。

由于比较器引脚与普通输入 / 输出脚共用，若比较器功能使能时，这些引脚的输入 / 输出寄存器将读为“0”（端口控制寄存器读为“1”）或读为端口数据寄存器的值（端口控制寄存器读为“0”）。

## 串行接口模块 – SIM

该单片机内有一个串行接口模块，包括两种易与外部设备通信的串行接口：四线 SPI 或两线 $I^2C$ 接口。这两种接口具有相当简单的通信协议，单片机可以通过这些接口与传感器、闪存或 EEPROM 内存等硬件设备通信。SIM 接口的引脚与其它的 I/O 引脚共用，所以要使用 SIM 功能时应先在配置选项中选中 SIM 功能。因为这两种接口共用引脚和寄存器，所以要通过一个 SIMC0 寄存器中的 SIM2~SIM0 位来选择哪一种通信接口。若 SIM 功能使能，可通过上拉电阻控制寄存器选择与输入 / 输出口共用的 SIM 脚的上拉电阻。

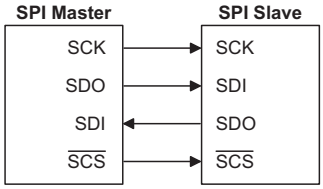
### SPI 接口

SPI 接口常用于与外部设备如传感器、闪存或 EEPROM 内存等通信。四线 SPI 接口最初是由摩托罗拉公司研制，是一个有相当简单的通信协议的串行数据接口，这个协议可以简化与外部硬件的编程要求。

SPI 通信模式为全双工模式，且能以主 / 从模式的工作方式进行通信，单片机既可以做为主机，也可以做为从机。虽然 SPI 接口理论上允许一个主机控制多个从机，但此处的 SPI 中只有一个片选信号引脚。若主机需要控制多个从机，可使用输入 / 输出引脚选择从机。

### SPI 接口操作

SPI 接口是一个全双工串行数据传输器。SPI 接口的四线为：SDI、SDO、SCK 和 $\overline{SCS}$。SDI 和 SDO 是数据的输入和输出线。SCK 是串行时钟线，$\overline{SCS}$ 是从机的选择线。SPI 的接口引脚与普通 I/O 口和 $I^2C$ 的功能脚共用。通过设定 SIM 配置选项和 SIMC0/SIMC2 寄存器的对应位，来使能 SPI 接口。SPI 配置选项设定好后，还可以通过 SIMC0 寄存器中的 SIMEN 位来除能或使能。连接到 SPI 接口的单片机以从主 / 从模式进行通信，且主机完成所有的数据传输初始化，并控制时钟信号。由于单片机只有一个 $\overline{SCS}$ 引脚，所以只能拥有一个从机设备。可通过软件控制 $\overline{SCS}$ 引脚使能与除能，设置 CSEN 位为“1”使能 $\overline{SCS}$ 功能，设置 CSEN 位为“0” $\overline{SCS}$ 引脚将处于浮空状态。



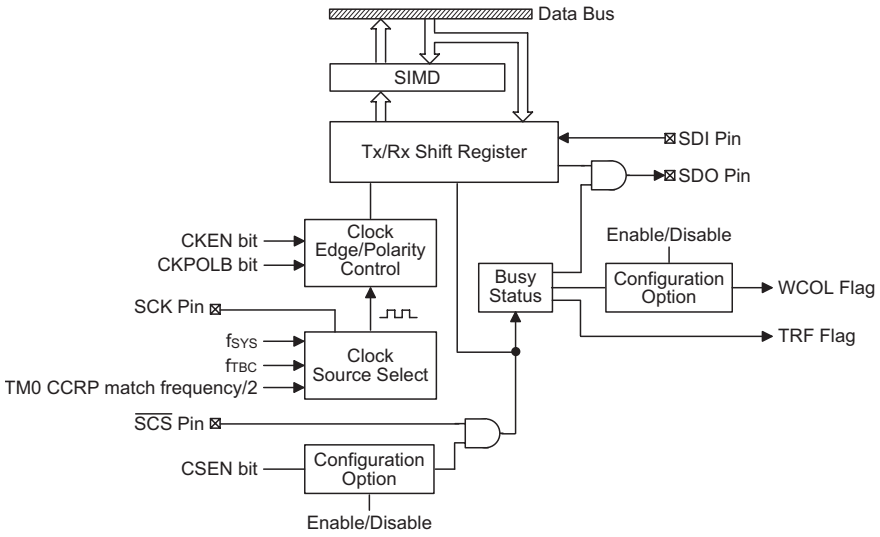

**SPI 主 / 从机连接方式**

该系列单片机的 SPI 功能具有以下特点：

- 全双工同步数据传输
- 主从模式
- 最低有效位先传或最高有效位先传的数据传输模式
- 传输完成标志位
- 时钟源上升沿或下降沿有效
- WCOL 和 CSEN 位使能或除能选择

SPI 接口状态受很多因素的影响，如单片机处于主机或从机的工作模式和 CSEN，SIMEN 位的状态。

配置选项中有几项与 SPI 接口功能相关。其中一项为使能 SIM 功能，共用引脚选择为 SIM 脚而非普通输入 / 输出脚。注意，若配置选项未选择 SIM 功能，SIMC0 寄存器中的 SIMEN 位的状态不会产生影响。另外两个 SPI 配置选项决定 CSEN 和 WCOL 位是否有用。




**SPI 方框图**

## SPI 寄存器

有三个内部寄存器用于控制 SPI 接口的所有操作，其中有一个数据寄存器 SIMD、两个控制寄存器 SIMC0 和 SIMC2。注意，SIMC1 寄存器仅用于 $I^2C$ 接口。

| 寄存器名称 | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SIMC0 | SIM2 | SIM1 | SIM0 | PCKEN | PCKP1 | PCKP0 | SIMEN | — |
| SIMD | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| SIMC2 | D7 | D6 | CKPOLB | CKEG | MLS | CSEN | WCOL | TRF |

**SIM 寄存器列表**

SIMD 用于存储发送和接收的数据。这个寄存器由 SPI 和 $I^2C$ 功能所共用。在单片机尚未将数据写入到 SPI 总线中时，要传输的数据应先存在 SIMD 中。SPI 总线接收到数据之后，单片机就可以从 SIMD 数据寄存器中读取。所有通过 SPI 传输或接收的数据都必须通过 SIMD 实现。

### SIMD 寄存器

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

“×”为未知

单片机中也有两个控制 SPI 接口功能的寄存器，SIMC0 和 SIMC2。应注意的是 SIMC2 与 $I^2C$ 接口功能中的的寄存器 SIMA 是同一个寄存器。SPI 功能不会用到寄存器 SIMC1，SIMC1 只适用于 $I^2C$ 中。寄存器 SIMC0 用于控制使能 / 除能功能和设置数据传输的时钟频率。虽然 SIMC0 与 SPI 功能关，但是也用于控制外部时钟分频。寄存器 SIMC2 用于其它的控制功能如 LSB/MSB 选择，写冲突标志位等。

### SIMC0 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SIM2 | SIM1 | SIM0 | PCKEN | PCKP1 | PCKP0 | SIMEN | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | — |
| POR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | — |

Bit 7~5 **SIM2~SIM0**：SIM 工作模式控制位
000：SPI 主机模式；SPI 时钟为 $f_{SYS}/4$
001：SPI 主机模式；SPI 时钟为 $f_{SYS}/16$
010：SPI 主机模式；SPI 时钟为 $f_{SYS}/64$
011：SPI 主机模式；SPI 时钟为 $f_{TBC}$
100：SPI 主机模式；SPI 时钟为 TM0 CCRP 匹配频率 /2
101：SPI 从机模式
110：$I^2C$ 从机模式
111：未使用模式
这几位用于设置 SIM 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率及 $I^2C$ 或 SPI 功能。SPI 时钟源可来自于系统时钟也可以选择来自 TM0。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 **PCKEN**：PCK 输出脚控制位
0：除能
1：使能

Bit 3~2 **PCKP1,PCKP0**：选择 PCK 输出脚的频率位
00：$f_{SYS}$
01：$f_{SYS}/4$
10：$f_{SYS}/8$
11：TM0 CCRP 匹配频率 /2

Bit 1 **SIMEN**：SIM 控制位
0：除能
1：使能
此位为 SIM 接口的开 / 关控制位。此位为“0”时，SIM 接口除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 脚处于浮空状态，SIM 工作电流减小到最小值。此位为“1”时，SIM 接口使能。配置选项中首先将 SIM 接口使能才能使此位有效。若 SIM 经由 SIM2~SIM0 位设置为工作在 SPI 接口，当 SIMEN 位由低到高转变时，SPI 控制寄存器中的设置不会发生变化，其首先应在应用程序中初始化。若 SIM 经由 SIM2~SIM0 位设置为工作在 $I^2C$ 接口，当 SIMEN 位由低到高转变时，$I^2C$ 控制寄存器中的设置，如 HXT 和 TXAK，将不会发生变化，其首先应在应用程序中初始化，此时相关 $I^2C$ 标志，如 HCF、HAAS、HBB、SRW 和 RXAK，将被设置为其默认状态。

Bit 0 未使用，读为“0”

## SIMC2 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | D7 | D6 | CKPOLB | CKEG | MLS | CSEN | WCOL | TRF |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 未定义位
用户可通过软件程序对这两位进行读写。

Bit 5 **CKPOLB**：时钟线的基础状态位
0：当时钟无效时，SCK 口为高电平
1：当时钟无效时，SCK 口为低电平
此位决定了时钟线的基础状态，当时钟无效时，若此位为高，SCK 为低电平，若此位为低，SCK 为高电平。

Bit 4 **CKEG**：SPI 的 SCK 有效时钟边沿类型位
CKPOLB=0
0：SCK 为高电平且在 SCK 上升沿抓取数据
1：SCK 为高电平且在 SCK 下降沿抓取数据
CKPOLB=1
0：SCK 为低电平且在 SCK 下降沿抓取数据
1：SCK 为低电平且在 SCK 上升沿抓取数据
CKEG 和 CKPOLB 位用于设置 SPI 总线上时钟信号输入和输出方式。在执行数据传输前，这两位必须被设置，否则将产生错误的时钟边沿信号。CKPOLB 位决定时钟线的基本状态，若时钟无效且此位为高，则 SCK 为低电平，若时钟无效且此位为低，则 SCK 为高电平。CKEG 位决定有效时钟边沿类型，取决于 CKPOLB 的状态。

Bit 3 **MLS**：SPI 数据移位命令位
0：LSB
1：MSB
数据移位选择位，用于选择数据传输时高位优先传输还是低位优先传输。此位设置为高时高位优先传输，为低时低位优先传输。

Bit 2 **CSEN**：SPI $\overline{SCS}$ 引脚控制位
0：除能
1：使能
CSEN 位用于 $\overline{SCS}$ 引脚的使能 / 除能控制。此位为低时，$\overline{SCS}$ 除能并处于浮空状态。此位为高时，SCS 作为选择脚。注意，CSEN 位的使能 / 除能可通过配置选项设置。

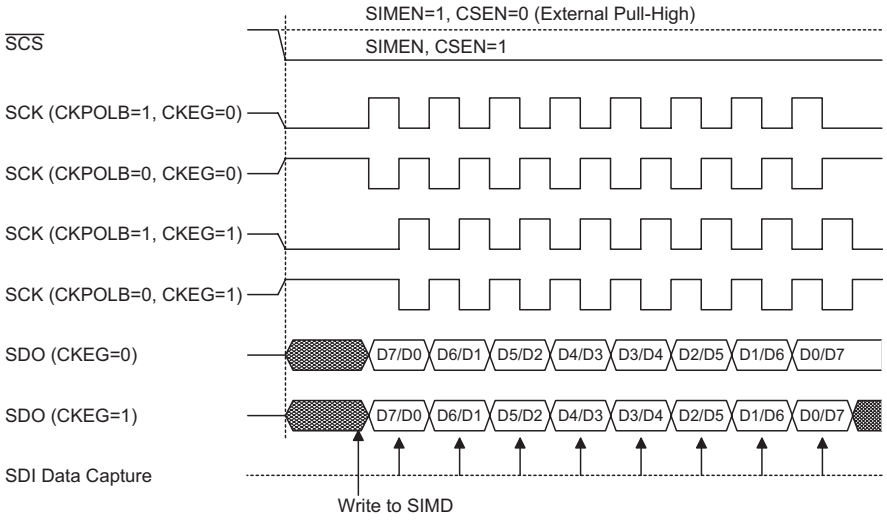
Bit 1 **WCOL**：SPI 写冲突标志位
0：无冲突
1：冲突
WCOL 标志位用于监测数据冲突的发生。此位为高时，数据在传输时被写入 SIMD 寄存器。若数据正在被传输时，此操作无效。此位可被应用程序清零。注意，WCOL 位的使能 / 除能可通过配置选项设置。

Bit 0 **TRF**：SPI 发送 / 接收结束标志位
0：数据正在发送
1：数据发送结束
TRF 位为发送 / 接收结束标志位，当 SPI 数据传输结束时，此位自动置为高，但须通过应用程序设置为“0”。此位也可用于产生中断。

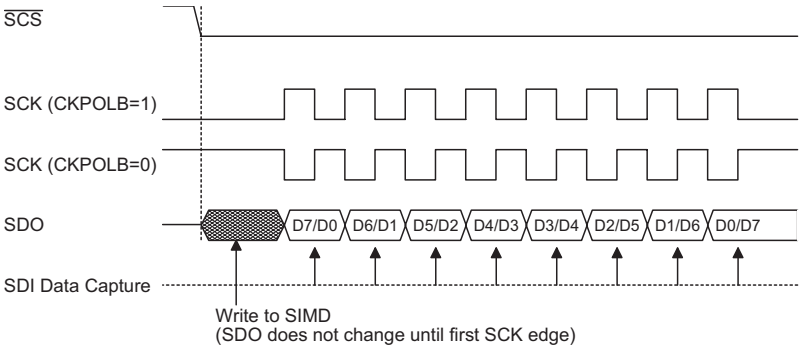
## SPI 通信

将 SIMEN 设置为高，使能 SPI 功能之后，单片机处于主机模式，当数据写入到寄存器 SIMD 的同时传输 / 接收开始进行。数据传输完成时，TRF 位将自动被置位但清除只能通过应用程序完成。单片机处于从机模式时，收到主机发来的信号之后，会传输 SIMD 中的数据，而且在 SDI 引脚上的数据也会被移位到 SIMD 寄存器中。主机应在输出时钟信号之前先输出一个 $\overline{SCS}$ 信号以使能从机，从机的数据传输功能也应在与 $\overline{SCS}$ 信号相关的适当时候准备就绪，这由 CKPOLB 和 CKEG 位决定。所附时序图表明了在 CKPOLB 和 CKEG 位各种设置情况下从机数据与 $\overline{SCS}$ 信号的关系。

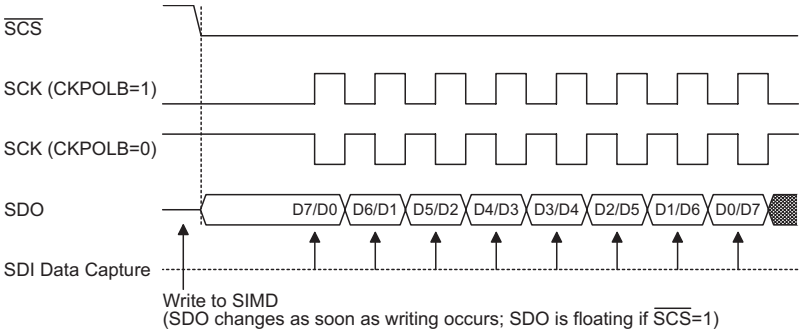
即使在单片机处于空闲模式，SPI 功能仍将继续执行。




**SPI 主机模式时序**

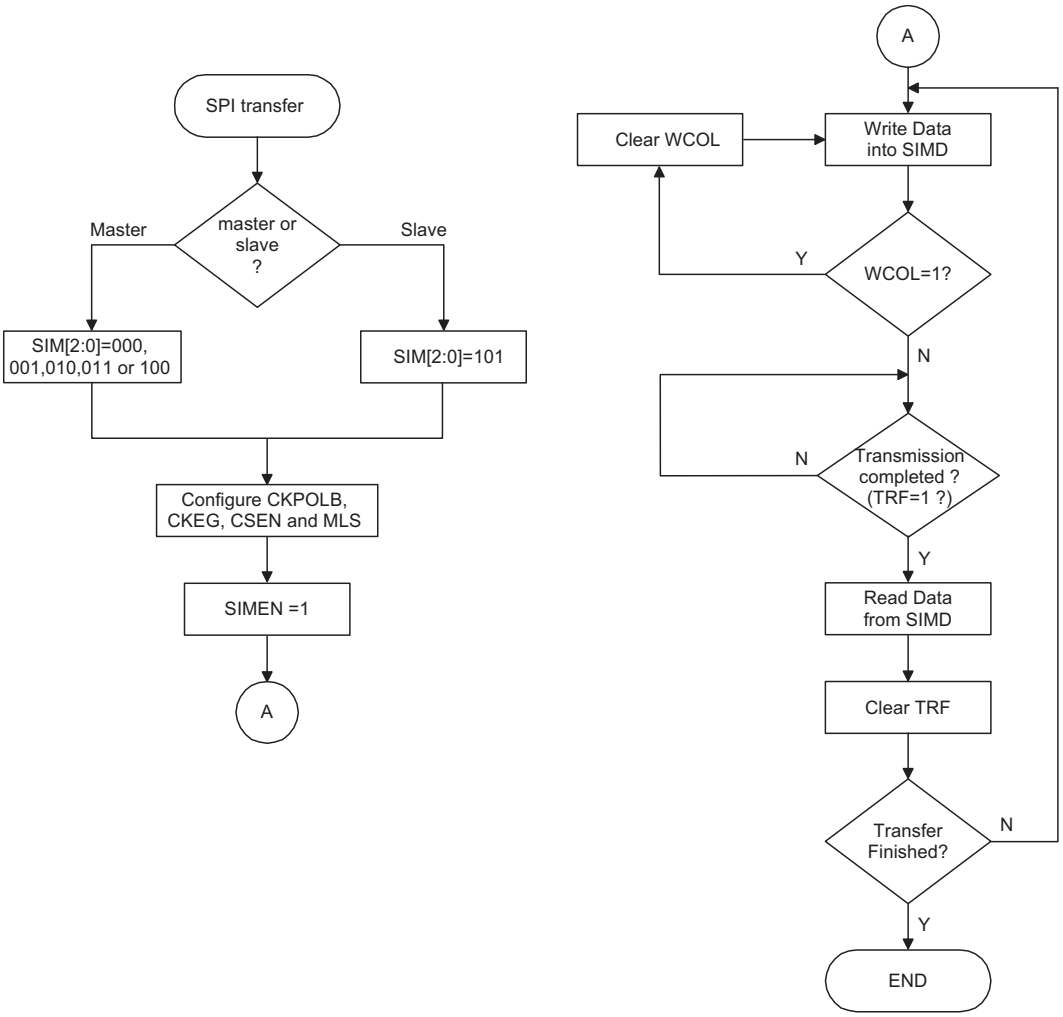



**SPI 从机模式时序 -- CKEG=0**




Note: For SPI slave mode, if SIMEN=1 and CSEN=0, SPI is always enabled and ignores the $\overline{SCS}$ level.

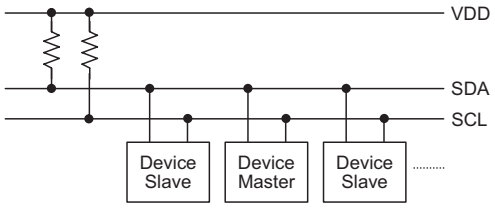
**SPI 从机模式时序 -- CKEG=1**




**SPI 传输控制流程图**

## I²C 接口

I²C 可以和传感器，EEPROM 内存等外部硬件接口进行通信。最初是由飞利浦公司研制，是适用于同步串行数据传输的双线式低速串行接口。I²C 接口具有两线通信，非常简单的通信协议和在同一总线上和多个设备进行通信的能力的优点，使之在很多的应用场合中大受欢迎。




**I²C 主从总线连接图**

## I²C 接口操作

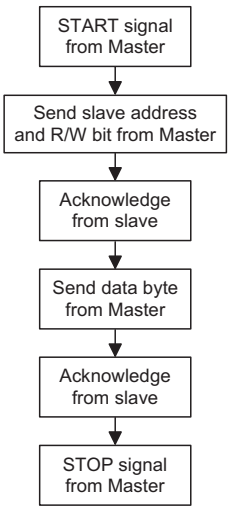
I²C 串行接口是一个双线的接口，有一条串行数据线 SDA 和一条串行时钟线 SCL。由于可能有多个设备在同一条总线上相互连接，所以这些设备的输出都是开漏型输出。因此应在这些输出口上都应加上拉电阻。应注意的是：I²C 总线上的每个设备都没有选择线，但分别与唯一的地址一一对应，用于 I²C 通信。

如果有两个设备通过双向的 I²C 总线进行通信，那么就存在一个主机和一个从机。主机和从机都可以用于传输和接收数据，但只有主机才可以控制总线动作。那些处于从机模式的设备，要在 I²C 总线上传输数据只有两种方式，一是从机发送模式，二是从机接收模式。

单片机中有几个和 I²C 接口相关的配置选项，其中之一是使能 I²C 功能，选择 SIM 引脚而非普通 I/O 口。应注意的是如果在配置选项中没有将 SIM 功能使能，那么寄存器 SIMC0 中的 SIMEN 位不起作用。配置选项中有一个选项允许使用其它的时钟来驱动 I²C 接口。另外有一个配置选项用于控制 I²C 接口的去抖时间大小。这个功能可以使用内部时钟在外部时钟上增加一个去抖间隔，会减小时钟线上毛刺发生的可能性，以避免单片机发生误动作。如果选择了这个功能，去抖时间可以选择 2 个或 4 个系统时钟。为了达到需要的 I²C 数据传输速度，系统时钟 $f_{SYS}$ 和 I²C 去抖时间之间存在一定的关系。I²C 标准模式或者快速模式下，用户需注意所选的系统时钟频率与标准匹配去抖时间的设置，其具体关系如下表所示。

| I²C 去抖时间选择 | I²C 标准模式 (100kHz) | I²C 快速模式 (400kHz) |
|---|---|---|
| 无去抖时间 | $f_{SYS}$ > 2MHz | $f_{SYS}$ > 5MHz |
| 2 个系统时钟去抖时间 | $f_{SYS}$ > 4MHz | $f_{SYS}$ > 10MHz |
| 4 个系统时钟去抖时间 | $f_{SYS}$ > 8MHz | $f_{SYS}$ > 20MHz |

**I²C 最小 $f_{SYS}$ 频率**



## I²C 寄存器

I²C 总线的三个控制寄存器是 SIMC0 和 SIMC1，SIMA 及一个数据寄存器 SIMD。SIMD 寄存器，SPI 章节中已有介绍，用于存储正在传输和接收的数据，当单片机将数据写入 I²C 总线之前，实际将被传输的数据存放在寄存器 SIMD 中。从 I²C 总线接收到数据之后，单片机就可以从寄存器 SIMD 中得到这个数据。I²C 总线上的所有传输或接收到的数据都必须通过 SIMD。应注意的是 SIMA 也有另外一个名字，SIMC2，使用 SPI 功能时会用到。I²C 接口会用到寄存器 SIMC0 中的 SIMEN 位和 SIM0~SIM2 位。

| 寄存器名称 | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SIMC0 | SIM2 | SIM1 | SIM0 | PCKEN | PCKP1 | PCKP0 | SIMEN | — |
| SIMC1 | HCF | HAAS | HBB | HTX | TXAK | SRW | IAMWU | RXAK |
| SIMD | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| SIMA | IICA6 | IICA5 | IICA4 | IICA3 | IICA2 | IICA1 | IICA0 | D0 |

**I²C 寄存器列表**

### SIMC0 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SIM2 | SIM1 | SIM0 | PCKEN | PCKP1 | PCKP0 | SIMEN | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | — |
| POR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | — |

Bit 7~5 **SIM2~SIM0**：SIM 工作模式控制位
000：SPI 主机模式；SPI 时钟为 $f_{SYS}/4$
001：SPI 主机模式；SPI 时钟为 $f_{SYS}/16$
010：SPI 主机模式；SPI 时钟为 $f_{SYS}/64$
011：SPI 主机模式；SPI 时钟为 $f_{TBC}$
100: SPI 主机模式；SPI 时钟为 TM0 CCRP 匹配频率 /2
101：SPI 从机模式
110：I²C 从机模式
111：未使用模式
这几位用于设置 SIM 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率及 I²C 或 SPI 功能。SPI 时钟源可来自于系统时钟也可以选择来自 TM0。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 **PCKEN**：PCK 输出脚控制位
0：除能
1：使能

Bit 3~2 **PCKP1,PCKP0**：选择 PCK 输出脚的频率位
00：$f_{SYS}$
01：$f_{SYS}/4$
10：$f_{SYS}/8$
11：TM0 CCRP 匹配频率 /2

Bit 1 **SIMEN**：SIM 控制位
0：除能
1：使能
此位为 SIM 接口的开 / 关控制位。此位为“0”时，SIM 接口除能，SDI、SDO、SCK 和 $\overline{SCS}$ 或 SDA 和 SCL 脚处于浮空状态，SIM 工作电流减小到最小值。此位为“1”时，SIM 接口使能。配置选项中首先将 SIM 接口使能才能使此位有效。若 SIM 经由 SIM2~SIM0 位设置为工作在 SPI 接口，当 SIMEN 位由低到高转变时，SPI 控制寄存器中的设置不会发生变化，其首先应在应用程序中初始化。若 SIM 经由 SIM2~SIM0 位设置为工作在 I²C 接口，当 SIMEN 位由低到高转变时，I²C 控制寄存器中的设置，如 HXT 和 TXAK，将不会发生变化，其首先应在应用程序中初始化，此时相关 I²C 标志，如 HCF、HAAS、HBB、SRW 和 RXAK，将被设置为其默认状态。

Bit 0 未使用，读为“0”

### SIMC1 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | HCF | HAAS | HBB | HTX | TXAK | SRW | IAMWU | RXAK |
| R/W | R | R | R | R/W | R/W | R | R/W | R |
| POR | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Bit 7 **HCF**：$I^2C$ 总线数据传输结束标志位
0：数据正在被传输
1：8 位数据传输完成
数据正在传输时该位为低。当 8 位数据传输完成时，此位为高并产生一个中断。

Bit 6 **HAAS**：$I^2C$ 地址匹配标志位
0：地址不匹配
1：地址匹配
此标志位用于决定从机地址是否与主机发送地址相同。若地址匹配此位为高，否则此位为低。

Bit 5 **HBB**：$I^2C$ 总线忙标志位
0：$I^2C$ 总线闲
1：$I^2C$ 总线忙
当检测到 START 信号时 $I^2C$ 忙，此位变为高电平。当检测到 STOP 信号时 $I^2C$ 总线停止，该位变为低电平。

Bit 4 **HTX**：从机处于发送或接收模式标志位
0：从机处于接收模式
1：从机处于发送模式

Bit 3 **TXAK**：$I^2C$ 总线发送确认标志位
0：从机发送确认标志
1：从机没有发送确认标志
单片机接收 8 位数据之后会将该位在第九个时钟传到总线上。如果单片机想要接收更多的数据，则应在接收数据之前将此位设置为“0”。

Bit 2 **SRW**：$I^2C$ 从机读 / 写位
0：从机应处于接收模式
1：从机应处于发送模式
SRW 位是从机读写位。决定主机是否希望传输或接收来自 $I^2C$ 总线的数据。当传输地址和从机的地址相同时，HAAS 位会被设置为高，主机将检测 SRW 位来决定进入发送模式还是接收模式。如果 SRW 位为高时，主机会请求从总线上读数据，此时设备处于传输模式。当 SRW 位为“0”时，主机往总线上写数据，设备处于接收模式以读取该数据。

Bit 1 **IAMWU**：$I^2C$ 地址匹配唤醒控制位
0：除能
1：使能
此位应设置为“1”使能 $I^2C$ 地址匹配以使系统从休眠或空闲模式中唤醒。若进入休眠或空闲模式前 IAMWU 已经设置以使能 $I^2C$ 地址匹配唤醒功能，在系统唤醒后须软件清除此位以确保单片机正确地运行。

Bit 0 **RXAK**：$I^2C$ 总线接收确认标志位
0：从机接收到确认标志
1：从机没有接收到确认标志
RXAK 位是接收确认标志位。如果 RXAK 位被重设为“0”即 8 位数据传输之后，设备在第九个时钟有接受到一个正确的确认位。如果单片机处于发送状态，发送方会检查 RXAK 位来判断接收方是否愿意继续接收下一个字节。因此直到 RXAK 为“1”时，传输方停止发送数据。这时，传输方将释放 SDA 线，主机发出停止信号。

SIMD 用于存储发送和接收的数据。这个寄存器由 SPI 和 I²C 功能所共用。在单片机尚未将数据写入到 SPI 总线中时，要传输的数据应存在 SIMD 中。SPI 总线接收到数据之后，单片机就可以从 SIMD 数据寄存器中读取。所有通过 SPI 传输或接收的数据都必须通过 SIMD 实现。

**SIMD 寄存器**

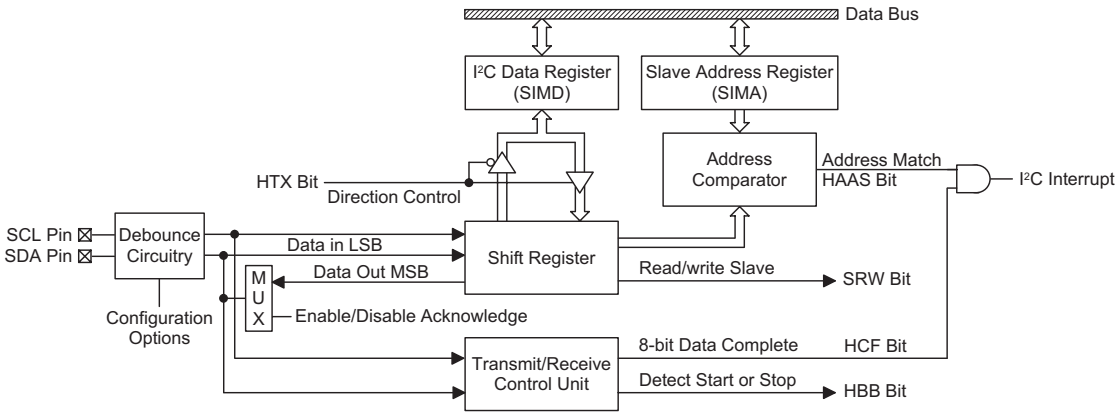
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | × | × | × | × | × | × | × | × |

**SIMA 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | IICA6 | IICA5 | IICA4 | IICA3 | IICA2 | IICA1 | IICA0 | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | — |
| POR | × | × | × | × | × | × | × | — |

Bit 7~1 **IICA6~IICA0**：I²C 从机地址位
IICA6~IICA0 是从机地址对应的 6~0 位。此寄存器也在 SPI 接口功能中使用，但其名称改为 SIMC2。SIMA 寄存器用于存放 7 位从机地址，寄存器 SIMA 中的第 7~1 位是单片机的从机地址，位 0 未定义。如果接至 I²C 的主机发送处的地址和寄存器 SIMA 中存储的地址相符，那么就选中了这个从机。应注意的是寄存器 SIMA 和 SPI 接口使用的寄存器 SIMC2 是同一个寄存器。

Bit 0 无定义
此位可通过软件程序进行读写。



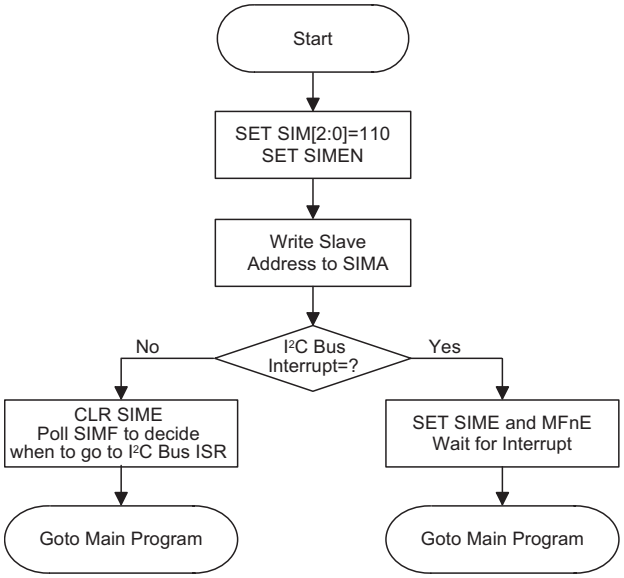

**I²C 方框图**

## I²C 总线通信

I²C 总线上的通信需要四步完成，一个起始信号，一个从机地址发送，一个数据传输，还有一个停止信号。当起始信号被写入 I²C 总线时，总线上的所有从机都会接收到这个起始信号并且被通知总线上会即将有数据到达。数据的前 7 位是从机地址，高位在前，低位在后。如果发出的地址和从机地址匹配，SIMC1 寄存器的 HAAS 位会被置位，同时产生 I²C 中断。进入中断服务程序后，系统要检测 HAAS 位，以判断 I²C 总线中断是来自从机地址匹配，还是来自 8 位数据传递完毕。在数据传递中，注意的是，在 7 位从机地址被发送后，接下来的一位，即第 8 位，是读 / 写控制位，该位的值会反映到 SRW 位中。从机通过检

测 SRW 位以确定主控制器是要进入发送模式还是接收模式。在 I²C 总线开始传送数据前，需要先初始化 I²C 总线，初始化 I²C 总线步骤如下：

- 步骤 1
  设置 SIMC0 寄存器中 SIM2~SIM0 和 SIMEN 位为“1”，以使能 I²C 总线
- 步骤 2
  向 I²C 总线地址寄存器 SIMA 写入从机地址。
- 步骤 3
  设置 SIME 位和中断控制寄存器中的 SIM 多功能中断使能位，以使能 SIM 中断和多功能中断。




**I²C 总线初始化流程图**

## I²C 总线起始信号

起始信号只能由连接 I²C 总线主机产生，而不是由只做从机的 MCU 产生。总线上的所有从机都可以侦测到起始信号。如果有从机侦测到起始信号，则表明 I²C 总线处于忙碌状态，并会置位 HBB。起始信号是指在 SCL 为高电平时，SDA 线上发生从高到低的电平变化。

## 从机地址

总线上的所有从机都会侦测由主机发出的起始信号。发送起始信号后，紧接着主机会发送从机地址以选择要进行数据传输的从机。所有在 I²C 总线上的从机接收到 7 位地址数据后，都会将其与各自内部的地址进行比较。如果从机从主机上接收到的地址与自身内部的地址相匹配，则会产生一个 I²C 总线中断信号。地址位接下来的一位为读 / 写状态位 ( 即第 8 位 )，将被保存到 SIMC1 寄存器的 SRW 位，随后发出一个低电平应答信号（即第 9 位）。当单片机从机的地址匹配时，会将状态标志位 HAAS 置位。I²C 总线有两个中断源，当程序运行至中断服务子程序时，通过检测 HAAS 位以确定 I²C 总线中断是来自从机地址匹配，还是来自 8 位数据传递完毕。当是从机地址匹配发生中断时，则从机或是用于发送模式并将数据写进 SIMD 寄存器，或是用于接收模式并从 SIMD 寄存器中读取空值以释放 SCL 线。

## I²C 总线读 / 写信号

SIMC1 寄存器的 SRW 位用来表示主机是要从 I²C 总线上读取数据还是要将数据写到 I²C 总线上。从机则通过检测该位以确定自己是作为发送方还是接收方。当 SRW 置“1”，表示主机要从 I²C 总线上读取数据，从机则作为发送方，将数据写到 I²C 总线；当 SRW 清“0”，表示主机要写数据到 I²C 总线上，从机则做为接收方，从 I²C 总线上读取数据。

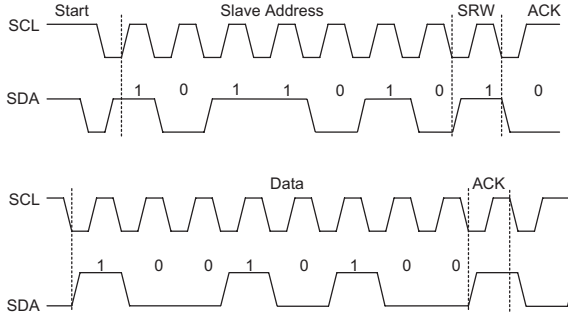
## I²C 总线从机地址确认信号

主机发送呼叫地址后，当 I²C 总线上的任何从机内部地址与其匹配时，会发送一个应答信号。此应答信号会通知主机有从机已经接收到了呼叫地址。如果主机没有收到应答信号，则主机必须发送停止 (STOP) 信号以结束通信。当 HAAS 为高时，表示从机接收到的地址与自己内部地址匹配，则从机需检查 SRW 位，以确定自己是作为发送方还是作为接收方。如果 SRW 位为高，从机须设置成发送方，这样会置位 SIMC1 寄存器的 HTX 位。如果 SRW 位为低，从机须设置成接收方，这样会清零 SIMC1 寄存器的 HTX 位。

## I²C 总线数据和确认信号

在从机确认接收到从地址后，会进行 8 位宽度的数据传输。这个数据传输顺序是的高位在前，低位在后。接收方在接收到 8 位数据后必须发出一个应答信号 (“0”) 以继续接收下一个数据。如果发送方没接收到应答信号，发送方将释放 SDA 线，同时，主机将发出 STOP 信号以释放 I²C 总线。所传送的数据存储在 SIMD 寄存器中。如果设置成发送方，从机必须先将欲传输的数据写到 SIMD 寄存器中；如果设置成接收方，从机必须从 SIMD 寄存器读取数据。

当接收器想要继续接收下一个数据时，必须在第 9 个时钟发出应答信号 (TXAK)。被设为发送方的从机将检测寄存器 SIMC1 中的 RXAK 位以判断是否传输下一个字节的数据，如果单片机不传输下一个字节，那么它将释放 SDA 线并等待接收主机的停止信号。

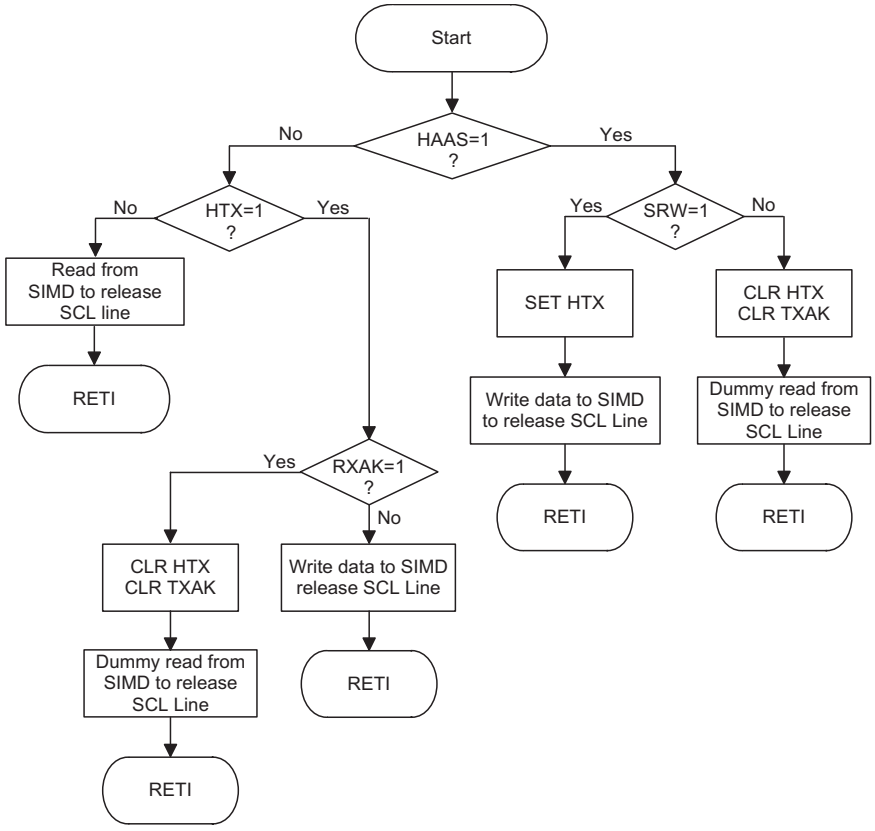



S=Start (1 bit)
SA=Slave Address (7 bits)
SR=SRW bit (1 bit)
M=Slave device send acknowledge bit (1 bit)
D=Data (8 bits)
A=ACK (RXAK bit for transmitter, TXAK bit for receiver 1 bit)
P=Stop (1 bit)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | SA | SR | M | D | A | D | A | ····· | S | SA | SR | M | D | A | D | A | ····· | P |

注：* 当从机地址匹配时，单片机必须选择设置为发送模式还是接收模式。若设置为发送模式，需写数据至 SIMD 寄存器；若设置为接收模式，需立即从 SIMD 寄存器中虚读数据以释放 SCL 线。

**I²C 通信时序图**




**I²C 总线 ISR 流程图**

## 外围时钟输出

外围时钟输出功能使单片机能够为外部硬件提供和单片机时钟同步的时钟信号。

### 外围时钟操作

外围时钟输出引脚 PCK 与输入 / 输出脚共用，可以通过 SIMC0 寄存器的 PCKEN 位来选择引脚功能。外围时钟功能由 SIMC0 寄存器控制。外围时钟输出的时钟源来自 TM0 CCRP 匹配频率 /2 或内部系统时钟分频。SIMC0 寄存器的 PCKEN 位是总的开 / 关控制位，当该位为高时使能外围时钟，为低时除能外围时钟。系统时钟所需要的分频比由同一个寄存器中的 PCKP0 和 PCKP1 位来选择。如果系统进入休眠模式，将除能外围时钟输出功能。

**SIMC0 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SIM2 | SIM1 | SIM0 | PCKEN | PCKP1 | PCKP0 | SIMEN | — |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | — |
| POR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | — |

Bit 7~5 **SIM2~SIM0**：SIM 工作模式控制位
000：SPI 主机模式；SPI 时钟为 $f_{SYS}/4$
001：SPI 主机模式；SPI 时钟为 $f_{SYS}/16$
010：SPI 主机模式；SPI 时钟为 $f_{SYS}/64$
011：SPI 主机模式；SPI 时钟为 $f_{TBC}$
100: SPI 主机模式；SPI 时钟为 TM0 CCRP 匹配频率 /2
101：SPI 从机模式
110：$I^2C$ 从机模式
111：未使用模式
这几位用于设置 SIM 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率及 $I^2C$ 或 SPI 功能。SPI 时钟源可来自于系统时钟也可以选择来自 TM0。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 **PCKEN**：PCK 输出脚控制位
0：除能
1：使能

Bit 3~2 **PCKP1,PCKP0**：选择 PCK 输出脚的频率位
00：$f_{SYS}$
01：$f_{SYS}/4$
10：$f_{SYS}/8$
11：TM0 CCRP 匹配频率 /2

Bit 1 **SIMEN**：SIM 控制位
0：除能
1：使能
此位为 SIM 接口的开 / 关控制位。此位为“0”时，SIM 接口除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 脚处于浮空状态，SIM 工作电流减小到最小值。此位为“1”时，SIM 接口使能。配置选项中首先将 SIM 接口使能才能使此位有效。注意，当 SIMEN 位由低到高转变时，SPI 控制寄存器的内容处于未知状态，其首先应在应用程序中初始化。

Bit 0 未使用，读为“0”

# 中断

中断是单片机一个重要功能。当外部事件或内部功能如定时器模块有效，并且产生中断时，系统会暂时中止当前的程序而转到执行相对应的中断服务程序。此单片机提供多个外部中断和内部中断功能，外部中断由 INT0~INT3 和 $\overline{\text{PINT}}$ 引脚动作产生，而内部中断由各种内部功能，如定时器模块、比较器、时基、LVD、EEPROM 和 SIM 等产生。

## 中断寄存器

中断控制基本上是在一定单片机条件发生时设置请求标志位，应用程序中中断使能位的设置是通过位于专用数据存储器中的一系列寄存器控制的。寄存器的数量由所选单片机的型号决定，但总的分为三类。第一类是 INTC0~INTC3 寄存器，用于设置基本的中断；第二类是 MFI0~MFI3 寄存器，用于设置多功能中断；最后一种有 INTEG 寄存器，用于设置外部中断边沿触发类型。

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

| 功能 | 使能位 | 请求标志 | 注释 |
|---|---|---|---|
| 总中断 | EMI | — | — |
| 比较器 | CPnE | CPnF | n=0 或 1 |
| INTn 脚 | INTnE | INTnF | n=0~3 |
| 多功能 | MFnE | MFnF | n=0~5 |
| 时基 | TBnE | TBnF | n=0 或 1 |
| SIM | SIME | SIMF | — |
| LVD | LVE | LVF | — |
| EEPROM | DEE | DEF | — |
| $\overline{\text{PINT}}$ 脚 | XPE | XPF | — |
| TM | TnPE | TnPF | n=0~3 |
| | TnAE | TnAF | |
| | TnBE | TnBF | |

中断寄存器位命名模式

中断寄存器内容

- **HT68F20**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTEG | — | — | — | — | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| INTC0 | — | CP0F | INT1F | INT0F | CP0E | INT1E | INT0E | EMI |
| INTC1 | — | MF1F | MF0F | CP1F | — | MF1E | MF0E | CP1E |
| INTC2 | MF3F | TB1F | TB0F | MF2F | MF3E | TB1E | TB0E | MF2E |
| MFI0 | — | — | T0AF | T0PF | — | — | T0AE | T0PE |
| MFI1 | — | — | T1AF | T1PF | — | — | T1AE | T1PE |
| MFI2 | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |

- **HT68F30**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTEG | — | — | — | — | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| INTC0 | — | CP0F | INT1F | INT0F | CP0E | INT1E | INT0E | EMI |
| INTC1 | — | MF1F | MF0F | CP1F | — | MF1E | MF0E | CP1E |
| INTC2 | MF3F | TB1F | TB0F | MF2F | MF3E | TB1E | TB0E | MF2E |
| MFI0 | — | — | T0AF | T0PF | — | — | T0AE | T0PE |
| MFI1 | — | T1BF | T1AF | T1PF | — | T1BE | T1AE | T1PE |
| MFI2 | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |

- **HT68F40**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTEG | — | — | — | — | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| INTC0 | — | CP0F | INT1F | INT0F | CP0E | INT1E | INT0E | EMI |
| INTC1 | — | MF1F | MF0F | CP1F | — | MF1E | MF0E | CP1E |
| INTC2 | MF3F | TB1F | TB0F | MF2F | MF3E | TB1E | TB0E | MF2E |
| MFI0 | T2AF | T2PF | T0AF | T0PF | T2AE | T2PE | T0AE | T0PE |
| MFI1 | — | T1BF | T1AF | T1PF | — | T1BE | T1AE | T1PE |
| MFI2 | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |

- **HT68F50**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTEG | — | — | — | — | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| INTC0 | — | CP0F | INT1F | INT0F | CP0E | INT1E | INT0E | EMI |
| INTC1 | — | MF1F | MF0F | CP1F | — | MF1E | MF0E | CP1E |
| INTC2 | MF3F | TB1F | TB0F | MF2F | MF3E | TB1E | TB0E | MF2E |
| MFI0 | T2AF | T2PF | T0AF | T0PF | T2AE | T2PE | T0AE | T0PE |
| MFI1 | — | T1BF | T1AF | T1PF | — | T1BE | T1AE | T1PE |
| MFI2 | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |
| MFI3 | — | — | T3AF | T3PF | — | — | T3AE | T3PE |

- **HT68F60**

| Name | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTEG | INT3S1 | INT3S0 | INT2S1 | INT2S0 | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| INTC0 | — | INT2F | INT1F | INT0F | INT2E | INT1E | INT0E | EMI |
| INTC1 | MF0F | CP1F | CP0F | INT3F | MF0E | CP1E | CP0E | INT3E |
| INTC2 | — | MF3F | MF2F | MF1F | — | MF3E | MF2E | MF1E |
| INTC3 | MF5F | TB1F | TB0F | MF4F | MF5E | TB1E | TB0E | MF4E |
| MFI0 | T2AF | T2PF | T0AF | T0PF | T2AE | T2PE | T0AE | T0PE |
| MFI1 | — | T1BF | T1AF | T1PF | — | T1BE | T1AE | T1PE |
| MFI2 | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |
| MFI3 | — | — | T3AF | T3PF | — | — | T3AE | T3PE |

## INTEG 寄存器

- **HT68F20/HT68F30/HT68F40/HT68F50**

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

Bit 7~4 未使用，读为“0”

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

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

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | INT3S1 | INT3S0 | INT2S1 | INT2S0 | INT1S1 | INT1S0 | INT0S1 | INT0S0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7~6 **INT3S1,INT3S0**：INT3 脚中断边沿控制位
00：除能
01：上升沿
10：下降沿
11：双沿

Bit5~4 **INT2S1,INT2S0**：INT2 脚中断边沿控制位
00：除能
01：上升沿
10：下降沿
11：双沿

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

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

### INTC0 寄存器

- **HT68F20/HT68F30/HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | CP0F | INT1F | INT0F | CP0E | INT1E | INT0E | EMI |
| R/W | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 未使用，读为“0”

Bit 6 **CP0F**：比较器 0 中断请求标志位
0：无请求
1：中断请求

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

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

Bit 3 **CP0E**：比较器 0 中断控制位
0：除能
1：使能

Bit 2 **INT1E**：INT1 中断控制位
0：除能
1：使能

Bit 1 **INT0E**：INT0 中断控制位
0：除能
1：使能

Bit 0 **EMI**：总中断控制位
0：除能
1：使能

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | INT2F | INT1F | INT0F | INT2E | INT1E | INT0E | EMI |
| R/W | — | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 未使用，读为“0”

Bit 6 **INT2F**：INT2 中断请求标志位
0：无请求
1：中断请求

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

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

Bit 3 **INT2E**：INT2 中断控制位
0：除能
1：使能

Bit 2 **INT1E**：INT1 中断控制位
0：除能
1：使能

Bit 1 **INT0E**：INT0 中断控制位
0：除能
1：使能

Bit 0 **EMI**：总中断控制位
0：除能
1：使能

## INTC1 寄存器

- **HT68F20/HT68F30/HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | MF1F | MF0F | CP1F | — | MF1E | MF0E | CP1E |
| R/W | — | R/W | R/W | R/W | — | R/W | R/W | R/W |
| POR | — | 0 | 0 | 0 | — | 0 | 0 | 0 |

Bit 7 未使用，读为“0”

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

Bit 5 **MF0F**：多功能中断 0 请求标志位
0：无请求
1：中断请求

Bit 4 **CP1F**：比较器 1 中断请求标志位
0：无请求
1：中断请求

Bit 3 未使用，读为“0”

Bit 2 **MF1E**：多功能中断 1 控制位
0：除能
1：使能

Bit 1 **MF0E**：多功能中断 0 控制位
0：除能
1：使能

Bit 0 **CP1E**：比较器 1 中断控制位
0：除能
1：使能

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MF0F | CP1F | CP0F | INT3F | MF0E | CP1E | CP0E | INT3E |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **MF0F**：多功能中断 0 请求标志位
0：无请求
1：中断请求

Bit 6 **CP1F**：比较器 1 中断请求标志位
0：无请求
1：中断请求

Bit 5 **CP0F**：比较器 0 中断请求标志位
0：无请求
1：中断请求

Bit 4 **INT3F**：INT3 中断请求标志位
0：无请求
1：中断请求

Bit 3 **MF0E**：多功能中断 0 控制位
0：除能
1：使能

Bit 2 **CP1E**：比较器 1 中断控制位
0：除能
1：使能

Bit 1 **CP0E**：比较器 0 中断控制位
0：除能
1：使能

Bit 0 **INT3E**：INT3 中断控制位
0：除能
1：使能

## INTC2 寄存器

- **HT68F20/HT68F30/HT68F40/HT68F50**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MF3F | TB1F | TB0F | MF2F | MF3E | TB1E | TB0E | MF2E |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **MF3F**：多功能中断 3 请求标志位
0：无请求
1：中断请求

Bit 6 **TB1F**：时基 1 中断请求标志位
0：无请求
1：中断请求

Bit 5 **TB0F**：时基 0 中断请求标志位
0：无请求
1：中断请求

Bit 4 **MF2F**：多功能中断 2 请求标志位
0：无请求
1：中断请求

Bit 3 **MF3E**：多功能中断 3 控制位
0：除能
1：使能

Bit 2 **TB1E**：时基 1 中断控制位
0：除能
1：使能

Bit 1 **TB0E**：时基 0 中断控制位
0：除能
1：使能

Bit 0 **MF2E**：多功能中断 2 控制位
0：除能
1：使能

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | MF3F | MF2F | MF1F | — | MF3E | MF2E | MF1E |
| R/W | — | R/W | R/W | R/W | — | R/W | R/W | R/W |
| POR | — | 0 | 0 | 0 | — | 0 | 0 | 0 |

Bit 7 未定义，读为“0”

Bit 6 **MF3F**：多功能中断 3 请求标志位
0：无请求
1：中断请求

Bit 5 **MF2F**：多功能中断 2 请求标志位
0：无请求
1：中断请求

Bit 4 **MF1F**：多功能中断 1 请求标志位
0：无请求
1：中断请求

Bit 3 未使用，读为“0”

Bit 2 **MF3E**：多功能中断 3 控制位
0：除能
1：使能

Bit 1 **MF2E**：多功能中断 2 控制位
0：除能
1：使能

Bit 0 **MF1E**：多功能中断 1 控制位
0：除能
1：使能

### INTC3 寄存器

- **HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MF5F | TB1F | TB0F | MF4F | MF5E | TB1E | TB0E | MF4E |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **MF5F**：多功能中断 5 请求标志位
0：无请求
1：中断请求

Bit 6 **TB1F**：时基 1 中断请求标志位
0：无请求
1：中断请求

Bit 5 **TB0F**：时基 0 中断请求标志位
0：无请求
1：中断请求

Bit 4 **MF4F**：多功能中断 4 请求标志位
0：无请求
1：中断请求

Bit 3 **MF5E**：多功能中断 5 控制位
0：除能
1：使能

Bit 2 **TB1E**：时基 1 中断控制位
0：除能
1：使能

Bit 1 **TB0E**：时基 0 中断控制位
0：除能
1：使能

Bit 0 **MF4E**：多功能中断 4 控制位
0：除能
1：使能

### MFI0 寄存器

- **HT68F20/HT68F30**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | T0AF | T0PF | — | — | T0AE | T0PE |
| R/W | — | — | R/W | R/W | — | — | R/W | R/W |
| POR | — | — | 0 | 0 | — | — | 0 | 0 |

Bit 7~6 未使用，读为“0”

Bit 5 **T0AF**：TM0 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 4 **T0PF**：TM0 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 3~2 未使用，读为“0”

Bit 1 **T0AE**：TM0 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **T0PE**：TM0 比较器 P 匹配中断控制位
0：除能
1：使能

- **HT68F40/HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T2AF | T2PF | T0AF | T0PF | T2AE | T2PE | T0AE | T0PE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **T2AF**：TM2 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 6 **T2PF**：TM2 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 5 **T0AF**：TM0 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 4 **T0PF**：TM0 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 3 **T2AE**：TM2 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 2 **T2PE**：TM2 比较器 P 匹配中断控制位
0：除能
1：使能

Bit 1 **T0AE**：TM0 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **T0PE**：TM0 比较器 P 匹配中断控制位
0：除能
1：使能

### MFI1 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | T1AF | T1PF | — | — | T1AE | T1PE |
| R/W | — | — | R/W | R/W | — | — | R/W | R/W |
| POR | — | — | 0 | 0 | — | — | 0 | 0 |

Bit 7~6 未使用，读为“0”

Bit 5 **T1AF**：TM1 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 4 **T1PF**：TM1 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 3~2 未使用，读为“0”

Bit 1 **T1AE**：TM1 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **T1PE**：TM1 比较器 P 匹配中断控制位
0：除能
1：使能

- **HT68F30/HT68F40/HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | T1BF | T1AF | T1PF | — | T1BE | T1AE | T1PE |
| R/W | — | R/W | R/W | R/W | — | R/W | R/W | R/W |
| POR | — | 0 | 0 | 0 | — | 0 | 0 | 0 |

Bit 7 未使用，读为“0”

Bit 6 **T1BF**：TM1 比较器 B 匹配中断请求标志位
0：无请求
1：中断请求

Bit 5 **T1AF**：TM1 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 4 **T1PF**：TM1 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 3 未使用，读为“0”

Bit 2 **T1BE**：TM1 比较器 B 匹配中断控制位
0：除能
1：使能

Bit 1 **T1AE**：TM1 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **T1PE**：TM1 比较器 P 匹配中断控制位
0：除能
1：使能

### MFI2 寄存器

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | DEF | LVF | XPF | SIMF | DEE | LVE | XPE | SIME |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 **DEF**：数据 EEPROM 中断请求标志位
0：无请求
1：中断请求

Bit 6 **LVF**：LVD 中断请求标志位
0：无请求
1：中断请求

Bit 5 **XPF**：外围中断请求标志位
0：无请求
1：中断请求

Bit 4 **SIMF**：SIM 中断请求标志位
0：无请求
1：中断请求

Bit 3 **DEE**：数据 EEPROM 中断控制位
0：除能
1：使能

Bit 2 **LVE**：LVD 中断控制位
0：除能
1：使能

Bit 1 **XPE**：外围中断控制位
0：除能
1：使能

Bit 0 **SIME**：SIM 中断控制位
0：除能
1：使能

### MFI3 寄存器

- **HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | T3AF | T3PF | — | — | T3AE | T3PE |
| R/W | — | — | R/W | R/W | — | — | R/W | R/W |
| POR | — | — | 0 | 0 | — | — | 0 | 0 |

Bit 7~6 未使用，读为“0”

Bit 5 **T3AF**：TM3 比较器 A 匹配中断请求标志位
0：无请求
1：中断请求

Bit 4 **T3PF**：TM3 比较器 P 匹配中断请求标志位
0：无请求
1：中断请求

Bit 3~2 未使用，读为“0”

Bit 1 **T3AE**：TM3 比较器 A 匹配中断控制位
0：除能
1：使能

Bit 0 **T3PE**：TM3 比较器 P 匹配中断控制位
0：除能
1：使能

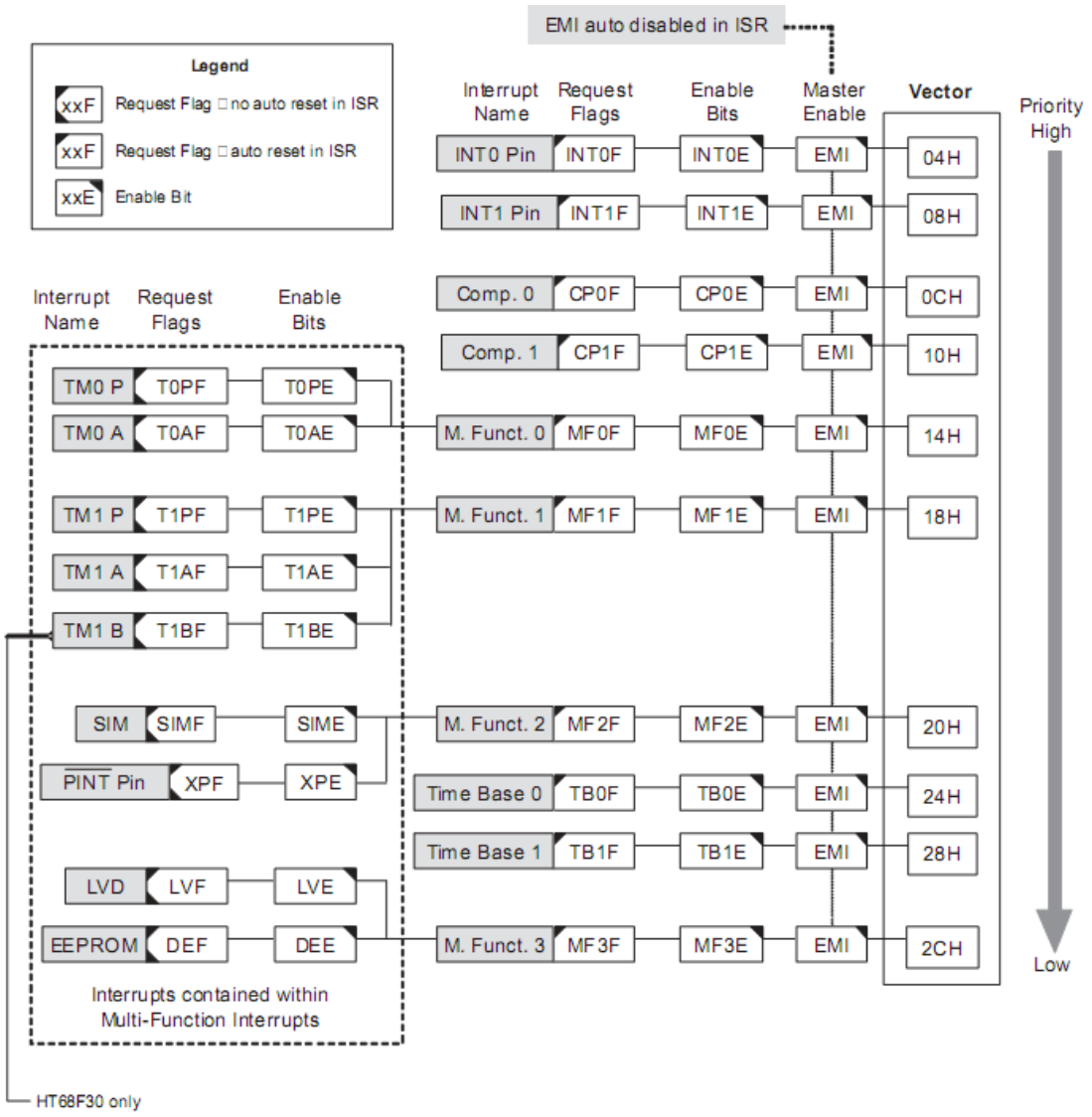
## 中断操作

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

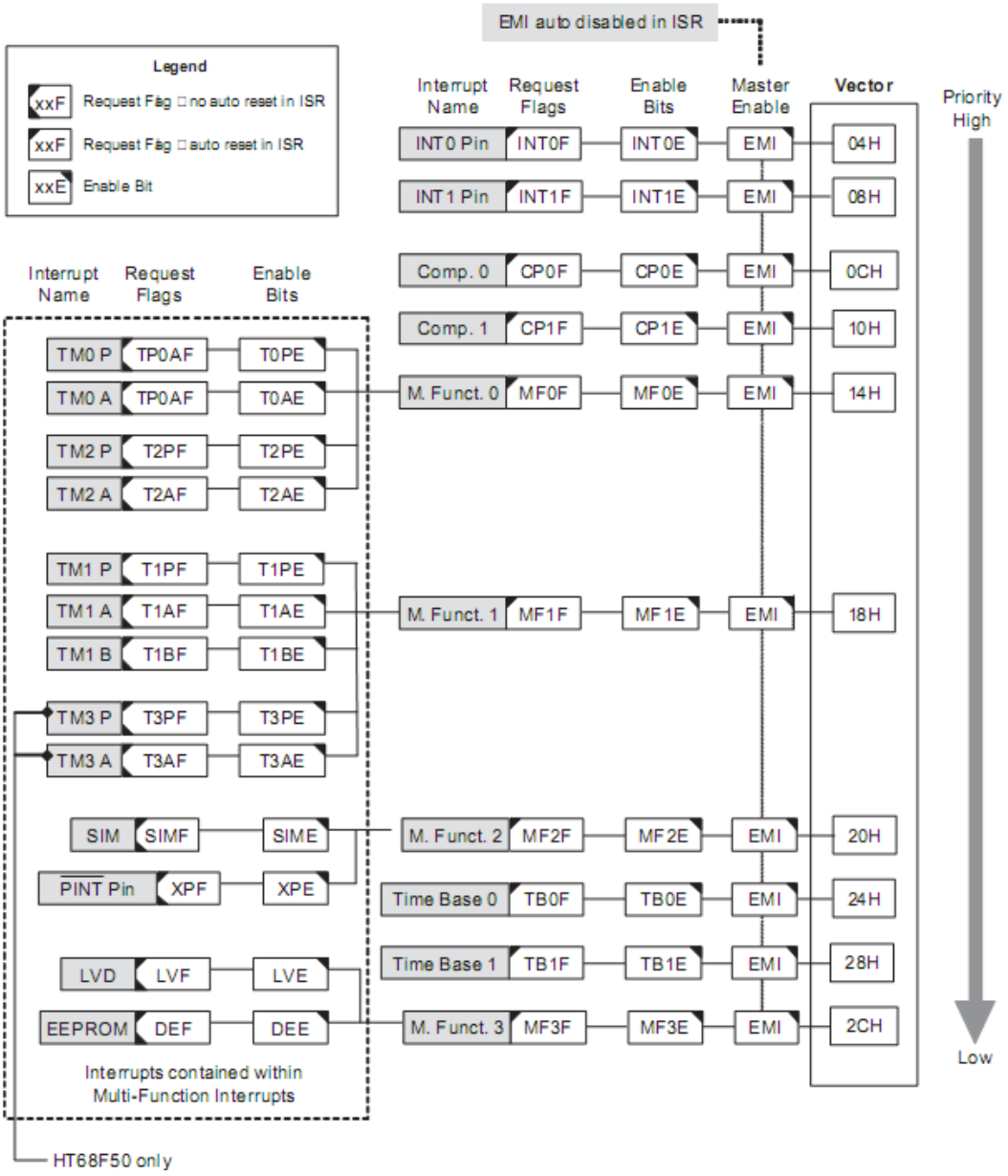
当中断发生时，下条指令的地址将被压入堆栈。相应的中断向量地址加载至 PC 中。系统将从此向量取下条指令。中断向量处通常为跳转指令，以跳转到相应的中断服务程序。中断服务程序必须以“RETI”指令返回至主程序，以继续执行原来的程序。

各个中断使能位以及相应的请求标志位，以优先级的次序显示在下图。一些中断源有自己的向量，但是有些中断却共用多功能中断向量。一旦中断子程序被响应，系统将自动清除 EMI 位，所有其它的中断将被屏蔽，这个方式可以防止任何进一步的中断嵌套。其它中断请求可能发生在此期间，虽然中断不会立即响应，但是中断请求标志位会被记录。

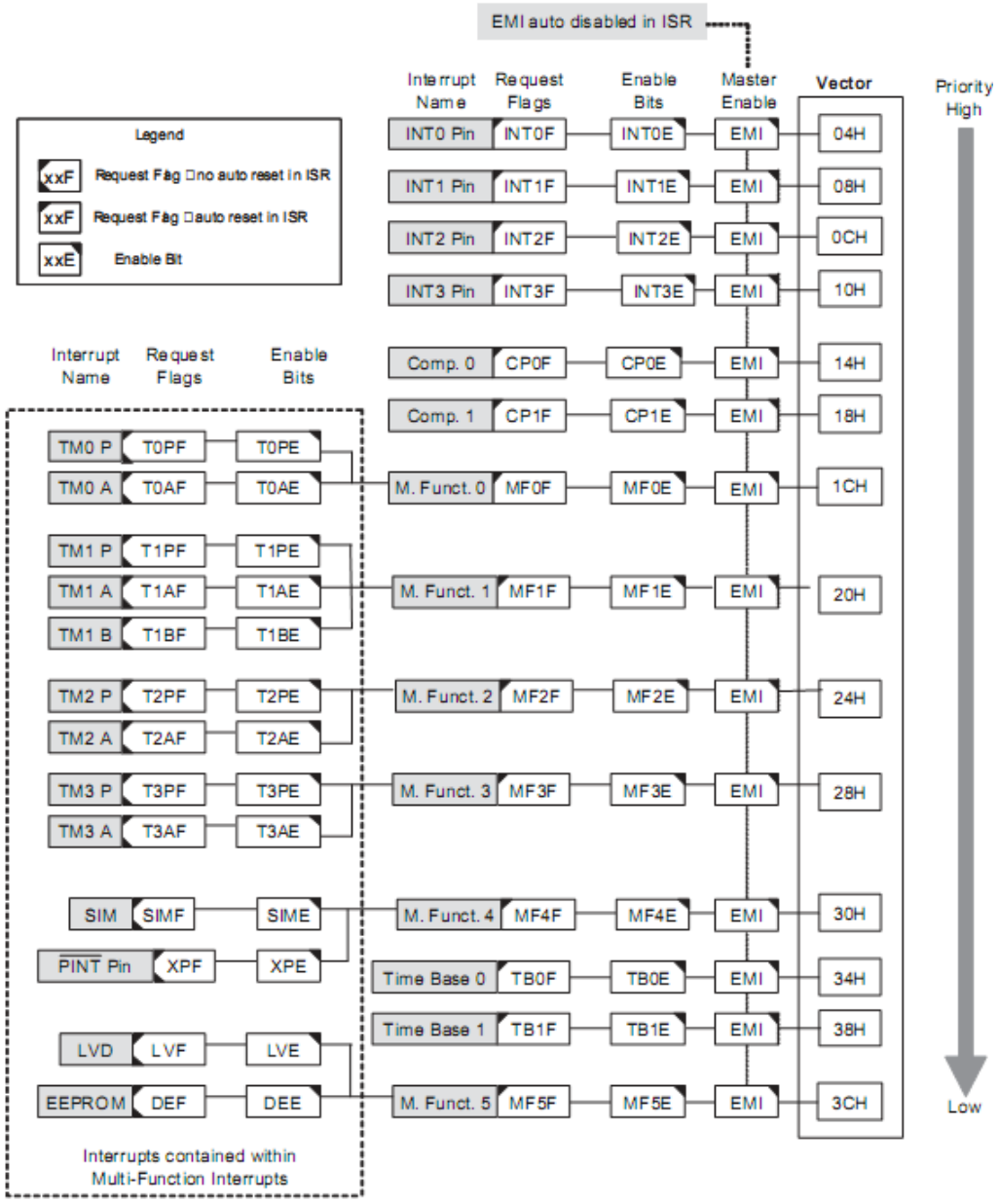
如果某个中断服务子程序正在执行时，有另一个中断要求立即响应，那么 EMI 位应在程序进入中断子程序后置位，以允许此中断嵌套。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到 SP 减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，执行优先级如下流程图所示。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应将相应的标志置起。




中断结构 – **HT68F20/HT68F30**




中断结构 – **HT68F40/HT68F50**




中断结构 – **HT68F60**

### 外部中断

通过 INT0~INT3 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，INT0~INT3 引脚的状态发生变化，外部中断请求标志 INT0F~INT3F 被置位时外部中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和相应中断使能位 INT0E~INT3E 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，此引脚将被作为外部中断脚使用。此时该引脚必须通过设置控制寄存器，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态改变，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INT0F~INT3F 会自动复位且 EMI 位会被清零以除能其它中断。注意，即使此引脚被用作外部中断输入，其配置选项中的上拉电阻仍保持有效。寄存器 INTEG 被用来选择有效的边沿类型，来触发外部中断。可以选择上升沿还是下降沿或双沿触发都产生外部中断。注意 INTEG 也可以用来除能外部中断功能。

### 比较器中断

比较器中断由两个内部比较器控制。当比较器输出状态改变，比较器中断请求标志 CP0F 或 CP1F 被置位，比较器中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和比较器中断使能位 CP0E 和 CP1E 需先被置位。当中断使能，堆栈未满并且比较器输入产生一个比较器输出变化时，将调用比较器中断向量子程序。当响应中断服务子程序时，外部中断请求标志位会自动复位且 EMI 位会被清零以除能其它中断。

### 多功能中断

此系列单片机中有多达六种多功能中断，与其它中断不同，它没有独立源，但由其它现有的中断源构成，即 TM 中断，SIM 中断，外部设备中断，LVD 中断和 EEPROM 中断。

当多功能中断中任何一种中断请求标志 MF0F~MF5F 被置位，多功能中断请求产生。当中断使能，堆栈未满，包括在多功能中断中的任意一个中断发生时，将调用多功能中断向量中的一个子程序。当响应中断服务子程序时，相关的多功能请求标志位会自动复位且 EMI 位会自动清零以除能其它中断。

但必须注意的是，在中断响应时，虽然多功能中断标志会自动复位，但多功能中断源的请求标志位，即 TM 中断，SIM 中断，外部设备中断，LVD 中断和 EEPROM 中断的请求标志位不会自动复位，必须由应用程序清零。

### 时基中断

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

时基中断的目的是提供一个固定周期的中断信号，时钟源来自内部时钟源 $f_{TB}$。$f_{TB}$ 输入时钟首先经过分频器，分频率由程序设置 TBC 寄存器相关位获取合适的分频值以提供更长的时基中断周期。控制时基中断频率 $f_{TB}$ 的时钟源有几种，如在系统工作模式章节所示。

**TBC 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TBON | TBCK | TB11 | TB10 | LXTLP | TB02 | TB01 | TB00 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

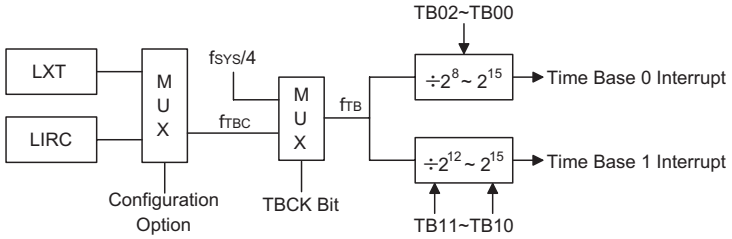
Bit 7 **TBON**：TB0 和 TB1 控制位
0：除能
1：使能

Bit 6 **TBCK**：选择 $f_{TB}$ 时钟位
0：$f_{TBC}$
1：$f_{SYS}/4$

Bit 5~4 **TB11~TB10**：选择时基 1 溢出周期位
00：$4096/f_{TB}$
01：$8192/f_{TB}$
10：$16384/f_{TB}$
11：$32768/f_{TB}$

Bit 3 **LXTLP**：LXT 低功耗控制位
0：除能
1：使能

Bit 2~0 **TB02~TB00**：选择时基 0 溢出周期位
000：$256/f_{TB}$
001：$512/f_{TB}$
010：$1024/f_{TB}$
011：$2048/f_{TB}$
100：$4096/f_{TB}$
101：$8192/f_{TB}$
110：$16384/f_{TB}$
111：$32768/f_{TB}$




时基中断

## 串行接口模块中断

串行接口模块中断，即 SIM 中断，属于多功能中断。当一个字节数据已由 SIM 接口接收或发送完，中断请求标志 SIMF 被置位，SIM 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、串行接口中断使能位 SIME 和多功能中断使能位需先被置位。当中断使能，堆栈未满且一个字节数据已被传送或接收完毕时，可跳转至相关多功能中断向量子程序中执行。当串行接口中断响应，EMI 将被自动清零以除能其它中断，多功能中断请求标志也可自动清除，但 SIMF 标志需在应用程序中手动清除。

### 外部设备中断

外部设备中断和外部中断工作方式类似，属于多功能中断。当 $\overline{\text{PINT}}$ 引脚出现一个下降沿跳变，此时外部中断请求标志位 XPF 被置位，外部设备发生中断。要使程序跳转至相应中断向量地址，总中断控制位 EMI、外部设备中断使能位 XPE 和相应多功能中断使能位必须先被置位。当中断使能，堆栈未满且外部设备中断脚出现一个下降沿跳变，单片机将调用位于多功能中断向量相应子程序。当响应外部设备中断服务子程序时，EMI 位会被清零以除能其它中断，多功能中断请求标志位也将自动清除。

XPF 标志位不会自动复位，由应用程序清零。外部设备中断引脚与其它引脚共用，需正确地设置以使能外部设备中断脚。

### EEPROM 中断

EEPROM 中断也属于多功能中断。当写或读周期结束，EEPROM 中断请求标志 DEF 被置位，EEPROM 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、EEPROM 中断使能位 DEE 和相应多功能中断使能位需先被置位。当中断使能，堆栈未满且 EEPROM 写或读周期结束时，可跳转至相关多功能中断向量子程序中执行。当 EEPROM 中断响应，EMI 将被自动清零以除能其它中断，多功能中断请求标志也可自动清除，但 DEF 标志需在应用程序中手动清除。

### LVD 中断

LVD 中断也属于多功能中断。当低电压检测功能检测到一个低电压时，LVD 中断请求标志 LVF 被置位，LVD 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、低电压中断使能位 LVE 和相应多功能中断使能位需先被置位。当中断使能，堆栈未满且低电压条件发生时，可跳转至相关多功能中断向量子程序中执行。当低电压中断响应，EMI 将被自动清零以除能其它中断，多功能中断请求标志也可自动清除，但 LVF 标志需在应用程序中手动清除。

### TM 中断

简易型和标准型 TM 各有两个中断，增强型 TM 有三个中断。所有的 TM 中断也属于多功能中断。简易型和标准型 TM 各有两个中断请求标志位 TnPF、TnAF 及两个使能位 TnPE、TnAE。增强型 TM 有三个中断请求标志 TnPF、TnAF、TnBF 及三个使能位 TnPE、TnAE、TnBE。当 TM 比较器 P、A、B 匹配情况发生时，任意 TM 中断请求标志被置位，TM 中断请求产生。

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

### 中断唤醒功能

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

### 编程注意事项

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

多功能中断中所含中断相应程序执行时，多功能中断请求标志 MF0F~MF5F 可以自动清零，但各自的请求标志需在应用程序中手动清除。

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

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

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

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

## 暂停模式和唤醒

### 进入空闲或休眠模式

只有在应用程序中执行“HALT”指令，单片机才可以进入休眠或空闲模式。执行这条指令之后，会有以下情形发生：

- 系统振荡器停止运作，且应用程序停止在“HALT”指令处。
- 数据存储器内容和寄存器内容保持当前值。
- 如果 WDT 使能且 WDT 时钟源来自 $f_{SUB}$，则 WDT 将被清零并重新计数；如果 WDT 时钟源来自系统时钟，则 WDT 停止计数。
- 输入 / 输出端口保持当前状态。
- 置位状态寄存器中的暂停标志位－ PDF，并清零看门狗溢出标志位－ TO。

## 静态电流的注意事项

要使系统静态电流降到最小，为微安级，除了需要单片机进入休眠或空闲模式，还要考虑到电路的设计。应该特别注意的是单片机的输入 / 输出口的状态。所有高阻抗输入脚都必须连接到固定的高或低电平，因为引脚浮空会造成内部振荡并导致耗电增加。这也应用于有不同封装的单片机，因为它们可能含有未引出的引脚，这些引脚也必须设为输出或带有上拉电阻的输入。另外还需注意单片机设为输出的 I/O 引脚上的负载。应将它们设置在有最小拉电流的状态或将它们和其它的 CMOS 输入一样接到没有拉电流的外部电路上。还应注意的是，如果使能配置选项中的 LIRC 振荡器，会导致耗电增加。

## 唤醒

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- 外部复位
- PA 口下降沿
- 系统中断
- WDT 溢出

若由外部 RES 引脚唤醒，系统会经过完全复位的过程；若由 WDT 溢出唤醒，则会发生看门狗定时器复位。这两种唤醒方式都会使系统复位，可以通过状态寄存器中 TO 和 PDF 位来判断它的唤醒源。系统上电或执行清除看门狗的指令，会清零 PDF；执行“HALT”指令，PDF 将被置位。看门狗计数器溢出将会置位 TO 标志并唤醒系统，这种复位会重置程序计数器和堆栈指针，其它标志保持原有状态。

PA 口中的每个引脚都可以通过 PAWU 寄存器使能下降沿唤醒功能。PA 端口唤醒后，程序将在“HALT”指令后继续执行。

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

## 低电压检测 – LVD

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

## LVD 寄存器

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

**LVDC 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | — | — | LVDO | LVDEN | — | VLVD2 | VLVD1 | VLVD0 |
| R/W | — | — | R | R/W | — | R/W | R/W | R/W |
| POR | — | — | 0 | 0 | — | 0 | 0 | 0 |

Bit 7~6　未定义，读为“0”

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

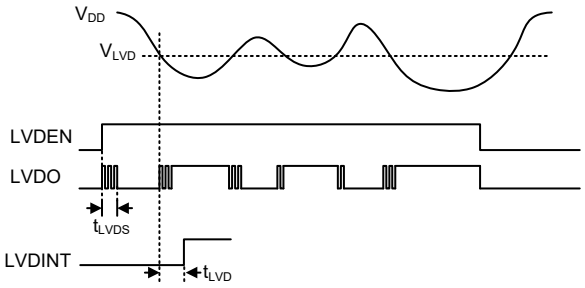
Bit 4　**LVDEN**：低电压检测控制位
0：除能
1：使能

Bit 3　未定义，读为“0”

Bit 2~0　**VLVD2~VLVD0**：选择 LVD 电压位
000：2.0V
001：2.2V
010：2.4V
011：2.7V
100：3.0V
101：3.3V
110：3.6V
111：4.4V

## LVD 操作

通过比较电源电压 $V_{DD}$ 与存储在 LVDC 寄存器中的预置电压值的结果，低电压检测功能工作。其设置的范围为 2.0V~4.4V。当电源电压 $V_{DD}$ 低于预置电压值时，LVDO 位被置为高，表明低电压产生。低电压检测功能由一个自动使能的参考电压提供。若 LVDEN 位为高，当单片机掉电时低电压检测器保持有效状态。低电压检测器使能后，读取 LVDO 位前，电路稳定需要一定的延时 $t_{LVDS}$。注意，$V_{DD}$ 电压可能上升或下降比较缓慢，在 $V_{LVD}$ 电压值附近时，LVDO 位可能有多种变化。




**LVD 操作**

低电压检测器也有自己的中断功能，也是属于多功能中断的一种，它是除了轮询 LVDO 位之外的另一种检测低电压的方法。中断条件产生置位 LVDO 并延时 $t_{LVD}$ 后，中断产生。若 LVDEN 位为高，当单片机掉电时低电压检测器保持有效状态。此种情况下，若 $V_{DD}$ 降至小于 LVD 预置电压值时，中断请求标志位 LVF 将被置位，中断产生，单片机将被从休眠或空闲模式中唤醒。若不要求低电压检测的唤醒功能使能，在单片机进入休眠或空闲模式前应将 LVF 标志置为高。

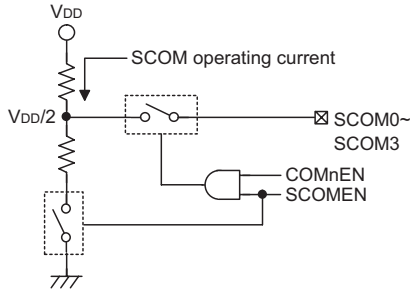
# 带 SCOM 功能的 LCD

单片机具有驱动外部 LCD 面板的能力。LCD 驱动的 COM 脚 SCOM0~SCOM3 与 PC0~PC3 或 PC0~PC1，PC6~PC7 口共用。LCD 控制信号 (COM & SEG) 由软件编程实现。

## LCD 操作

单片机通过设置 PC0~PC3 或 PC0~PC1，PC6~PC7 作为 COM 引脚，其它输出口作为 SEG 引脚，以驱动外部的液晶面板。LCD 驱动功能是由 SCOMC 寄存器来控制，另外，该寄存器可设置 LCD 的开启和关闭以及输出偏压值等功能，使得 COM 口输出 $V_{DD}/2$ 的电压，从而实现 1/2bias LCD 的显示。

SCOMC 寄存器中的 SCOMEN 位是 LCD 驱动的主控制位，它与 COMnEN 位搭配共同选择哪些 PC 端口用于 LCD 驱动。需注意的是，端口控制寄存器不需要设置为输出以使能 LCD 驱动操作。




**LCD COM 偏压**

| SCOMEN | COMnEN | 引脚功能 | O/P Level |
|---|---|---|---|
| 0 | × | I/O | 0 或 1 |
| 1 | 0 | I/O | 0 或 1 |
| 1 | 1 | SCOMn | $V_{DD}/2$ |

**输出控制**

## LCD 偏压控制

LCD 驱动器可以提供多种驱动电流选择以适应不同 LCD 面板的需求。通过设置 SCOMC 寄存器中 ISEL0 位和 ISEL1 位可以配置不同的偏压电阻。

### SCOMC 寄存器

- **HT68F20**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | D7 | ISEL1 | ISEL0 | SCOMEN | COM3EN | COM2EN | COM1EN | COM0EN |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7　　保留位
0：正确电平值 -- 该位需清除为 0
1：不可预测的操作 -- 该位不能设置为高

Bit 6~5 **ISEL1,ISEL0**：选择 SCOM 典型偏压电流 ($V_{DD}$=5V) 位
00: 25μA
01: 50μA
10: 100μA
11: 200μA

Bit 4 **SCOMEN**：SCOM 模块控制位
0：除能
1：使能

Bit 3 **COM3EN**：选择 PC3 或 SCOM3 位
0：GPIO
1：SCOM3

Bit 2 **COM2EN**：选择 PC2 或 SCOM2 位
0：GPIO
1：SCOM2

Bit 1 **COM1EN**：选择 PC1 或 SCOM1 位
0：GPIO
1：SCOM1

Bit 0 **COM0EN**：选择 PC0 或 SCOM0 位
0：GPIO
1：SCOM0

- **HT68F30/HT68F40/HT68F50/HT68F60**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | D7 | ISEL1 | ISEL0 | SCOMEN | COM3EN | COM2EN | COM1EN | COM0EN |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 保留位
0：正确电平值 -- 该位需清除为 0
1：不可预测的操作 -- 该位不能设置为高

Bit 6~5 **ISEL1,ISEL0**：选择 SCOM 典型偏压电流 ($V_{DD}$=5V) 位
00: 25μA
01: 50μA
10: 100μA
11: 200μA

Bit 4 **SCOMEN**：SCOM 模块控制位
0：除能
1：使能

Bit 3 **COM3EN**：选择 PC7 或 SCOM3 位
0：GPIO
1：SCOM3

Bit 2 **COM2EN**：选择 PC6 或 SCOM2 位
0：GPIO
1：SCOM2

Bit 1 **COM1EN**：选择 PC1 或 SCOM1 位
0：GPIO
1：SCOM1

Bit 0 **COM0EN**：选择 PC0 或 SCOM0 位
0：GPIO
1：SCOM0

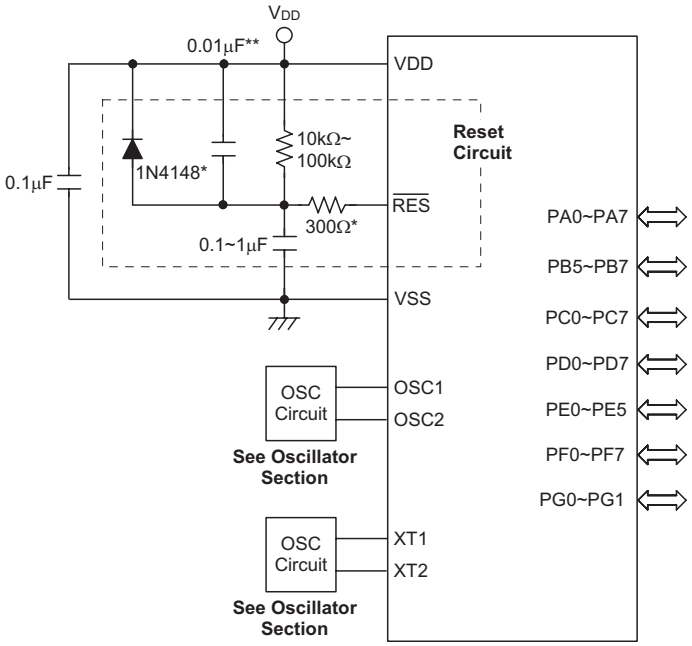
# 配置选项

配置选项在烧写程序时写入芯片。通过 HT-IDE 的软件开发环境，使用者在开发过程中可以选择配置选项。当配置选项烧入单片机后，无法再通过应用程序修改。所有位必须按系统的需要定义，具体内容可参考下表：

| 序号 | 选项 |
|---|---|
| **振荡器选项** | |
| 1 | 高速振荡器类型选择 -- $f_H$：<br>1. HXT<br>2. ERC<br>3. HIRC |
| 2 | 低速振荡器类型选择 -- $f_L$：<br>1. LXT<br>2. LIRC |
| 3 | WDT 时钟选择 -- $f_S$：<br>1. $f_{SUB}$<br>2. $f_{SYS}/4$ |
| 4 | HIRC 频率选择：<br>1. 4MHz<br>2. 8MHz<br>3. 12MHz |
| 注：$f_{SUB}$ 和 $f_{TBC}$ 时钟源通过 $f_L$ 的配置选项选择为 LXT 或 LIRC | |
| **复位脚选项** | |
| 5 | PB0/$\overline{RES}$ 脚选项：<br>1. $\overline{RES}$ 脚<br>2. I/O 脚 |
| **看门狗选项** | |
| 6 | 看门狗定时器功能：<br>1. 使能<br>2. 除能 |
| 7 | 清看门狗指令选择：<br>1. 1 条指令<br>2. 2 个指令 |
| **LVR 选项** | |
| 8 | LVR 功能：<br>1. 使能<br>2. 除能 |
| 9 | LVR 电压选择：<br>1. 2.10V<br>2. 2.55V<br>3. 3.15V<br>4. 4.20V |

| 序号 | 选项 |
|---|---|
| **SIM 选项** | |
| 10 | SIM 功能：<br>1. 使能<br>2. 除能 |
| 11 | SPI -- WCOL 位：<br>1. 使能<br>2. 除能 |
| 12 | SPI -- CSEN 位：<br>1. 使能<br>2. 除能 |
| 13 | $I^2C$ 去抖时间选择：<br>1. 无去抖<br>2. 1 个系统时钟去抖<br>3. 2 个系统时钟去抖 |

## 应用电路




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

# UART 模块串行接口

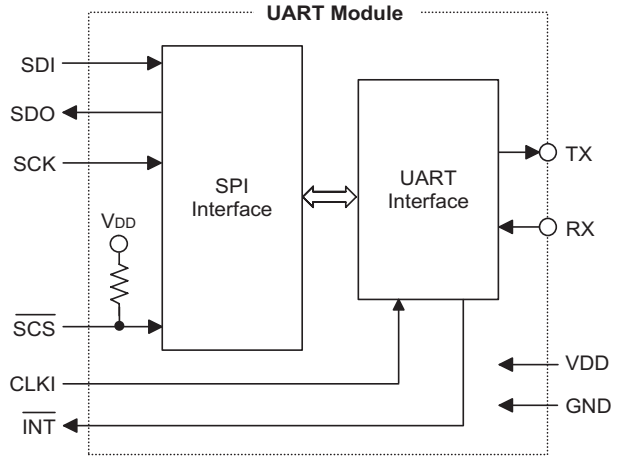
## UART 模块特性

- 由 SPI 接口实现 Holtek MCU 间的通信
- 全双工通用异步接收器 / 发送器 (UART)
  – 8 位或 9 位传输格式
  – 奇校验、偶校验或无校验
  – 1 位或 2 位停止位
  – 8 位预分频的波特率发生器
  – 奇偶、帧、噪声和溢出检测
  – 支持地址匹配中断（最后一位 =1）
  – 独立的发送和接收使能
  – 4-byte FIFO 接收缓冲器
  – 发送和接收中断源：
    ♦ 发送完成
    ♦ 发送器空闲
    ♦ 接收完成
    ♦ 接收器溢出
    ♦ 地址匹配
- UART 传输模块除能时，TX 脚处于高阻抗状态
- UART 接收模块除能时，RX 脚处于高阻抗状态
- $V_{DD}$=5V 时，CMOS 时钟输入脚 CLKI 频率高达 20MHz

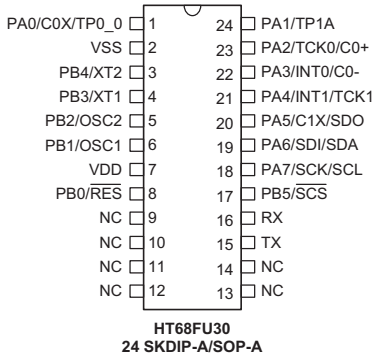
## UART 模块概述

此系列单片机具有一个全双工的异步串行通信接口——UART，可以很方便的与其它具有串行口的芯片通信。它可以支持诸如单片机之间的数据通信网络，低成本 PC 和外部设备间数据连接，便携式和电池供电设备间的通信，工厂自动化及过程控制等。

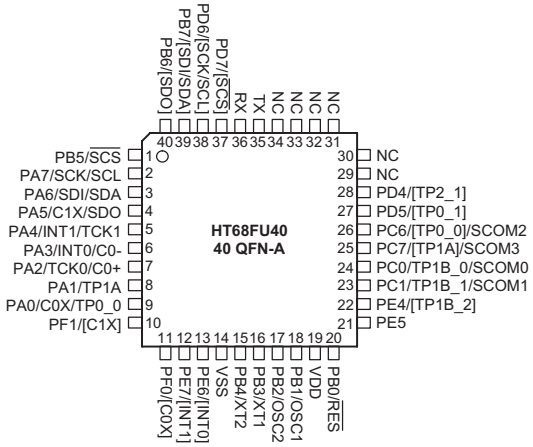
## UART 模块方框图



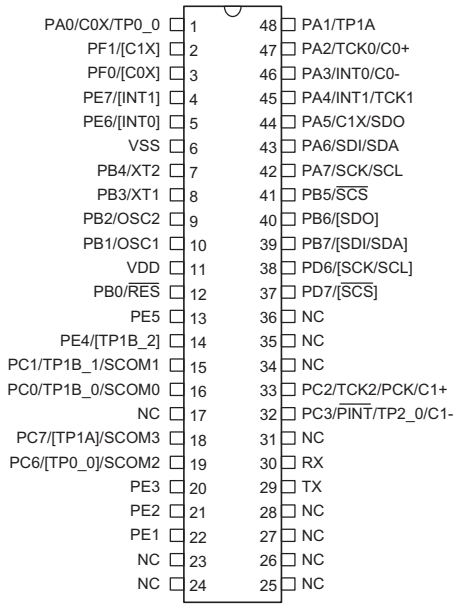
## 引脚图



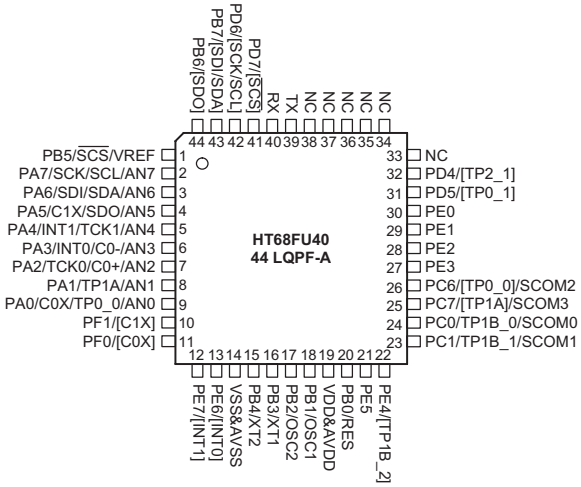

**HT68FU30**
**24 SKDIP-A/SOP-A**



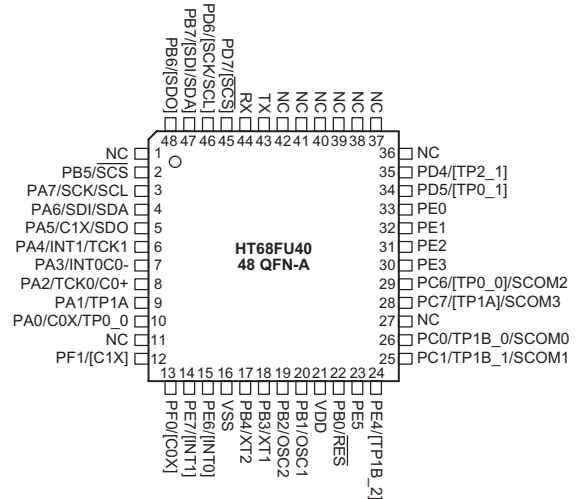

**HT68FU40**
**40 QFN-A**



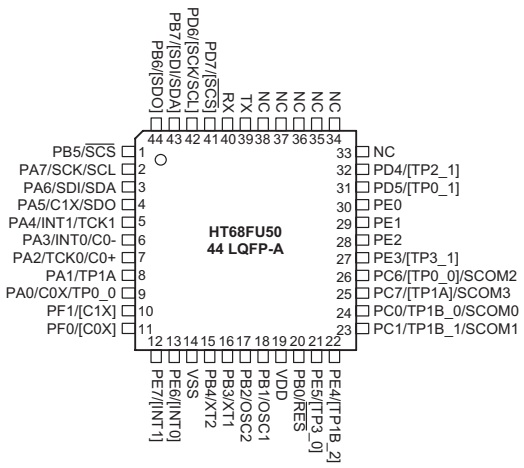

**HT68FU40**
**48 SSOP-A**

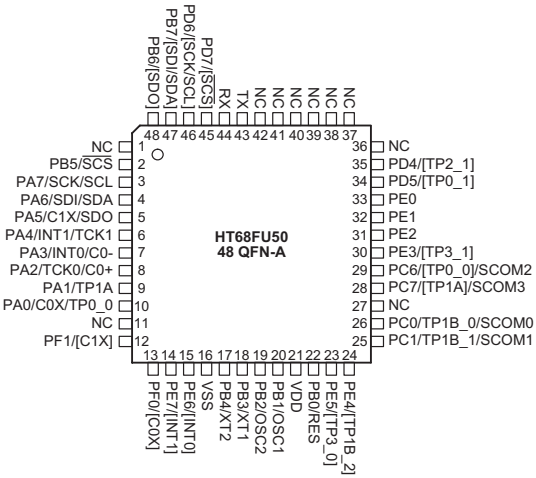


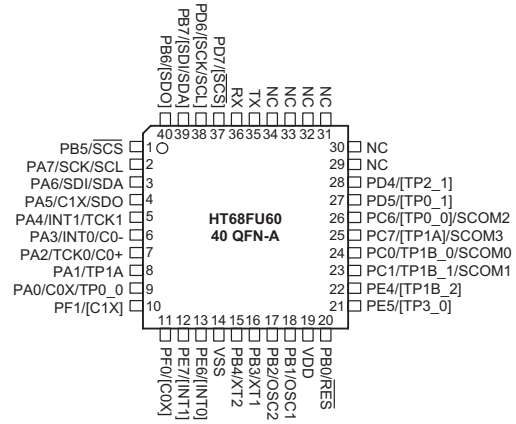

**HT68FU40**
**44 LQPF-A**

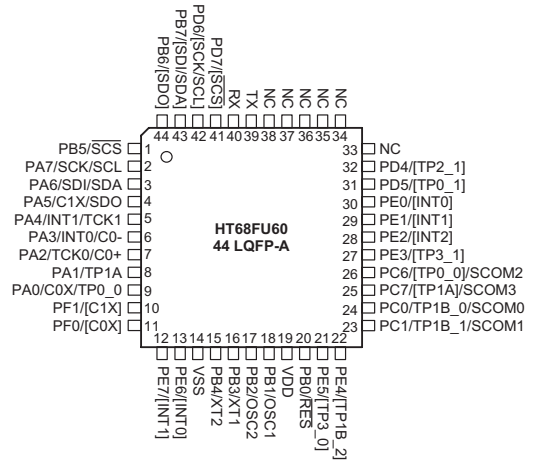


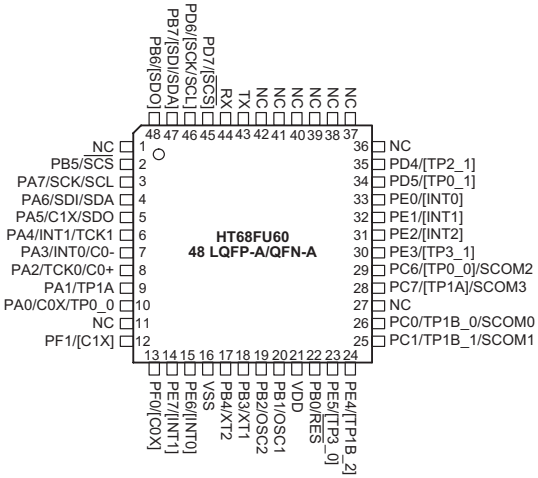

**HT68FU40**
**48 QFN-A**


HT68FU50
44 LQFP-A
PB5/SCS
PA7/SCK/SCL
PA6/SDI/SDA
PA5/C1X/SDO
PA4/INT1/TCK1
PA3/INT0/C0-
PA2/TCK0/C0+
PA1/TP1A
PA0/C0X/TP0_0
PF1/[C1X]
PF0/[C0X]
NC
PD4/[TP2_1]
PD5/[TP0_1]
PE0
PE1
PE2
PE3/[TP3_1]
PC6/[TP0_0]/SCOM2
PC7/[TP1A]/SCOM3
PC0/TP1B_0/SCOM0
PC1/TP1B_1/SCOM1
PE4/[TP1B_2]
PE5/[TP3_0]
PB0/RES
PB1/OSC1
PB2/OSC2
PB3/XT1
PB4/XT2
VSS
PE6/[INT0]
PE7/[INT1]
PB6/[SDO]
PB7/[SDI/SDA]
PD6/[SCK/SCL]
PD7/[SCS]
RX
TX


HT68FU50
48 QFN-A
NC
PB5/SCS
PA7/SCK/SCL
PA6/SDI/SDA
PA5/C1X/SDO
PA4/INT1/TCK1
PA3/INT0/C0-
PA2/TCK0/C0+
PA1/TP1A
PA0/C0X/TP0_0
PF1/[C1X]
PF0/[C0X]
PD4/[TP2_1]
PD5/[TP0_1]
PE0
PE1
PE2
PE3/[TP3_1]
PC6/[TP0_0]/SCOM2
PC7/[TP1A]/SCOM3
PC0/TP1B_0/SCOM0
PC1/TP1B_1/SCOM1
PE4/[TP1B_2]
PE5/[TP3_0]
PB0/RES
PB1/OSC1
PB2/OSC2
PB3/XT1
PB4/XT2
VSS
PE6/[INT0]
PE7/[INT1]
PB6/[SDO]
PB7/[SDI/SDA]
PD6/[SCK/SCL]
PD7/[SCS]
RX
TX


HT68FU60
40 QFN-A
PB5/SCS
PA7/SCK/SCL
PA6/SDI/SDA
PA5/C1X/SDO
PA4/INT1/TCK1
PA3/INT0/C0-
PA2/TCK0/C0+
PA1/TP1A
PA0/C0X/TP0_0
PF1/[C1X]
PF0/[C0X]
NC
PD4/[TP2_1]
PD5/[TP0_1]
PC6/[TP0_0]/SCOM2
PC7/[TP1A]/SCOM3
PC0/TP1B_0/SCOM0
PC1/TP1B_1/SCOM1
PE4/[TP1B_2]
PE5/[TP3_0]
PB0/RES
PB1/OSC1
PB2/OSC2
PB3/XT1
PB4/XT2
VSS
PE6/[INT0]
PE7/[INT1]
PB6/[SDO]
PB7/[SDI/SDA]
PD6/[SCK/SCL]
PD7/[SCS]
RX
TX


HT68FU60
44 LQFP-A
PB5/SCS
PA7/SCK/SCL
PA6/SDI/SDA
PA5/C1X/SDO
PA4/INT1/TCK1
PA3/INT0/C0-
PA2/TCK0/C0+
PA1/TP1A
PA0/C0X/TP0_0
PF1/[C1X]
PF0/[C0X]
NC
PD4/[TP2_1]
PD5/[TP0_1]
PE0/[INT0]
PE1/[INT1]
PE2/[INT2]
PE3/[TP3_1]
PC6/[TP0_0]/SCOM2
PC7/[TP1A]/SCOM3
PC0/TP1B_0/SCOM0
PC1/TP1B_1/SCOM1
PE4/[TP1B_2]
PE5/[TP3_0]
PB0/RES
PB1/OSC1
PB2/OSC2
PB3/XT1
PB4/XT2
VSS
PE6/[INT0]
PE7/[INT1]
PB6/[SDO]
PB7/[SDI/SDA]
PD6/[SCK/SCL]
PD7/[SCS]
RX
TX


HT68FU60
48 LQFP-A/QFN-A
NC
PB5/SCS
PA7/SCK/SCL
PA6/SDI/SDA
PA5/C1X/SDO
PA4/INT1/TCK1
PA3/INT0/C0-
PA2/TCK0/C0+
PA1/TP1A
PA0/C0X/TP0_0
NC
PF1/[C1X]
PF0/[C0X]
PD4/[TP2_1]
PD5/[TP0_1]
PE0/[INT0]
PE1/[INT1]
PE2/[INT2]
PE3/[TP3_1]
PC6/[TP0_0]/SCOM2
PC7/[TP1A]/SCOM3
NC
PC0/TP1B_0/SCOM0
PC1/TP1B_1/SCOM1
PE4/[TP1B_2]
PE5/[TP3_0]
PB0/RES
PB1/OSC1
PB2/OSC2
PB3/XT1
PB4/XT2
VSS
PE6/[INT0]
PE7/[INT1]
PB6/[SDO]
PB7/[SDI/SDA]
PD6/[SCK/SCL]
PD7/[SCS]
RX
TX

## UART 模块引脚说明

| 引脚名称 | I/O | 共用引脚映射说明 |
|---|---|---|
| RX | I | UART RX 串行数据输入脚<br>若 UARTEN=1 且 RXEN=1，RX 为 UART 串行数据输入脚<br>若 UARTEN=0 或 RXEN=0，RX 处于高阻抗状态 |
| TX | O | UART TX 串行数据输出脚<br>若 UARTEN=1 且 TXEN=1，TX 为 UART 串行数据输出脚<br>若 UARTEN=0 或 TXEN=0，TX 处于高阻抗状态 |
| SDI | I | 从机 SPI 串行数据输入信号脚<br>内连 MCU 主机 SPI 模块 SDO 输出信号 |
| SDO | O | 从机 SPI 串行数据输出信号脚<br>内连 MCU 主机 SPI 模块 SDI 输入信号 |
| SCK | I | 从机 SPI 串行时钟输入信号脚<br>内连 MCU 主机 SPI 时钟 SCK 输出信号 |
| $\overline{SCS}$ | I | 从机 SPI 设备选择输入信号脚<br>内连 MCU 主机 SPI 模块 $\overline{SCS}$ 输出信号——连接上拉电阻 |
| CLKI | I | 时钟输入信号脚<br>内连 MCU 主机 PCK 输出信号 |
| $\overline{PINT}$ | O | UART 中断输出信号脚<br>内连 MCU 主机 $\overline{PINT}$ 输入信号<br>在此信号线上，UART 相关中断将产生一个低脉冲信号 |
| NC | — | 未使用 |

注：除 UART TX 和 RX 脚外的所有引脚的描述见 datasheet 其它章节。

## UART 模块直流电气特性

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $I_{DD1}$ | 工作电流 *<br>(SPI 使能 ,UART 除能 ) | 3.0V | $f_{CLKI}$=12MHz,<br>SCK=$f_{CLKI}$/4,<br>输出无负载 | — | — | 1.0 | mA |
| | | 5.0V | $f_{CLKI}$=16MHz,<br>SCK=$f_{CLKI}$/4,<br>输出无负载 | — | — | 2.0 | mA |
| $I_{DD2}$ | 工作电流 *<br>(SPI 使能 ,UART 使能 ) | 2.2V | $f_{CLKI}$=6MHz,<br>SCK=$f_{CLKI}$/4,<br>输出无负载 | — | — | 4.2 | mA |
| | | 5.0V | $f_{CLKI}$=12MHz,<br>SCK=$f_{CLKI}$/4,<br>输出无负载 | — | — | 4.8 | mA |

| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $I_{STB}$ | 静态电流 * (SPI 除能 ,UART 除能 ) | 5.0V | $f_{CLKI}$=16MHz, SCK=$f_{CLKI}$/4, $\overline{SCS}$=$V_{DD}$, UARTEN=0, TXEN=1,RXEN=1, SDI=H,RX=H, 输出无负载 | — | — | 0.6 | μA |
| $V_{IL}$ | RX 端口的低电平输入电压 | — | — | 0 | — | $0.3V_{DD}$ | V |
| $V_{IH}$ | RX 端口的高电平输入电压 | — | — | $0.7V_{DD}$ | — | $V_{DD}$ | V |
| $I_{OL}$ | TX 端口灌电流 | 3.0V | $V_O$=$0.1V_{DD}$ | 2.5 | 5.0 | — | mA |
| | | 5.0V | | 10.0 | 25.0 | — | mA |
| $I_{OH}$ | RX 端口源电流 | 3.0V | $V_O$=$0.9V_{DD}$ | -1.5 | -3.0 | — | mA |
| | | 5.0V | | -5.0 | -8.0 | — | mA |
| $R_{PH}$ | $\overline{SCS}$ 脚上拉电阻 | 3.0V | — | 20 | 60 | 100 | kΩ |
| | | 5.0V | | 10 | 30 | 50 | KΩ |

注：“*”表示的工作电流 $I_{DD1}$ 为从机 SPI 接口使能且 UART 接口除能时额外消耗的电流。同样，工作电流 $I_{DD2}$ 为从机 SPI 接口和 UART 接口都使能时额外消耗的电流。若 UART 模块使能，在计算单片机相关工作电流时需加入 $I_{DD1}$ 或 $I_{DD2}$。整机的静态电流计算时需加入上表中的静态电流值。

## UART 模块交流电气特性

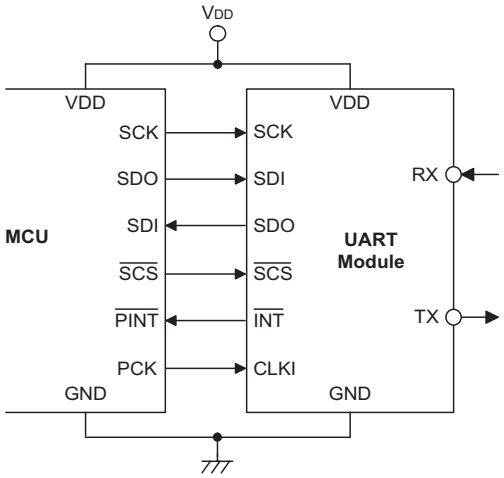
| 符号 | 参数 | 测试条件 | | 最小 | 典型 | 最大 | 单位 |
|---|---|---|---|---|---|---|---|
| | | $V_{DD}$ | 条件 | | | | |
| $t_{CP}$ | SCK 周期 ($t_{CH}$+ $t_{CL}$) | 3.0V | — | 62.5 | — | — | ns |
| | | 5.0V | — | 50.0 | — | — | ns |
| $t_{CH}$ | SCK 高电平宽度 | 3.0V | — | 28 | — | — | ns |
| | | 5.0V | — | 22 | — | — | ns |
| $t_{CL}$ | SCK 低电平宽度 | 3.0V | — | 28 | — | — | ns |
| | | 5.0V | — | 22 | — | — | ns |
| $t_{CSW}$ | $\overline{SCS}$ 高电平宽度 | 3.0V | — | 500 | — | — | ns |
| | | 5.0V | — | 400 | — | — | ns |
| $t_{CSS}$ | $\overline{SCS}$ 到 SCK 的设置时间 | — | — | 100 | — | — | ns |
| $t_{CSH}$ | $\overline{SCS}$ 到 SCK 的保持时间 | — | — | 0 | — | — | ns |
| $t_{SDS}$ | SDI 到 SCK 的设置时间 | — | — | 100 | — | — | ns |
| $t_{SDH}$ | SDI 到 SCK 的保持时间 | — | — | 0 | — | — | ns |
| $t_R$ | SPI 输出上升时间 | — | — | — | 10 | — | ns |
| $t_F$ | SPI 输出下降时间 | — | — | — | 10 | — | ns |
| $t_W$ | SPI 数据输出延迟时间 | — | — | 0 | — | — | ns |

## UART 模块功能描述

内嵌的 UART 模块具有一个全双工的异步串行通信口，可以很方便的与其它具有串行口的芯片通信。UART 具有许多功能特性，发送或接收串行数据时，将数据组成一个 8 位或 9 位的数据块，连同数据特征位一并传输。MCU 主机 SPI 接口与 UART 从机 SPI 接口内部相连，从而实现 MCU 与 UART 模块间的内部连接。MCU 与 UART 模块间包括 UART 命令在内的所有数据的发送与接收都通过该 SPI 接口实现。MCU 通过其主机 SPI 串行接口控制 UART 功能。UART 功能占用一个内部中断向量，当接收到数据或数据发送结束，触发 UART 中断。

## UART 模块内部信号

除了 TX 和 RX 引脚外，MCU 和 UART 模块还可经其它连线进行通信，如下表所示。注意，这些线位于单片机内部，未引出。




**MCU 与 UART 内部连接图**

注：PCK 输出频率必须 ≤12MHz

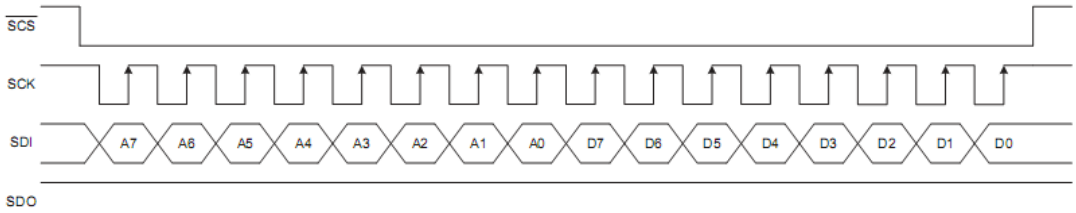
## UART 模块 SPI 接口

MCU 与 UART 模块的通信经由内部 SPI 接口实现。SPI 接口由四根信号线组成：$\overline{SCS}$(SPI 片选 )、SCK(SPI 时钟 )、SDI( 串行数据输入 ) 和 SDO( 串行数据输出 )。$\overline{SCS}$ 为低后，SPI 主机开始数据传输周期。首个 8-bit 数据传输后，$\overline{SCS}$ 继续保持在低电平状态直至整个 16-bit 数据传输完成。若在 16-bit 数据未传输完成时 $\overline{SCS}$ 为高电平状态，则该数据将被 UART 模块 SPI 从机丢弃。

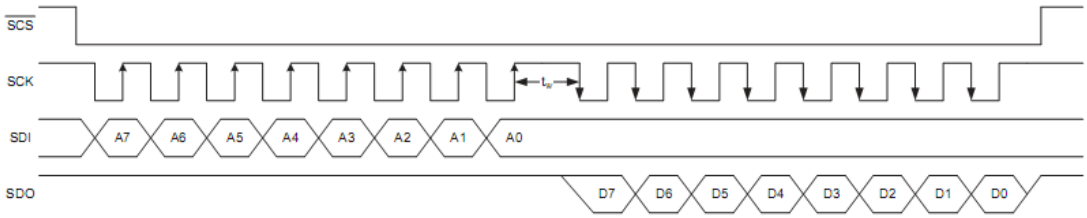
## SPI 时序

SPI 通用接口读写操作格式如下：

- 写类型格式：8-bit 命令输入 +8-bit 数据输入
- 读类型格式：8-bit 命令输入 +8-bit 数据输出




**写类型格式：8-bit 命令输入 +8-bit 数据输入**




**读类型格式：8-bit 命令输入 +8-bit 数据输出**

MCU 主机 SPI 需先将 $\overline{SCS}$ 设为低电平然后将 SCK 设为低电平可以初始化数据传输。SDI 输入数据位需在下个 SCK 上升沿到达前稳定下来，因为在下一个 SCK 上升沿处 MCU 会将 SDI 的状态锁存。在 SCK 下降沿时 SDO 线更新输出数据位。下个 SCK 下降沿到来前，主机需获取到线的状态。

SPI 接口上每次发送与（或）接收的数据为 16 位，该 16 位的数据由一个命令帧和一个数据帧组成。$\overline{SCS}$ 为高电平时，SPI 接口除能且 SDO 为高阻抗状态。

一个完整的记录传输完成需 16 个 SCK 时钟周期，主机需设置 $\overline{SCS}$ 为高电平状态，为下个记录的传输做好准备。

写操作中 $\overline{SCS}$ 由主机再次设为高时，仅在接收 16-bit 串行数据序列后单片机开始执行命令。

读操作中单片机开始执行接收到 8-bit 读命令，而后准备输出数据。若有必要，主机可在此过程中任何时间改变 $\overline{SCS}$ 的状态以中止读操作，且丢弃已读的数据部分。

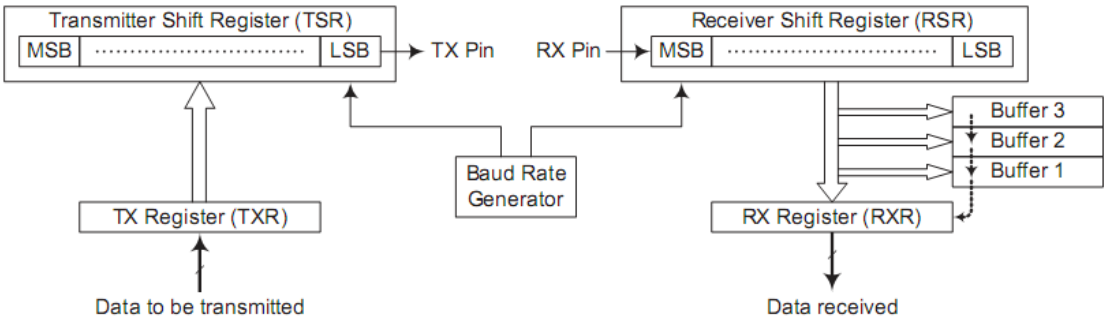
## UART 模块外部引脚接口

内部 UART 有两个外部引脚 TX 和 RX，可与外部串行接口进行通信。若 UARTEN(UCR1.7) 和 TXEN(UCR2.7) 均为“1”时，TX 脚为发送串行数据输出脚，RX 为接收串行数据输入脚。若 UARTEN 或 TXEN 为“0”时，TX 和 RX 均为高阻抗状态。

## UART 数据传输方案

下图显示了 UART 的整体结构。需要发送的数据首先写入 TXR 寄存器，接着此数据被传输到发送移位寄存器 TSR 中，然后在波特率发生器的控制下将 TSR 寄存器中数据一位位地移到 TX 引脚上，低位在前。TXR 寄存器被映射到单片机的数据存储器中，而发送移位寄存器没有实际地址，所以发送移位寄存器不可直接操作。

数据在波特率发生器的控制下，低位在前高位在后，从外部引脚 RX 进入接收移位寄存器 RSR。当数据接收完成，数据从接收移位寄存器移入可被用户程序操作的 RXR 寄存器中。RXR 寄存器被映射到单片机数据存储器中，而接收移位寄存器没有实际地址，所以接收移位寄存器不可直接操作。需要注意的是，上述发送寄存器 TXR 和接收寄存器 RXR，其实是共享一个地址的数据寄存器 TXR/RXR 寄存器。




**UART 数据传输方案**

## UART 命令

UART 模块中有读和写命令，读写的命令和地址信息为一个单独的字节，读写的格式如下表所示。

| 命令类型 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 读 FIFO | 0 | 0 | 0 | 0 | 0 | × | × | × |
| 读寄存器 | 0 | 0 | 0 | 1 | 0 | A2 | A1 | A0 |
| 写 FIFO | 0 | 0 | 0 | 0 | 1 | × | × | × |
| 写寄存器 | 0 | 0 | 0 | 1 | 1 | A2 | A1 | A0 |

注：“×”此处为“无关”

## UART 状态和控制寄存器

UART 功能相关的有六个寄存器——控制 UART 模块整体功能的 USR、UCR1、UCR2 和 UCR3 寄存器，控制波特率的 BRG 寄存器，管理发送和接收数据的数据寄存器 TXR/RXT。

| A[2:0] | 名称 | 复位 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 00H | USR | 0000 1011 | PERR | NF | FERR | OERR | RIDLE | RXIF | TIDLE | TXIF |
| 01H | UCR1 | 0000 0x00 | UARTEN | BNO | PREN | PRT | STOPS | TXBRK | RX8 | TX8 |
| 02H | UCR2 | 0000 0000 | TXEN | RXEN | BRGH | ADDEN | WAKE | RIE | TIIE | TEIE |
| 03H | BRG | xxxx xxxx | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 |
| 04H | UCR3 | 0--- ---- | URST | — | — | — | — | — | — | — |
| 05H~ 07H | 未使用 | ---- ---- | 保留位 | | | | | | | |

**UART 寄存器列表**

## USR 寄存器

寄存器 USR 是 UART 的状态寄存器，可以通过程序读取。所有 USR 位是只读的。

详细解释如下：

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | PERR | NF | FERR | OERR | RIDLE | RXIF | TIDLE | TXIF |
| R/W | R | R | R | R | R | R | R | R |
| POR | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

Bit 7 **PERR**：奇偶校验出错标志位
0：奇偶校验正确
1：奇偶校验出错
PERR 是奇偶校验出错标志位。若 PERR=0，奇偶校验正确；若 PERR=1，接收到的数据奇偶校验出错。只有使能了奇偶校验此位才有效。可使用软件清除该标志位，即先读取 USR 寄存器再读 RXR 寄存器来清除此位。

Bit 6 **NF**：噪声干扰标志位
0：没有受到噪声干扰
1：受到噪声干扰
NF 是噪声干扰标志位。若 NF=0，没有受到噪声干扰；若 NF=1，UART 接收数据时受到噪声干扰。它与 RXIF 在同周期内置位，但不会与溢出标志位同时置位。可使用软件清除该标志位，即先读取 USR 寄存器再读 RXR 寄存器将清除此标志位。

Bit 5 **FERR**：帧错误标志位
0：无帧错误发生
1：有帧错误发生
FREE 是帧错误标志位。若 FREE=0，没有帧错误发生；若 FREE=1，当前的数据发生了帧错误。可使用软件清除该标志位，即先读取 USR 寄存器再读 RXR 寄存器来清除此位。

Bit 4 **OERR**：溢出错误标志位
0：无溢出错误发生
1：有溢出错误发生
OERR 是溢出错误标志位，表示接收缓冲器是否溢出。若 OERR=0，没有溢出错误；若 OERR=1，发生了溢出错误，它将影响下一组数据的接收。可通过软件清除该标志位，即先读取 USR 寄存器再读 RXR 寄存器将清除此标志位。

Bit 3 **RIDLE**：接收状态标志位
0：正在接收数据
1：接收器空闲
RIDLE 是接收状态标志位。若 RIDLE=0，正在接收数据；若 RIDLE=1，接收器空闲。在接收到停止位和下一个数据的起始位之间，RIDLE 被置位，表明 UART 空闲，RX 脚处于逻辑高状态。

Bit 2 **RXIF**：接收寄存器状态标志位
0：RXR 寄存器为空
1：RXR 寄存器含有有效数据
RXIF 是接收寄存器状态标志位。当 RXIF=0，RXR 寄存器为空；当 RXIF=1，RXR 寄存器接收到新数据。当数据从移位寄存器加载到 RXR 寄存器中，如果 UCR2 寄存器中的 RIE=1，则会触发中断。当接收数据时检测到一个或多个错误时，相应的标志位 NF、FERR 或 PERR 会在同一周期内置位。读取 USR 寄存器再读 RXR 寄存器，如果 RXR 寄存器中没有新的数据，那么将清除 RXIF 标志。

Bit 1 **TIDLE**：数据发送完成标志位
0：数据传输中
1：无数据传输
TIDLE 是数据发送完成标志位。若 TIDLE=0，数据传输中。当 TXIF=1 且数据发送完毕或者暂停字被发送时，TIDLE 置位。TIDLE=1，TX 引脚空闲且处于逻辑高状态。读取 USR 寄存器再写 TXR 寄存器将清除 TIDLE 位。数据字符或暂停字就绪时，不会产生该标志位。

Bit 0 **TXIF**：发送数据寄存器 TXR 状态位
0：数据还没有从缓冲器加载到移位寄存器中
1：数据已从缓冲器中加载到移位寄存器中（TXR 数据寄存器为空）
TXIF 是发送数据寄存器为空标志位。若 TXIF=0，数据还没有从缓冲器加载到移位寄存器中；若 TXIF=1，数据已从缓冲器中加载到移位寄存器中。读取 USR 存器再写 TXR 寄存器将清除 TXIF。当 TXEN 被置位，由于发送缓冲器未满，TXIF 也会被置位。

### UCR1 寄存器

UCR1 和 UCR2 是 UART 的两个控制寄存器，用来定义各种 UART 功能，例如 UART 的使能与除能、奇偶校验控制和传输数据的长度等等。

详细解释如下：

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | UARTEN | BNO | PREN | PRT | STOPS | TXBRK | RX8 | TX8 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | W |
| POR | 0 | 0 | 0 | 0 | 0 | 0 | × | 0 |

“×”未知

Bit 7 **UARTEN**：UART 功能使能位
0：UART 除能，TX 和 RX 脚为高阻抗状态
1：UART 使能，TX 和 RX 脚作为 UART 功能引脚
此位为 UART 的使能位。UARTEN=0，UART 除能，RX 和 TX 可用作普通的输入输出口；UARTEN=1，UART 使能，TX 和 RX 将分别由 TXEN 和 RXEN 控制。当 UART 被除能将清除缓冲器，所有缓冲器中的数据将被忽略，另外波特率计数器、错误和状态标志位被复位，TXEN、RXEN、TXBRK、RXIF、OERR、FERR、PERR 和 NF 清零而 TIDLE、TXIF 和 RIDLE 置位，UCR1、UCR2 和 BRG 寄存器中的其它位保持不变。若 UART 工作时 UARTEN 清零，所有发送和接收将停止，模块也将复位成上述状态。当 UART 再次使能时，它将在上次配置下重新工作。

Bit 6 **BNO**：发送数据位数选择位
0：8-bit 传输数据
1：9-bit 传输数据
BNO 是发送数据位数选择位。BNO=1，传输数据为 9 位；BNO=0，传输数据为 8 位。若选择了 9 位数据传输格式，RX8 和 TX8 将分别存储接收和发送数据的第 9 位。

Bit 5 **PREN**：奇偶校验使能位
0：奇偶校验除能
1：奇偶校验使能
此位为奇偶校验使能位。PREN=1，使能奇偶校验；PREN=0，除能奇偶校验。

Bit 4 **PRT**：奇偶校验选择位
0：偶校验
1：奇校验
奇偶校验选择位。PRT=1，奇校验；PRT=0，偶校验。

Bit 3 **STOPS**：停止位的长度选择位
0：有一位停止位
1：有两位停止位
此位用来设置停止位的长度。STOP=1，有两位停止位；STOP=0，只有一位停止位。

Bit 2 **TXBRK**：暂停字发送控制位
0：没有暂停字要发送
1：发送暂停字
TXBRK 是暂停字发送控制位。TXBRK=0，没有暂停字要发送，TX 引脚正常操作；TXBRK=1，将会发送暂停字，发送器将发送逻辑“0”。若 TXBRK 为高，缓冲器中数据发送完毕后，发送器将至少保持 13 位宽的低电平直至 TXBRK 复位。

Bit 1 **RX8**：接收 9-bit 数据传输格式中第 8 位（只读）
此位只有在传输数据为 9 位的格式中有效，用来存储接收数据的第 9 位。BNO 是用来控制传输位数是 8 位还是 9 位。

Bit 0 **TX8**：发送 9-bit 数据传输格式中的第 8 位（只写）
此位只有在传输数据为 9 位的格式中有效，用来存储发送数据的第 9 位。BNO 是用来控制传输位数是 8 位还是 9 位。

### UCR2 寄存器

UCR2 是 UART 的第二个控制寄存器，它的主要功能是控制发送器、接收器以及各种 UART 中断源的使能或除能。它也可用来控制波特率，使能接收唤醒和地址侦测。

详细解释如下：

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TXEN | RXEN | BRGH | ADDEN | WAKE | RIE | TIIE | TEIE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | W |
| POR | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

Bit 7 **TXEN**：UART 发送使能位
0：UART 发送除能
1：UART 发送使能
此位为发送使能位。TXEN=0，发送将被除能，发送器立刻停止工作。另外缓冲器将被复位，此时 TX 引脚处于高阻抗状态。若 TXEN=1 且 UARTEN=1，则发送将被使能，TX 引脚将由 UART 来控制。在数据传输时清除 TXEN 将中止数据发送且复位发送器，此时 TX 引脚作为普通的输入输出端口使用。

Bit 6 **RXEN**：UART 接收使能位
0：UART 接收除能
1：UART 接收使能
此位为接收使能位。RXEN=0，接收将被除能，接收器立刻停止工作。另外缓冲器将被复位，此时 RX 引脚处于高阻抗状态。若 RXEN=1 且 UARTEN=1，则接收将被使能，RX 引脚将由 UART 来控制。在数据传输时清除 RXEN 将中止数据接收且复位接收器，此时 RX 引脚可作为普通输入输出端口使用。

Bit 5 **BRGH**：波特率发生器高低速选择位
0：低速波特率
1：高速波特率
此位为波特率发生器高低速选择位，它和 BRG 寄存器一起控制 UART 的波特率。BRGH=1，为高速模式；BRGH=0，为低速模式。

Bit 4 **ADDEN**：地址检测使能位
0：地址检测除能
1：地址检测使能
此位为地址检测使能和除能位。ADDEN=1，地址检测使能，此时数据的第 8 位 (BON=0) 或第 9 位 (BON=1) 为高，那么接到的是地址而非数据。若相应的中断使能且接收到的值最高位为 1，那么中断请求标志将会被置位，若最高位为 0，那么将不会产生中断且收到的数据也会被忽略。

Bit 3 **WAKE**：RX 脚下降沿唤醒功能使能位
0：RX 脚下降沿唤醒功能除能
1：RX 脚下降沿唤醒功能使能
此位为接收唤醒功能的使能和除能位。若 WAKE=1 且在 IDLE 或 SLEEP 模式下，RX 引脚的下降沿将唤醒单片机。若 WAKE=0 且在 IDLE 或 SLEEP 模式下，RX 引脚的任何边沿都不能唤醒单片机。

Bit 2 **RIE**：接收中断使能位
0：接收中断除能
1：接收中断使能
此位为接收中断使能或除能位。若 RIE=1，当 OERR 或 RXIF 置位时，UART 的中断请求标志置位；若 RIE=0，UART 中断请求标志不受 OERR 和 RXIF 影响。

Bit 1 **TIIE**：发送器空闲中断使能位
0：发送器空闲中断除能
1：发送器空闲中断使能
此位为发送器空闲中断的使能或除能位。若 TIIE=1，当 TIDLE 置位时，UART 的中断请求标志置位；若 TIIE=0，UART 中断请求标志不受 TIDLE 的影响。

Bit 0 **TEIE**：发送寄存器为空中断使能位
0：发送寄存器为空中断除能
1：发送寄存器为空中断使能
此位为发送寄存器为空中断的使能或除能位。若 TEIE=1，当 TXIF 置位时，UART 的中断请求标志置位；若 TEIE=0，UART 中断请求标志不受 TXIF 的影响。

### UCR3 寄存器

UCR3 是 UART 的最后一个控制寄存器，它的主要功能是 UART 模块的软件复位操作。UCR3 中唯一可用的位是 URST，是软件复位控制位。此位为“0”时，UART 正常运行；若此位为“1”，整个 UART 模块复位。此时，发送和接收器也复位，UART 相关寄存器都保持为复位发生后的 POR 状态。

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | URST | — | — | — | — | — | — | — |
| R/W | R/W | — | — | — | — | — | — | — |
| POR | 0 | — | — | — | — | — | — | — |

Bit 7 **URST**：UART 软件复位位
0：UART 不发生复位
1：UART 发生复位

Bit 6~0 未使用，读为“0”

## 波特率发生器

UART 自身具有一个波特率发生器，通过它可以设定数据传输速率。波特率是由一个独立的内部 8 位计数器产生，它由 BRG 寄存器和 UCR2 寄存器的第 2 位 BRGH 来控制。BRGH 是决定波特率发生器处于高速模式还是低速模式，从而决定计算公式的选用。BRG 寄存器的值 N 可根据下表中的公式计算，N 的范围是 0 到 255。

| UCR2 的 BRGH 位 | 0 | 1 |
|---|---|---|
| 波特率 (BR) | $\frac{f_{CLKI}}{[64(N+1)]}$ | $\frac{f_{CLKI}}{[16(N+1)]}$ |

为得到相应的波特率，首先需要设置 BRGH 来选择相应的计算公式从而算出 BRG 的值。由于 BRG 的值不连续，所以实际波特率和理论值之间有一个偏差。下面举例怎样计算 BRG 寄存器中的值 N 和误差。

### 波特率和误差的计算

系统选用 4M 晶振且 BRGH=0，若期望的波特率为 4800，计算它的 BRG 寄存器的值 N，实际波特率和误差。

$$\text{根据上表，波特率 BR}= \frac{f_{CLKI}}{[64(N+1)]}$$

$$\text{转换后的公式 N}= \frac{f_{CLKI}}{(BR\times 64)}-1$$

$$\text{代入参数 N} = \frac{4000000}{(4800\times 64)}-1 \ =12.0208$$

取最接近的值，十进制 12 写入 BRG 寄存器，实际波特率如下

$$\text{BR}= \frac{4000000}{[64(12+1)]} =4808$$

$$\text{误差} \ \frac{4808-4800}{4800} \ =0.16\%$$

下面两表给出 BRGH 取不同值时的实际波特率和误差。

| 波特率 K/BPS | BRGH=0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $f_{CLKI}$=4MHz | | | $f_{CLKI}$=3.579545MHz | | | $f_{CLKI}$=7.159MHz | | |
| | BRG | Kbaud | 误差 (%) | BRG | Kbaud | 误差 (%) | BRG | Kbaud | 误差 (%) |
| 0.3 | 207 | 0.300 | 0.16 | 185 | 0.300 | 0.00 | — | — | — |
| 1.2 | 51 | 1.202 | 0.16 | 46 | 1.190 | -0.83 | 92 | 1.203 | 0.23 |
| 2.4 | 25 | 2.404 | 0.16 | 22 | 2.432 | 1.32 | 46 | 2.380 | -0.83 |
| 4.8 | 12 | 4.808 | 0.16 | 11 | 4.661 | -2.90 | 22 | 4.863 | 1.32 |
| 9.6 | 6 | 8.929 | -6.99 | 5 | 9.321 | -2.90 | 11 | 9.322 | -2.90 |
| 19.2 | 2 | 20.833 | 8.51 | 2 | 18.643 | -2.90 | 5 | 18.643 | -2.90 |
| 38.4 | — | — | — | — | — | — | 2 | 32.286 | -2.90 |
| 57.6 | 0 | 62.500 | 8.51 | 0 | 55.930 | -2.90 | 1 | 55.930 | -2.90 |
| 115.2 | — | — | — | — | — | — | 0 | 111.859 | -2.90 |

**BRGH=0 时的波特率和误差**

| 波特率 K/BPS | BRGH=1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $f_{CLKI}$=4MHz | | | $f_{CLKI}$=3.579545MHz | | | $f_{CLKI}$=7.159MHz | | |
| | BRG | Kbaud | 误差 (%) | BRG | Kbaud | 误差 (%) | BRG | Kbaud | 误差 (%) |
| 0.3 | — | — | — | — | — | — | — | — | — |
| 1.2 | 207 | 1.202 | 0.16 | 185 | 1.203 | 0.23 | — | — | — |
| 2.4 | 103 | 2.404 | 0.16 | 92 | 2.406 | 0.23 | 185 | 2.406 | 0.23 |
| 4.8 | 51 | 4.808 | 0.16 | 46 | 4.76 | -0.83 | 92 | 4.811 | 0.23 |
| 9.6 | 25 | 9.615 | 0.16 | 22 | 9.727 | 1.32 | 46 | 9.520 | -0.83 |
| 19.2 | 12 | 19.231 | 0.16 | 11 | 18.643 | -2.90 | 22 | 19.454 | 1.32 |
| 38.4 | 6 | 35.714 | -6.99 | 5 | 37.286 | -2.90 | 11 | 37.286 | -2.90 |
| 57.6 | 3 | 62.5 | 8.51 | 3 | 55.930 | -2.90 | 7 | 55.930 | -2.90 |
| 115.2 | 1 | 125 | 8.51 | 1 | 111.86 | -2.90 | 3 | 111.86 | -2.90 |
| 250 | 0 | 250 | 0 | — | — | — | — | — | — |

**BRGH=1 时的波特率和误差**

**BRG 寄存器**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | BRG7 | BRG6 | BRG5 | BRG4 | BRG3 | BRG2 | BRG1 | BRG0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR | × | × | × | × | × | × | × | × |

“×”未知

Bit 7~0 **BRG7~BRG0：**波特率值

软件设置 BRGH 位（设置波特率发生器的速度）和 BRG 寄存器（设置波特率的值），一起控制 UART 的波特率。

## UART 模块的设置与控制

UART 采用标准的不归零码传输数据，这种方法通常被称为 NRZ 法。它由 1 位起初位，8 位或 9 位数据位和 1 位或者两位停止位组成。奇偶校验是由硬件自动完成的，可设置成奇校验、偶校验和无校验三种格式。常用的数据传输格式由 8 位数据位，1 位停止位，无校验组成，用 8、N、1 表示，它是系统上电的默认格式。数据位数、停止位数和奇偶校验由 UCR1 寄存器的 BNO、PRT、PREN 和 STOPS 设定。用于数据发送和接收的波特率由一个内部的 8 位波特率发送器产生，数据传输时低位在前高位在后。尽管 UART 发送器和接收器在功能上相互独立，但它们使用相同的数据传输格式和波特率，在任何情况下，停止位是必须的。

### UART 的使能和除能

UART 是由 UCR1 寄存器的 UARTEN 位来使能和除能的。若 UARTEN、TXEN 和 RXEN 都为高，则 TX 和 RX 分别为 UART 的发送端口和接收端口。若没有数据发送，TX 引脚默认状态为高电平。

UARTEN 清零将除能 TX 和 RX，使其处于高阻抗状态。当 UART 被除能时将清空缓冲器，所有缓冲器中的数据将被忽略，另外错误和状态标志位被复位，TXEN、RXEN、TXBRK、RXIF、OERR、FERR、PERR 和 NF 清零而 TIDLE、TXIF 和 RIDLE 置位，UCR1、UCR2 和 BRG 寄存器中的其它位保持不变。若 UART 工作时 UARTEN 清零，所有发送和接收将停止，模块也将复位成上述状态。当 UART 再次使能时，它将在上次配置下重新工作。

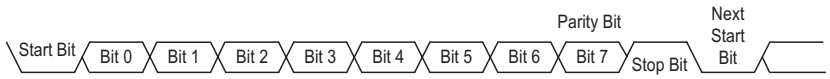
### 数据位、停止位位数以及奇偶校验的选择

数据传输格式由数据长度、是否校验、校验类型、地址位以及停止位长度组成。它们都是由 UCR1 寄存器的各个位控制的。BNO 决定数据传输是 8 位还是 9 位；PRT 决定校验类型；PRTEN 决定是否选择奇偶校验；而 STOPS 决定选用 1 位还是 2 位停止位。下表列出了各种数据传输格式。地址位用来确定此帧是否为地址。停止位的长度和数据位的长度无关。

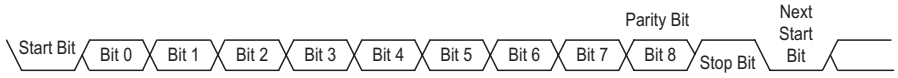
| 起始位 | 数据位 | 地址位 | 校验位 | 停止位 |
|---|---|---|---|---|
| **8 位数据位** | | | | |
| 1 | 8 | 0 | 0 | 1 |
| 1 | 7 | 0 | 1 | 1 |
| 1 | 7 | 1 | 0 | 1 |
| **9 位数据位** | | | | |
| 1 | 9 | 0 | 0 | 1 |
| 1 | 8 | 0 | 1 | 1 |
| 1 | 8 | 1 | 0 | 1 |

发送和接收数据格式

下图是传输 8 位和 9 位数据的波形。




**8-Bit Data Format**




**9-Bit Data Format**

### UART 发送器

UCR1 寄存器的 BNO 位是控制数据传输的长度。BNO=1 其长度为 9 位，第 9 位 MSB 存储在 UCR1 寄存器的 TX8 中。发送器的核心是发送移位寄存器 TSR，它的数据由发送寄存器 TXR 提供，应用程序只须将发送数据写入 TXR 寄存器。上组数据的停止位发出前，TSR 寄存器禁止写入。如果还有新的数据要发送，一旦停止位发出，待发数据将会从 TXR 寄存器加载到 TSR 寄存器。TSR 不像其它寄存器一样映射到数据存储器，所以应用程序不能对其进行读写操作。TXEN=1，发送使能，但若 TXR 寄存器没有数据或者波特率没有设置，发送器将不会工作。先写 TXR 寄存器再写 TXEN 也会触发发送。当发送器使能，若 TSR 寄存器为空，数据写入 TXR 寄存器将会直接加载到 TSR 寄存器中。发送器工作时，TXEN 清零，发送器将立刻停止工作并且复位，此时 TX 引脚处于高阻抗状态。

#### 发送数据

当 UART 发送数据时，数据从移位寄存器中移到 TX 引脚上，其低位在前高位在后。在发送模式中，TXR 寄存器在内部总线和发送移位寄存器间形成一个缓冲。如果选择 9 位数据传输格式，最高位 MSB 存储在 UCR1 寄存器的 TX8 中。

发送器初始化可由如下步骤完成：

- 正确地设置 BNO、PRT、PREN 和 STOPS 位以确定数据长度、校验类型和停止位长度。
- 设置 BRG 寄存器，选择期望的波特率。
- 置高 TXEN，使能 UART 发送器且使 TX 作为 UART 的发送端。
- 读取 USR 寄存器，然后将待发数据写入 TXR 寄存器。注意，此步骤会清除 TXIF 标志位。

如果要发送多个数据只需重复上一步骤。

当 TXIF=0 时，数据将禁止写入 TXR 寄存器。可以通过以下步骤来清除 TXIF：

- 读取 USR 寄存器
- 写 TXR 寄存器

只读标志位 TXIF 由 UART 硬件置位。若 TXIF=1，TXR 寄存器为空，其它数据可以写入而不会覆盖以前的数据。若 TEIE=1，TXIF 标志位会影响中断。在数据传输时，写 TXR 指令会将待发数据暂存在 TXR 寄存器中，当前数据发送完毕后，待发数据被加载到发送移位寄存器中。当发送器空闲时，写 TXR 指令会将数据直接加载到 TSR 寄存器中，数据传输立刻开始且 TXIF 置位。当一帧数据发送完毕，TIDLE 将被置位。

可以通过以下步骤来清除 TIDLE：

- 读取 USR 寄存器
- 写 TXR 寄存器

清除 TXIF 和 TIDLE 软件执行次序相同。

### 发送暂停字

若 TXBRK=1，下一帧将会发送暂停字。它是同一个起始位、13 × N（N=1，2……）位逻辑 0 以及停止位组成。置位 TXBRK 将会发送暂停字，而清除 TXBRK 将产生停止位，传输暂停字不会产生中断。需要注意思的是，暂停字至少 13 位宽。若 TXBRK 持续为高，那么发送器会一直发送暂停字；当应用程序清除了 TXBRK，发送器将传输最后一帧暂停字再加上一位或者两位停止位。暂停字后的高电平保证下一帧数据起始位的检测。

## UART 接收器

UART 接收器支持 8 位或者 9 位数据接收。若 BNO=1，数据长度为 9 位，而最高位 MSB 存放在 UCR1 寄存器的 RX8 中。接收器的核心是串行移位寄存器 RSR。RX 引脚上的数据送入数据恢复器中，它在 16 倍波特率的频率下工作，而串行移位器工作在正常波特率下。当在 RX 引脚上检测到停止位，数据从 RSR 寄存器中加载到 RXR 寄存器。RX 引脚上的每一位数据会被采样三次以判断其逻辑状态。RSR 不像其它寄存器一样映射在数据存储器，所以应用程序不能对其进行读写操作。

### 接收数据

当 UART 接收数据时，数据低位在前高位在后，连续地从 RX 引脚进入。RXR 寄存器在内部总线和接收移位寄存器间形成一个缓冲。RXR 寄存器是一个两层的 FIFO 缓冲器，它能保存两帧数据的同时接收第三帧数据，应用程序必须保证在接收完第三帧前读取 RXR 寄存器，否则忽略第三帧数据并且发生溢出错误。

接收器的初始化可由如下步骤完成：

- 正确地设置 BNO、PRT、PREN 和 STOPS 位以确定数据长度、校验类型和停止位长度。
- 设置 BRG 寄存器，选择期望的波特率。
- 置高 TXEN，使能 UART 发送器且使 RX 作为 UART 的发送端。
- 此时接收器被使能并检测起始位。
- 接收数据将会发生如下事件：
- 当 RXR 寄存器中有一帧以上的数据时，USR 寄存器中的 RXIF 位将会置位。
- 若 RIE=1，数据从 RSR 寄存器加载到 RXR 寄存器中将产生中断。
- 若接收器检测到没到帧错误、噪声干扰错误、奇偶出错或溢出错误，那么相应的错误标志位置位。

可以通过如下步骤来清除 RXIF：

- 读取 USR 寄存器
- 读取 RXR 寄存器

### 接收暂停字

UART 接收任何暂停字都会当作帧错误处理。接收器只根据 BNO 和 STOPS 位确定一帧数据的长度。若暂停字位数大于 BNO 和 STOPS 位指定的长度，接收器认为接收已完毕，RXIF 和 FERR 置位，RXR 寄存器清 0，若相应的中断允许且 RIDLE 为高将会产生中断。若暂停字较长，接收器收到起始位、数据位将会置位 FERR 标志，且在下一起始位前必须检测到有效的停止位。暂停字只会被认为包含信息 0 且会置位 FERR 标志。暂停字将会加载到缓冲器中，在接收到停止位前不会再接收数据，没有检测到停止位也会置位只读标志位 RIDLE。

UART 接收到暂停字会产生以下事件：

- 帧错误标志位 FERR 置位。
- RXR 寄存器清零。
- OERR、NF、PERR、RIDLE 或 RXIF 可能会置位。

### 空闲状态

当 UART 接收数据时，即在起初位和停止位之间，USR 寄存器的接收标志位 RIDLE 清零。在停止位和下一帧数据的起始位之间，RIDLE 被置位，表示接收器空闲。

### 接收中断

USR 寄存器的只读标志位 RXIF 由接收器的边缘触发置位。若 RIE=1，数据从移位寄存器 RSR 加载到 RXR 寄存器时产生中断，同样地，溢出也会产生中断。

### 接收错误处理

UART 会产生几种接收错误，下面部分将描述各错误以及怎样处理。

### 溢出 – OERR 标志

RXR 寄存器是一个两层的 FIFO 缓冲器，它能保存两帧数据的同时接收第三帧数据，应用程序必须保证在接收完第三帧前读取 RXR 寄存器，否则发生溢出错误。

产生溢出错误时将会发生以下事件：

- USR 寄存器中 OERR 被置位。
- RXR 寄存器中数据不会丢失。
- RSR 寄存器数据将会被覆盖。
- 若 RIE=1，将会产生中断。

### 噪声干扰 – NF 标志

数据恢复时多次采样可以有效的鉴别出噪声干扰。当检测到数据受到噪声干扰时将会发生以下事件：

- 在 RXIF 上升沿，USR 寄存器中只读标志位 NF 置位。
- 数据从 RSR 寄存器加载到 RXR 寄存器中。
- 不产生中断，此位置位的同时由 RXIF 请求中断。

先读取 USR 寄存器再读取 RXR 寄存器将复位 NF。

### 帧错误 – FERR 标志

若在停止位上检测到 0，USR 寄存器中只读标志 FERR 置位。若选择两位停止位，此两位都必须为高，否则将置位 FERR。它同数据一起存储在缓冲器中，可被任何复位清零。

### 奇偶校验错误 – PERR 标志

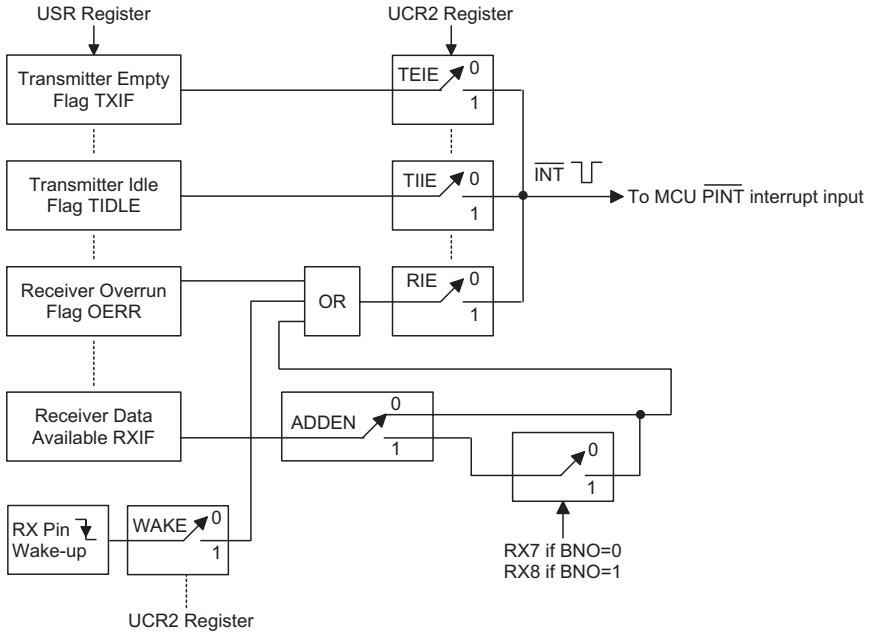
若接收到数据出现奇偶校验错误，USR 寄存器中只读标志 PERR 置位。只有使能了奇偶校验，选择了校验类型，此标志位才有效。它同数据一起存储在缓冲器中，可被任何复位清除。注意，FERR 和 PERR 与相应的数据一起存储有缓冲器中，在读取数据之前必须先访问错误标志位。

## UART 模块中断结构

UART 拥有几个单独的中断，这几个中断产生时，$\overline{\text{PINT}}$ 脚上会产生一个低脉冲信号。发送寄存器为空、发送器空闲、接收器数据有效、溢出和地址检测和 RX 引脚唤醒都会产生中断。若 UART 中断允许且堆栈未满，程序将会跳转到相应的中断向量执行中断服务程序，而后再返回主程序。其中四种含有与 USR 寄存器相关的标志位，若 UCR2 寄存器中相应中断允许位被置位，USR 寄存器中标志位将会产生中断。发送器有两个相应的中断允许位而接收器共用一个中断允许位。这些允许位可用于禁止个别的 UART 中断源。

地址检测也是 UART 的中断源，它没有相应的标志位，若 UCR2 寄存器中 ADDEN=1，当检测到地址将会产生 UART 中断。RX 引脚唤醒也可以产生 UART 中断，它没有相应的标志位，当 UXR2 中的 WAKE 和 RIE 位被置位，RX 引脚上有下降沿可以唤醒单片机。应注意，RX 唤醒中断发生时，系统必须延时 $t_{SST}$ 才能正常工作。

注意，USR 寄存器标志位为只读状态，软件不能对其进行设置，在进入相应中断服务程序时也不能清除这些标志位，其它中断亦是如此。这些标志位仅在 UART 特定动作发生时才会自动被清除，详细解释见 UART 寄存器章节。整体 UART 中断的使能或除能可由中断控制寄存器中的相关中断使能控制位控制，其中断请求由 UART 模块决定。




**UART 中断框图**

### 地址检测模式

置位 UCR2 寄存器中的 ADDEN 将启动地址检测模式。若此位为“1”，可产生接收数据有效中断，其请求标志位为 RXIF。若 ADDEN 有效，只有在接收到数据最高位为 1 才会产生中断，中断允许位 EURI 和 EMI 也要使能才会产生中断。地址的最高位为第 9 位 (BNO=1) 或第 8 位 (BNO=0)，若此位为高，则接收到的是地址而非数据。只有接收的数据的最后位为高才会产生中断。若 ADDEN 除能，每接收到一个有效数据便会置位 RXIF，而不用考虑数据的最后一位。地址检测和奇偶校验在功能上相互排斥，若地址检测模式使能，必须保证操作的正确，同时必须将奇偶检验使能位清零，除能奇偶校验。

| ADDEN | Bit 9(BNO=1)<br>Bit 8(BNO=0) | 产生 UART 中断 |
|---|---|---|
| 0 | 0 | √ |
| | 1 | √ |
| 1 | 0 | × |
| | 1 | √ |

**ADDEN 位功能**

## UART 模块暂停和唤醒

MCU 和 UART 模块分别进入暂停模式。MCU 进入暂停模式的方法已在 datasheet 前面有所描述。芯片进入暂停模式前，UART 模块需先行进入暂停模式。首先清除 UARTEN 位以除能 UART 模块电路，然后设置 $\overline{SCS}$ 为高以除能 SPI 接口电路。当 UART 和 SPI 接口均进入暂停模式后，UART 模式的 SCK 和 CLKI 时钟源将除能。清除 $\overline{SCS}$，然后置位 UARTEN 可使 UART 模块进入上电状态。当传送数据时，UART 进入暂停模式，发送将停止并且 TX 引脚保持高电平。同样地，当接收数据时 UART 进入暂停模式，数据接收也会停止。当 UART 电路进入暂停模式，USR、UCR1、UCR2 和 UCR3、接收 / 发送寄存器、BRG 寄存器都不会受到影响。

UART 功能中包括了 RX 引脚的唤醒功能，由 UCR2 寄存器中 WAKE 位控制。进入暂停模式前，若该标志位与 UART 允许位 UARTEN、接收器允许位 RXEN 和接收器中断位 RIE 都被置位，则 RX 引脚的下降沿可唤醒单片机。唤醒后系统需延时 $t_{SST}$ 才能正常工作，在此期间，RX 引脚上的任何数据将被忽略。

若要唤醒并产生 UART 中断，除了唤醒使能控制位和接收中断使能控制位需置位外，全局中断允许位 EMI 和相关中断使能控制位也必须置位；若这两标志位没有被置位，那么，单片机将可以被唤醒但不会产生中断。同样唤醒后系统需一定的延时才能正常工作，然后才会产生 UART 中断。

## UART 功能的使用

若使用 UART 功能，需完成下列几个步骤以使 UART 模块能够正常工作。

- SPI 的功能性引脚用于 UART 模块数据的发送与接收时，其引脚重置功能需正确设置。
  为使 MCU 主机 SPI 和 UART 模块从机 SPI 连接正确，SIM 引脚重置寄存器 PRM0 中 PCK 和 PINTB 引脚的设置需同下表中的值保持一致。
  - ♦ HT68FU30
    – PRM0 寄存器 – PCK 和 $\overline{PINT}$ 引脚重置设置

| 位 | 1 | 0 |
|---|---|---|
| 名称 | SIMPS0 | PCKPS |
| 设置值 | 1 | 1 |

  - ♦ HT68FU40/HT68FU50
    – PRM0 寄存器 – PCK 和 $\overline{PINT}$ 引脚重置设置

| 位 | 2 | 1 | 0 |
|---|---|---|---|
| 名称 | SIMPS1 | SIMPS0 | PCKPS |
| 设置值 | 0 | 1 | 1 |

♦ HT68FU60

– PRM0 寄存器—PCK 和 $\overline{\text{PINT}}$ 引脚重置设置

| 位 | 2 | 1 | 0 |
|---|---|---|---|
| 名称 | SIMPS1 | SIMPS0 | PCKPS |
| 设置值 | 1 | 1 | 1 |

- 需设置 SIM 工作模式控制位 SIM2~SIM0(SIMC0.7~5) 使能 SIM，使其以不同的 SPI 时钟频率工作在 SPI 主机模式。

♦ SIMC0 寄存器中的 SIM 工作模式控制位 SIM2~SIM0

| 位 | 2 | 1 | 0 |
|---|---|---|---|
| 名称 | SIM2 | SIM1 | SIM0 |
| 值 | 100,011,010,001,000 | | |

000：SPI 主机模式；SPI 时钟为 $f_{SYS}/4$

001：SPI 主机模式；SPI 时钟为 $f_{SYS}/16$

010：SPI 主机模式；SPI 时钟为 $f_{SYS}/64$

011：SPI 主机模式；SPI 时钟为 $f_{TBC}$

100：SPI 主机模式；SPI 时钟为 TM0 CCRP 匹配频率 /2

101~111：未使用

- PCK 控制位设置为“1”可使能 PCK 输出，此举可为 UART 波特率发生器提供各种 PCK 输出频率，而输出频率则由 SIMC0 寄存器中的 PCKP1 和 PCKP0 位决定。

♦ SIMC0 寄存器中的 PCK 输出频率选择位 PCKP1~PCKP0

| 位 | 3 | 2 |
|---|---|---|
| 名称 | PCKP1 | PCKP0 |
| 值 | 11,10,01,00 | |

00：PCK 输出频率为 $f_{SYS}$

01：PCK 输出频率为 $f_{SYS}/4$

10：PCK 输出频率为 $f_{SYS}/8$

11：PCK 输出频率为 TM0 CCRP 匹配频率 /2

♦ SIMC0 寄存器中的 PCK 输出使能控制位 PCKEN

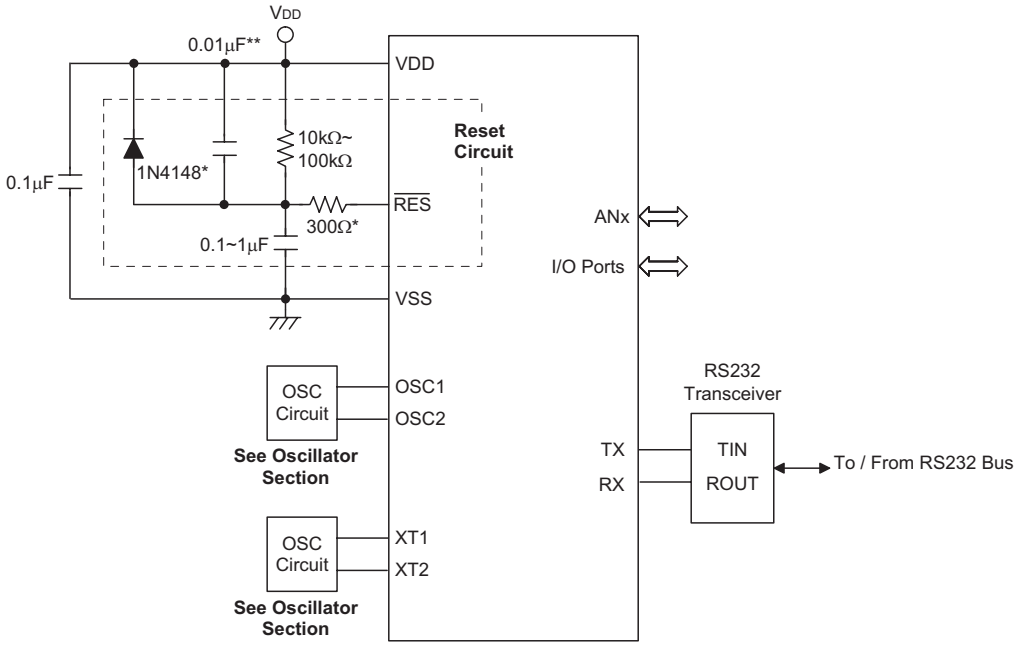
| 位 | 4 |
|---|---|
| 名称 | PCKEN |
| 值 | 1 |

0：除能 PCK 输出

1：使能 PCK 输出

上述条件设置完成后，MCU 可通过设置 SIMEN 位为高使能 SIM 接口，使用 SPI 接口实现与外部 UART 的通信。MCU 主机 SPI 功能的详细描述请参考 datasheet 中串行接口模块章节。

## UART 模块应用电路




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

# 指令集

## 简介

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

为了更加容易理解各种各样的指令码，接下来按功能分组介绍它们。

## 指令周期

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

## 数据的传送

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

## 算术运算

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

## 逻辑和移位运算

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

## 分支和控制转换

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

## 位运算

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

## 查表运算

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

## 其它运算

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

# 指令集概要

下表中说明了按功能分类的指令集，用户可以将该表作为基本的指令参考。

## 惯例

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

| 助记符 | 说明 | 指令周期 | 影响标志位 |
|---|---|---|---|
| **算术运算** | | | |
| ADD A,[m] | ACC 与数据存储器相加，结果放入 ACC | 1 | Z,C,AC,OV |
| ADDM A,[m] | ACC 与数据存储器相加，结果放入数据存储器 | 1[注] | Z,C,AC,OV |
| ADD A, x | ACC 与立即数相加，结果放入 ACC | 1 | Z,C,AC,OV |
| ADC A,[m] | ACC 与数据存储器、进位标志相加，结果放入 ACC | 1 | Z,C,AC,OV |
| ADCM A,[m] | ACC 与数据存储器、进位标志相加，结果放入数据存储器 | 1[注] | Z,C,AC,OV |
| SUB A, x | ACC 与立即数相减，结果放入 ACC | 1 | Z,C,AC,OV |
| SUB A,[m] | ACC 与数据存储器相减，结果放入 ACC | 1 | Z,C,AC,OV |
| SUBM A,[m] | ACC 与数据存储器相减，结果放入数据存储器 | 1[注] | Z,C,AC,OV |
| SBC A,[m] | ACC 与数据存储器、进位标志的反相减，结果放入 ACC | 1 | Z,C,AC,OV |
| SBCM A,[m] | ACC 与数据存储器、进位标志相减，结果放入数据存储器 | 1[注] | Z,C,AC,OV |
| DAA [m] | 将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器 | 1[注] | C |
| **逻辑运算** | | | |
| AND A,[m] | ACC 与数据存储器做“与”运算，结果放入 ACC | 1 | Z |
| OR A,[m] | ACC 与数据存储器做“或”运算，结果放入 ACC | 1 | Z |
| XOR A,[m] | ACC 与数据存储器做“异或”运算，结果放入 ACC | 1 | Z |
| ANDM A,[m] | ACC 与数据存储器做“与”运算，结果放入数据存储器 | 1[注] | Z |
| ORM A,[m] | ACC 与数据存储器做“或”运算，结果放入数据存储器 | 1[注] | Z |
| XORM A,[m] | ACC 与数据存储器做“异或”运算，结果放入数据存储器 | 1[注] | Z |
| AND A, x | ACC 与立即数做“与”运算，结果放入 ACC | 1 | Z |
| OR A, x | ACC 与立即数做“或”运算，结果放入 ACC | 1 | Z |
| XOR A, x | ACC 与立即数做“异或”运算，结果放入 ACC | 1 | Z |
| CPL [m] | 对数据存储器取反，结果放入数据存储器 | 1[注] | Z |
| CPLA [m] | 对数据存储器取反，结果放入 ACC | 1 | Z |
| **递增和递减** | | | |
| INCA [m] | 递增数据存储器，结果放入 ACC | 1 | Z |
| INC [m] | 递增数据存储器，结果放入数据存储器 | 1[注] | Z |
| DECA [m] | 递减数据存储器，结果放入 ACC | 1 | Z |
| DEC [m] | 递减数据存储器，结果放入数据存储器 | 1[注] | Z |
| **移位** | | | |
| RRA [m] | 数据存储器右移一位，结果放入 ACC | 1 | 无 |

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

| 助记符 | 说明 | 指令周期 | 影响标志位 |
|---|---|---|---|
| CLR WDT1 | 预清除看门狗定时器 | 1 | TO,PDF |
| CLR WDT2 | 预清除看门狗定时器 | 1 | TO,PDF |
| SWAP [m] | 交换数据存储器的高低字节，结果放入数据存储器 | 1 注 | 无 |
| SWAPA [m] | 交换数据存储器的高低字节，结果放入 ACC | 1 | 无 |
| HALT | 进入暂停模式 | 1 | TO,PDF |

注：1、对跳转指令而言，如果比较的结果牵涉到跳转即需 2 个周期，如果没有发生跳转，则只需一个周期。
2、任何指令若要改变 PCL 的内容将需要 2 个周期来执行。
3、对于“CLR WDT1”或“CLR WDT2”指令而言，TO 和 PDF 标志位也许会受执行结果影响，“CLR WDT1”和“CLR WDT2”被连续地执行后，TO 和 PDF 标志位会被清除，否则 TO 和 PDF 标志位保持不变。

## 指令定义

**ADC A, [m]** Add data memory and carry to the accumulator
说明：将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。
运算过程：ACC ← ACC+[m]+C
影响标志位：OV、Z、AC、C

**ADCM A, [m]** Add the accumulator and carry to the accumulator
说明：将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。
运算过程：[m] ← ACC+[m]+C
影响标志位：OV、Z、AC、C

**ADD A, [m]** Add data memory to the accumulator
说明：将指定的数据存储器和累加器内容相加，结果存放到累加器。
运算过程：ACC ← ACC+[m]
影响标志位：OV、Z、AC、C

**ADD A, x** Add immediate data to the accumulator
说明：将累加器和立即数相加，结果存放到累加器。
运算过程：ACC ← ACC+x
影响标志位：OV、Z、AC、C

**ADDM A, [m]** Add the accumulator to the data memory
说明：将指定的数据存储器和累加器内容相加，结果存放到指定的数据存储器。
运算过程：[m] ← ACC+[m]
影响标志位：OV、Z、AC、C

**AND A, [m]** Logical AND accumulator with data memory
说明：将累加器中的数据和指定数据存储器内容做逻辑与，结果存放到累加器。
运算过程：ACC ← ACC “AND” [m]
影响标志位：Z

**AND A, x** Logical AND immediate data to the accumulator
说明：将累加器中的数据和立即数做逻辑与，结果存放到累加器。
运算过程：ACC ← ACC “AND” x
影响标志位：Z

**ANDM A, [m]** Logical AND data memory with the accumulator
说明：将指定数据存储器内容和累加器中的数据做逻辑与，结果存放到数据存储器。
运算过程：[m] ← ACC “AND” [m]
影响标志位：Z

**CALL addr** Subroutine call
说明：无条件的调用指定地址的子程序，此时程序计数器先加 1 获得下一个要执行的指令地址并压入堆栈，接着载入指定地址并从新地址执行程序。由于指令需要额外的运算，所以此指令为 2 个周期。
运算过程：Stack ← Program Counter+1
Program Counter ← addr
影响标志位：无

**CLR [m]** Clear data memory
说明：将指定数据存储器的内容清零。
运算过程：[m] ← 00H
影响标志位：无

**CLR [m].i** Clear bit of data memory
说明：将指定数据存储器的 i 位内容清零。
运算过程：[m].i ← 0
影响标志位：无

**CLR WDT** Clear Watchdog Timer
说明：WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。
运算过程：WDT ← 00H
PDF & TO ← 0
影响标志位：TO、PDF

**CLR WDT1** Preclear Watchdog Timer
说明：PDF 和 TO 标志位都被清 0。必须配合 CLR WDT2 一起使用清除 WDT 计时器。当程序仅执行 CLR WDT1，而没有执行 CLR WDT2 时，PDF 与 TO 保留原状态不变。
运算过程：WDT ← 00H
PDF & TO ← 0
影响标志位：TO、PDF

**CLR WDT2** Preclear Watchdog Timer
说明： PDF 和 TO 标志位都被清 0。必须配合 CLR WDT1 一起使用清除 WDT 计时器。当程序仅执行 CLR WDT2，而没有执行 CLR WDT1 时，PDF 与 TO 保留原状态不变。
运算过程： WDT ← 00H
PDF & TO ← 0
影响标志位： TO、PDF

**CPL [m]** Complement data memory
说明： 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或从 0 变 1。
运算过程： [m] ← $\overline{[m]}$
影响标志位： Z

**CPLA [m]** Complement data memory
说明： 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或从 0 变 1，结果被存放回累加器且数据寄存器的内容保持不变。
运算过程： ACC ← $\overline{[m]}$
影响标志位： Z

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

**DEC [m]** Decrement data memory
说明： 将指定数据存储器的内容减 1。
运算过程： [m] ← [m]-1
影响标志位： Z

**DECA [m]** Decrement data memory and place result in the accumulator
说明： 将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器的内容不变。
运算过程： ACC ← [m]-1
影响标志位： Z

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

**INC [m]** Increment data memory
说明： 将指定数据存储器的内容加 1。
运算过程： [m] ← [m]+1
影响标志位： Z

**INCA [m]** Increment data memory and place result in the accumulator
说明： 将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。
运算过程： ACC ← [m]+1
影响标志位： Z

**JMP addr** Directly jump
说明： 程序计数器的内容无条件地由被指定的地址取代，程序由新的地址继续执行。当新的地址被加载时，必须插入一个空指令周期，所以此指令为 2 个周期的指令。
运算过程： PC ← addr
影响标志位： 无

**MOV A, [m]** Move data memory to the accumulator
说明： 将指定数据存储器的内容复制到累加器。
运算过程： ACC ← [m]
影响标志位： 无

**MOV A, x** Move immediate data to the accumulator
说明： 将 8 位立即数载入累加器。
运算过程： ACC ← x
影响标志位： 无

**MOV [m], A** Move the accumulator data to memory
说明： 将累加器的内容复制到指定的数据存储器。
运算过程： [m] ← ACC
影响标志位： 无

**NOP** No operation
说明： 空操作，顺序执行下一条指令。
运算过程： PC ← PC+1
影响标志位： 无

**OR A, [m]** Logical OR accumulator with data memory
说明： 将累加器中的数据和指定的数据存储器内容逻辑或，结果存放到累加器。
运算过程： ACC ← ACC “OR” [m]
影响标志位： Z

**OR A, x** Logical OR immediate data to the accumulator
说明： 将累加器中的数据和立即数逻辑或，结果存放到累加器。
运算过程： ACC ← ACC “OR” x
影响标志位： Z

**ORM A, [m]** Logical OR data memory with accumulator
说明： 将存在指定数据存储器中的数据和累加器逻辑或，结果放到数据存储器。
运算过程： [m] ← ACC “OR” [m]
影响标志位： Z

**RET** Return from subroutine
说明： 将堆栈寄存器中的程序计数器值恢复，程序由取回的地址继续执行。
运算过程： PC ← Stack
影响标志位： 无

**RET A, x** Return and place immediate data in the accumulator
说明： 将堆栈寄存器中的程序计数器值恢复且累加器载入指定的立即数，程序由取回的地址继续执行。
运算过程： PC ← Stack
ACC ← x
影响标志位： 无

**RETI** Return from interrupt
说明： 将堆栈寄存器中的程序计数器值恢复且中断功能通过设置 EMI 位重新使能。EMI 是控制中断使能的主控制位。如果在执行 RETI 指令之前还有中断未被相应，则这个中断将在返回主程序之前被相应。
运算过程： PC ← Stack
EMI ← 1
影响标志位： 无

**RL [m]** Rotate data memory left
说明： 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
运算过程： [m].(i+1) ← [m].i (i=0~6)
[m].0 ← [m].7
影响标志位： 无

**RLA [m]** Rotate data memory left and place result in the accumulator
说明： 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，结果送到累加器，而指定数据存储器的内容保持不变。
运算过程： ACC.(i+1) ← [m].i (i=0~6)
ACC.0 ← [m].7
影响标志位： 无

**RLC [m]** Rotate data memory left through carry
说明： 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位。
运算过程： [m].(i+1) ← [m].i (i=0~6)
[m].0 ← C
C ← [m].7
影响标志位： C

**RLCA [m]** Rotate left through carry and place result in the accumulator
说明： 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位 , 移位结果送回累加器，但是指定数据寄存器的内容保持不变。
运算过程： ACC.(i+1) ← [m].i (i=0~6)
ACC.0 ← C
C ← [m].7
影响标志位： C

**RR [m]** Rotate data memory right
说明： 将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。
运算过程： [m].i ← [m].(i+1) (i=0~6)
[m].7 ← [m].0,
影响标志位： 无

**RRA [m]** Rotate right and place result in the accumulator
说明： 将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。
运算过程： ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← [m].0
影响标志位： 无

**RRC [m]** Rotate data memory right through carry
说明： 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。
运算过程： [m].i ← [m].(i+1) (i=0~6)
[m]. 7 ← C
C ← [m].0
影响标志位： C

**RRCA [m]** Rotate right through carry and place result in the accumulator
说明： 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位 , 移位结果送回累加器，但是指定数据寄存器的内容保持不变。
运算过程： ACC.i ← [m].(i+1) (i=0~6)
ACC.7 ← C
C ← [m].0
影响标志位： C

**SBC A, [m]** Subtract data memory and carry from the accumulator
说明： 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
运算过程： ACC ← ACC − [m] − $\overline{C}$
影响标志位： OV、Z、AC、C

**SBCM A, [m]** Subtract data memory and carry from the accumulator
说明： 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
运算过程： [m] ← ACC − [m] − $\overline{C}$
影响标志位： OV、Z、AC、C

**SDZ [m]** Skip if decrement data memory is 0
说明： 将指定的数据存储器的内容减 1，判断是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
运算过程： [m] ← [m] -1，如果 [m]=0 跳过下一条指令执行
影响标志位： 无

**SDZA [m]** Decrement data memory and place result in ACC,skip if 0
说明： 将指定数据存储器内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
运算过程： ACC ← [m]-1，如果 ACC=0 跳过下一条指令执行
影响标志位： 无

**SET [m]** Set data memory
说明： 将指定数据存储器的每一位设置为 1。
运算过程： [m] ← FFH
影响标志位： 无

**SET [m]. i** Set bit of data memory
说明： 将指定数据存储器的第 i 位设置为 1。
运算过程： [m].i ← 1
影响标志位： 无

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

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

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

**SUB A, [m]** Subtract data memory from the accumulator
说明： 将累加器的内容减去指定的数据存储器的数据，把结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
运算过程： ACC ← ACC － [m]
影响标志位： OV、Z、AC、C

**SUBM A, [m]** Subtract data memory from the accumulator
说明： 将累加器的内容减去指定数据存储器的数据，结果存放到指定的数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
运算过程： [m] ← ACC － [m]
影响标志位： OV、Z、AC、C

**SUB A, x** Subtract immediate data from the accumulator
说明：将累加器的内容减去立即数，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
运算过程：ACC ← ACC － x
影响标志位：OV、Z、AC、C

**SWAP [m]** Swap nibbles within the data memory
说明：将指定数据存储器的低 4 位和高 4 位互相交换。
运算过程：[m].3~[m].0 ↔ [m].7~[m].4
影响标志位：无

**SWAPA [m]** Swap data memory and place result in the accumulator
说明：将指定数据存储器的低 4 位和高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。
运算过程：ACC.3~ACC.0 ← [m].7~[m].4
ACC.7~ACC.4 ← [m].3~[m].0
影响标志位：无

**SZ [m]** Skip if data memory is 0
说明：判断指定数据存储器的内容是否为 0，若为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
运算过程：如果 [m] = 0, 跳过下一条指令执行
影响标志位：无

**SZA [m]** Move data memory to ACC,skip if 0
说明：将指定数据存储器内容复制到累加器，并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
运算过程：ACC ← [m]，如果 [m] = 0，跳过下一条指令执行
影响标志位：无

**SZ [m]. i** Skip if bit I of the data memory is 0
说明：判断指定数据存储器的第 i 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
运算过程：如果 [m].i = 0，跳过下一条指令执行
影响标志位：无

**TABRD [m]** Move the ROM code(specific page) to TBLH and data memory
指令说明 将表格指针 TBLP 和 TBHP 所指的程序代码低字节（特定页）移至指定的数据存储器且将高字节移至 TBLH。
功能表示 [m] ← 程序代码 ( 低字节 )
TBLH ← 程序代码 ( 高字节 )
影响标志位 无

**TABRDC [m]** Move the ROM code(current page) to TBLH and data memory
指令说明 将表格指针 TBLP 所指的程序代码低字节（当前页）移至指定的数据存储器且将高字节移至 TBLH。
功能表示 [m] ← 程序代码 ( 低字节 )
TBLH ← 程序代码 ( 高字节 )
影响标志位 无

**TABRDL [m]** Move the ROM code(last page) to TBLH and data memory
指令说明 将表格指针 TBLP 所指的程序代码低字节（最后一页）移至指定的数据存储器且将高字节移至 TBLH。
功能表示 [m] ← 程序代码 ( 低字节 )
TBLH ← 程序代码 ( 高字节 )
影响标志位 无

**XOR A, [m]** Logical XOR accumulator with data memory
说明： 将累加器的数据和指定的数据存储器内容逻辑异或，结果存放到累加器。
运算过程： ACC ← ACC “XOR” [m]
影响标志位： Z

**XORM A, [m]** Logical XOR data memory with accumulator
说明： 将累加器的数据和指定的数据存储器内容逻辑异或，结果放到数据存储器。
运算过程： [m] ← ACC “XOR” [m]
影响标志位： Z

**XOR A, x** Logical XOR immediate data to the accumulator
说明： 将累加器的数据与立即数逻辑异或，结果存放到累加器。
运算过程： ACC ← ACC “XOR” x
影响标志位： Z

# 封装信息

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

封装信息的相关内容如下所示，点击可链接至 Holtek 网站相关信息页面。

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

## 16-pin DIP (300mil) 外形尺寸

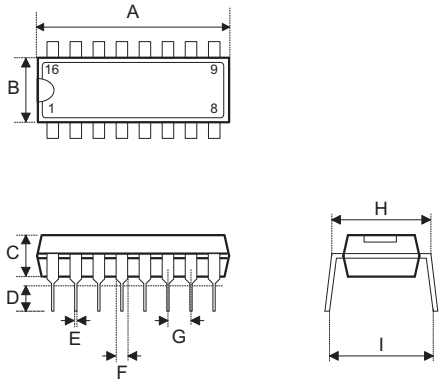



Fig1. Full Lead Packages

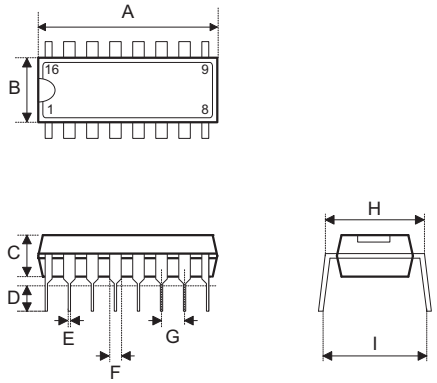



Fig2. 1/2 Lead Packages

见 fig1

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.780 | 0.790 | 0.800 |
| B | 0.240 | 0.250 | 0.280 |
| C | 0.115 | 0.130 | 0.195 |
| D | 0.115 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.060 | 0.070 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 19.81 | 20.07 | 20.32 |
| B | 6.10 | 6.35 | 7.11 |
| C | 2.92 | 3.30 | 4.95 |
| D | 2.92 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.52 | 1.78 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

见 **fig2 – Type 1**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.745 | 0.765 | 0.785 |
| B | 0.275 | 0.285 | 0.295 |
| C | 0.120 | 0.135 | 0.150 |
| D | 0.110 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.050 | 0.060 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

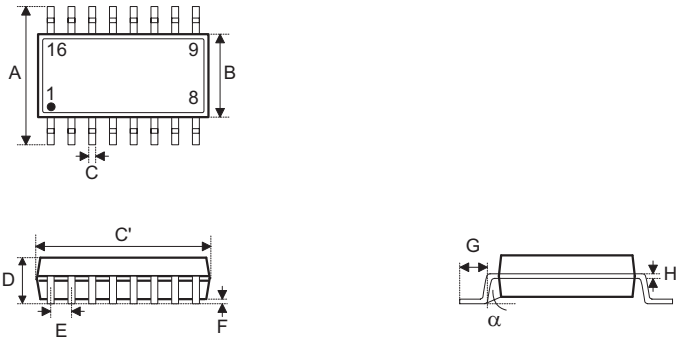
| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 18.92 | 19.43 | 19.94 |
| B | 6.99 | 7.24 | 7.49 |
| C | 3.05 | 3.43 | 3.81 |
| D | 2.79 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.27 | 1.52 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

见 **fig2 – Type 2**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.735 | 0.755 | 0.775 |
| B | 0.240 | 0.250 | 0.280 |
| C | 0.115 | 0.130 | 0.195 |
| D | 0.115 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.060 | 0.070 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 18.67 | 19.18 | 19.69 |
| B | 6.10 | 6.35 | 7.11 |
| C | 2.92 | 3.30 | 4.95 |
| D | 2.92 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.52 | 1.78 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

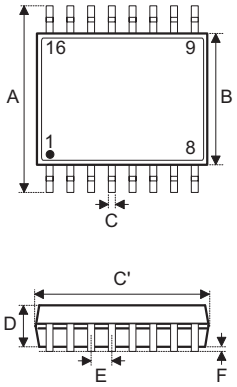
## 16-pin NSOP (150mil) 外形尺寸




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

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

## 16-pin SSOP (150mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.236 BSC | |
| B | | 0.154 BSC | |
| C | 0.008 | — | 0.012 |
| C’ | | 0.193 BSC | |
| D | — | — | 0.069 |
| E | | 0.025 BSC | |
| F | 0.004 | — | 0.010 |
| G | 0.016 | — | 0.050 |
| H | 0.004 | — | 0.010 |
| α | 0° | — | 8° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 6.00 BSC | |
| B | | 3.90 BSC | |
| C | 0.20 | — | 0.30 |
| C’ | | 4.90 BSC | |
| D | — | — | 1.75 |
| E | | 0.635 BSC | |
| F | 0.10 | — | 0.25 |
| G | 0.41 | — | 1.27 |
| H | 0.10 | — | 0.25 |
| α | 0° | — | 8° |

## 20-pin DIP (300mil) 外形尺寸

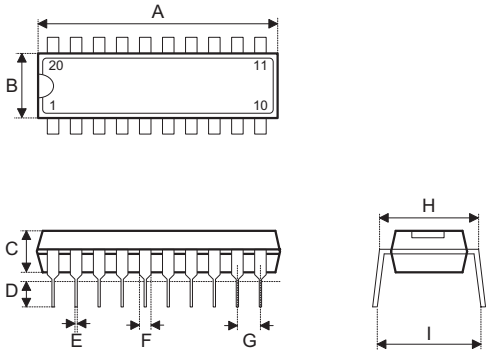



Fig1. Full Lead Packages

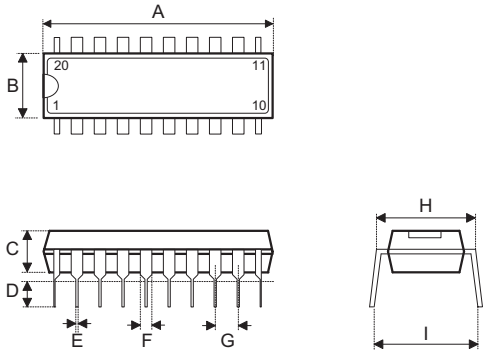



Fig2. 1/2 Lead Packages

见 **fig1**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.980 | 1.030 | 1.060 |
| B | 0.240 | 0.250 | 0.280 |
| C | 0.115 | 0.130 | 0.195 |
| D | 0.115 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.060 | 0.070 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

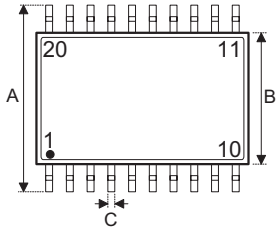
| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 24.89 | 26.16 | 26.92 |
| B | 6.10 | 6.35 | 7.11 |
| C | 2.92 | 3.30 | 4.95 |
| D | 2.92 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.52 | 1.78 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

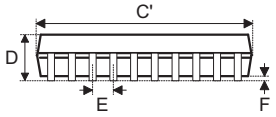
见 **fig2**

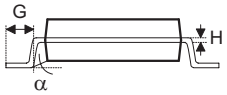
| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.945 | 0.965 | 0.985 |
| B | 0.275 | 0.285 | 0.295 |
| C | 0.120 | 0.135 | 0.150 |
| D | 0.110 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.050 | 0.060 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 24.00 | 24.51 | 25.02 |
| B | 6.99 | 7.24 | 7.49 |
| C | 3.05 | 3.43 | 3.81 |
| D | 2.79 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.27 | 1.52 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

## 20-pin SOP (300mil) 外形尺寸



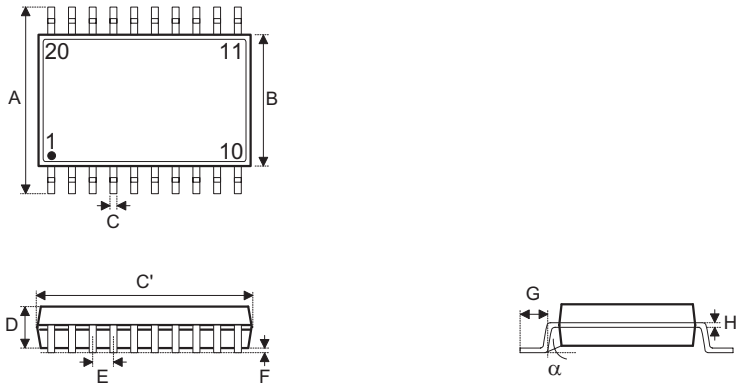




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.406 BSC | |
| B | | 0.295 BSC | |
| C | 0.012 | — | 0.020 |
| C’ | | 0.504 BSC | |
| D | — | — | 0.104 |
| E | | 0.050 BSC | |
| F | 0.004 | — | 0.012 |
| G | 0.016 | — | 0.050 |
| H | 0.008 | — | 0.013 |
| α | 0° | — | 8° |

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

## 20-pin SSOP (150mil) 外形尺寸




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

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

## 24-pin SKDIP (300mil) 外形尺寸

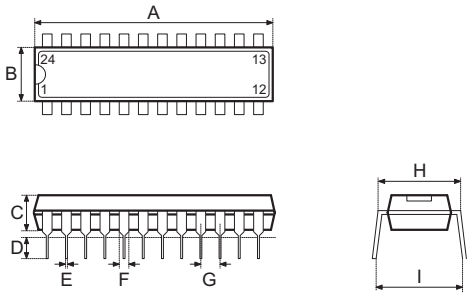



**Fig1. Full Lead Packages**

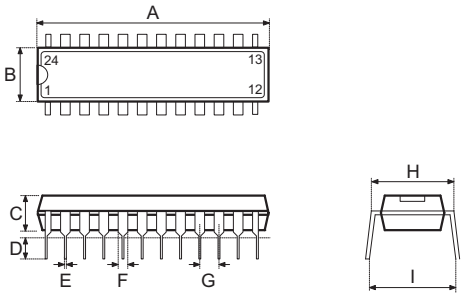



**Fig2. 1/2 Lead Packages**

见 **fig1**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 1.230 | 1.250 | 1.280 |
| B | 0.240 | 0.250 | 0.280 |
| C | 0.115 | 0.130 | 0.195 |
| D | 0.115 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.060 | 0.070 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 31.24 | 31.75 | 32.51 |
| B | 6.10 | 6.35 | 7.11 |
| C | 2.92 | 3.30 | 4.95 |
| D | 2.92 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.52 | 1.78 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

见 **fig2 – Type 1**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 1.145 | 1.165 | 1.185 |
| B | 0.275 | 0.285 | 0.295 |
| C | 0.120 | 0.135 | 0.150 |
| D | 0.110 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.050 | 0.060 |
| G | | 0.100 BSC | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

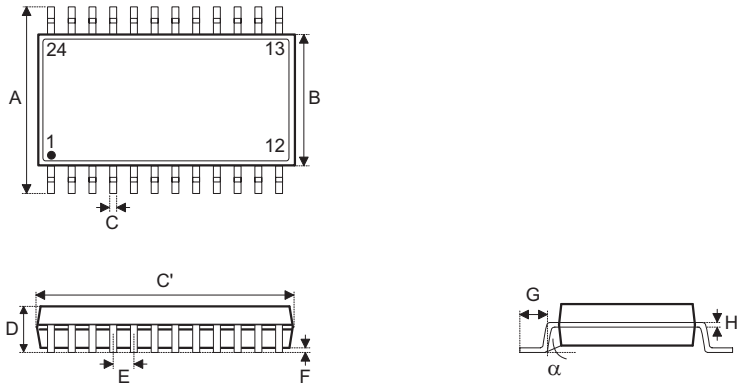
| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 29.08 | 29.59 | 30.10 |
| B | 6.99 | 7.24 | 7.49 |
| C | 3.05 | 3.43 | 3.81 |
| D | 2.79 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.27 | 1.52 |
| G | | 2.54 BSC | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

见 **fig2 – Type 2**

| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 1.160 | 1.185 | 1.195 |
| B | 0.240 | 0.250 | 0.280 |
| C | 0.115 | 0.130 | 0.195 |
| D | 0.115 | 0.130 | 0.150 |
| E | 0.014 | 0.018 | 0.022 |
| F | 0.045 | 0.060 | 0.070 |
| G | 0.100 BSC | | |
| H | 0.300 | 0.310 | 0.325 |
| I | — | — | 0.430 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 29.46 | 30.10 | 30.35 |
| B | 6.10 | 6.35 | 7.11 |
| C | 2.92 | 3.30 | 4.95 |
| D | 2.92 | 3.30 | 3.81 |
| E | 0.36 | 0.46 | 0.56 |
| F | 1.14 | 1.52 | 1.78 |
| G | 2.54 BSC | | |
| H | 7.62 | 7.87 | 8.26 |
| I | — | — | 10.92 |

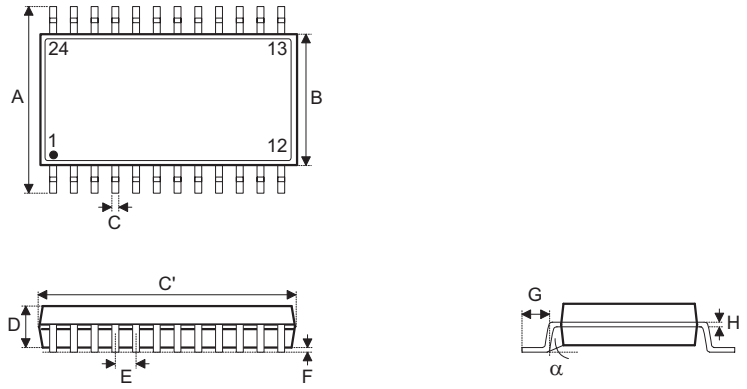
## 24-pin SOP (300mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.406 BSC | |
| B | | 0.295 BSC | |
| C | 0.012 | — | 0.020 |
| C’ | | 0.606 BSC | |
| D | — | — | 0.104 |
| E | | 0.050 BSC | |
| F | 0.004 | — | 0.012 |
| G | 0.016 | — | 0.050 |
| H | 0.008 | — | 0.013 |
| α | 0° | — | 8° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 10.30 BSC | |
| B | | 7.50 BSC | |
| C | 0.31 | — | 0.51 |
| C’ | | 15.40 BSC | |
| D | — | — | 2.65 |
| E | | 1.27 BSC | |
| F | 0.10 | — | 0.30 |
| G | 0.40 | — | 1.27 |
| H | 0.20 | — | 0.33 |
| α | 0° | — | 8° |

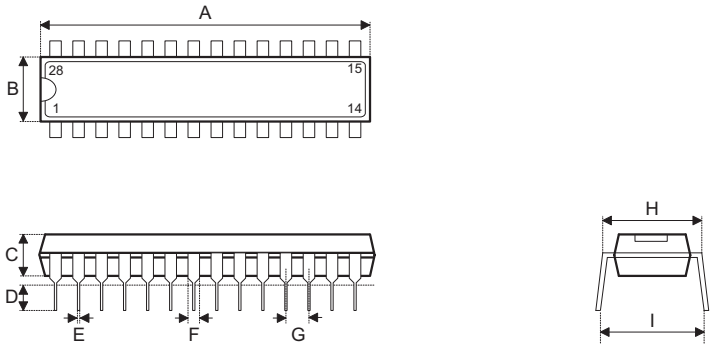
## 24-pin SSOP (150mil) 外形尺寸




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

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

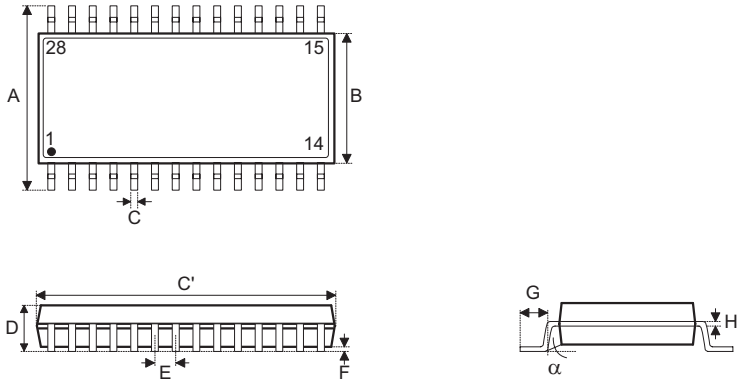
## 28-pin SKDIP (300mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 1.380 | — | 1.420 |
| B | 0.280 | — | 0.310 |
| C | 0.060 | — | 0.130 |
| D | 0.125 | — | 0.200 |
| E | 0.015 | — | 0.022 |
| F | 0.045 | — | 0.065 |
| G | 0.100 BSC | | |
| H | 0.300 | — | 0.325 |
| I | — | — | 0.400 |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 35.05 | — | 36.07 |
| B | 7.11 | — | 7.87 |
| C | 1.52 | — | 3.30 |
| D | 3.18 | — | 5.08 |
| E | 0.38 | — | 0.56 |
| F | 1.14 | — | 1.65 |
| G | 2.54 BSC | | |
| H | 7.62 | — | 8.26 |
| I | — | — | 10.16 |

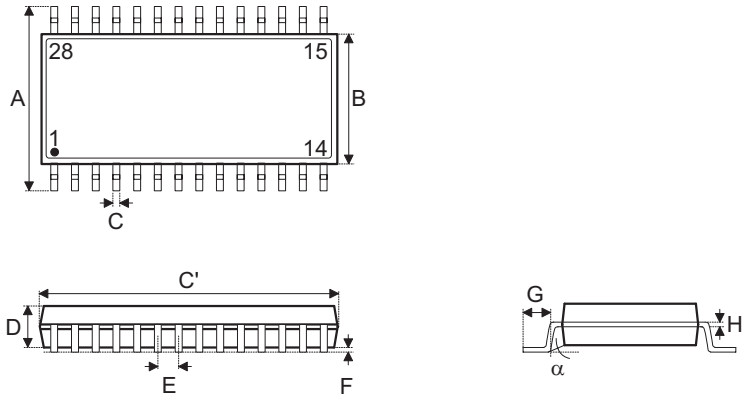
## 28-pin SOP (300mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.406 BSC | |
| B | | 0.295 BSC | |
| C | 0.012 | — | 0.020 |
| C’ | | 0.705 BSC | |
| D | — | — | 0.104 |
| E | | 0.050 BSC | |
| F | 0.004 | — | 0.012 |
| G | 0.016 | — | 0.050 |
| H | 0.008 | — | 0.013 |
| α | 0° | — | 8° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 10.30 BSC | |
| B | | 7.50 BSC | |
| C | 0.31 | — | 0.51 |
| C’ | | 17.90 BSC | |
| D | — | — | 2.65 |
| E | | 1.27 BSC | |
| F | 0.10 | — | 0.30 |
| G | 0.40 | — | 1.27 |
| H | 0.20 | — | 0.33 |
| α | 0° | — | 8° |

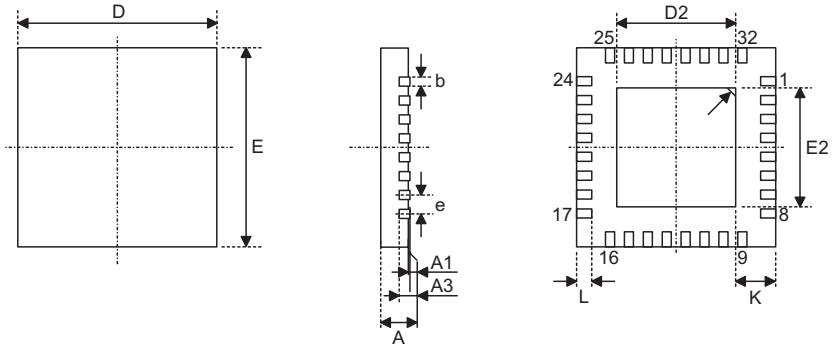
## 28-pin SSOP (150mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.236 BSC | |
| B | | 0.154 BSC | |
| C | 0.008 | — | 0.012 |
| C’ | | 0.390 BSC | |
| D | — | — | 0.069 |
| E | | 0.025 BSC | |
| F | 0.004 | — | 0.010 |
| G | 0.016 | — | 0.050 |
| H | 0.004 | — | 0.010 |
| α | 0° | — | 8° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 6.00 BSC | |
| B | | 3.90 BSC | |
| C | 0.20 | — | 0.30 |
| C’ | | 9.90 BSC | |
| D | — | — | 1.75 |
| E | | 0.635 BSC | |
| F | 0.10 | — | 0.25 |
| G | 0.41 | — | 1.27 |
| H | 0.10 | — | 0.25 |
| α | 0° | — | 8° |

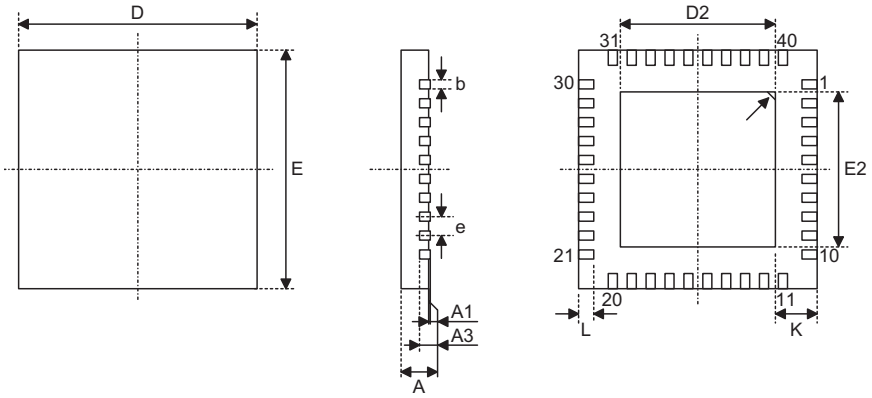
## SAW Type 32-pin QFN (5mm×5mm×0.75mm) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.028 | 0.030 | 0.031 |
| A1 | 0.000 | 0.001 | 0.002 |
| A3 | 0.008 REF | | |
| b | 0.007 | 0.010 | 0.012 |
| D | 0.197 BSC | | |
| E | 0.197 BSC | | |
| e | 0.020 BSC | | |
| D2 | 0.122 | — | 0.142 |
| E2 | 0.122 | — | 0.142 |
| L | 0.014 | 0.016 | 0.018 |
| K | 0.008 | — | — |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.70 | 0.75 | 0.80 |
| A1 | 0.00 | 0.02 | 0.05 |
| A3 | 0.203 REF | | |
| b | 0.18 | 0.25 | 0.30 |
| D | 5.00 BSC | | |
| E | 5.00 BSC | | |
| e | 0.50 BSC | | |
| D2 | 3.10 | — | 3.60 |
| E2 | 3.10 | — | 3.60 |
| L | 0.35 | 0.40 | 0.45 |
| K | 0.20 | — | — |

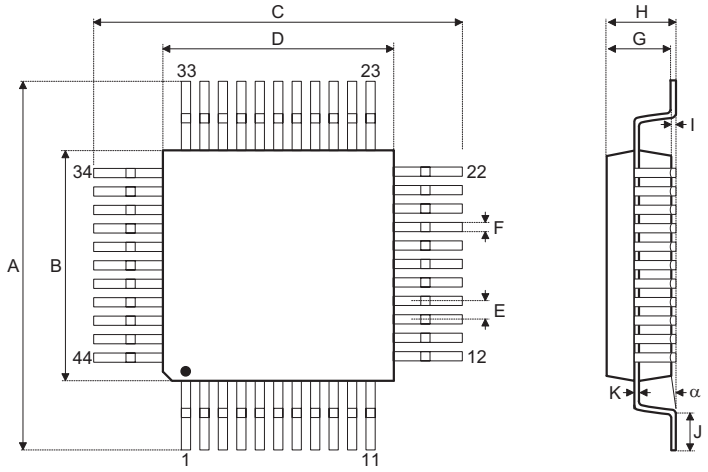
## SAW Type 40-pin QFN (6mm×6mm×0.75mm) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.028 | 0.030 | 0.031 |
| A1 | 0.000 | 0.001 | 0.002 |
| A3 | | 0.008 REF | |
| b | 0.007 | 0.010 | 0.012 |
| D | | 0.236 BSC | |
| E | | 0.236 BSC | |
| e | | 0.020 BSC | |
| D2 | 0.173 | — | 0.181 |
| E2 | 0.173 | — | 0.181 |
| L | 0.014 | 0.016 | 0.018 |
| K | 0.008 | — | — |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.70 | 0.75 | 0.80 |
| A1 | 0.00 | 0.02 | 0.05 |
| A3 | | 0.203 REF | |
| b | 0.18 | 0.25 | 0.30 |
| D | | 6.00 BSC | |
| E | | 6.00 BSC | |
| e | | 0.50 BSC | |
| D2 | 4.40 | — | 4.60 |
| E2 | 4.40 | — | 4.60 |
| L | 0.35 | 0.40 | 0.45 |
| K | 0.20 | — | — |

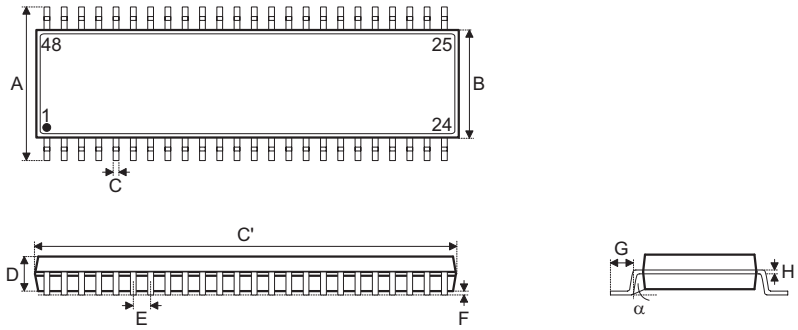
## 44-pin LQFP (10mm×10mm) (FP 2.0mm) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 0.472 BSC | |
| B | | 0.394 BSC | |
| C | | 0.472 BSC | |
| D | | 0.394 BSC | |
| E | | 0.032 BSC | |
| F | 0.012 | 0.015 | 0.018 |
| G | 0.053 | 0.055 | 0.057 |
| H | — | — | 0.063 |
| I | 0.002 | — | 0.006 |
| J | 0.018 | 0.024 | 0.030 |
| K | 0.004 | — | 0.008 |
| α | 0° | — | 7° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | | 12.00 BSC | |
| B | | 10.00 BSC | |
| C | | 12.00 BSC | |
| D | | 10.00 BSC | |
| E | | 0.80 BSC | |
| F | 0.30 | 0.37 | 0.45 |
| G | 1.35 | 1.40 | 1.45 |
| H | — | — | 1.60 |
| I | 0.05 | — | 0.15 |
| J | 0.45 | 0.60 | 0.75 |
| K | 0.09 | — | 0.20 |
| α | 0° | — | 7° |

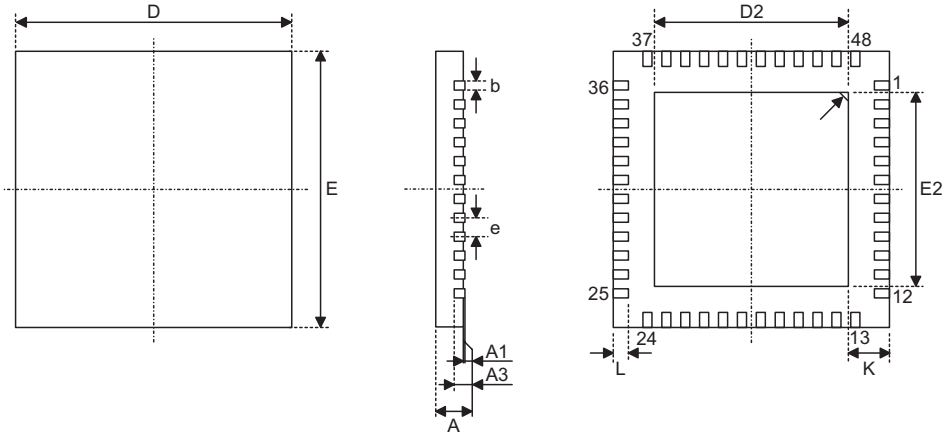
## 48-pin SSOP (300mil) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.395 | — | 0.420 |
| B | 0.291 | 0.295 | 0.299 |
| C | 0.008 | — | 0.014 |
| C’ | 0.620 | 0.625 | 0.630 |
| D | 0.095 | 0.102 | 0.110 |
| E | | 0.025 BSC | |
| F | 0.008 | 0.012 | 0.016 |
| G | 0.020 | — | 0.040 |
| H | 0.005 | — | 0.010 |
| α | 0° | — | 8° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 10.03 | — | 10.67 |
| B | 7.39 | 7.49 | 7.59 |
| C | 0.20 | — | 0.34 |
| C’ | 15.75 | 15.88 | 16.00 |
| D | 2.41 | 2.59 | 2.79 |
| E | | 0.635 BSC | |
| F | 0.20 | 0.30 | 0.41 |
| G | 0.51 | — | 1.02 |
| H | 0.13 | — | 0.25 |
| α | 0° | — | 8° |

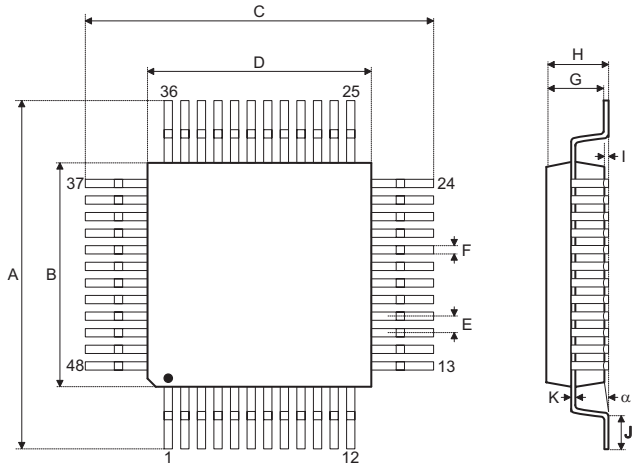
## SAW Type 48-pin QFN (7mm×7mm×0.85mm) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.031 | 0.033 | 0.035 |
| A1 | 0.000 | 0.001 | 0.002 |
| A3 | | 0.008 REF | |
| b | 0.008 | 0.010 | 0.012 |
| D | | 0.276 BSC | |
| E | | 0.276 BSC | |
| e | | 0.020 BSC | |
| D2 | 0.203 | — | 0.211 |
| E2 | 0.203 | — | 0.211 |
| L | 0.014 | 0.016 | 0.018 |
| K | — | — | — |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 正常 | 最大 |
| A | 0.80 | 0.85 | 0.90 |
| A1 | 0.00 | 0.04 | 0.05 |
| A3 | | 0.203 REF | |
| b | 0.20 | 0.25 | 0.30 |
| D | | 7.00 BSC | |
| E | | 7.00 BSC | |
| e | | 0.50 BSC | |
| D2 | 5.15 | — | 5.35 |
| E2 | 5.15 | — | 5.35 |
| L | 0.35 | 0.40 | 0.45 |
| K | — | — | — |

## 48-pin LQFP (7mm×7mm) 外形尺寸




| 符号 | 尺寸 ( 单位：inch) | | |
|---|---|---|---|
| | 最小 | 一般 | 最大 |
| A | | 0.354 BSC | |
| B | | 0.276 BSC | |
| C | | 0.354 BSC | |
| D | | 0.276 BSC | |
| E | | 0.020 BSC | |
| F | 0.007 | 0.009 | 0.011 |
| G | 0.053 | 0.055 | 0.057 |
| H | — | — | 0.063 |
| I | 0.002 | — | 0.006 |
| J | 0.018 | 0.024 | 0.030 |
| K | 0.004 | — | 0.008 |
| α | 0° | — | 7° |

| 符号 | 尺寸 ( 单位：mm) | | |
|---|---|---|---|
| | 最小 | 一般 | 最大 |
| A | | 9.00 BSC | |
| B | | 7.00 BSC | |
| C | | 9.00 BSC | |
| D | | 7.00 BSC | |
| E | | 0.50 BSC | |
| F | 0.17 | 0.22 | 0.27 |
| G | 1.35 | 1.40 | 1.45 |
| H | — | — | 1.60 |
| I | 0.05 | — | 0.15 |
| J | 0.45 | 0.60 | 0.75 |
| K | 0.09 | — | 0.20 |
| α | 0° | — | 7° |

Copyright© 2024 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

本文件出版时 HOLTEK 已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。HOLTEK 不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。HOLTEK 就文中提到的信息及该信息之应用，不承担任何法律责任。此外，HOLTEK 并不推荐将 HOLTEK 的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。HOLTEK 特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用 HOLTEK 产品的风险完全由买方承担，如因该等使用导致 HOLTEK 遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使 HOLTEK 免受损害。HOLTEK ( 及其授权方，如适用 ) 拥有本文件所提供信息 ( 包括但不限于内容、数据、示例、材料、图形、商标 ) 的知识产权，且该信息受著作权法和其他知识产权法的保护。HOLTEK 在此并未明示或暗示授予任何知识产权。HOLTEK 拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。