



内置 LCD & EEPROM 低功耗 Flash 单片机

HT67F370/HT69F340

HT69F350/HT69F360

版本: V1.62 日期: 2025-02-26

www.holtek.com

目录

特性	7
CPU 特性	7
周边特性	7
开发工具	8
概述	8
选型表	8
方框图	9
引脚图	10
引脚说明	16
极限参数	38
直流电气特性	38
交流电气特性	41
A/D 转换器电气特性 – HT67F370	42
LVD & LVR 电气特性	44
内部参考电压电气特性 – HT67F370	45
LCD 直流电气特性	45
上电复位电气特性	46
系统结构	47
时序和流水线结构	47
程序计数器	48
堆栈	48
算术逻辑单元 – ALU	49
Flash 程序存储器	50
结构	50
特殊向量	50
查表	50
查表范例	51
在线烧录 – ICP	52
片上调试 – OCDS	53
在线应用烧录 – IAP	53
数据存储器	62
结构	62
数据存储器寻址	63
通用数据存储器	63
特殊功能数据存储器	63
LCD 显示数据存储器	64
特殊功能寄存器描述	65
间接寻址寄存器 – IAR0, IAR1, IAR2	65
存储器指针 – MP0, MP1L, MP1H, MP2L, MP2H	65

程序存储区指针 – PBP	66
累加器 – ACC	67
程序计数器低字节寄存器 – PCL	67
查表寄存器 – TBLP, TBHP, TBLH	67
状态寄存器 – STATUS	68
EEPROM 数据存储器	69
EEPROM 数据存储器结构	69
EEPROM 寄存器	69
从 EEPROM 中读取数据	71
写数据到 EEPROM 中	71
写保护	71
EEPROM 中断	71
编程注意事项	72
振荡器	73
振荡器概述	73
系统时钟配置	73
外部晶体 / 陶瓷振荡器 – HXT	74
内部 RC 振荡器 – HIRC	74
外部 32.768kHz 晶体振荡器 – LXT	74
内部 32kHz 振荡器 – LIRC	76
辅助振荡器	76
工作模式和系统时钟	76
系统时钟	76
系统工作模式	77
控制寄存器	78
工作模式转换	80
待机电流的注意事项	83
唤醒	83
看门狗定时器	84
看门狗定时器时钟源	84
看门狗定时器控制寄存器	84
看门狗定时器操作	85
复位和初始化	86
复位功能	86
复位初始状态	89
输入 / 输出端口	102
上拉电阻	105
PA 口唤醒	105
输入 / 输出端口控制寄存器	105
引脚共用功能	105
输入 / 输出引脚结构	123
编程注意事项	123
定时器模块 – TM	124
简介	124

TM 操作	124
TM 时钟源	124
TM 中断	125
TM 外部引脚	125
编程注意事项	125
简易型 TM – CTM.....	127
简易型 TM 操作	127
简易型 TM 寄存器介绍	127
简易型 TM 工作模式	131
标准型 TM – STM	137
标准型 TM 操作	137
标准型 TM 寄存器介绍	137
标准型 TM 工作模式	141
周期型 TM – PTM.....	151
周期型 TM 操作	151
周期型 TM 寄存器介绍	151
周期型 TM 工作模式	155
A/D 转换器 – HT67F370	164
A/D 转换器简介	164
A/D 转换器寄存器介绍	165
A/D 转换器操作	167
A/D 转换器参考电压	168
A/D 转换器输入信号	169
A/D 转换率和时序图	169
A/D 转换步骤概述	170
编程注意事项	171
A/D 转换功能	171
A/D 程序范例	171
串行接口模块 – SIM	173
SPI 接口	173
I ² C 接口	179
UART 接口	187
UART 外部接口.....	187
UART 数据传输方案.....	188
UART 状态和控制寄存器.....	188
波特率发生器	192
UART 设置与控制.....	194
UART 发送器.....	195
UART 接收器.....	196
接收错误处理	197
UART 中断结构.....	198
UART 暂停和唤醒.....	199
中断	200
中断寄存器	200

中断操作	207
外部中断	210
时基中断	210
多功能中断	211
EEPROM 中断	211
LVD 中断	211
TM 中断	212
串行接口模块中断	212
UART 中断	212
A/D 转换器中断	212
中断唤醒功能	212
编程注意事项	213
低电压检测 – LVD	213
LVD 寄存器	213
LVD 操作	214
LCD 驱动	215
LCD 显示存储器	215
LCD 时钟源	216
LCD 寄存器	216
LCD 复位功能	218
LCD 驱动输出	218
LCD 电压源偏压	218
LCD 波形时序图	221
编程注意事项	226
配置选项	226
应用电路	227
指令集	230
简介	230
指令周期	230
数据的传送	230
算术运算	230
逻辑和移位运算	230
分支和控制转换	231
位运算	231
查表运算	231
其它运算	231
指令集概要	232
惯例	232
扩展指令集	235
指令定义	237
扩展指令定义	249
封装信息	259
48-pin LQFP (7mm × 7mm) 外形尺寸	260
64-pin LQFP (7mm × 7mm) 外形尺寸	261

80-pin LQFP (10mm × 10mm) 外形尺寸	262
--------------------------------------	-----

特性

CPU 特性

- 工作电压：
 - ◆ $f_{SYS}=4\text{MHz}$: 1.8V~5.5V
 - ◆ $f_{SYS}=8\text{MHz}$ (HT69F340/350/360): 2.0V~5.5V
 - ◆ $f_{SYS}=8\text{MHz}$ (HT67F370): 1.8V~5.5V
 - ◆ $f_{SYS}=12\text{MHz}$: 2.7V~5.5V
 - ◆ $f_{SYS}=16\text{MHz}$ (HT69F340/350/360): 4.5V~5.5V
 - ◆ $f_{SYS}=16\text{MHz}$ (HT67F370): 3.3V~5.5V
- $V_{DD}=5\text{V}$, 系统时钟为 16MHz 时, 指令周期为 $0.25\mu\text{s}$
- 提供暂停和唤醒功能, 以降低功耗
- 振荡器类型:
 - ◆ 外部高速晶振 – HXT
 - ◆ 内部高速 4/8/12MHz RC – HIRC
 - ◆ 外部低速 32.768kHz 晶振 – LXT
 - ◆ 内部低速 32kHz RC – LIRC
- 内部集成振荡器, 无需外接元件
- 多种工作模式: 正常、低速、空闲和休眠
- 所有指令都可在 1~3 个指令周期内完成
- 查表指令
- 115 条指令
- 8 层堆栈
- 位操作指令

周边特性

- Flash 程序存储器: $4\text{K}\times 16 \sim 32\text{K}\times 16$
- RAM 数据存储器: $256\times 8 \sim 2048\times 8$
- True EEPROM 存储器: $64\times 8 \sim 256\times 8$
- 提供在线应用烧录功能 – IAP
- 看门狗定时器功能
- 多达 63 个双向 I/O 口
- LCD 驱动器功能
- 两个与 I/O 口共用的外部中断引脚
- 多个定时器模块用于时间测量、输入捕捉、比较匹配输出、PWM 输出及单脉冲输出
- 双时基功能, 可提供固定时间的中断信号
- 串行接口模块 – SIM, 用于 SPI 或 I²C 通信
- 1 个全双工异步通信接口 – UART (存在于 HT69F360 和 HT67F370)
- 12 个外部通道 12-bit 分辨精度的 A/D 转换器, 具有内部参考电压 V_{VR}
- 低电压复位功能

- 低电压检测功能
- Flash 程序存储器烧录可达 10,000 次
- Flash 程序存储器数据可保存 10 年以上
- True EEPROM 数据存储器烧录可达 100,000 次
- True EEPROM 数据存储器数据可保存 10 年以上
- 封装类型：48/64/80-pin LQFP

开发工具

为加快产品开发并简化单片机参数设置，Holtek 提供相关开发工具，用户可通过以下链接下载：

https://www.holtek.com.cn/page/tool-detail/dev_plat/ha/IR_Remote_Controller_Workshop

概述

本系列单片机是一款 A/D 型具有 8 位高性能精简指令集的 Flash 单片机，专为带有 LCD 接口的应用而设计。

在存储器特性方面，Flash 存储器可多次编程的特性给用户提供了较大的方便。此外还包含了一个 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 True EEPROM 存储器。此外通过使用 IAP 功能，便于用户直接将测量的数据存储至程序存储器中或进行应用程序更新。

在模拟特性方面，该系列单片机包含一个多通道 12-bit A/D 转换器。还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、捕捉输入、比较匹配输出及 PWM 产生功能。内建完整的 SPI、I²C 和 UART 接口功能，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。

此系列单片机提供了内部和外部，高速和低速振荡器功能选项，内部振荡器完全内建，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。

包含 I/O 使用灵活、时基功能等其它特性，使该系列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

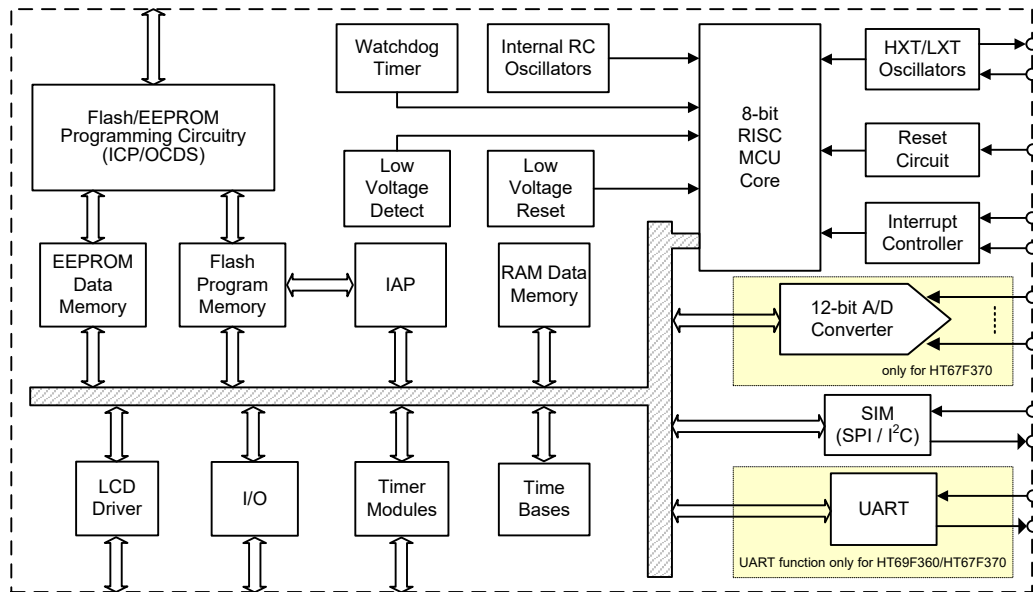
对此系列的单片机而言，大多数的特性参数都是一样的。主要差异在于程序存储器的容量、数据存储器容量、EEPROM 容量、I/O 数、TM 特性、A/D 转换器、LCD 显示和封装类型。下表列出了各单片机的主要特性。

型号	V _{DD}	程序存储器	数据存储器	数据 EEPROM	I/O	外部中断	堆栈
HT69F340	1.8V~5.5V	4K×16	256×8	64×8	39	2	8
HT69F350	1.8V~5.5V	8K×16	512×8	64×8	55	2	8
HT69F360	1.8V~5.5V	16K×16	1024×8	128×8	63	2	8
HT67F370	1.8V~5.5V	32K×16	2048×8	256×8	63	2	8

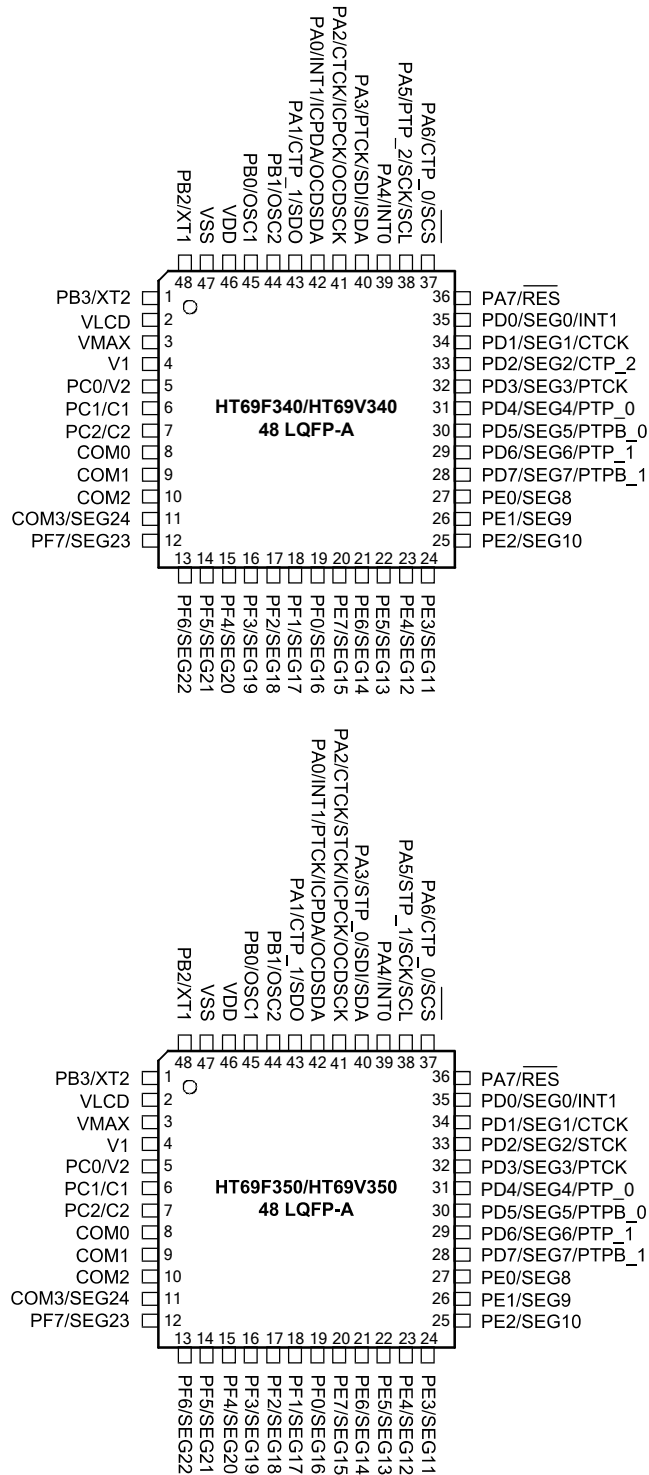
型号	TM 模块	接口 (SPI/I ² C)	UART	A/D 转换器	时基	LCD 驱动	封装
HT69F340	10-bit CTM × 1 10-bit PTM × 1	√	—	—	2	24×4/25×3	48LQFP
HT69F350	10-bit CTM × 1 16-bit STM × 1 10-bit PTM × 1	√	—	—	2	36×4/37×3	48/64LQFP
HT69F360	10-bit CTM × 1 16-bit STM × 1 10-bit PTM × 2	√	√	—	2	48×4/49×3	64/80LQFP
HT67F370	10-bit CTM × 1 16-bit STM × 1 10-bit PTM × 2	√	√	12-bit×12	2	48×4/49×3	64/80LQFP

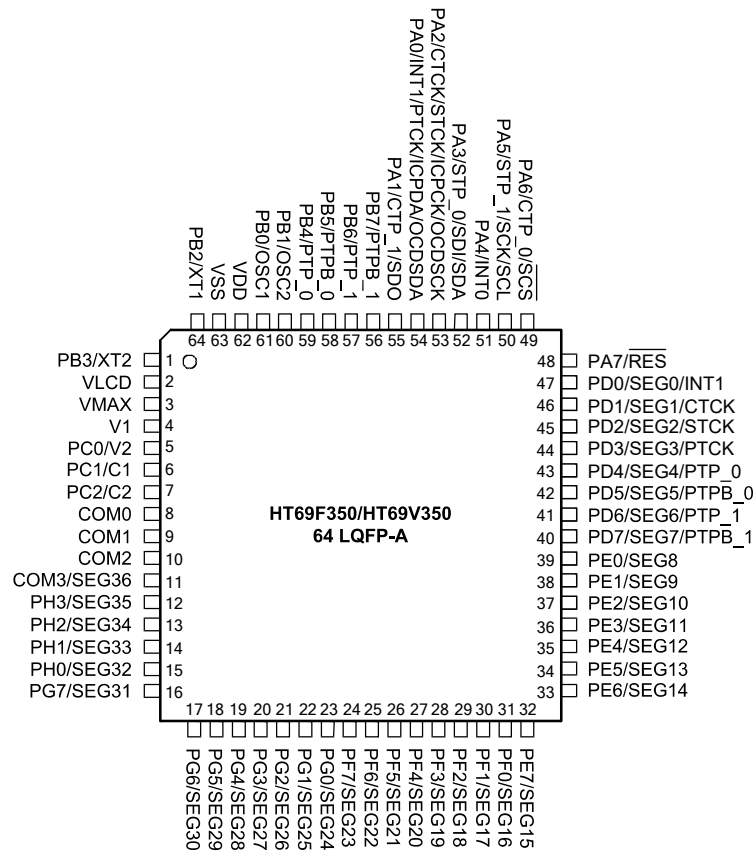
注：由于单片机存在多种封装形式，选型表反映最多引脚封装的情况。

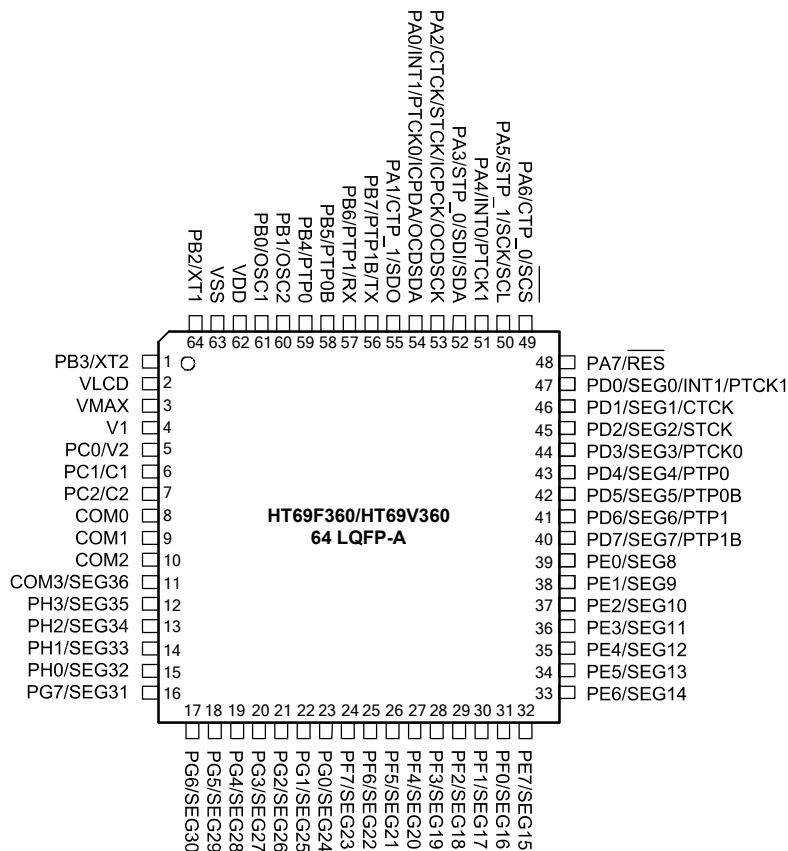
方框图

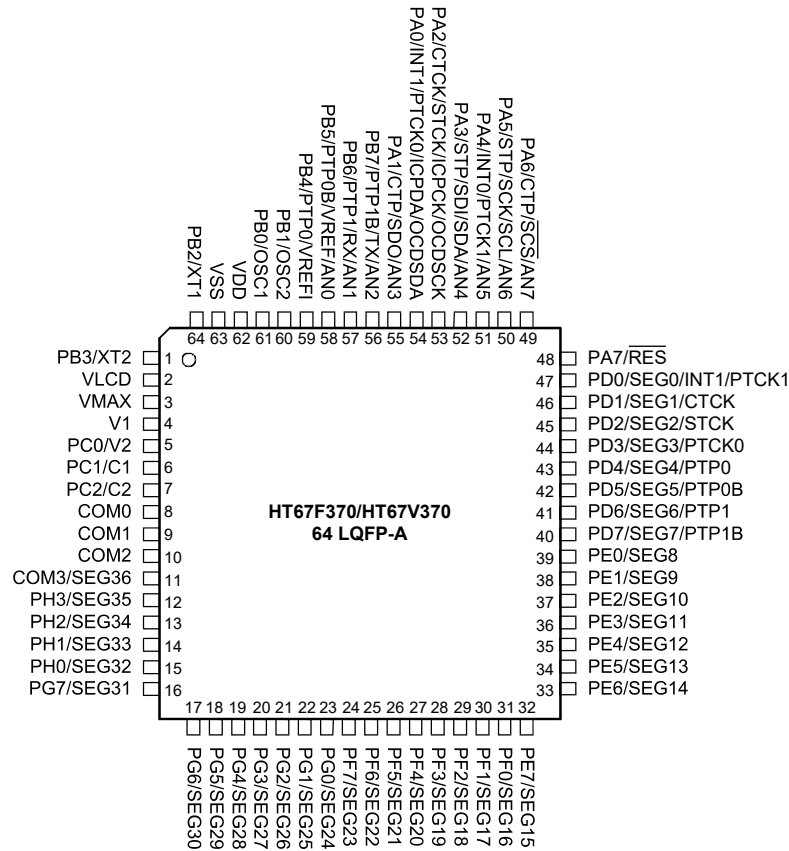


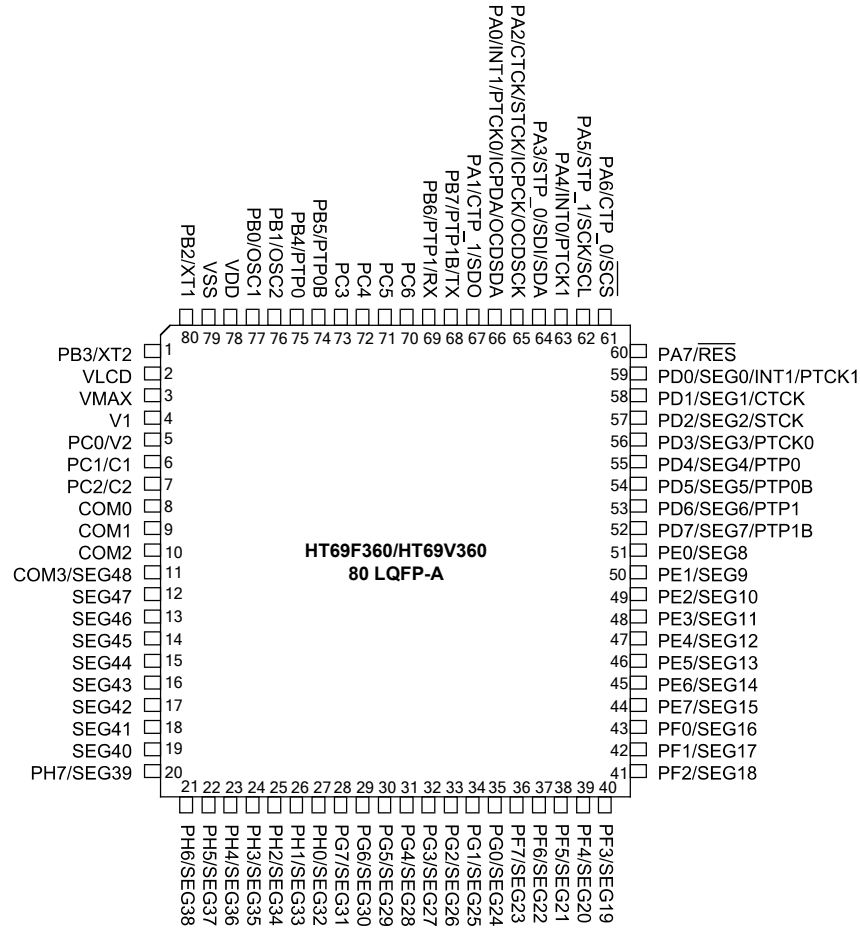
引脚图

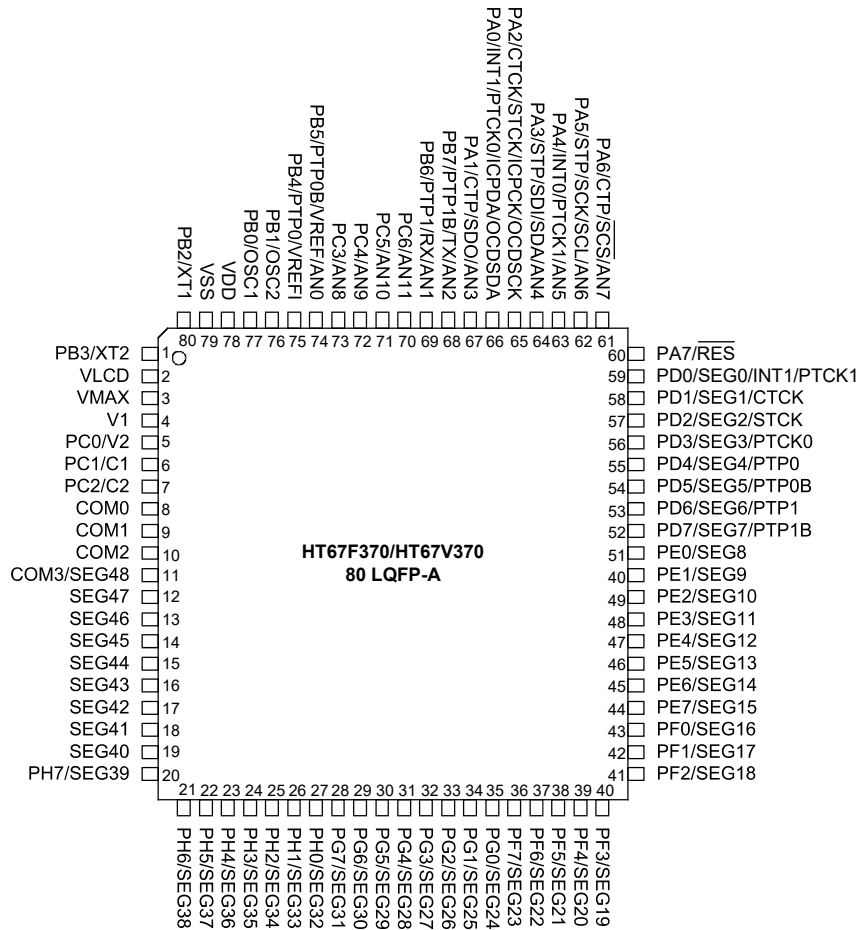












- 注：1. 若共用脚同时有多种输出，引脚共用功能可由相关的软件控制位决定。
2. OCSDA 和 OCDSCK 引脚专门用于 OCDS 功能引脚，适用于此系列单片机的 EV 芯片。EV 芯片支持“片上调试”功能，在开发过程中，通过 OCSDA 和 OCDSCK 连接至 Holtek HT-IDE 开发工具进行调试。
3. 在较小封装中可能含有未引出的引脚，需合理设置其状态以避免输入浮空造成额外耗电，详见“待机电流注意事项”和“输入 / 输出端口”章节。

引脚说明

除了电源引脚和一些相关的转换器控制位外，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 PA0、PA1 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，如定时器模块引脚等。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。由于该系列单片机存在多种封装类型，该表格反映的是较大封装的情况。

HT69F340

引脚名称	功能	OPT	I/T	O/T	说明
PA0/INT1/ICPDA/ OCSDA	PA0	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT1	INTEG INTC0 SFS	ST	—	外部中断 1
	ICPDA	—	ST	CMOS	ICP 地址 / 数据
	OCSDA	—	ST	CMOS	OCDS 地址 / 数据，仅用于 EV 芯片
PA1/CTP_1/SDO	PA1	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_1	PAFS	—	CMOS	CTM 输出
	SDO	PAFS	—	CMOS	SPI 数据输出
PA2/CTCK/ ICPCK/OCDSCK	PA2	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTCK	SFS	ST	—	CTM 时钟输入
	ICPCK	—	ST	—	ICP 时钟引脚
	OCDSCK	—	ST	—	OCDS 时钟引脚，仅用于 EV 芯片
PA3/PTCK/SDI/ SDA	PA3	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTCK	SFS	ST	—	PTM 时钟输入
	SDI	PAFS	ST	—	SPI 数据输入
	SDA	PAFS	ST	NMOS	I ² C 数据线
PA4/INT0	PA4	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	INTEG INTC0	ST	—	外部中断 0
PA5/PTP_2/SCK/ SCL	PA5	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTP_2	PAFS	—	CMOS	PTM 输出
	SCK	PAFS	ST	CMOS	SPI 串行时钟
	SCL	PAFS	ST	NMOS	I ² C 时钟

引脚名称	功能	OPT	I/T	O/T	说明
PA6/CTP_0/ $\overline{\text{SCS}}$	PA6	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_0	PAFS	—	CMOS	CTM 输出
	$\overline{\text{SCS}}$	PAFS	ST	CMOS	SPI 从机选择
PA7/ $\overline{\text{RES}}$	PA7	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{RES}}$	CO	ST	—	外部复位引脚
PB0/OSC1	PB0	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC1	SCC HXTC PBFS	HXT	—	HXT 振荡器引脚
PB1/OSC2	PB1	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC2	SCC HXTC PBFS	—	HXT	HXT 振荡器引脚
PB2/XT1	PB2	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT1	SCC LXTC PBFS	LXT	—	LXT 振荡器引脚
PB3/XT2	PB3	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT2	SCC LXTC PBFS	—	LXT	LXT 振荡器引脚
PC0/V2	PC0	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	V2	—	—	AO	LCD 电压泵
PC1/C1	PC1	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C1	—	—	AO	LCD 电压泵
PC2/C2	PC2	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C2	—	—	AO	LCD 电压泵
PD0/SEG0/INT1	PD0	PDPU PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG0	PDFS	—	AO	LCD SEG 驱动输出
	INT1	SFS	ST	—	外部中断 1
PD1/SEG1/CTCK	PD1	PDPU PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG1	PDFS	—	AO	LCD SEG 驱动输出
	CTCK	SFS	ST	—	CTM 时钟输入

引脚名称	功能	OPT	I/T	O/T	说明
PD2/SEG2/CTP_2	PD2	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG2	PDFSSFS	—	AO	LCD SEG 驱动输出
	CTP_2	PDFSSFS	—	CMOS	CTM 输出
PD3/SEG3/PTCK	PD3	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG3	PDFSSFS	—	AO	LCD SEG 驱动输出
	PTCK	SFS	ST	—	PTM 时钟输入
PD4/SEG4/PTP_0	PD4	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG4	PDFSSFS	—	AO	LCD SEG 驱动输出
	PTP_0	PDFSSFS	—	CMOS	PTM 输出
PD5/SEG5/PTPB_0	PD5	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG5	PDFSSFS	—	AO	LCD SEG 驱动输出
	PTPB_0	PDFSSFS	—	CMOS	PTM 反相输出
PD6/SEG6/PTP_1	PD6	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG6	PDFSSFS	—	AO	LCD SEG 驱动输出
	PTP_1	PDFSSFS	—	CMOS	PTM 输出
PD7/SEG7/PTPB_1	PD7	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG7	PDFSSFS	—	AO	LCD SEG 驱动输出
	PTPB_1	PDFSSFS	—	CMOS	PTM 反相输出
PE0/SEG8	PE0	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG8	PEFS	—	AO	LCD SEG 驱动输出
PE1/SEG9	PE1	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG9	PEFS	—	AO	LCD SEG 驱动输出
PE2/SEG10	PE2	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG10	PEFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PE3/SEG11	PE3	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG11	PEFS	—	AO	LCD SEG 驱动输出
PE4/SEG12	PE4	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG12	PEFS	—	AO	LCD SEG 驱动输出
PE5/SEG13	PE5	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG13	PEFS	—	AO	LCD SEG 驱动输出
PE6/SEG14	PE6	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG14	PEFS	—	AO	LCD SEG 驱动输出
PE7/SEG15	PE7	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG15	PEFS	—	AO	LCD SEG 驱动输出
PF0/SEG16	PF0	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG16	PFFS	—	AO	LCD SEG 驱动输出
PF1/SEG17	PF1	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG17	PFFS	—	AO	LCD SEG 驱动输出
PF2/SEG18	PF2	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG18	PFFS	—	AO	LCD SEG 驱动输出
PF3/SEG19	PF3	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG19	PFFS	—	AO	LCD SEG 驱动输出
PF4/SEG20	PF4	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG20	PFFS	—	AO	LCD SEG 驱动输出
PF5/SEG21	PF5	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG21	PFFS	—	AO	LCD SEG 驱动输出
PF6/SEG22	PF6	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG22	PFFS	—	AO	LCD SEG 驱动输出
PF7/SEG23	PF7	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG23	PFFS	—	AO	LCD SEG 驱动输出
COM0~COM2	COMn	—	—	AO	LCD COM 驱动输出
COM3/SEG24	COM3	—	—	AO	LCD COM 驱动输出
	SEG24	—	—	AO	LCD SEG 驱动输出
V1	V1	—	—	AO	LCD 电压泵
VLCD	VLCD	—	PWR	—	LCD 电源电压

引脚名称	功能	OPT	I/T	O/T	说明
VMAX	VMAX	—	PWR	—	IC 最大电压，连接到 VDD，VLCD 或 V1
VDD	VDD	—	PWR	—	电源电压
VSS	VSS	—	PWR	—	地

注：I/T：输入类型； O/T：输出类型
 OPT：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； AO：模拟输出；
 HXT：高速晶体振荡器； LXT：低速晶体振荡器。

HT69F350

引脚名称	功能	OPT	I/T	O/T	说明
PA0/INT1/PTCK/ ICPDA/OCSDA	PA0	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT1	INTEG INTC0 SFSR	ST	—	外部中断 1
	PTCK	SFSR	ST	—	PTM 时钟输入
	ICPDA	—	ST	CMOS	ICP 地址 / 数据
	OCSDA	—	ST	CMOS	OCDS 地址 / 数据，仅用于 EV 芯片
PA1/CTP_1/SDO	PA1	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_1	PAFS	—	CMOS	CTM 输出
	SDO	PAFS	—	CMOS	SPI 数据输出
PA2/CTCK/STCK/ ICPCK/OCDSCK	PA2	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTCK	SFSR	ST	—	CTM 时钟输入
	STCK	SFSR	ST	—	STM 时钟输入
	ICPCK	—	ST	—	ICP 时钟引脚
PA3/STP_0/SDI/ SDA	OCDSCK	—	ST	—	OCDS 时钟引脚，仅用于 EV 芯片
	PA3	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP_0	PAFS	—	CMOS	STM 输出
	SDI	PAFS	ST	—	SPI 数据输入
PA4/INT0	SDA	PAFS	ST	NMOS	I ² C 数据线
	PA4	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	INTEG INTC0	ST	—	外部中断 0

引脚名称	功能	OPT	I/T	O/T	说明
PA5/STP_1/SCK/ SCL	PA5	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP_1	PAFS	—	CMOS	STM 输出
	SCK	PAFS	ST	CMOS	SPI 串行时钟
	SCL	PAFS	ST	NMOS	I ² C 时钟
PA6/CTP_0/ $\overline{\text{SCS}}$	PA6	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_0	PAFS	—	CMOS	CTM 输出
	$\overline{\text{SCS}}$	PAFS	ST	CMOS	SPI 从机选择
PA7/ $\overline{\text{RES}}$	PA7	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{RES}}$	CO	ST	—	外部复位引脚
PB0/OSC1	PB0	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC1	SCC HXTC PBFS	HXT	—	HXT 振荡器引脚
PB1/OSC2	PB1	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC2	SCC HXTC PBFS	—	HXT	HXT 振荡器引脚
PB2/XT1	PB2	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT1	SCC LXTC PBFS	LXT	—	LXT 振荡器引脚
PB3/XT2	PB3	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT2	SCC LXTC PBFS	—	LXT	LXT 振荡器引脚
PB4/PTP_0	PB4	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP_0	PBFS	—	CMOS	PTM 输出
PB5/PTPB_0	PB5	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTPB_0	PBFS	—	CMOS	PTM 反相输出
PB6/PTP_1	PB6	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP_1	PBFS	—	CMOS	PTM 输出

引脚名称	功能	OPT	I/T	O/T	说明
PB7/PTPB_1	PB7	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTPB_1	PBFS	—	CMOS	PTM 反相输出
PC0/V2	PC0	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	V2	—	—	AO	LCD 电压泵
PC1/C1	PC1	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C1	—	—	AO	LCD 电压泵
PC2/C2	PC2	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C2	—	—	AO	LCD 电压泵
PD0/SEG0/INT1	PD0	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG0	PDFS	—	AO	LCD SEG 驱动输出
	INT1	INTEG INTC0 SFSR	ST	—	外部中断 1
PD1/SEG1/CTCK	PD1	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG1	PDFS	—	AO	LCD SEG 驱动输出
	CTCK	SFSR	ST	—	CTM 时钟输入
PD2/SEG2/STCK	PD2	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG2	PDFS	—	AO	LCD SEG 驱动输出
	STCK	SFSR	ST	—	STM 时钟输入
PD3/SEG3/PTCK	PD3	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG3	PDFS	—	AO	LCD SEG 驱动输出
	PTCK	SFSR	ST	—	PTM 时钟输入
PD4/SEG4/PTP_0	PD4	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG4	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP_0	PDFS SFSR	—	CMOS	PTM 输出
PD5/SEG5/PTPB_0	PD5	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG5	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTPB_0	PDFS SFSR	—	CMOS	PTM 反相输出

引脚名称	功能	OPT	I/T	O/T	说明
PD6/SEG6/PTP_1	PD6	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG6	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP_1	PDFS SFSR	—	CMOS	PTM 输出
PD7/SEG7/PTPB_1	PD7	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG7	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTPB_1	PDFS SFSR	—	CMOS	PTM 反相输出
PE0/SEG8	PE0	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG8	PEFS	—	AO	LCD SEG 驱动输出
PE1/SEG9	PE1	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG9	PEFS	—	AO	LCD SEG 驱动输出
PE2/SEG10	PE2	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG10	PEFS	—	AO	LCD SEG 驱动输出
PE3/SEG11	PE3	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG11	PEFS	—	AO	LCD SEG 驱动输出
PE4/SEG12	PE4	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG12	PEFS	—	AO	LCD SEG 驱动输出
PE5/SEG13	PE5	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG13	PEFS	—	AO	LCD SEG 驱动输出
PE6/SEG14	PE6	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG14	PEFS	—	AO	LCD SEG 驱动输出
PE7/SEG15	PE7	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG15	PEFS	—	AO	LCD SEG 驱动输出
PF0/SEG16	PF0	PFPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG16	PDFS	—	AO	LCD SEG 驱动输出
PF1/SEG17	PF1	PFPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG17	PDFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PF2/SEG18	PF2	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG18	PFFS	—	AO	LCD SEG 驱动输出
PF3/SEG19	PF3	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG19	PFFS	—	AO	LCD SEG 驱动输出
PF4/SEG20	PF4	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG20	PFFS	—	AO	LCD SEG 驱动输出
PF5/SEG21	PF5	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG21	PFFS	—	AO	LCD SEG 驱动输出
PF6/SEG22	PF6	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG22	PFFS	—	AO	LCD SEG 驱动输出
PF7/SEG23	PF7	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG23	PFFS	—	AO	LCD SEG 驱动输出
PG0/SEG24	PG0	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG24	PGFS	—	AO	LCD SEG 驱动输出
PG1/SEG25	PG1	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG25	PGFS	—	AO	LCD SEG 驱动输出
PG2/SEG26	PG2	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG26	PGFS	—	AO	LCD SEG 驱动输出
PG3/SEG27	PG3	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG27	PGFS	—	AO	LCD SEG 驱动输出
PG4/SEG28	PG4	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG28	PGFS	—	AO	LCD SEG 驱动输出
PG5/SEG29	PG5	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG29	PGFS	—	AO	LCD SEG 驱动输出
PG6/SEG30	PG6	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG30	PGFS	—	AO	LCD SEG 驱动输出
PG7/SEG31	PG7	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG31	PGFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PH0/SEG32	PH0	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG32	PHFS	—	AO	LCD SEG 驱动输出
PH1/SEG33	PH1	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG33	PHFS	—	AO	LCD SEG 驱动输出
PH2/SEG34	PH2	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG34	PHFS	—	AO	LCD SEG 驱动输出
PH3/SEG35	PH3	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG35	PHFS	—	AO	LCD SEG 驱动输出
COM3/SEGn	COM3	—	—	AO	LCD COM 驱动输出
	SEGn	—	—	AO	LCD SEG 驱动输出 n=24 (48 脚封装) n=36 (64 脚封装)
COM0~COM2	COMn	—	—	AO	LCD COM 驱动输出
V1	V1	—	—	AO	LCD 电压泵
VMAX	VMAX	—	PWR	—	IC 最大电压，连接到 VDD，VLCD 或 V1
VLCD	VLCD	—	PWR	—	LCD 电源
VDD	VDD	—	PWR	—	电源电压
VSS	VSS	—	PWR	—	地

注：I/T：输入类型； O/T：输出类型
 OPT：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； AO：模拟输出；
 HXT：高速晶体振荡器； LXT：低速晶体振荡器。

HT69F360

引脚名称	功能	OPT	I/T	O/T	说明
PA0/INT1/PTCK0/ICPDA/OCSDA	PA0	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT1	INTEG INTC0 SFSR	ST	—	外部中断 1
	PTCK0	SFSR	ST	—	PTM0 时钟输入
	ICPDA	—	ST	CMOS	ICP 地址 / 数据
	OCSDA	—	ST	CMOS	OCDS 地址 / 数据，仅用于 EV 芯片
PA1/CTP_1/SDO	PA1	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_1	PAFS	—	CMOS	CTM 输出
	SDO	PAFS	—	CMOS	SPI 数据输出
PA2/CTCK/STCK/ICPCK/OCDSCK	PA2	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTCK	SFSR	ST	—	CTM 时钟输入
	STCK	SFSR	ST	—	STM 时钟输入
	ICPCK	—	ST	—	ICP 时钟引脚
PA3/STP_0/SDI/SDA	PA3	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP_0	PAFS	—	CMOS	STM 输出
	SDI	PAFS	ST	—	SPI 数据输入
	SDA	PAFS	ST	NMOS	I ² C 数据线
PA4/INT0/PTCK1	PA4	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	INTEG INTC0	ST	—	外部中断 0
	PTCK1	—	ST	—	PTM1 时钟输入
PA5/STP_1/SCK/SCL	PA5	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP_1	PAFS	—	CMOS	STM 输出
	SCK	PAFS	ST	CMOS	SPI 串行时钟
	SCL	PAFS	ST	NMOS	I ² C 时钟
PA6/CTP_0/ $\overline{\text{SCS}}$	PA6	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP_0	PAFS	—	CMOS	CTM 输出
	$\overline{\text{SCS}}$	PAFS	ST	CMOS	SPI 从机选择
PA7/ $\overline{\text{RES}}$	PA7	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{RES}}$	CO	ST	—	外部复位引脚

引脚名称	功能	OPT	I/T	O/T	说明
PB0/OSC1	PB0	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC1	SCC HXTC PBFS	HXT	—	HXT 振荡器引脚
PB1/OSC2	PB1	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	OSC2	SCC HXTC PBFS	—	HXT	HXT 振荡器引脚
PB2/XT1	PB2	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT1	SCC LXTC PBFS	LXT	—	LXT 振荡器引脚
PB3/XT2	PB3	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	XT2	SCC LXTC PBFS	—	LXT	LXT 振荡器引脚
PB4/PTP0	PB4	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP0	PBFS	—	CMOS	PTM0 输出
PB5/PTP0B	PB5	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP0B	PBFS	—	CMOS	PTM0 反相输出
PB6/PTP1/RX	PB6	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP1	PBFS	—	CMOS	PTM1 输出
	RX	SFSR1	ST	—	UART 数据接收输入
PB7/PTP1B /TX	PB7	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	PTP1B	PBFS	—	CMOS	PTM1 反相输出
	TX	PBFS SFSR1	—	CMOS	UART 数据发送输出
PC0/V2	PC0	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	V2	—	—	AO	LCD 电压泵
PC1/C1	PC1	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C1	—	—	AO	LCD 电压泵
PC2/C2	PC2	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	C2	—	—	AO	LCD 电压泵
PC3~PC6	PC3~PC6	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能

引脚名称	功能	OPT	I/T	O/T	说明
PD0/SEG0/INT1/ PTCK1	PD0	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG0	PDFS	—	AO	LCD SEG 驱动输出
	INT1	INTEG INTC0 SFSR	ST	—	外部中断 1
	PTCK1	SFSR1	ST	—	PTM1 时钟输入
PD1/SEG1/CTCK	PD1	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG1	PDFS	—	AO	LCD SEG 驱动输出
	CTCK	SFSR	ST	—	CTM 时钟输入
PD2/SEG2/STCK	PD2	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG2	PDFS	—	AO	LCD SEG 驱动输出
	STCK	SFSR	ST	—	STM 时钟输入
PD3/SEG3/PTCK0	PD3	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG3	PDFS	—	AO	LCD SEG 驱动输出
	PTCK0	SFSR	ST	—	PTM0 时钟输入
PD4/SEG4/PTP0	PD4	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG4	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP0	PDFS SFSR	—	CMOS	PTM0 输出
PD5/SEG5/PTP0B	PD5	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG5	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP0B	PDFS SFSR	—	CMOS	PTM0 反相输出
PD6/SEG6/PTP1	PD6	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG6	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP1	PDFS SFSR	—	CMOS	PTM1 输出
PD7/SEG7/PTP1B	PD7	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG7	PDFS SFSR	—	AO	LCD SEG 驱动输出
	PTP1B	PDFS SFSR	—	CMOS	PTM1 反相输出
PE0/SEG8	PE0	PEP PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG8	PEFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PE1/SEG9	PE1	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG9	PEFS	—	AO	LCD SEG 驱动输出
PE2/SEG10	PE2	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG10	PEFS	—	AO	LCD SEG 驱动输出
PE3/SEG11	PE3	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG11	PEFS	—	AO	LCD SEG 驱动输出
PE4/SEG12	PE4	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG12	PEFS	—	AO	LCD SEG 驱动输出
PE5/SEG13	PE5	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG13	PEFS	—	AO	LCD SEG 驱动输出
PE6/SEG14	PE6	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG14	PEFS	—	AO	LCD SEG 驱动输出
PE7/SEG15	PE7	PEPU PEFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG15	PEFS	—	AO	LCD SEG 驱动输出
PF0/SEG16	PF0	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG16	PFFS	—	AO	LCD SEG 驱动输出
PF1/SEG17	PF1	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG17	PFFS	—	AO	LCD SEG 驱动输出
PF2/SEG18	PF2	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG18	PFFS	—	AO	LCD SEG 驱动输出
PF3/SEG19	PF3	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG19	PFFS	—	AO	LCD SEG 驱动输出
PF4/SEG20	PF4	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG20	PFFS	—	AO	LCD SEG 驱动输出
PF5/SEG21	PF5	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG21	PFFS	—	AO	LCD SEG 驱动输出
PF6/SEG22	PF6	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG22	PFFS	—	AO	LCD SEG 驱动输出
PF7/SEG23	PF7	PFPU PFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG23	PFFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PG0/SEG24	PG0	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG24	PGFS	—	AO	LCD SEG 驱动输出
PG1/SEG25	PG1	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG25	PGFS	—	AO	LCD SEG 驱动输出
PG2/SEG26	PG2	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG26	PGFS	—	AO	LCD SEG 驱动输出
PG3/SEG27	PG3	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG27	PGFS	—	AO	LCD SEG 驱动输出
PG4/SEG28	PG4	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG28	PGFS	—	AO	LCD SEG 驱动输出
PG5/SEG29	PG5	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG29	PGFS	—	AO	LCD SEG 驱动输出
PG6/SEG30	PG6	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG30	PGFS	—	AO	LCD SEG 驱动输出
PG7/SEG31	PG7	PGPU PGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG31	PGFS	—	AO	LCD SEG 驱动输出
PH0/SEG32	PH0	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG32	PHFS	—	AO	LCD SEG 驱动输出
PH1/SEG33	PH1	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG33	PHFS	—	AO	LCD SEG 驱动输出
PH2/SEG34	PH2	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG34	PHFS	—	AO	LCD SEG 驱动输出
PH3/SEG35	PH3	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG35	PHFS	—	AO	LCD SEG 驱动输出
PH4/SEG36	PH4	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG36	PHFS	—	AO	LCD SEG 驱动输出
PH5/SEG37	PH5	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG37	PHFS	—	AO	LCD SEG 驱动输出
PH6/SEG38	PH6	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG38	PHFS	—	AO	LCD SEG 驱动输出

引脚名称	功能	OPT	I/T	O/T	说明
PH7/SEG39	PH7	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻功能
	SEG39	PHFS	—	AO	LCD SEG 驱动输出
SEG40~SEG47	SEGn	—	—	AO	LCD SEG 驱动输出
COM3/SEGn	COM3	—	—	AO	LCD COM 驱动输出
	SEGn	—	—	AO	LCD SEG 驱动输出 n=36 (64 脚封装) n=48 (80 脚封装)
COM0~COM2	COMn	—	—	AO	LCD COM 驱动输出
V1	V1	—	—	AO	LCD 电压泵
VMAX	VMAX	—	PWR	—	IC 最大电压，连接到 VDD，VLCD 或 V1
VLCD	VLCD	—	PWR	—	LCD 电源
VDD	VDD	—	PWR	—	电源电压
VSS	VSS	—	PWR	—	地

注：I/T：输入类型； O/T：输出类型
 OPT：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； AO：模拟输出；
 HXT：高速晶体振荡器； LXT：低速晶体振荡器。

HT67F370

引脚名称	功能	OPT	I/T	O/T	说明
PA0/INT1/PTCK0/ ICPDA/OCSDA	PA0	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT1	INTEG INTC0 SFSR	ST	—	外部中断输入 1
	PTCK0	SFSR	ST	—	PTM0 时钟输入
	ICPDA	—	ST	CMOS	ICP 地址 / 数据
	OCSDA	—	ST	CMOS	OCDS 地址 / 数据，仅用于 EV 芯片
PA1/CTP/ SDO/AN3	PA1	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP	PAFS	—	CMOS	CTM 输出
	SDO	PAFS	—	CMOS	SPI 串行数据输出
	AN3	PAFS	AN	—	A/D 转换器外部输入 3
PA2/CTCK/STCK/ ICPCK/OCDSCK	PA2	PAWU PAPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTCK	SFSR	ST	—	CTM 时钟输入
	STCK	SFSR	ST	—	STM 时钟输入
	ICPCK	—	ST	—	ICP 时钟引脚
	OCDSCK	—	ST	—	OCDS 时钟，仅用于 EV 芯片

引脚名称	功能	OPT	I/T	O/T	说明
PA3/STP/SDI/ SDA/AN4	PA3	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP	PAFS	ST	CMOS	STM 输入 / 输出
	SDI	PAFS	ST	—	SPI 串行数据输入
	SDA	PAFS	ST	NMOS	I ² C 数据线
	AN4	PAFS	AN	—	A/D 转换器外部输入 4
PA4/INT0/ PTCK1/AN5	PA4	PAWU PAPU SFSR1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	INTEG INTC0 SFSR1	ST	—	外部中断输入 0
	PTCK1	SFSR1	ST	—	PTM1 时钟输入
	AN5	SFSR1	AN	—	A/D 转换器外部输入 5
PA5/STP/SCK/ SCL/AN6	PA5	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP	PAFS	ST	CMOS	STM 输入 / 输出
	SCK	PAFS	ST	CMOS	SPI 串行时钟
	SCL	PAFS	ST	NMOS	I ² C 时钟线
	AN6	PAFS	AN	—	A/D 转换器外部输入 6
PA6/CTP/ SCS/AN7	PA6	PAWU PAPU PAFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP	PAFS	—	CMOS	CTM 输出
	SCS	PAFS	ST	CMOS	SPI 从机选择
	AN7	PAFS	AN	—	A/D 转换器外部输入 7
PA7/ $\overline{\text{RES}}$	PA7	PAWU PAPU CO	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{RES}}$	CO	ST	—	外部复位引脚
PB0/OSC1	PB0	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OSC1	SCC HXTC PBFS	HXT	—	HXT 振荡器引脚
PB1/OSC2	PB1	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OSC2	SCC HXTC PBFS	—	HXT	HXT 振荡器引脚

引脚名称	功能	OPT	I/T	O/T	说明
PB2/XT1	PB2	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	XT1	SCC LXTC PBFS	LXT	—	LXT 振荡器引脚
PB3/XT2	PB3	PBPU PBFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	XT2	SCC LXTC PBFS	—	LXT	LXT 振荡器引脚
PB4/PTP0/VREFI	PB4	PBPU PBFS SFSR1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP0	PBFS SFSR1	ST	CMOS	PTM0 输入 / 输出
	VREFI	PBFS SFSR1	AN	—	A/D 转换器 PGA 输入
PB5/PTP0B/ VREF/AN0	PB5	PBPU PBFS SFSR1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP0B	PBFS SFSR1	ST	CMOS	PTM0 输入 / 反相输出
	VREF	PBFS SFSR1	AN	—	A/D 转换器外部参考电压输入
	AN0	PBFS SFSR1	AN	—	A/D 转换器外部输入 0
PB6/PTP1/RX/ AN1	PB6	PBPU PBFS SFSR1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP1	PBFS SFSR1	ST	CMOS	PTM1 输入 / 输出
	RX	PBFS SFSR1	ST	—	UART 数据接收引脚
	AN1	PBFS SFSR1	AN	—	A/D 转换器外部输入 1
PB7/PTP1B/TX/ AN2	PB7	PBPU PBFS SFSR1	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP1B	PBFS SFSR1	ST	CMOS	PTM1 输入 / 反相输出
	TX	PBFS SFSR1	—	CMOS	UART 数据传送输出
	AN2	PBFS SFSR1	AN	—	A/D 转换器外部输入 2
PC0/V2	PC0	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	V2	—	—	AN	LCD 电压泵

引脚名称	功能	OPT	I/T	O/T	说明
PC1/C1	PC1	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	C1	—	—	AN	LCD 电压泵
PC2/C2	PC2	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	C2	—	—	AN	LCD 电压泵
PC3/AN8	PC3	PCPU PCFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	AN8	PCFS	AN	—	A/D 转换器外部输入 8
PC4/AN9	PC4	PCPU PCFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	AN9	PCFS	AN	—	A/D 转换器外部输入 9
PC5/AN10	PC5	PCPU PCFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	AN10	PCFS	AN	—	A/D 转换器外部输入 10
PC6/AN11	PC6	PCPU PCFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	AN11	PCFS	AN	—	A/D 转换器外部输入 11
PD0/SEG0/ INT1/PTCK1	PD0	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG0	PDFS	—	AN	LCD SEG 输出
	INT1	INTEG INTC0 PDFS SFSR	ST	—	外部中断 1
	PTCK1	PDFS SFSR1	ST	—	PTM1 时钟输入
PD1/SEG1/CTCK	PD1	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG1	PDFS	—	AN	LCD SEG 输出
	CTCK	PDFS SFSR	ST	—	CTM 时钟输入
PD2/SEG2/STCK	PD2	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG2	PDFS	—	AN	LCD SEG 输出
	STCK	PDFS SFSR	ST	—	STM 时钟输入
PD3/SEG3/PTCK0	PD3	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG3	PDFS	—	AN	LCD SEG 输出
	PTCK0	PDFS SFSR	ST	—	PTM0 时钟输入
PD4/SEG4/PTP0	PD4	PDP PDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG4	PDFS SFSR	—	AN	LCD SEG 输出
	PTP0	PDFS SFSR	ST	CMOS	PTM0 输入 / 输出

引脚名称	功能	OPT	I/T	O/T	说明
PD5/SEG5/PTP0B	PD5	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG5	PDFS SFSR	—	AN	LCD SEG 输出
	PTP0B	PDFS SFSR	ST	CMOS	PTM0 输入 / 反相输出
PD6/SEG6/PTP1	PD6	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG6	PDFS SFSR	—	AN	LCD SEG 输出
	PTP1	PDFS SFSR	ST	CMOS	PTM1 输入 / 输出
PD7/SEG7/PTP1B	PD7	PDPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG7	PDFS SFSR	—	AN	LCD SEG 输出
	PTP1B	PDFS SFSR	ST	CMOS	PTM1 输入 / 反相输出
PE0/SEG8	PE0	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG8	PEFS	—	AN	LCD SEG 输出
PE1/SEG9	PE1	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG9	PEFS	—	AN	LCD SEG 输出
PE2/SEG10	PE2	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG10	PEFS	—	AN	LCD SEG 输出
PE3/SEG11	PE3	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG11	PEFS	—	AN	LCD SEG 输出
PE4/SEG12	PE4	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG12	PEFS	—	AN	LCD SEG 输出
PE5/SEG13	PE5	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG13	PEFS	—	AN	LCD SEG 输出
PE6/SEG14	PE6	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG14	PEFS	—	AN	LCD SEG 输出
PE7/SEG15	PE7	PEPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG15	PEFS	—	AN	LCD SEG 输出
PF0/SEG16	PF0	PFPUPDFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG16	PFFS	—	AN	LCD SEG 输出

引脚名称	功能	OPT	I/T	O/T	说明
PF1/SEG17	PF1	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG17	PFFS	—	AN	LCD SEG 输出
PF2/SEG18	PF2	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG18	PFFS	—	AN	LCD SEG 输出
PF3/SEG19	PF3	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG19	PFFS	—	AN	LCD SEG 输出
PF4/SEG20	PF4	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG20	PFFS	—	AN	LCD SEG 输出
PF5/SEG21	PF5	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG21	PFFS	—	AN	LCD SEG 输出
PF6/SEG22	PF6	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG22	PFFS	—	AN	LCD SEG 输出
PF7/SEG23	PF7	PFPUPFFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG23	PFFS	—	AN	LCD SEG 输出
PG0/SEG24	PG0	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG24	PGFS	—	AN	LCD SEG 输出
PG1/SEG25	PG1	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG25	PGFS	—	AN	LCD SEG 输出
PG2/SEG26	PG2	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG26	PGFS	—	AN	LCD SEG 输出
PG3/SEG27	PG3	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG27	PGFS	—	AN	LCD SEG 输出
PG4/SEG28	PG4	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG28	PGFS	—	AN	LCD SEG 输出
PG5/SEG29	PG5	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG29	PGFS	—	AN	LCD SEG 输出
PG6/SEG30	PG6	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG30	PGFS	—	AN	LCD SEG 输出
PG7/SEG31	PG7	PGPUPGFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG31	PGFS	—	AN	LCD SEG 输出

引脚名称	功能	OPT	I/T	O/T	说明
PH0/SEG32	PH0	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG32	PHFS	—	AN	LCD SEG 输出
PH1/SEG33	PH1	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG33	PHFS	—	AN	LCD SEG 输出
PH2/SEG34	PH2	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG34	PHFS	—	AN	LCD SEG 输出
PH3/SEG35	PH3	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG35	PHFS	—	AN	LCD SEG 输出
PH4/SEG36	PH4	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG36	PHFS	—	AN	LCD SEG 输出
PH5/SEG37	PH5	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG37	PHFS	—	AN	LCD SEG 输出
PH6/SEG38	PH6	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG38	PHFS	—	AN	LCD SEG 输出
PH7/SEG39	PH7	PHPU PHFS	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	SEG39	PHFS	—	AN	LCD SEG 输出
SEG40~SEG47	SEGn	—	—	AN	LCD SEG 输出
COM0~COM2	COMn	—	—	AN	LCD COM 输出
COM3/SEGn	COM3	—	—	AN	LCD COM 输出
	SEGn	—	—	AN	LCD SEG 输出 64-pin 封装 n=36, 80-pin 封装 n=48
V1	V1	—	—	AN	LCD 电压泵
VMAX	VMAX	—	PWR	—	IC 最大电压，连接到 VDD, VLCD 或 V1
VLCD	VLCD	—	PWR	—	LCD 电源电压
VDD	VDD	—	PWR	—	电源电压
VSS	VSS	—	PWR	—	负电源电压，地

注：I/T：输入类型； O/T：输出类型；
OPT：通过配置选项 (CO) 或者寄存器选项来配置；
PWR：电源； CO：配置选项；
ST：斯密特触发输入； NMOS：NMOS 输出；
CMOS：CMOS 输出； AN：模拟信号；
HXT：高速晶体振荡器； LXT：低速晶体振荡器。

极限参数

电源供应电压	$V_{SS}-0.3V \sim V_{SS}+6.0V$
端口输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3V$
储存温度	$-60^{\circ}C \sim 150^{\circ}C$
工作温度	$-40^{\circ}C \sim 85^{\circ}C$
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

$T_a=25^{\circ}C$

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)	—	$f_{SYS}=f_{HXT}=4MHz$	1.8	—	5.5	V
			$f_{SYS}=f_{HXT}=8MHz$ (HT69F340/350/360)	2.0	—	5.5	V
			$f_{SYS}=f_{HXT}=8MHz$ (HT67F370)	1.8	—	5.5	V
			$f_{SYS}=f_{HXT}=12MHz$	2.7	—	5.5	V
			$f_{SYS}=f_{HXT}=16MHz$ (HT69F340/350/360)	4.5	—	5.5	V
			$f_{SYS}=f_{HXT}=16MHz$ (HT67F370)	3.3	—	5.5	V
V_{DD}	工作电压 (HIRC)	—	$f_{SYS}=f_{HIRC}=4MHz$	1.8	—	5.5	V
			$f_{SYS}=f_{HIRC}=8MHz$ (HT69F340/350/360)	2.0	—	5.5	V
			$f_{SYS}=f_{HXT}=8MHz$ (HT67F370)	1.8	—	5.5	V
			$f_{SYS}=f_{HIRC}=12MHz$	2.7	—	5.5	V
V_{IL}	I/O 口的低电平输入电压	5V	—	0.0	—	1.5	V
	低电平输入电压 (RES)	—	—	0	—	$0.2V_{DD}$	V
	低电平输入电压 (RES)	—	—	0	—	$0.4V_{DD}$	V
V_{IH}	I/O 口的高电平输入电压	5V	—	3.5	—	5.0	V
	高电平输入电压 (RES)	—	—	$0.8V_{DD}$	—	V_{DD}	V
	高电平输入电压 (RES)	—	—	$0.9V_{DD}$	—	V_{DD}	V

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{DD}	工作电流 (HXT)	3V	f _H =4MHz, 无负载,	—	0.50	0.75	mA
		5V	所有外设关闭	—	1.0	1.5	mA
		3V	f _H =8MHz, 无负载,	—	1.0	1.5	mA
		5V	所有外设关闭	—	2	3	mA
		3V	f _H =12MHz, 无负载,	—	1.50	2.75	mA
		5V	所有外设关闭	—	3.0	4.5	mA
	工作电流 (HIRC)	5V	f _H =16MHz, 无负载,	—	4	6	mA
		3V	f _H =4MHz, 无负载,	—	0.4	0.6	mA
		5V	所有外设关闭	—	0.8	1.2	mA
		3V	f _H =8MHz, 无负载,	—	0.8	1.2	mA
		5V	所有外设关闭	—	1.6	2.4	mA
		3V	f _H =12MHz, 无负载,	—	1.2	1.8	mA
		5V	所有外设关闭	—	2.4	3.6	mA
	工作电流 (LXT)	3V	f _{sys} =32.768kHz, 无负载,	—	10	20	μA
		5V	所有外设关闭	—	30	50	μA
	工作电流 (LIRC)	3V	f _{sys} =32kHz, 无负载,	—	10	20	μA
		5V	所有外设关闭	—	30	50	μA
I _{STB}	待机电流 (SLEEP 模式)	3V	WDT off, 无负载,	—	0.2	0.8	μA
		5V	所有外设关闭	—	0.5	1.0	μA
		3V	WDT on, 无负载,	—	1.5	3.0	μA
		5V	所有外设关闭	—	3	5	μA
	待机电流 (IDLE0 模式)	3V	f _{SUB} on, 无负载,	—	3	5	μA
		5V	所有外设关闭	—	5	10	μA
	待机电流 (IDLE1 模式, HIRC)	3V	f _H =4MHz, f _{SUB} on,	—	180	270	μA
		5V	无负载, 所有外设关闭	—	400	600	μA
		3V	f _H =8MHz, f _{SUB} on,	—	360	500	μA
		5V	无负载, 所有外设关闭	—	600	800	μA
		3V	f _H =12MHz, f _{SUB} on,	—	540	750	μA
		5V	无负载, 所有外设关闭	—	800	1200	μA

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{STB}	待机电流 (IDLE1 模式, HXT)	3V	f _H =4MHz, f _{SUB} on, 无负载, 所有外设关闭	—	180	270	μA
		5V	(HT69F340/350/360)	—	400	600	μA
		3V	f _H =4MHz, f _{SUB} on, 无负载, 所有外设关闭	—	180	250	μA
		5V	(HT67F370)	—	400	600	μA
		3V	f _H =8MHz, f _{SUB} on, 无负载, 所有外设关闭	—	360	500	μA
		5V		—	600	800	μA
		3V	f _H =12MHz, f _{SUB} on, 无负载, 所有外设关闭	—	540	750	μA
		5V		—	800	1200	μA
		5V	f _H =16MHz, f _{SUB} on, 无负载, 所有外设关闭	—	1.10	1.65	mA
		5V	(HT69F340/350/360)	—	1.4	2.0	mA
I _{OL}	I/O 口灌电流 (HT67F370)	1.8V	V _{OL} =0.1V _{DD}	7	14	—	mA
		3V		16	32	—	
		5V		32	64	—	
	输入 / 输出口灌电流 (HT69F340/350/360)	3V	V _{OL} =0.1V _{DD}	15	30	—	mA
		5V		30	60	—	mA
I _{OH}	I/O 口源电流 (HT67F370)	1.8V	V _{OH} =0.9V _{DD}	-1.4	-2.8	—	mA
		3V		-4	-8	—	
		5V		-10	-20	—	
	输入 / 输出口源电流 (HT69F340/350/360)	3V	V _{OH} =0.9V _{DD}	3.5	7.0	—	mA
		5V		7.5	15.0	—	mA
R _{PH}	I/O 口上拉电阻 (HT67F370)	3V	LVPU=0	40	120	200	kΩ
		5V		30	60	120	
		3V	LVPU=1	10	30	50	kΩ
		5V		5	15	25	
	输入 / 输出口上拉电阻 (HT69F340/350/360)	3V	LVPU=0	60	120	240	kΩ
		5V		30	60	120	
		3V	LVPU=1	15	30	60	kΩ
		5V		7.5	15.0	30	
V _{OH}	输入 / 输出口的高电平输出电压	3V	I _{OH} =2mA	2.7	—	—	V
		5V	I _{OH} =5mA	4.5	—	—	V
V _{OL}	输入 / 输出口的低电平输出电压	3V	I _{OL} =4mA	—	—	0.3	V
		5V	I _{OL} =10mA	—	—	0.5	V

交流电气特性

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
f _{sys}	系统时钟 (HXT)	1.8V ~5.5V	—	0.4	—	4.0	MHz
		2.0V ~5.5V		0.4	—	8.0	MHz
		2.7V ~5.5V		0.4	—	12.0	MHz
		4.5V ~5.5V		0.4	—	16.0	MHz
	系统时钟 (HIRC)	3V/5V	Ta=25°C	-2%	4	+2%	MHz
		3V/5V	Ta=25°C	-2%	8	+2%	MHz
		5V	Ta=25°C	-2%	12	+2%	MHz
		2.2V ~3.6V	Ta=-40°C~85°C	-8%	4	+8%	MHz
		3.0V ~5.5V	Ta=-40°C~85°C	-8%	4	+8%	MHz
		2.2V ~3.6V	Ta=-40°C~85°C	-8%	8	+8%	MHz
		3.0V ~5.5V	Ta=-40°C~85°C	-8%	8	+8%	MHz
		3.0V ~5.5V	Ta=-40°C~85°C	-11%	12	+11%	MHz
	系统时钟 (LIRC)	5V	Ta=25°C	-10%	32	+10%	kHz
		2.2V ~5.5V	Ta=-40°C~85°C	-50%	32	+60%	kHz
t _{EE RD}	EEPROM 读周期	—	—	1	2	4	t _{sys}
t _{EE WR}	EEPROM 写周期	—	—	1	2	4	ms
t _{TIMER}	xTCKn 和 TM 捕捉输入 最小脉宽	—	—	0.3	—	—	μs
t _{RES}	外部复位最小脉宽	—	—	10	—	—	μs
t _{INT}	外部中断最小脉宽	—	—	10	—	—	μs

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
t _{SST}	系统启动时间 (从暂停模式中唤醒, 暂停状态下 f _{SYS} off, 低速模式 → 正常模式 正常模式 → 低速模式)	—	f _{SYS} =f _{HXT} 或 f _{LXT} (低速模式 → 正常模式 (f _{HXT}), 正常模式 → 低速模式 (f _{LXT}))	1024	—	—	t _{SYS}
			f _{SYS} =f _{HXT} (从暂停模式中唤醒, 暂停状态下 f _{SYS} off)	128	—	—	t _{SYS}
			f _{SYS} =f _{LXT} (从暂停模式中唤醒, 暂停状态下 f _{SYS} off)	1024	—	—	t _{SYS}
		—	f _{SYS} =f _{HIRC}	16	—	—	t _{SYS}
		—	f _{SYS} =f _{LIRC}	2	—	—	t _{SYS}
	系统启动时间 (从暂停模式中唤醒, 暂停状态下 f _{SYS} on)	—	—	2	—	—	t _{SYS}
t _{RSTD}	系统复位延迟时间 (上电复位, LVR 硬件 复位, LVR 软件复位, WDT 软件复位)	—	—	25	50	100	ms
	系统复位延迟时间 (RES 复位, WDT 溢出 硬件复位)	—	—	8.3	16.7	33.3	ms

注: 1. t_{SYS}=1/f_{SYS}

2. 为保持内部 HIRC 振荡器频率的精确度, 在 VDD 和 VSS 之间应连接上一个 0.1μF 的去耦电容, 并离设备尽可能近。

A/D 转换器电气特性 – HT67F370

Ta=-40°C~85°C, 除非特别说明

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	—	—	1.8	—	5.5	V
V _{ADI}	输入电压	—	—	0	—	V _{REF}	V
V _{REF}	参考电压	—	—	1.8	—	V _{DD}	V
DNL	非线性微分误差	1.8V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =2.0μs	—	—	±3	LSB
		2V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =0.5μs				
		3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =0.5μs				
		5V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =0.5μs				
		1.8V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =10μs				
		3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =10μs				
		5V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =10μs				

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
INL	非线性积分误差	1.8V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, V _{REF} =V _{DD} , t _{ADCK} =2.0μs	—	—	±4	LSB
		2V	SAINS[3:0]=0000B,				
		3V	SAVRS[1:0]=01B,				
		5V	V _{REF} =V _{DD} , t _{ADCK} =0.5μs				
		1.8V	SAINS[3:0]=0000B,				
		3V	SAVRS[1:0]=01B,				
		5V	V _{REF} =V _{DD} , t _{ADCK} =10μs				
I _{ADC}	A/D 转换器使能的额外电流	1.8V	无负载, t _{ADCK} =2.0μs	—	0.5	1.0	mA
		3V	无负载, t _{ADCK} =0.5μs	—	1	2	
		5V	无负载, t _{ADCK} =0.5μs	—	1.5	3.0	
t _{ADCK}	时钟周期	—	1.8V≤V _{DD} <2.0V	2.0	—	10.0	μs
			2.0V≤V _{DD} ≤5.5V	0.5	—	10.0	
t _{ON2ST}	A/D 转换器 On-to-Start 时间	—	—	4	—	—	μs
t _{ADS}	采样时间	—	—	—	4	—	t _{ADCK}
t _{ADC}	转换时间 (包括采样和保持时间)	—	—	—	16	—	t _{ADCK}
I _{PGA}	PGA 使能的额外电流	2.2V	无负载	—	250	400	μA
		3V		—	300	450	
		5V		—	400	500	
V _{CM}	PGA 共模电压范围	2.2V	—	V _{SS}	—	V _{DD} -1.4	V
		3V		V _{SS}	—	V _{DD} -1.4	
		5V		V _{SS}	—	V _{DD} -1.4	
V _{OR}	PGA 最大输出电压范围	2.2V	—	V _{SS} +0.1	—	V _{DD} -0.1	V
		3V		V _{SS} +0.1	—	V _{DD} -0.1	
		5V		V _{SS} +0.1	—	V _{DD} -0.1	
V _{VR}	PGA 固定输出电压	2.2V~5.5V	Ta=-40°C~85°C	-1%	2	+1%	V
		3.2V~5.5V		-1%	3	+1%	
		4.2V~5.5V		-1%	4	+1%	

LVD & LVR 电气特性

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{LVR}	低电压复位电压 (HT69F340/350/360)	—	LVR 使能, 选择 1.7V	-5%	1.7	+5%	V
			LVR 使能, 选择 1.9V		1.9		
			LVR 使能, 选择 2.55V		2.55		
			LVR 使能, 选择 3.15V		3.15		
			LVR 使能, 选择 3.8V		3.8		
	低电压复位电压 (HT67F370)	—	LVR 使能, 电压选择 2.55V	-3%	2.55	+3%	
LVR 使能, 电压选择 3.15V			3.15				
LVR 使能, 电压选择 3.8V			3.8				
V _{LVD}	低电压检测电压	—	LVDEN=1, V _{LVD} =1.8V	-5%	1.8	+5%	V
			LVDEN=1, V _{LVD} =2.0V		2.0		V
			LVDEN=1, V _{LVD} =2.4V		2.4		V
			LVDEN=1, V _{LVD} =2.7V		2.7		V
			LVDEN=1, V _{LVD} =3.0V		3.0		V
			LVDEN=1, V _{LVD} =3.3V		3.3		V
			LVDEN=1, V _{LVD} =3.6V		3.6		V
			LVDEN=1, V _{LVD} =4.0V		4.0		V
I _{LVR}	使用 LVR 的额外功耗 (HT69F340/350/360)	3V	LVR 除能 → LVR 使能	—	10	20	μA
	5V	—		15	30	μA	
	使能 LVR 的额外功耗 (HT67F370)	5V	LVD 除能, PGAIS=0	—	—	8	μA
I _{LVD}	使用 LVD 的额外功耗 (HT69F340/350/360)	3V	LVD 除能 → LVD 使能 (LVR 除能)	—	10	20	μA
		5V		—	15	30	μA
		3V	LVD 除能 → LVD 使能 (LVR 使能)	—	1	2	μA
		5V		—	2	4	μA
	使能 LVD 的额外功耗 (HT67F370)	5V	LVR 除能, PGAIS=0	—	—	8	μA
t _{LVR}	产生 LVR 复位的低电压最短保持时间	—	—	120	240	480	μs
t _{LVD}	产生 LVD 中断的低电压最短保持时间	—	—	60	120	240	μs
t _{LVDS}	LVDO 稳定时间 (HT69F340/350/360)	—	由于 LVR 使能, LVD off→on	—	—	15	μs
		—	由于 LVR 除能, LVD off→on	—	—	150	μs
	LVDO 稳定时间 (HT67F370)	—	LVR 使能, PGAIS=0, LVD off → on	—	—	15	μs
			LVR 除能, PGAIS=0, LVD off → on	—	—	150	μs

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{LVR/LVDBG}	工作电流 (HT67F370)	3V	LVD 使能, LVR 使能, V _{LVR} =1.9V, V _{LVD} =2V, PGAIS=0	—	—	10	μA
		5V		—	8	15	
		3V	LVD 使能, LVR 使能, V _{LVR} =1.9V, V _{LVD} =2V, PGAIS=1	—	—	100	μA
		5V		—	80	115	
t _{SRESET}	软件复位最短时间	—	—	45	90	120	μs

内部参考电压电气特性 – HT67F370

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{BG}	Bandgap 参考电压	—	—	-3%	1.2	+3%	V
I _{BG}	Bandgap 参考电压使能的额外功耗	—	LVR 除能, LVD 除能	—	—	110	μA
t _{BGS}	V _{BG} 开启稳定时间	—	—	—	—	50	μs

注：1. 以上参数均应在无负载的条件下测得，除非有特别说明。

2. V_{DD} 引脚需连接一个 0.1μF 陶瓷电容到地。

3. V_{BG} 电压可以用作 A/D 转换器参考电压输入。

LCD 直流电气特性

Ta=25°C

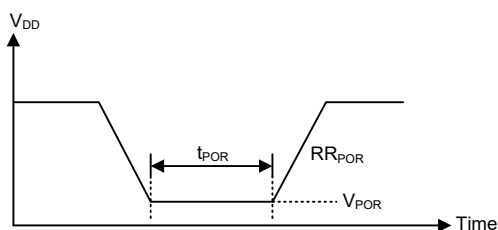
符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{STB}	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, C 型, V _{LCD} =V _{DD} , 1/2 偏压	—	6	10	μA
		5V		—	10	15	μA
	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, C 型, V _{LCD} =V _{DD} , 1/3 偏压	—	6	10	μA
		5V		—	10	15	μA
	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/2 偏压 (I _{BIAS} =7.5μA)	—	13.5	20.0	μA
		5V		—	22.5	40.0	μA
	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/2 偏压 (I _{BIAS} =15μA)	—	21	40	μA
		5V		—	35	60	μA
	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/2 偏压 (I _{BIAS} =45μA)	—	51	80	μA
		5V		—	85	160	μA
	IDLE 模式下待机电流 (f _{SYS} , f _{WDT} off, f _S =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/2 偏压 (I _{BIAS} =90μA)	—	96	160	μA
		5V		—	160	320	μA

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{STB}	IDLE 模式下待机电流 (f _{sys} , f _{WDT} off, f _s =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/3 偏压 (I _{BIAS} =7.5μA)	—	11	20	μA
		5V		—	18.3	40.0	μA
	IDLE 模式下待机电流 (f _{sys} , f _{WDT} off, f _s =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/3 偏压 (I _{BIAS} =15μA)	—	16	25	μA
		5V		—	26.6	50.0	μA
	IDLE 模式下待机电流 (f _{sys} , f _{WDT} off, f _s =f _{SUB} =32768 或 32K RC OSC)	3V	无负载, 系统暂停, LCD on, WDT off, R 型, V _{LCD} =V _{DD} , 1/3 偏压 (I _{BIAS} =45μA)	—	36	50	μA
		5V		—	60	100	μA
I _{OL}	LCD COM 和 SEG 引脚灌 电流	3V	V _{OL} =0.1V _{LCD}	—	66	100	μA
		5V		—	110	200	μA
I _{OH}	LCD COM 和 SEG 引脚源 电流	3V	V _{OH} =0.9V _{LCD}	—	210	420	μA
		5V		—	350	700	μA

上电复位电气特性

Ta=25°C

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 _{POR}	上电复位电压速率	—	—	0.035	—	—	V/ms
t _{POR}	V _{DD} 保持为 V _{POR} 的最小时间	—	—	1	—	—	ms

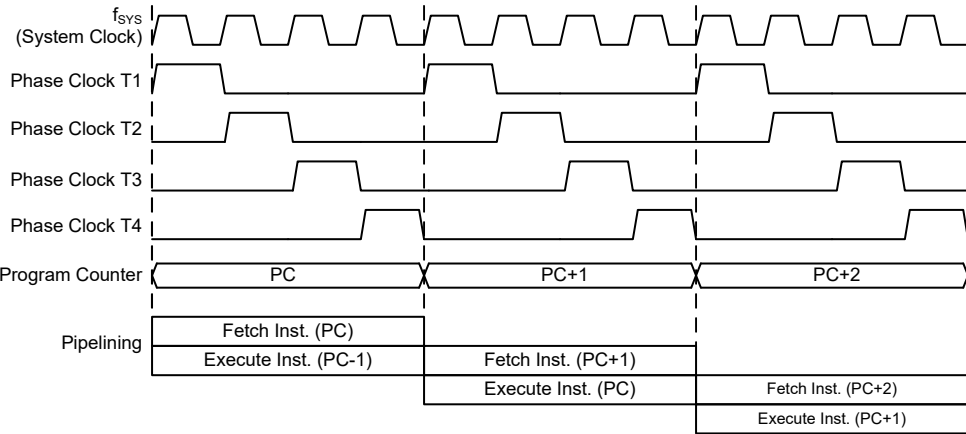


系统结构

内部系统结构是 Holtek 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需多一个指令周期外，其它大部分标准指令或扩展指令都能分别在一个或两个指令周期内完成。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 和 A/D (仅存在于 HT67F370) 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。

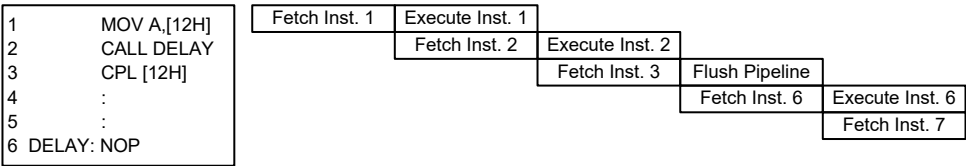
时序和流水线结构

主系统时钟由 HXT，LXT，HIRC 或 LIRC 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 开始时，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。



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的地址。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。对于存储器容量大于 8K 个字的单片机，程序存储器地址可能位于某一程序存储区，可通过程序存储区指针的 PBP1~PBP0 位来选择。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。

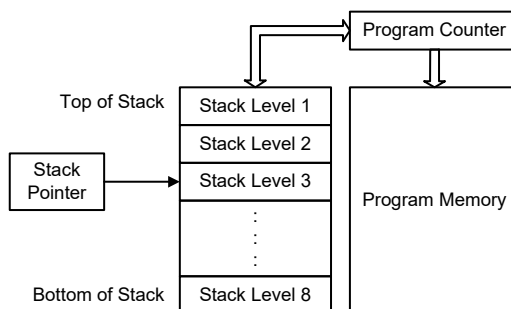
程序计数器的低字节，即程序计数器的低字节寄存器 PCL，可以通过程序控制，且它是可读可写的寄存器。通过直接将数据传进这个寄存器，一个程序短跳转可直接执行，然而只有低字节的操作是有效的，跳转被限制在存储器的当前页中，即 256 个存储器地址范围内，需注意当这样一个程序跳转要执行时，会插入一个空指令周期。PCL 寄存器的使用可能引起程序跳转，因此需要一个额外的指令周期去预取。

单片机型号	程序计数器	
	程序计数器高字节	PCL 寄存器
HT69F340	PC11~PC8	PCL7~PCL0
HT69F350	PC12~PC8	PCL7~PCL0
HT69F360	PBP0, PC12~PC8	PCL7~PCL0
HT67F370	PBP1~PBP0, PC12~PC8	PCL7~PCL0

堆栈

堆栈是一个特殊的存储空间，只用于存储程序计数器的内容。堆栈既不是数据部分也非程序空间部分，既不可读也不可写。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。



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
LADD, LADDM, LADC, LADCM, LSUB, LSUBM, LSBC, LSBCM, LDAA
- 逻辑运算：
AND, OR, XOR, ANDM, ORM, XORM, CPL, CPLA
LAND, LANDM, LOR, LORM, LXOR, LXORM, LCPL, LCPLA
- 移位运算：
RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
LRR, LRRR, LRRCA, LRRR, LRLA, LRL, LRLCA, LRLC
- 递增和递减：
INCA, INC, DECA, DEC, LINCA, LINC, LDECA, LDEC
- 分支判断：
JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA, CALL, RET, RETI,
LSNZ, LSZ, LSZA, LSIZ, LSIZA, LSDZ, LSDZA

Flash 程序存储器

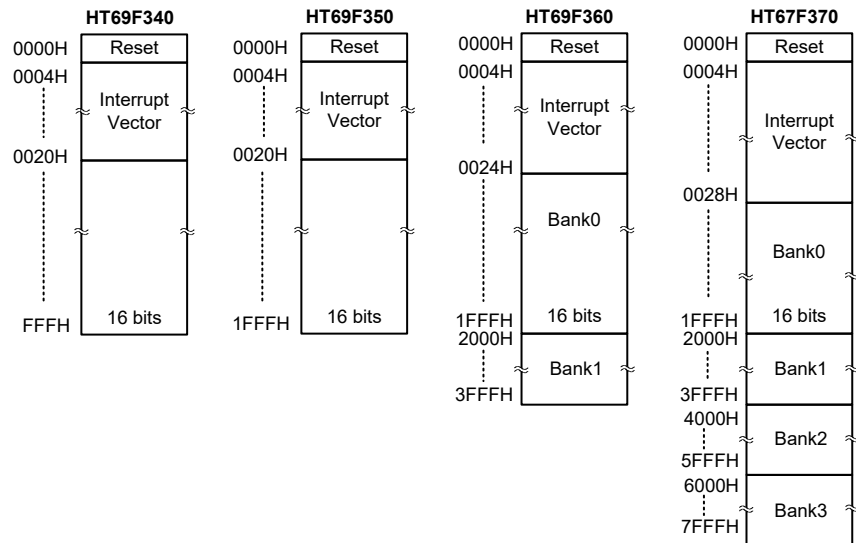
程序存储器用来存放用户代码即储存程序。此系列单片机的程序存储器为 FLASH 类型意味着可以多次重复烧录，方便用户使用同一芯片进行程序代码的修改。使用适当的单片机烧录工具，此系列所有单片机提供用户灵活便利的调试方法和项目开发规划及更新。

结构

程序存储器的容量为 $4K \times 16$ 位到 $32K \times 16$ 位，程序存储器用程序计数器来寻址，其中也包含数据、表格信息和中断入口。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
HT69F340	$4K \times 16$	—
HT69F350	$8K \times 16$	—
HT69F360	$16K \times 16$	0~1
HT67F370	$32K \times 16$	0~3

HT69F360 单片机的程序存储器分为两个 Bank，分别为 Bank0~Bank1。通过 PBP 寄存器中的 Bit 0 位选择所需要的 Bank。HT67F370 单片机的程序存储器分为四个 Bank，分别为 Bank0~Bank3。通过 PBP 寄存器中的 Bit 0~1 位选择所需要的 Bank。



程序存储器结构

特殊向量

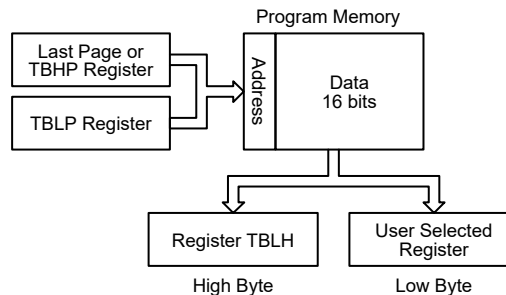
程序存储器内部某些地址保留用作诸如复位和中断入口等特殊用途。地址 0000H 是芯片复位后的程序起始地址。在芯片复位之后，程序将跳到这个地址并开始执行。

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。这两个寄存器定义表格总的地址。

在设置完表格指针后，当数据存储器 [m] 位于数据存储器 Sector 0，表格数据可以使用“TABRD [m]”或“TABRDL [m]”指令分别从程序存储器查表读取。如果存储器 [m] 位于数据存储器其它 Sector，表格数据可以使用“LTABRD [m]”或“LTABRDL [m]”指令分别从程序存储器查表读取。当这些指令执行时，程序存储器中表格数据低字节，将被传送到使用者在指令中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。

下图是查表中寻址 / 数据流程：



查表范例

以下范例说明如何定义并从单片机中获取表格指针和表格数据。这个例子使用的是利用“ORG”指令储存在程序存储器中的一个原始数据表。ORG 指令的值“1F00H”位于 HT67F370 单片机中 32K 程序存储器内最后一页的起始地址。表格指针设置的初始值为“06H”，以确保从数据表格读取的第一笔数据位于程序存储器地址“7F06H”，或是指定页起始地址后的第六个地址。值得注意的是，若“TABRD [m]”或“LTABRD [m]”指令被使用，则表格指针指向 TBHP 和 TBLP 指定的地址。在这个例子中，表格数据的高字节等于零，而当“TABRD [m]”或“LTABRD [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

TBLH 寄存器为可读 / 写寄存器，可以重新储存，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。

表格读取程序范例

```

rombank 3 code3
ds .section 'data'
tempreg1 db ?      ; temporary register #1 in current page
tempreg2 db ?      ; temporary register #2 in current page
:
:
mov a,06h           ; initialise low table pointer - note that this address
                    ; is referenced
mov tblp,a          ; to the last page or the page that tbhp pointed
mov a,7fh           ; initialise high table pointer
mov tbhp,a          ; it is not necessary to set tbhp if executing tabrdl
:
:
tabrd tempreg1
tabrdl tempreg1     ; transfers value in table referenced by table pointer
                    ; data at program memory address 7F06H transferred to
                    ; tempreg1 and TBLH

```

```

dec tblp          ; reduce value of table pointer by one
tabrd tempreg2    ; transfers value in table referenced by table pointer
tabrdl tempreg2    ; data at program memory address 7F05H transferred to
                  ; tempreg2 and TBLH.
                  ; In this example the data "1AH" is transferred to
                  ; tempreg1 and data "0FH" to register tempreg2 while the
                  ; value 00H will be transferred to the high byte
                  ; register TBLH
:
:
org 1F00h          ; sets initial address of lastpage
dc 000Ah,000Bh,000Ch,000Dh,000Eh,000Fh,001Ah,001Bh
:
:

```

在线烧录 – ICP

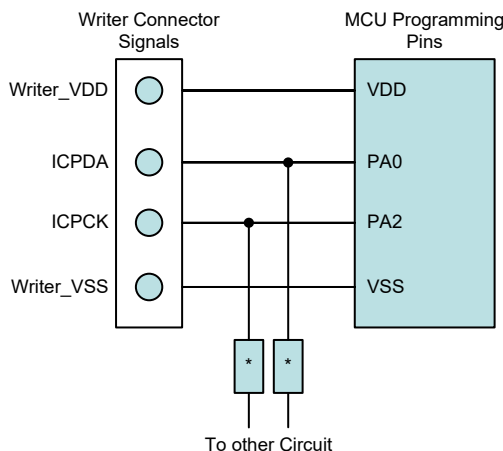
Flash 型程序存储器的提供使得用户可以方便简单地在同一芯片上进行程序的更新和修改。

另外，Holtek 单片机提供 4 线接口的在线烧录方式。用户可将进行过烧录或未经过烧录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。

Holtek 烧录器引脚	MCU 在线烧录引脚	功能
ICPDA	PA0	串行数据 / 地址烧录
ICPCK	PA2	时钟烧录
VDD	VDD	电源
VSS	VSS	地

程序存储器可以通过 4 线的接口在线进行烧录。其中一个单独的引脚用于数据串行下载或上传、一条用于串行时钟、两条用于提供电源。芯片在线烧录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。

在烧录过程中，用户控制 ICPDA 和 ICPCK 脚进行数据和时钟烧录，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，* 的电阻值必须大于 1kΩ。若为电容，* 的电容值必须小于 1nF。

片上调试 – OCDS

EV 芯片 HT69V3x0 和 HT67V370 分别用于 HT69F3x0 和 HT67F370 单片机的仿真。此 EV 芯片提供片上调试功能 (OCDS) 用于开发过程中该系列单片机的调试。除了片上调试功能，EV 芯片和实际单片机在功能上几乎是兼容的。用户可将 OCSDA 和 OCDSCK 引脚连接至 Holtek HT-IDE 开发工具，用 EV 芯片来仿真实际单片机芯片的性能。OCSDA 引脚为 OCDS 数据 / 地址输入 / 输出脚，OCDSCK 引脚为 OCDS 时钟输入脚。当用户用 EV 芯片进行调试时，OCSDA 和 OCDSCK 引脚上的其它共用功能对 EV 芯片无效。这两个 OCDS 引脚与 ICP 引脚共用，因此在线烧录时仍用作 Flash 存储器烧录引脚。关于 OCDS 功能的详细描述，参考“Holtek e-Link for 8-bit MCU OCDS User's Guide”文档。

Holtek e-Link 引脚	EV 芯片 OCDS 引脚	引脚描述
OCSDA	OCSDA	片上调试数据 / 地址输入 / 输出
OCDSCK	OCDSCK	片上调试时钟输入
VDD	VDD	电源
GND	VSS	地

在线应用烧录 – IAP

单片机提供 IAP 功能来对 Flash ROM 进行数据和程序更新。用户可自行定义 IAP 的 ROM 地址，但是用户在使用 IAP 功能时必须注意几个特点。注意，HT69F340 支持“块擦除”功能不支持“页擦除”功能。

HT69F340 IAP 配置		HT69F350 IAP 配置		HT69F360/HT67F370 IAP 配置	
块擦除	256 个字 / 块	页擦除	32 个字 / 页	页擦除	64 个字 / 页
写	4 个字 / 次	写	32 个字 / 次	写	64 个字 / 次
读	1 个字 / 次	读	1 个字 / 次	读	1 个字 / 次

在应用烧录控制寄存器

地址寄存器 FARL/FARH 和数据寄存器 FD0L/FD0H、FD1L/FD1H、FD2L/FD2H 和 FD3L/FD3H，以及位于数据存储器所有 Sector 的控制寄存器 FC0、FC1 和 FC2，都是与 IAP 相关的 Flash 存取寄存器。如果在除 sector0 外的其他 sector 中用间接寻址访问 FC0、FC1 和 FC2 寄存器，所有与这些寄存器相关的读写操作必须使用间接寻址寄存器 IAR1 和 IAR2，和一对存储器指针 MP1L/MP1H 或 MP2L/MP2H 来执行。举例来说，若想要访问 sector1 中的控制寄存器 FC0，FC1 和 FC2，由于这些寄存器的地址为数据存储 sector1 中的 67H~69H，则位于 67H 到 69H 地址范围的值必须首先被写入 MP1L 或 MP2L 存储器指针低字节，而“01H”必须写入 MP1H 或 MP2H 存储器指针高字节。

寄存器名称	位							
	7	6	5	4	3	2	1	0
FC0	CFWEN	FMOD2	FMOD1	FMOD0	FWPEN	FWT	FRDEN	FRD
FC1	D7	D6	D5	D4	D3	D2	D1	D0
FC2 (HT69F350/HT69F360/ HT67F370)	—	—	—	—	—	—	—	CLWB
FARL	A7	A6	A5	A4	A3	A2	A1	A0
FARH (HT69F340)	—	—	—	—	A11	A10	A9	A8

寄存器名称	位							
	7	6	5	4	3	2	1	0
FARH (HT69F350)	—	—	—	A12	A11	A10	A9	A8
FARH (HT69F360/HT67F370)	—	—	A13	A12	A11	A10	A9	A8
FD0L	D7	D6	D5	D4	D3	D2	D1	D0
FD0H	D15	D14	D13	D12	D11	D10	D9	D8
FD1L	D7	D6	D5	D4	D3	D2	D1	D0
FD1H	D15	D14	D13	D12	D11	D10	D9	D8
FD2L	D7	D6	D5	D4	D3	D2	D1	D0
FD2H	D15	D14	D13	D12	D11	D10	D9	D8
FD3L	D7	D6	D5	D4	D3	D2	D1	D0
FD3H	D15	D14	D13	D12	D11	D10	D9	D8

IAP 寄存器列表

● FC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CFWEN	FMOD2	FMOD1	FMOD0	FWPEN	FWT	FRDEN	FRD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	1	0	0	0	0

- Bit 7 CFWEN:** Flash 存储器写使能控制位
 0: Flash 存储器写功能除能
 1: Flash 存储器写功能已被成功使能
 当此位由应用程序清零后, Flash 存储器写功能除能。注意, 此位写“1”则会导致无效操作。此位被用来说明 Flash 存储器写功能状态。当此位由硬件置为“1”时, 就意味着 Flash 存储器写功能已经成功使能, 否则此位为“0”, 该功能除能。
- Bit 6~4 FMOD2~FMOD0:** 模式选择位
 000: 写程序存储器
 001: 块 / 页擦除程序存储器
 010: 保留位
 011: 读程序存储器
 100: 保留位
 101: 保留位
 110: FWEN 模式 – Flash 存储器写功能使能模式
 111: 保留位
 当这几位设置为“001”时, HT69F340 选择块擦除模式, 而 HT69F350/HT69F360/HT67F370 选择页擦除模式。
- Bit 3 FWPEN:** Flash 存储器写步骤使能控制
 0: 除能
 1: 使能
 当此位置为“1”且 FMOD 位段设为“110”时, IAP 控制器将执行“Flash 存储器写功能使能”步骤。一旦 Flash 存储器写功能成功使能, 无需再设置 FWPEN 位。
- Bit 2 FWT:** Flash 存储器写开始控制位
 0: 不开始 Flash 存储器写或 Flash 存储器写过程已完成
 1: 开始 Flash 存储器写过程
 此位由软件置“1”。当 Flash 程序存储器写过程完成, 由硬件清零。

- Bit 1 **FRDEN**: Flash 存储器读使能位
0: Flash 存储器读除能
1: Flash 存储器读使能
- Bit 0 **FRD**: Flash 存储器读开始控制位
0: 不开始 Flash 存储器读或 Flash 存储器读过程已完成
1: 开始 Flash 存储器读过程
此位由软件置“1”。当 Flash 程序存储器写过程完成, 由硬件清零。

注: 1. 在同一条指令中 FWT、FRDEN 和 FRD 位不可同时设置为“1”。
2. 确保 f_{SUB} 时钟在执行擦或写动作前已稳定。
3. 当读、擦或写动作成功启动后, CPU 相关操作将停止。
4. 确保读、擦或写动作成功完成后才可执行其它操作。

● FC1 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/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 **D7~D0**: 整个芯片复位模式
当用户写“55H”到该寄存器, 将产生一个复位信号将整个单片机复位。

● FC2 寄存器 – HT69F350/HT69F360/HT67F370

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	CLWB
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

- Bit 7~1 未定义, 读为“0”
- Bit 0 **CLWB**: Flash 存储器写缓冲清除控制位
0: 不开始写缓冲区清除或写缓冲清除过程已完成
1: 开始写缓冲区清除过程
在对页执行写操作之前, 用户必须设置 CLWB 位来清除写缓冲。
当写缓冲区清除过程完成, 此位由软件置“1”, 由硬件清零。

● FARL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	A7	A6	A5	A4	A3	A2	A1	A0
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 Flash 存储器地址 [7:0]

● FARH 寄存器 – HT69F340

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	A11	A10	A9	A8
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 Flash 存储器地址 [11:8]

● FARH 寄存器 – HT69F350

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	A12	A11	A10	A9	A8
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 未定义，读为“0”

Bit 4~0 Flash 存储器地址 [12:8]

● FARH 寄存器 – HT69F360/HT67F370

Bit	7	6	5	4	3	2	1	0
Name	—	—	A13	A12	A11	A10	A9	A8
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 Flash 存储器地址 [13:8]

● FD0L 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 第一个 Flash 存储器数据 [7:0]

● FD0H 寄存器

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 第一个 Flash 存储器数据 [15:8]

● FD1L 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 第二个 Flash 存储器数据 [7:0]

● FD1H 寄存器

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 第二个 Flash 存储器数据 [15:8]

● FD2L 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 第三个 Flash 存储器数据 [7:0]

● FD2H 寄存器

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 第三个 Flash 存储器数据 [15:8]

● FD3L 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 第四个 Flash 存储器数据 [7:0]

● FD3H 寄存器

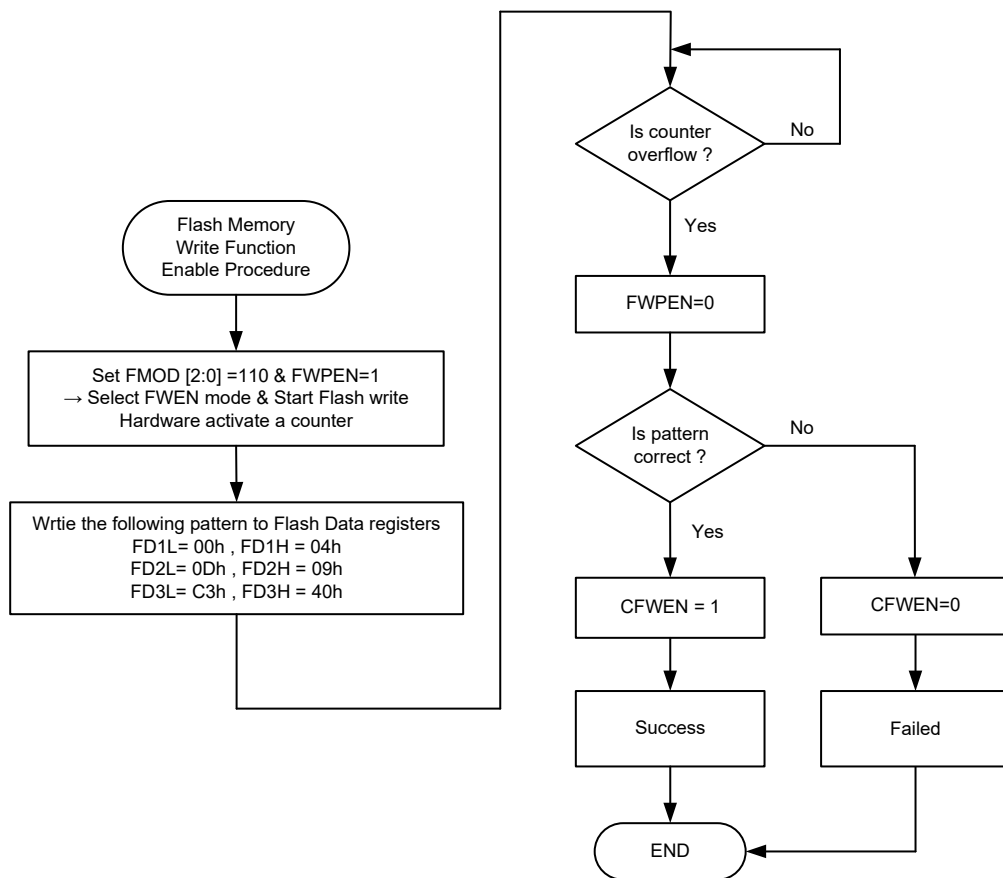
Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 第四个 Flash 存储器数据 [15:8]

Flash 存储器写功能使能步骤

为使用户可以通过 IAP 控制寄存器来更改 Flash 存储器数据，用户必须首先使能 Flash 存储器写操作，步骤如下：

1. 写“110”到 FMODE2~FMODE0 位，选择 FWEN 模式。
2. FWPEN 置为“1”。步骤 1 和步骤 2 可同时执行。
3. 数据序列 00H、04H、0DH、09H、C3H 和 40H 必须分别写入寄存器 FD1L、FD1H、FD2L、FD2H、FD3L 和 FD3H。
4. 溢出周期为 300μs 的计数器将进行有效计时，此时允许用户将正确的数据序列写入 FD1L/FD1H~FD3L/FD3H 寄存器对。计数器时钟来自 f_{SUB} 。
5. 如果计数器溢出或数据出错，Flash 存储器写操作将不被使能，用户必须重复以上步骤。
6. 如果在计数器溢出前数据正确，Flash 存储器写操作将被使能，CFWEN 位由硬件置为“1”，表明 Flash 存储器写操作成功使能。
7. 一旦 Flash 存储器写操作使能，用户可通过 Flash 控制寄存器更改 Flash ROM 数据。
8. 用户可以清零 CFWEN 位来除能 Flash 存储器写操作。



Flash 存储器写功能使能步骤

Flash 存储器写步骤

通过前面的 IAP 步骤成功使能 Flash 存储器写功能后，用户必须先擦除相应的 Flash 存储块，然后启动 Flash 写操作。对于 HT69F340，块擦除操作的数量是每块 256 个字，有效的块擦除地址只能由 FARH 寄存器指定，FARL 寄存器不用来指定块擦除地址。对于 HT69F350 单片机来说，由于页擦除操作的数量是每页 32 个字，有效的页擦除地址由 FARH 寄存器和 FARL 寄存器的 bit 7~bit 5 位来指定。对于 HT69F360/HT67F370 单片机来说，由于页擦除操作的数量是每页 64 个字，有效的页擦除地址由 FARH 寄存器和 FARL 寄存器的 bit 7~bit 6 位来指定。

擦除块	HT69F340	
	FARH [3:0]	FARL [7:0]
0	0000	XXXX XXXX
1	0001	XXXX XXXX
2	0010	XXXX XXXX
3	0011	XXXX XXXX
4	0100	XXXX XXXX
5	0101	XXXX XXXX
6	0110	XXXX XXXX
7	0111	XXXX XXXX

擦除块	HT69F340	
	FARH [3:0]	FARL [7:0]
8	1000	xxxx xxxx
9	1001	xxxx xxxx
10	1010	xxxx xxxx
11	1011	xxxx xxxx
12	1100	xxxx xxxx
13	1101	xxxx xxxx
14	1110	xxxx xxxx
15	1111	xxxx xxxx

“x”：无关

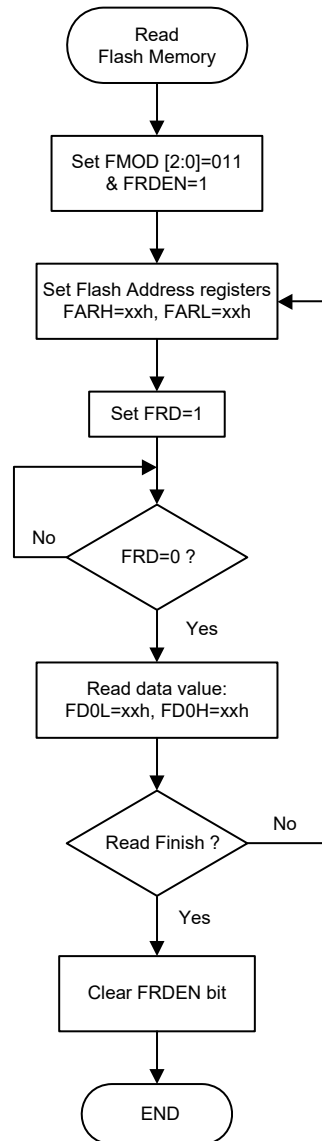
HT69F340 块擦除数量和选择

擦除页	HT69F350		HT69F360/HT67F370	
	FARH	FARL	FARH	FARL
0	0000 0000	000x xxxx	0000 0000	00xx xxxx
1	0000 0000	001x xxxx	0000 0001	01xx xxxx
2	0000 0000	010x xxxx	0000 0010	10xx xxxx
3	0000 0000	011x xxxx	0000 0011	11xx xxxx
4	0000 0000	100x xxxx	0000 0100	00xx xxxx
5	0000 0000	101x xxxx	0000 0101	01xx xxxx
6	0000 0000	110x xxxx	0000 0110	10xx xxxx
7	0000 0000	111x xxxx	0000 0111	11xx xxxx
8	0000 0001	000x xxxx	0000 1000	00xx xxxx
9	0000 0001	001x xxxx	0000 1001	01xx xxxx
:	:	:	:	:
:	:	:	:	:
252	0001 1111	100x xxxx	0011 1111	00xx xxxx
253	0001 1111	101x xxxx	0011 1111	01xx xxxx
254	0001 1111	110x xxxx	0011 1111	10xx xxxx
255	0001 1111	111x xxxx	0011 1111	11xx xxxx

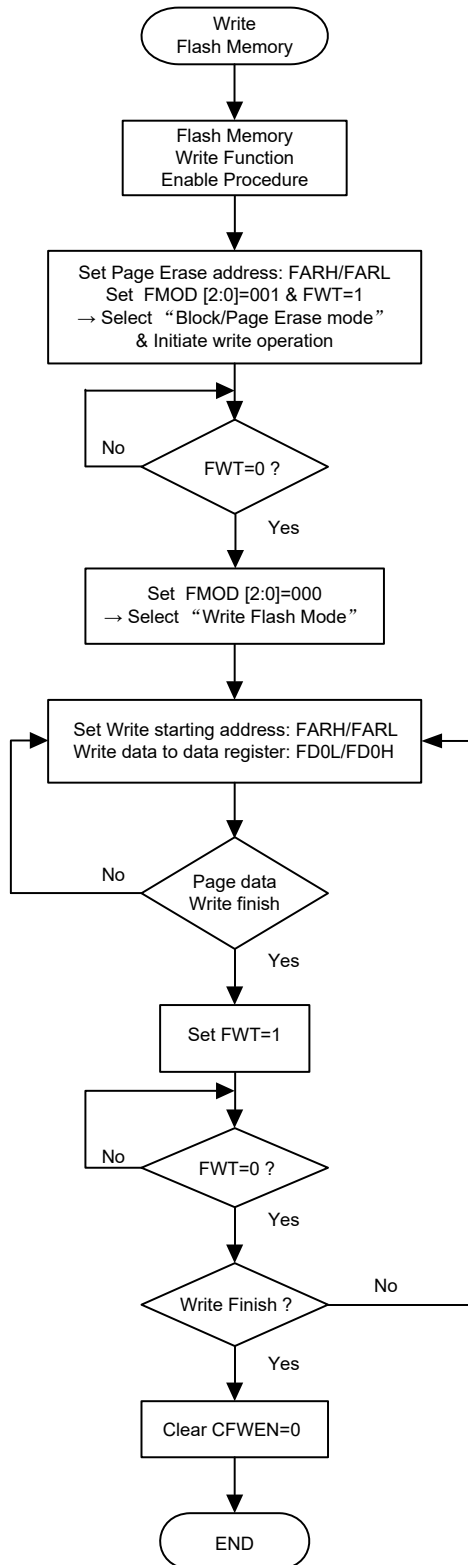
“x”：无关

注：HT69F350/360/HT67F370 单片机中有 256 个 IAP 擦除页。

HT69F350/HT69F360/HT67F370 页擦除数量和选择



读 Flash 存储器步骤



写 Flash 存储器步骤

注：当 FWT 或 FRD 位为“1”时，MCU 停止。

数据存储

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。

数据存储器分为三部分，第一部分是特殊功能寄存器，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。第二部分是通用数据存储器，所有地址都可在程序的控制下进行读取和写入。第三部分是为 LCD 存储器保留的。这个特殊数据存储器的地址直接映射到 LCD 显示器，写入这部分存储器的数据将直接影响显示的数据。

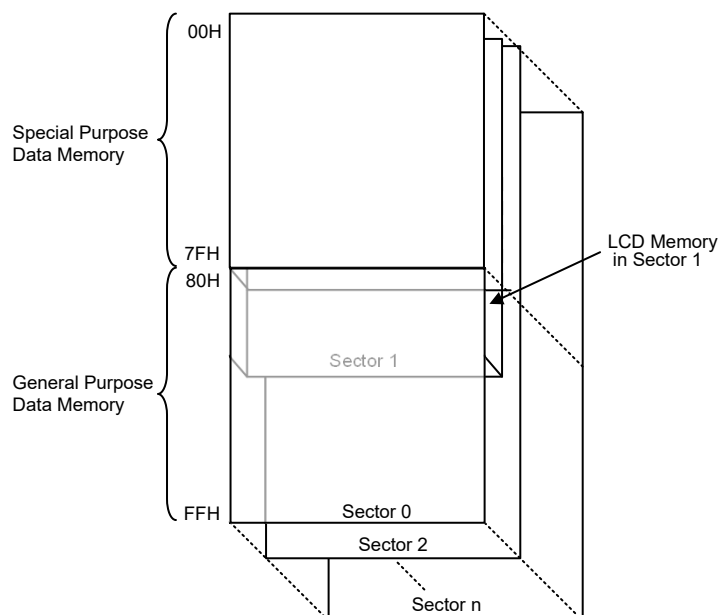
结构

数据存储器被分为几个 Sector，都位于 8 位数据存储器中。每个数据存储器 Sector 分为两类，特殊功能数据存储器 and 通用数据存储器。HT69F340 的 LCD 存储器地址范围为 Sector1 中的 80H~98H，HT69F350 的 LCD 存储器地址范围为 Sector1 中的 80H~A4H，而 HT69F360/HT67F370 则为 Sector1 中的 80H~B0H。

单片机数据存储器的起始地址都是 00H，而通用数据存储器的起始地址为 80H。特殊功能寄存器在数据存储器中的地址为 00H~7FH，是所有 Sector 共有且可访问的，而 EEC 寄存器则为例外，它只能在 Sector1 中被寻址。

单片机 型号	特殊功能数据存储器		LCD 显示数据存储器		通用数据存储器	
	容量	Sectors	容量	Sectors	容量	Sectors
HT69F340	128×8	0~2: 00H~7FH	25×8	1: 80H~98H	256×8	0: 80H~FFH 2: 80H~FFH
HT69F350	128×8	0~4: 00H~7FH	37×4	1: 80H~A4H	512×8	0: 80H~FFH 2: 80H~FFH 3: 80H~FFH 4: 80H~FFH
HT69F360	128×8	0~8: 00H~7FH	49×4	1: 80H~B0H	1024×8	0: 80H~FFH 2: 80H~FFH : 8: 80H~FFH
HT69F370	128×8	0~16: 00H~7FH	49×8	1: 80H~B0H	2048×8	0: 80H~FFH 2: 80H~FFH : 16: 80H~FFH

数据存储器概要



注：n=2 for HT69F340，n=4 for HT69F350，n=8 for HT69F360，n=16 for HT67F370。

数据存储器结构

数据存储器寻址

此系列单片机支持扩展指令架构，它并没有可用于数据存储器的存储区指针。存储区指针 PBP 仅适用于 HT69F360/HT67F370 单片机的程序存储器。对于数据存储器所需的 Sector 是通过 MP1H 或 MP2H 寄存器指定，而所选 Sector 的某一数据存储器地址使用间接寻址访问方式时，通过 MP1L 或 MP2L 寄存器指定。直接寻址可用于所有 Sector，通过相应的指令可以寻址所有可用的数据存储器空间。所访问的数据存储器位于除 Sector 0 外的任何数据存储器 Sector，扩展指令可代替间接寻址方式用来访问数据存储器。标准指令和扩展指令的主要区别在于扩展指令中的数据存储器地址“m”由高达 13 个有效位组成，高字节表示 Sector，低字节表示指定的地址。

通用数据存储器

所有的单片机程序需要一个读/写的存储区，让临时数据可以被储存和再使用，该 RAM 区域就是通用数据存储器。这个数据存储区可让使用者进行读取和写入的操作。在程序控制下使用位操作指令可对个别的位做置位或复位的操作，较大地方便了用户在数据存储器内进行位操作。

特殊功能数据存储器

这个区域的数据存储器是存放特殊寄存器的，这些寄存器与单片机的正确操作密切相关。大多数的寄存器可进行读取和写入，只有一些是被写保护而只能读取的，相关细节的介绍请参看有关特殊功能寄存器的部分。要注意的是，任何读取指令对存储器中未定义的地址进行读取将返回“00H”。

LCD 显示数据存储区

这个区域的数据存储器中存放的是要显示在 LCD 显示屏上的数据。通过写入数据到这部分 RAM，显示输出可直接由应用程序进行控制。由于 LCD 存储器地址与通用数据存储器的一部分相重叠，其数据将存储在其独立的 Sector1 区域中，用于 LCD 显示。对于 HT69F360/HT67F370，这部分地址即为 Sector1 中的 80H~B0H。要访问显示存储器，首先要将“01H”写入 MP1H 或 MP2H 中以选择 Sector1。而后存储器可利用 MP1L 或 MP2L 进行间接寻址。选中 Sector1，再利用 MP1L 或 MP2L 来读取或写入 80H~B0H 的存储器区域，即完成对 LCD 存储器的操作。对显示存储器进行直接寻址并不可行，这会导致数据进入到 Sector0 通用数据存储器。

Sector 0-2		Sector 0, 2 Sector 1		Sector 0-4		Sector 0, 2-4 Sector 1		Sector 0-8		Sector 0, 2-8 Sector 1		Sector 0-16		Sector 0, 2-16 Sector 1	
00H	IAR0	40H	EEA	00H	IAR0	40H	EEA	00H	IAR0	40H	EEA	00H	IAR0	40H	EEA
01H	MP0	41H		01H	MP0	41H		01H	MP0	41H		01H	MP0	41H	
02H	IAR1	42H	EED	02H	IAR1	42H	EED	02H	IAR1	42H	EED	02H	IAR1	42H	EED
03H	MP1L	43H		03H	MP1L	43H		03H	MP1L	43H		03H	MP1L	43H	
04H	MP1H	44H		04H	MP1H	44H		04H	MP1H	44H		04H	MP1H	44H	
05H	ACC	45H		05H	ACC	45H		05H	ACC	45H		05H	ACC	45H	
06H	PCL	46H		06H	PCL	46H		06H	PCL	46H		06H	PCL	46H	
07H	TBLP	47H		07H	TBLP	47H		07H	TBLP	47H		07H	TBLP	47H	
08H	TBLH	48H		08H	TBLH	48H		08H	TBLH	48H		08H	TBLH	48H	
09H	TBHP	49H		09H	TBHP	49H		09H	TBHP	49H		09H	TBHP	49H	
0AH	STATUS	4AH		0AH	STATUS	4AH		0AH	STATUS	4AH		0AH	STATUS	4AH	
0BH		4BH		0BH		4BH		0BH	PBP	4BH		0BH	PBP	4BH	
0CH	IAR2	4CH		0CH	IAR2	4CH		0CH	IAR2	4CH		0CH	IAR2	4CH	
0DH	MP2L	4DH		0DH	MP2L	4DH		0DH	MP2L	4DH		0DH	MP2L	4DH	
0EH	MP2H	4EH		0EH	MP2H	4EH		0EH	MP2H	4EH		0EH	MP2H	4EH	
0FH	TBC	4FH		0FH	TBC	4FH		0FH	TBC	4FH		0FH	TBC	4FH	
10H	INTC0	50H		10H	INTC0	50H		10H	INTC0	50H		10H	INTC0	50H	
11H	INTC1	51H		11H	INTC1	51H		11H	INTC1	51H		11H	INTC1	51H	
12H	SMOD1	52H		12H	SMOD1	52H		12H	SMOD1	52H		12H	SMOD1	52H	
13H	LVRC	53H		13H	LVRC	53H		13H	LVRC	53H		13H	LVRC	53H	
14H	MF0	54H		14H	MF0	54H		14H	MF0	54H		14H	MF0	54H	
15H	MF1	55H		15H	MF1	55H		15H	MF1	55H		15H	MF1	55H	
16H	MF2	56H		16H	MF2	56H		16H	MF2	56H		16H	MF2	56H	
17H	INTC2	57H		17H	INTC2	57H		17H	INTC2	57H		17H	INTC2	57H	
18H	PAWU	58H		18H	PAWU	58H		18H	PAWU	58H		18H	PAWU	58H	
19H	PAFU	59H		19H	PAFU	59H		19H	PAFU	59H		19H	PAFU	59H	
1AH	PA	5AH		1AH	PA	5AH		1AH	PA	5AH		1AH	PA	5AH	
1BH	PAC	5BH		1BH	PAC	5BH		1BH	PAC	5BH		1BH	PAC	5BH	
1CH	PBPU	5CH		1CH	PBPU	5CH		1CH	PBPU	5CH		1CH	PBPU	5CH	
1DH	PB	5DH		1DH	PB	5DH		1DH	PB	5DH		1DH	PB	5DH	
1EH	PBC	5EH		1EH	PBC	5EH		1EH	PBC	5EH		1EH	PBC	5EH	
1FH	PCPU	5FH		1FH	PCPU	5FH		1FH	PCPU	5FH		1FH	PCPU	5FH	
20H	PC	60H		20H	PC	60H		20H	PC	60H		20H	PC	60H	
21H	PCC	61H		21H	PCC	61H		21H	PCC	61H		21H	PCC	61H	
22H	PDPU	62H		22H	PDPU	62H		22H	PDPU	62H		22H	PDPU	62H	
23H	PD	63H		23H	PD	63H		23H	PD	63H		23H	PD	63H	
24H	PDC	64H		24H	PDC	64H		24H	PDC	64H		24H	PDC	64H	
25H	PEPU	65H		25H	PEPU	65H		25H	PEPU	65H		25H	PEPU	65H	
26H	PE	66H		26H	PE	66H		26H	PE	66H		26H	PE	66H	
27H	PEC	67H		27H	PEC	67H		27H	PEC	67H		27H	PEC	67H	
28H	PEPU	68H		28H	PEPU	68H		28H	PEPU	68H		28H	PEPU	68H	
29H	PF	69H		29H	PF	69H		29H	PF	69H		29H	PF	69H	
2AH	PFC	6AH		2AH	PFC	6AH		2AH	PFC	6AH		2AH	PFC	6AH	
2BH		6BH		2BH	PGPU	6BH		2BH	PGPU	6BH		2BH	PGPU	6BH	
2CH		6CH		2CH	PG	6CH		2CH	PG	6CH		2CH	PG	6CH	
2DH		6DH		2DH	PGC	6DH		2DH	PGC	6DH		2DH	PGC	6DH	
2EH		6EH		2EH	PHPU	6EH		2EH	PHPU	6EH		2EH	PHPU	6EH	
2FH		6FH		2FH	PH	6FH		2FH	PH	6FH		2FH	PH	6FH	
30H		70H		30H	PHC	70H		30H	PHC	70H		30H	PHC	70H	
31H	PAFS	71H		31H	PAFS	71H		31H	PAFS	71H		31H	PAFS	71H	
32H	PBFS	72H		32H	PBFS	72H		32H	PBFS	72H		32H	PBFS	72H	
33H		73H		33H		73H		33H		73H		33H		73H	
34H	PDFS	74H		34H	PDFS	74H		34H	PDFS	74H		34H	PDFS	74H	
35H	PEFS	75H		35H	PEFS	75H		35H	PEFS	75H		35H	PEFS	75H	
36H	PFFS	76H		36H	PFFS	76H		36H	PFFS	76H		36H	PFFS	76H	
37H		77H		37H	PGFS	77H		37H	PGFS	77H		37H	PGFS	77H	
38H		78H		38H	PHFS	78H		38H	PHFS	78H		38H	PHFS	78H	
39H		79H		39H	SFSR	79H		39H	SFSR	79H		39H	SFSR	79H	
3AH	CTMC0	7AH		3AH	CTMC0	7AH		3AH	CTMC0	7AH		3AH	CTMC0	7AH	
3BH	CTMC1	7BH		3BH	CTMC1	7BH		3BH	CTMC1	7BH		3BH	CTMC1	7BH	
3CH	CTMDL	7CH		3CH	CTMDL	7CH		3CH	CTMDL	7CH		3CH	CTMDL	7CH	
3DH	CTMDH	7DH		3DH	CTMDH	7DH		3DH	CTMDH	7DH		3DH	CTMDH	7DH	
3EH	CTMAL	7EH		3EH	CTMAL	7EH		3EH	CTMAL	7EH		3EH	CTMAL	7EH	
3FH	CTMAH	7FH		3FH	CTMAH	7FH		3FH	CTMAH	7FH		3FH	CTMAH	7FH	

□ : Unused, read as 00H

HT69F340

□ : Unused, read as 00H

HT69F350

□ : Unused, read as 00H

HT69F360

□ : Unused, read as 00H

HT67F370

特殊功能数据存储区

特殊功能寄存器描述

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, IAR2

间接寻址寄存器 IAR0、IAR1 和 IAR2 的地址虽位于 RAM 寄存器区，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寄存器和存储器指针做数据操作，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、IAR1 和 IAR2 上的任何动作，将对间接寻址指针 MP0、MP1L/MP1H 或 MP2L/MP2H 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 Sector 0，而 IAR1 和 MP1L/MP1H、IAR2 和 MP2L/MP2H 可以访问任何数据存储器 Sector。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。

存储器指针 – MP0, MP1L, MP1H, MP2L, MP2H

该系列单片机提供五个存储器指针，即 MP0、MP1L、MP1H、MP2L 和 MP2H。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 Sector 0，而 MP1L/MP1H 和 IAR1、MP2L/MP2H 和 IAR2 根据 MP1H 或 MP2H 寄存器可以访问所有的 Sector。直接寻址通过相关的数据存储器寻址指令来访问所有的数据 Sector。

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

间接寻址程序范例 1

```
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:
:
```

间接寻址程序范例 2

```
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,01h                ; setup the memory sector
    mov mp1h,a
    mov a,offset adres1      ; Accumulator loaded with first RAM address
    mov mp1l,a                ; setup memory pointer with first RAM address
loop:
    clr IAR1                  ; clear the data at address defined by MP1
    inc mp1l                  ; increment memory pointer MP1L
    sdz block                 ; check if last memory location has been cleared
    jmp loop
continue:
```

需注意，上述范例并没有确定 RAM 地址。

使用扩展指令直接寻址程序举例

```
data .section 'data'
temp db ?
code .section at 0 'code'
org 00h
start:
    lmov a,[m]                ; move [m] data to acc
    lsub a,[m+1]              ; compare [m] and [m+1] data
    snz c                     ; [m]>[m+1]?
    jmp continue              ; no
    lmov a,[m]                ; yes, exchange [m] and [m+1] data
    mov temp,a
    lmov a,[m+1]
    lmov [m],a
    mov a,temp
    lmov [m+1],a
continue:
```

注：这里的“m”是位于任何数据存储器 Sector 的某一地址。例如，m=1F0H 表示 Sector 1 中的地址 0F0H。

程序存储区指针 – PBP

该单片机数据存储器被分为几个 Bank，可以通过设置程序存储区指针 PBP 来访问不同的程序存储区。PBP 寄存器应在单片机使用“JMP”或“CALL”指令执行“分支”操作前正确地配置。在分支指令执行后会跳转到一个非连续的程序存储器地址，此地址位于程序存储区指针所选 Bank 内。

● PBP 寄存器 – HT69F360

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	PBP0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 未定义，读为“0”

Bit 0 **PBP0**: 程序存储区选择位
0: Bank 0
1: Bank 1

● PBP 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PBP1	PBP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **PBP1~PBP0**: 程序存储区指针 bit1 ~ bit 0
00: Bank 0
01: Bank 1
10: Bank 2
11: Bank 3

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。程序员对此寄存器进行操作，可以很容易地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指定的程序存储器地址，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。

查表寄存器 – 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 SC 标志位、CZ 标志位、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 标志位外，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、C、SC 和 CZ 标志位通常反映最近运算的状态。

- SC: OV 与当前指令操作结果 MSB 执行“XOR”所得结果。
- CZ: 不同指令不同标志位的操作结果。详细资料请参考寄存器定义部分。
- 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	SC	CZ	TO	PDF	OV	Z	AC	C
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
POR	x	x	0	0	x	x	x	x

“x”：未知

Bit 7 SC: OV 与当前指令操作结果 MSB 执行“XOR”所得结果

Bit 6 CZ: 不同指令不同标志位的操作结果

对于 SUB/SUBM/LSUB/LSUBM 指令，CZ 等于 Z 标志位。

对于 SBC/SBCM/LSBC/LSBCM 指令，CZ 等于上一个 CZ 标志位与当前零标志位执行“AND”所得结果。对于其他指令，CZ 标志位无影响。

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 数据存储器

此系列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的更简单。

EEPROM 数据存储器结构

HT69F340 和 HT69F350 的 EEPROM 数据存储器容量为 64×8，HT69F360 的 EEPROM 容量则为 128×8，而 HT67F370 的 EEPROM 容量则为 256×8。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。使用所有 Sector 中的一个地址和数据寄存器以及 Sector 1 中的一个控制寄存器，可以实现对 EEPROM 的单字节读写操作。

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 位于所有 Sector 中，它们能像其它特殊功能寄存器一样直接被访问。EEC 位于 Sector 1 中，可通过 MP1L/MP1H 或 MP2L/MP2H 存储器指针对和间接寻址寄存器 IAR1 或 IAR2 进行间接读取或写入。由于 EEC 控制寄存器位于 Sector 1 中的“40H”，在 EEC 寄存器上的任何操作被执行前，存储器指针低字节寄存器 MP1L 或 MP2L 必须先设为“40H”，MP1H 或 MP2H 被设为“01H”。

寄存器名称	位							
	7	6	5	4	3	2	1	0
EEA (HT67F370)	D7	D6	D5	D4	D3	D2	D1	D0
EEA (HT69F340/350)	—	—	D5	D4	D3	D2	D1	D0
EEA (HT69F360)	—	D6	D5	D4	D3	D2	D1	D0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

EEPROM 控制寄存器列表

● EEA 寄存器 – HT67F370

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 **D7~D0:** 数据 EEPROM 地址 Bit 7~Bit 0

● EEA 寄存器 – HT69F340/HT69F350

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 **D5~D0:** 数据 EEPROM 地址
数据 EEPROM 地址 Bit 5~Bit 0

● EEA 寄存器 – HT69F360

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	—	0	0	0	0	0	0	0

Bit 7 未定义，读为“0”

Bit 6~0 **D6~D0:** 数据 EEPROM 地址
数据 EEPROM 地址 Bit 6~Bit 0

● EED 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 **D7~D0:** 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: 激活写周期
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: 激活读周期
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 中

写数据至 EEPROM, 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。EEC 寄存器中的写使能位 WREN 先置为高以使能写功能, 然后 EEC 寄存器中的 WR 位需立即置高以开始写操作, 这两条指令必须连续执行。总中断位 EMI 在写周期开始前应当被清零, 写周期开始后再将其使能。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 写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。上电后存储器高字节寄存器 MP1L/MP1H 或 MP2L/MP2H 将重置为“0”, 这意味着数据存储器 Sector 0 被选中。由于 EEPROM 控制寄存器位于 Sector 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 写周期结束后将产生 EEPROM 写中断, 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。更多详情请参见中断章节。

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达到定期保护功能。存储器指针高字节寄存器也可以正常清零以阻止进入 EEPROM 控制寄存器存在的 Sector 1。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。WREN 位置位后，EEC 寄存器中的 WR 位需立即置位，以确保写周期正确地执行。写周期执行前总中断位 EMI 应先清零，在一个有效的写启动步骤完成之后再将此位重新使能。注意，单片机不应在 EEPROM 读或写操作完全完成之前进入空闲或休眠模式，否则 EEPROM 读或写操作将失败。

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, 40H                ; setup memory pointer low byte MP1L
MOV MP1L, A               ; MP1L points to EEC register
MOV A, 01H                ; setup Memory Pointer high byte MP1H
MOV MP1H, 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 write
CLR MP1H
MOV A, EED                ; 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, 40H                ; setup memory pointer low byte MP1L
MOV MP1L, A               ; MP1L points to EEC register
MOV A, 01H                ; setup Memory Pointer high byte MP1H
MOV MP1H, A
CLR EMI
SET IAR1.3                ; set WREN bit, enable write operations
SET IAR1.2                ; start Write Cycle - set WR bit - executed
                        ; immediately after setting WREN bit

SET EMI
BACK:
SZ IAR1.2                 ; check for write cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM write
CLR MP1H
```

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相关的控制寄存器共同完成的。

振荡器概述

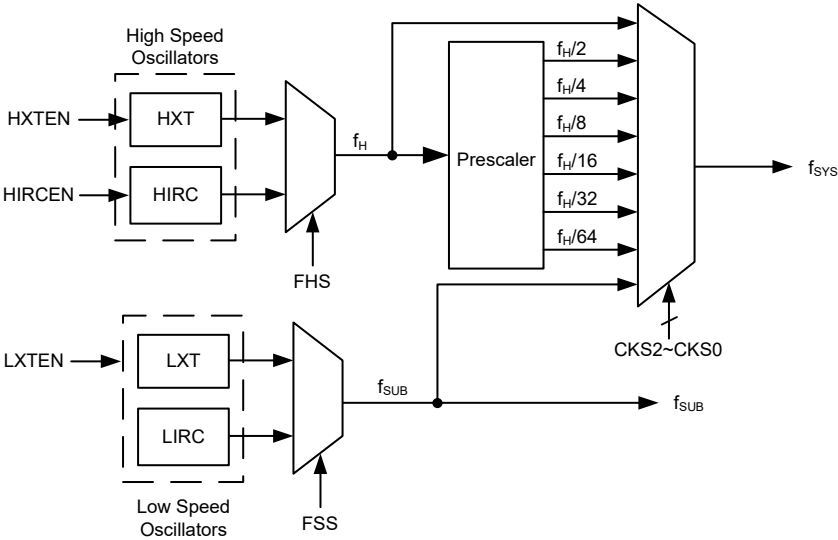
振荡器除了作为系统时钟源，还作为看门狗定时器和时基中断的时钟源。外部振荡器需要一些外围器件，而完全集成的内部振荡器不需要任何外围器件。它们提供的高速和低速系统振荡器具有较宽的频率范围。较高频率的振荡器提供更高的性能，但要求有更高的功率，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活而优化的性能 / 功耗比，此特性对功耗敏感的应用领域尤为重要。

类型	名称	频率	引脚
外部高速晶振	HXT	400kHz~16MHz	OSC1/OSC2
内部高速 RC	HIRC	4/8/12MHz	—
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

该系列单片机有四个系统振荡器，包括两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 HXT 和内部 4/8/12MHz 高速振荡器 HIRC，低速振荡器有内部 32kHz 低速振荡器 LIRC 和外部 32.768kHz 晶体振荡器。使用高速或低速振荡器作为系统时钟的选择是通过设置 SCC 寄存器中的 CKS2~CKS0 位决定的，系统时钟可动态选择。请注意，两个振荡器必须做出选择，即选择高速或选择低速振荡器。

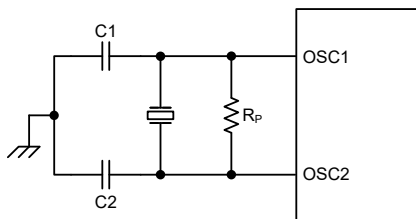


系统时钟配置

外部晶体 / 陶瓷振荡器 – HXT

外部高频晶体 / 陶瓷振荡器作为高频振荡器选项之一，可通过 FHS 位选择，由 HXTEN 位使能。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振或谐振器有关。

为了确保振荡器的稳定性及减少噪声和串扰的影响，晶体振荡器及其相关的电阻和电容以及它们之间的连线都应尽可能地接近单片机。



Note: 1. R_p 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 / 谐振振荡器

晶体振荡器 C1 和 C2 值		
晶体频率	C1	C2
16MHz	0pF	0pF
12MHz	0pF	0pF
8MHz	0pF	0pF
4MHz	0pF	0pF
1MHz	100pF	100pF
注：C1 和 C2 数值仅作参考		

晶体振荡器电容建议值

内部 RC 振荡器 – HIRC

内部 RC 振荡器是一个高度集成的系统振荡器，不需其它外部器件。作为高频振荡器的选项之一，内部 RC 振荡器由 FHS 位选择，由 HIRCEN 位使能，它具有三种固定的频率：4MHz，8MHz 和 12MHz，可通过 HIRCC 寄存器中的 HIRC1~HIRC0 位进行选择。为了确保能达到交流电气特性里描述的精准度，HIRC1~HIRC0 位需要与配置选项中选择频率吻合。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因电源电压、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 5V 及温度为 25°C 的条件下，4MHz、8MHz、12MHz 这三个固定频率的容差为 2%。应注意如果选择了该内部时钟，其操作无需额外的引脚。

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，由 FSS 位对其进行选择，由 LXTEN 位使能。时钟频率固定为 32.768kHz，此时 XT1 和 XT2 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 32.768kHz 晶振以帮助起振。对于那些要求精确频率的应用中，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 LXTEN 位置“1”，LXT 振荡器使能后，其启动需要一定的延时。

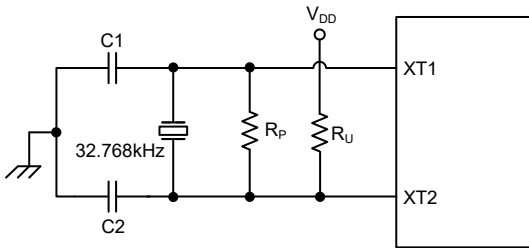
当系统进入休眠或空闲模式，系统时钟关闭以降低功耗。然而很多单片机应用在空闲 / 休眠模式下仍要保持内部定时器功能，此时必须提供额外的时钟。

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。

引脚共用的软件控制位决定 XT1/XT2 脚是用于 LXT 还是作为普通 I/O 口或其它共用功能使用。

- 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 口或其它共用功能使用。
- 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。

为了确保振荡器的稳定性及减少噪声和串扰的影响，晶体振荡器及其相关的电阻和电容以及它们之间的连线都应尽可能地接近单片机。



Note: 1. R_p, R_U, C1 and C2 are required.
2. Although not shown XT1/XT2 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 2. 建议阻值 R _p =5MΩ~10MΩ		

32.768kHz 晶振电容推荐值

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 LXTC 寄存器中的 LXTSP 位进行模式选择。

LXTSP	LXT 工作模式
0	低功耗
1	快速启动

当 LXTSP 位置高，LXT 快速启动模式使能。在这个模式下 LXT 振荡器将很快启动并达到稳定。而在 LXE 振荡器完全启动后，可以通过清零 LXTSP 位将其转入低功耗模式，此时振荡器将继续运行，同时减少电流损耗。需要注意的是 LXT 操作模式的转换必须在 LXT 振荡器时钟选为系统时钟源之前适当地控制。一旦 LXT 振荡器时钟利用 CKS 位段和 SCC 寄存器中的 FSS 位选为系统时钟源，LXT 振荡器的操作模式将无法改变。

需注意的是，不管 LXTSP 设置为什么状态，LXT 振荡器将一直正常运行。唯一的不同是在低功耗状态下 LXT 振荡器需要花更多的时间启动。

内部 32kHz 振荡器 – LIRC

内部 32kHz 系统振荡器也是一个低频振荡器，通过配置 FSS 位进行选择。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°C 温度 5V 电压下的容限保持在 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，也用来为其他单片机功能提供时钟源，如 WDT，LCD 驱动器和时基中断等功能。

工作模式和系统时钟

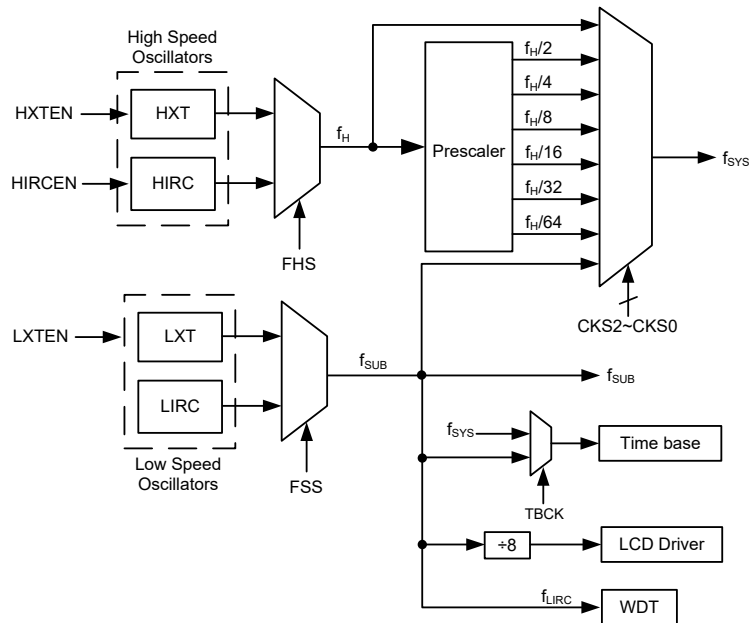
现今的应用要求单片机具有较高的性能及尽可能低的功耗，这种矛盾的要求在便携式电池供电的应用领域尤为明显。高性能所需要的高速时钟将增加功耗，反之亦然。Holtek 单片机了提供高、低速两种时钟源，它们之间可以动态切换，用户可通过优化单片机操作来获得较佳性能 / 功耗比。

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_{SUB} ，通过 SCC 寄存器中的 CKS2~CKS0 位进行选择。高频时钟来自 HXT 或 HIRC 振荡器，可通过配置 FHS 位选择，低频系统时钟源来自 LXT 或 LIRC 振荡器可通过配置 FSS 位选择。其它系统时钟还有高速系统振荡器的分频 $f_H/2 \sim f_H/64$ 。

通过配置相关使能位 HXTEN，HIRCEN 和 LXTEN 可使能 HXT，HIRC 和 LXT 振荡器。当 CPU 进入休眠模式时 LIRC 将关闭，WDT 关闭。



驱动时钟配置

注：当系统时钟源 f_{SYS} 由 f_H 到 f_{SUB} 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 $f_H \sim f_H/64$ 的频率，这通过配置相应的高速振荡器使能控制位确定。

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：空闲模式 0，空闲模式 1，空闲模式 2 以及休眠模式用于单片机 CPU 关闭时以节省耗电。

工作模式	CPU	相关寄存器设定			说明		
		FHIDEN	FSIDEN	CKS2~CKS0	f _{sys}	f _H	f _{SUB}
正常模式	On	x	x	000~110	f _H ~f _H /64	On	On
低速模式	On	x	x	111	f _{SUB}	On/Off ⁽¹⁾	On
空闲模式 0	Off	0	1	000~110	Off	Off	On
				111	f _{SUB}		
空闲模式 1	Off	1	1	xxx	On	On	On
空闲模式 2	Off	1	0	000~110	On	On	On/Off ⁽²⁾
				111	f _{SUB} ⁽²⁾		
休眠模式	Off	0	0	xxx	Off	Off	On/Off ⁽²⁾

“x”：无关

注：1. 在低速模式中，f_H 时钟通过配置相应的振荡器使能位控制其启动和关闭。
2. 在空闲模式 2 或休眠模式中，f_{SUB} 时钟通过 WDT 功能的使能或除能来控制其启动和关闭。

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 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 SCC 寄存器中的 CKS2~CKS0 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 f_{SUB}，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。

休眠模式

在 HALT 指令执行后且 FHIDEN 和 FSIDEN 位都为低时，系统进入休眠模式。在休眠模式中，CPU 停止运行，而 f_{SUB} 时钟的开启或关闭由 WDT 功能的使能或除能控制。

空闲模式 0

执行 HALT 指令后且 SCC 寄存器中 FSIDEN 位为高，SCC 寄存器中的 FSIDEN 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和 TM 将继续工作。在空闲模式 0 中，系统振荡器停止，但低速振荡器将开启以驱动一些外围功能。

空闲模式 1

执行 HALT 指令后且 SCC 寄存器中 FHIDEN 位为高，FSIDEN 位为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但高速和低速振荡器都将继续开启以为一些外部功能操作提供时钟源。

空闲模式 2

执行 HALT 指令后且 SCC 寄存器中的 FHIDEN 位为高，FSIDEN 位为低时，系统进入空闲模式 2。在空闲模式 2 中，CPU 停止，而高速振荡器将开启为一些外围功能操作提供时钟源。若 WDT 使能， f_{SUB} 将继续开启为 WDT 提供时钟。若 WDT 除能，则 f_{SUB} 也将关闭。

控制寄存器

SCC 寄存器用于单片机中内部时钟的整体控制。HXTC 寄存器用于 HXT 振荡器控制，而 HIRCC 则用于 HIRC 振荡器控制。

• SCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	FHS	FSS	FHIDEN	FSIDEN
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 7~5 **CKS2~CKS0:** 系统时钟选择位

000: f_H
001: $f_H/2$
010: $f_H/4$
011: $f_H/8$
100: $f_H/16$
101: $f_H/32$
110: $f_H/64$
111: f_{SUB}

这三位用于选择系统时钟源。除了以 f_H 或 f_{SUB} 作为系统时钟源外，也可使用高频振荡器的分频作为系统时钟。

Bit 4 未定义，读为“0”

Bit 3 **FHS:** 高频率时钟选择位

0: HIRC
1: HXT

Bit 2 **FSS:** 低频率时钟选择位

0: LIRC
1: LXT

Bit 1 **FHIDEN:** CPU 关闭时高频振荡器控制位

0: 除能
1: 使能

Bit 0 **FSIDEN:** CPU 关闭时低频振荡器控制位

0: 除能
1: 使能

CPU 在执行 HALT 指令后关闭，此时利用 FSIDEN 来控制低速振荡器的启动或停止。当 LIRC 振荡器被选为低速振荡器，其也将通过此位以及 WDT 功能使能控制位共同控制。当此位清零，而 WDT 功能使能，LIRC 振荡器将使能。

注：使用 CKS2~CKS0 位、FHS 位或 FSS 位进行时钟切换设置之后，在相关时钟成功切换至目标时钟源之前需要一定的延时。因此，若接下来执行的操作需要目标时钟源立即响应，则在此之前必须规划适当的延迟时间。

时钟切换延迟时间 = $4 \times t_{SYS} + [0 \sim (1.5 \times t_{CURR} + 0.5 \times t_{TAR})]$ ，其中 t_{CURR} 指代当前的时钟周期， t_{TAR} 指代目标时钟周期， t_{SYS} 指代当前系统时钟周期。

● HXTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	HXTM	HXTF	HXTEN
R/W	—	—	—	—	—	R/W	R	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 未定义，读为“0”

Bit 2 **HXTM**: HXT 模式选择位

0: HXT 频率 ≤ 10MHz (灌电流 / 源电流较小)

1: HXT 频率 > 10MHz (灌电流 / 源电流较大)

注意，此位须根据所使用的 HXT 频率正确设置。若 HXTM=0 而 HXT 频率大于 10MHz，则低压时振荡性能可能不佳。若 HXTM=1 而 HXT 频率小于 10MHz，则振荡频率和电流可能异常。

此位必须在 HXT 使能前正确地配置。当 OSC1 和 OSC2 引脚功能已通过相关引脚共用控制位使能，且 HXTEN 位已置高使能 HXT 振荡器，此时再改变 HXTM 设定值是无效的。若 OSC1 或 OSC2 引脚功能除能，此时无论 HXTEN 位为何值，可通过软件改写 HXTM 位值。

Bit 1 **HXTF**: HXT 时钟稳定标志位

0: 不稳定

1: 稳定

此位用于说明 HXT 振荡器是否稳定。当 HXTEN 位置高以使能 HXT 振荡器，HXTF 位将清零，并在 HXT 时钟稳定后置高。

Bit 0 **HXTEN**: HXT 振荡器使能控制位

0: 除能

1: 使能

● HIRCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	HIRC1	HIRC0	HIRCF	HIRCEN
R/W	—	—	—	—	R/W	R/W	R	R/W
POR	—	—	—	—	0	0	0	1

Bit 7~4 未定义，读为“0”

Bit 3~2 **HIRC1~HIRC0**: HIRC 频率选择位

00: 4MHz

01: 4MHz

10: 8MHz

11: 12MHz

当 HIRC 振荡器使能，且 HIRC 频率通过应用程序改变，时钟频率将在 HIRCF 标志位置 1 后自动改变。

建议这里选择的频率与配置选项中选定的频率保持一致，以确保能够达到交流电气特性中标示的 HIRC 频率精准度。

Bit 1 **HIRCF**: HIRC 时钟稳定标志位

0: 不稳定

1: 稳定

此位用于说明 HIRC 振荡器是否稳定。当 HIRC 频率交换，CPU 将停止 16 个 HIRC 时钟。当 HIRCF 位置高以使能 HIRC 振荡器，此位将被清零，并在 HIRC 时钟稳定后置高。在 HIRCF 位使能后 HIRC 稳定时间需要 16 个时钟。

Bit 0 **HIRCEN**: HIRC 振荡器使能控制位

0: 除能

1: 使能

• LXTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LXTSP	LXTF	LXTEN
R/W	—	—	—	—	—	R/W	R	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 未定义，读为“0”

Bit 2 **LXTSP**: LXT 快速启动控制位

- 0: 除能 – 低功耗
- 1: 使能 – 快速启动

此位用于控制 LXT 振荡器工作模式为低功耗或快速启动模式。当 LXTSP 位置高，LXT 振荡器将快速振荡，但需要更大的功耗。若 LXTSP 位清零，LXT 振荡器所产生的功耗较少，但需要较长时间才可以稳定。需要注意的是在 LXT 振荡器通过配置 SCC 寄存器中的 CKS1~CKS0 以及 FSS 位被选择系统时钟源时，此位无法改变。

Bit 1 **LXTF**: LXT 时钟稳定标志位

- 0: 不稳定
- 1: 稳定

此位用于说明 LXT 振荡器是否稳定。当 LXTEN 位置高使能 LXT 振荡器，此位将清零，并在 LXT 时钟稳定后置高。当 LXT 使能，其稳定时间需要 1024 的时钟。

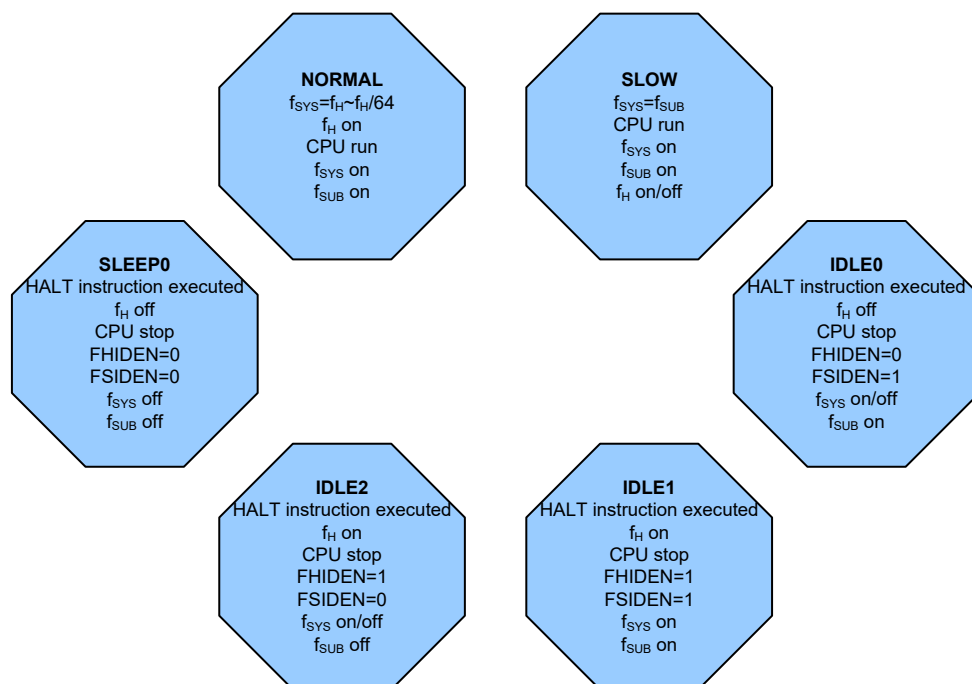
Bit 0 **LXTEN**: LXT 振荡器使能控制位

- 0: 除能
- 1: 使能

工作模式转换

该系列单片机可在操作模式之间动态切换，使用户得以根据手中应用的要求选择较佳的性能 / 功率比率。如此一来，无需高性能的单片机操作便可使用较低速的时钟，从而减少工作电流，延长移动应用的电池寿命。

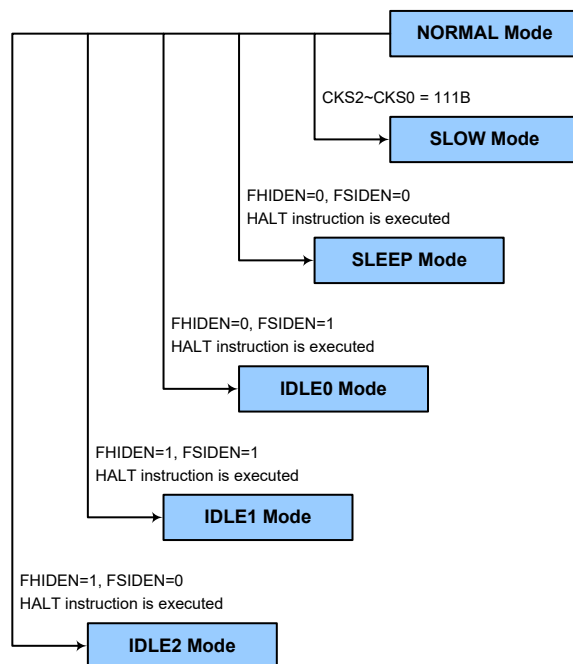
简而言之，正常模式和低速模式之间的转换通过配置 SCC 寄存器中的 CKS2~CKS0 位来执行，而从正常 / 低速模式切换到空闲模式则通过执行 HALT 指令实现。当执行 HALT 指令，由 SCC 寄存器中的 FHIDEN 和 FSIDEN 位的状态来决定单片机应进入空闲模式 0，空闲模式 1，空闲模式 2 或休眠模式。



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 SCC 寄存器中的 CKS2~CKS0 位为“111”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。

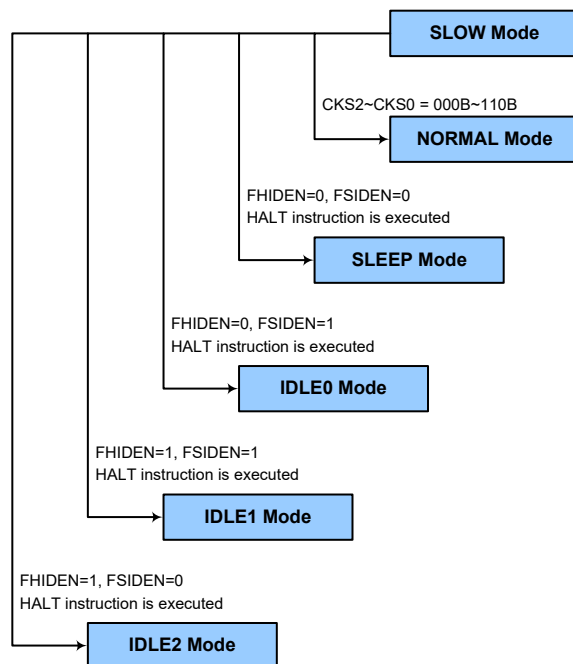
低速模式的时钟源来自 LXT 或 LIRC 振荡器，因此要求这些振荡器在所有模式切换动作发生前稳定下来。



低速模式切换到正常模式

在低速模式中系统时钟来自低速时钟 f_{SUB} 。当系统时钟切换到高速系统时钟振荡器的正常模式需将 $CKS[2:0]$ 设为“000”~“110”之间的值，此时系统时钟将被切换到 $f_H \sim f_H/64$ 。

但若 f_H 时钟未在低速模式中被用到，则需要一定的时间来重振并达到稳定。这由 HXTC 寄存器中的 HXTF 位或 HIRCC 寄存器中的 HIRCF 位监测。高速系统时钟振荡器稳定所需的时钟数量取决于所使用的高速系统振荡器类型。



进入休眠模式

进入休眠模式的方法仅有一种——在应用程序中执行“HALT”指令且设置寄存器 SCC 中的 FHIDEN 位和 FSIDEN 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。
- 数据存储器中的内容和寄存器将保持当前值。
- 输入 / 输出将保持当前值。
- 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。
- 若 WDT 功能使能，WDT 将被清除并重新开始计数。若 WDT 功能除能，WDT 将被清除并停止运行。

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”指令且设置寄存器 SCC 中的 FHIDEN 位为“0”，FSIDEN 位为“1”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 时钟将关闭， f_{SUB} 时钟继续运行，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。
- 输入 / 输出将保持当前值。

- 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。
- 若 WDT 功能使能，WDT 将被清除并重新开始计数。若 WDT 功能除能，WDT 将被清除并停止运行。

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”指令且设置寄存器 SCC 中的 FHIDEN 位为“1”，FSIDEN 位为“1”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 时钟和 f_{SUB} 时钟继续运行，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。
- 输入 / 输出将保持当前值。
- 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。
- 若 WDT 功能使能，WDT 将被清除并重新开始计数。若 WDT 功能除能，WDT 将被清除并停止运行。

进入空闲模式 2

进入空闲模式 1 的方法仅有一种——应用程序中执行“HALT”指令且设置寄存器 SCC 中的 FHIDEN 位为“1”，FSIDEN 位为“0”。在上述条件下执行该指令后，将发生的情况如下：

- f_H 时钟继续运行， f_{SUB} 时钟关闭，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。
- 输入 / 输出将保持当前值。
- 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。
- 若 WDT 功能使能，WDT 将被清除并重新开始计数。若 WDT 功能除能，WDT 将被清除并停止运行。

待机电流的注意事项

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，可能到只有几个微安的级别（空闲模式 1 和空闲模式 2 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 LXT 或 LIRC 振荡器，会导致耗电增加。

在空闲模式 1 和空闲模式 2 中，高速振荡器开启。若系统时钟来自高速系统振荡器，额外的待机电流也可能会有几百微安。

唤醒

单片机进入休眠模式或空闲模式后，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。

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

若由外部复位唤醒，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。

看门狗定时器

看门狗定时器的功能在于防止如电噪声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时钟 f_s ，而时钟 f_s 又由 f_{SUB} 时钟提供，通过配置 SCC 寄存器中的 FSS 位选择时钟 f_{SUB} 时钟的来源为 LXT 或 LIRC。LIRC 内部振荡器频率约为 32kHz 且这个指定的内部时钟周期随 V_{DD} 、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。看门狗定时器的时钟源可分频为 $2^8 \sim 2^{18}$ 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。

看门狗定时器控制寄存器

定时器所需溢出周期以及使能 / 除能操作由 WDT 寄存器控制。该寄存器控制看门狗定时器的所有操作。

● WDTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0**: WDT 功能软件控制位

10101: 除能
01010: 使能
其它值: MCU 复位

若这些位受不利的环境因素影响，单片机将复位。复位操作将在 2~3 个 LIRC 时钟周期后启动，且 SMOD1 寄存器中的 WRF 位将置一。

Bit 2~0 **WS2~WS0:** WDT 溢出周期选择位
000: $2^8/f_{SUB}$
001: $2^{10}/f_{SUB}$
010: $2^{12}/f_{SUB}$
011: $2^{14}/f_{SUB}$
100: $2^{15}/f_{SUB}$
101: $2^{16}/f_{SUB}$
110: $2^{17}/f_{SUB}$
111: $2^{18}/f_{SUB}$

这三位确定了看门狗定时器时钟源的分频比，也就是确定了溢出周期。

● **SMOD1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	LRF	WRF
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	x	0	0

“x”：未知

Bit 7~3 未定义，读为“0”
Bit 2 **LVRF:** LVR 功能复位标志位
 详见其他章节
Bit 1 **LRF:** LVRC 寄存器软件复位标志位
 详见其他章节
Bit 0 **WRF:** WDT 寄存器软件复位标志位
 0: 未发生
 1: 发生

此位通过 WDT 控制寄存器软件复位置一，由应用程序清零。注意此位只能由应用程序清零。

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除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。

看门狗定时器控制寄存器 WDTC 中有五个位 WE4~WE0 可提供使能 / 除能控制以及控制看门狗定时器复位操作。当 WE4~WE0 设置为“10101B”时除能 WDT 功能，而当设置为“01010B”时使能 WDT 功能。如果 WE4~WE0 设置为除“01010B”和“10101B”以外的值时，单片机将在 2~3 个 f_{SUB} 时钟周期后复位。上电后这些位初始化为“01010B”。

WE4~WE0 位	WDT 功能
10101B	除能
01010B	使能
其它值	MCU 复位

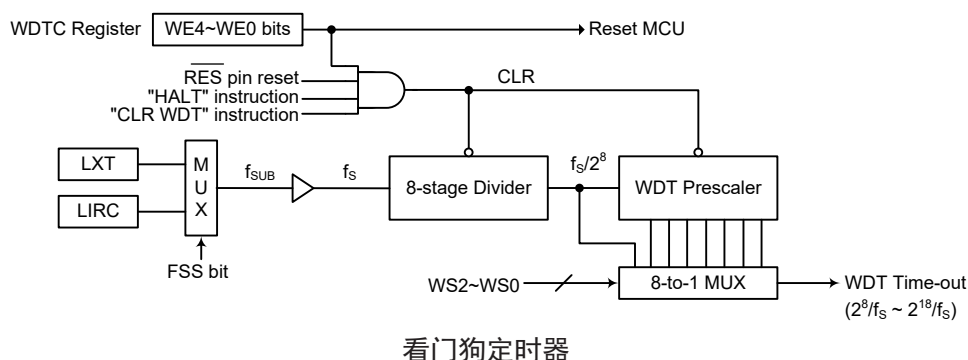
看门狗定时器使能 / 除能控制

在正常编程操作下，看门狗定时器溢出将启动单片机复位，并置位状态位 TO。但若系统工作在休眠或空闲模式，在发生看门狗定时器溢出时，状态寄存器中的 TO 和 PDF 位将被设置，只有程序计数器和堆栈指针将被复位。有四种方法可以清除看门狗定时器的内容。

第一种是 WDT 复位，即将除 01010B 和 10101B 外的任意值写入 WE4~WE0 位段。第二种是利用看门狗定时器软件清除指令。第三种是通过执行 HALT 指令。而第四种则是利用外部硬件复位，即在 RES 引脚上加上低电平。

只有一种方法可以利用软件指令清除看门狗定时器，那就是使用一个“CLR WDT”指令来清除 WDT。

当选择 2^{18} 分频比时溢出周期最大。举例来说，当以 32kHz LIRC 振荡器作为时钟源，分频比为 2^{18} 时，可得约为 8 秒的看门狗最大溢出周期，而分频比为 2^8 时得到约为 7.8ms 的最小溢出周期。



复位和初始化

复位功能是整个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定好的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，RES 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。

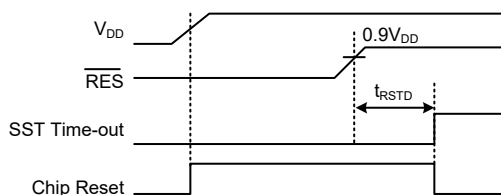
另一种复位为看门狗溢出单片机复位。不同方式的复位操作会对寄存器产生不同的影响。另一种复位为低电压复位即 LVR 复位，在电源供应电压低于 LVR 设定值时，系统会产生 LVR 复位，这种复位与 RES 脚拉低复位方式相似。

复位功能

包括内部和外部事件触发复位，单片机共有以下几种复位方式：

上电复位

这是最基本且不可避免的复位，发生在单片机首次上电后。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。



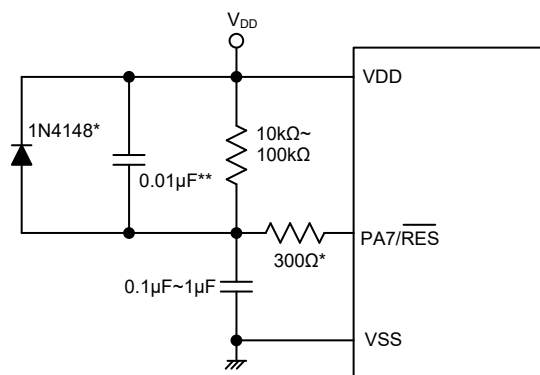
上电复位时序图

RES 引脚复位

由于复位引脚与 I/O 口共用，复位功能必须通过配置选项选择尽管单片机自带内部 RC 复位功能，若在上电时 V_{DD} 供电上升时间不够快或不能快速稳定，内部复位功能可能无法提供复位操作因此建议使用连接到 RES 引脚的外部 RC 网络，其附加时间延迟将确保 RES 在一个扩展周期内保持低电平，使供电可以达到稳定。在此期间，单片机的正常操作将被禁止。在 RES 线达到一定电压值时，复位延时时间 t_{RST} 将被调用以提供一个额外的延迟时间，而后单片机开始进行正常操作。图中的 SST 为系统启动定时器 (System Start-up Timer) 的缩写。

对大多数应用来说，在 V_{DD} 和 RES 引脚之间连接一个电阻，且 VSS 和 RES 引脚间连接一个电容将提供一个合适的外部复位电路。任何连接到 RES 的线应尽可能短以将杂散噪声干扰较大程度地降低。

对于工作在较多噪声环境下的应用，建议使用下图的增强型复位电路。



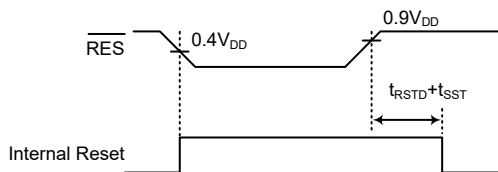
外部 RES 电路

注：* 建议加上此部分以增强 ESD 保护。

** 建议在电源有较强干扰场合加上此部分。

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。

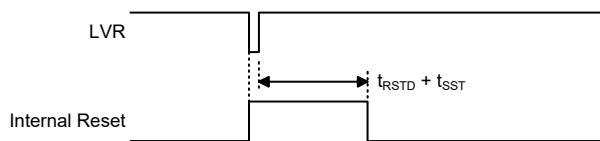
RES 引脚通过外部硬件强制拉至低电平时，此种复位形式即会发生。这种复位方式和其它的复位方式一样，程序计数器会被清除为零且程序从头开始执行。



RES 复位时序图

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。在正常模式和低速模式工作在一个特定的 LVR 电压 V_{LVR} 下，通过写一个相应的值到 LVRC 寄存器中可以使能 LVR 功能。例如在更换电池的情况下，单片机供应的电压可能会落在 $0.9V \sim V_{LVR}$ 的范围内，这时 LVR 将会自动复位单片机且 SMOD1 寄存器的 LVRF 位也被设置为“1”。有效的 LVR 信号，即在 $0.9V \sim V_{LVR}$ 的低电压状态的时间，必须超过 LVD/LVR 电气特性中 t_{LVR} 参数的值。如果低电压存在时间不超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。实际的 V_{LVR} 参数值可通过 LVRC 寄存器中的 LVS 位进行选择。如果由于不利的环境干扰因素使得 LVS7~LVS0 为其它值，LVR 将在 2~3 个 f_{SUB} 时钟周期后复位单片机。此时，SMOD1 寄存器中的 LRF 位被设置为“1”。寄存器 LVRC 上电后初始化为“01100110B”。当单片机进入省电模式时 LVR 功能将自动除能。



低电压复位时序图

• LVRC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
POR	0	1	1	0	0	1	1	0

Bit 7~0 **LVS7~LVS0**: LVR 电压选择控制位

01100110: 1.7V (默认值)

01010101: 1.9V

00110011: 2.55V

10011001: 3.15V

10101010: 3.8V

11110000: LVR 除能

其它值: 复位单片机—寄存器复位为 POR 值。

当低电压条件发生时，以上五个已定义的任何 LVR 电压值都会使单片机复位。复位动作将在 2~3 个 f_{SUB} 时钟周期后有效。此时复位后的寄存器内容将保持不变。

任何除上述五个已定义的寄存器值，也会导致单片机复位。复位动作将在 2~3 个 f_{SUB} 时钟周期后有效。但此时寄存器内容将复位为 POR 值。

• SMOD1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	LRF	WRF
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	x	0	0

“x”: 未知

Bit 7~3 未定义，读为“0”

Bit 2 **LVRF**: LVR 功能复位标志位

0: 未发生

1: 发生

此位在发生指定低电压复位的情况下置一。且只能由用户程序清零。

- Bit 1

LRF: LVRC 寄存器软件复位标志位

0: 未发生

1: 发生

若 LVRC 寄存器中包含任何未定义的 LVR 寄存器值，此位置一。其功能与软件复位相似。此位只能由应用程序清零。
- Bit 0

WRF: WDTIC 寄存器软件复位标志位

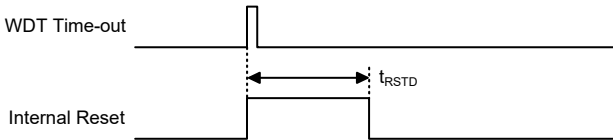
详情参见其他章节

IAP 复位

当写值“55H”至 FC1 寄存器时，将产生一个复位信号将整个单片机复位。详见在线应用编程章节。

正常运行下的看门狗溢出复位

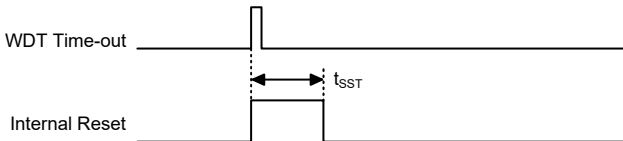
除了看门狗溢出标志位 TO 将被设为“1”之外，正常运行时看门狗溢出复位和 RES 复位相同。



正常运行时看门狗溢出时序图

休眠或空闲模式下的看门狗溢出复位

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 和 PDF 位被设为“1”外，绝大部分的条件保持不变。图中 tSST 的详细说明请参考交流电气特性。



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定时器等几种控制器操作控制。复位标志位如下所示：

TO	PDF	复位条件
0	0	上电复位
u	u	正常模式或低速模式时的 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”表示未改变

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, 时基	复位后清零, WDT 重新开始计数
定时器模块	所有定时器模块关闭
输入 / 输出	I/O 口设为输入模式
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。注意若芯片有多种封装类型, 表格反映较大的封装的情况。

HT69F340

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu
STATUS	xx00 xxxx	uuuu uuuu	uu01 uuuu	xx1u uuuu	uu11 uuuu
IAR2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SMOD1	---- -x00	---- -x00	---- -x00	---- -x00	---- -uuu
LVRC	0110 0110	0110 0110	0110 0110	0110 0110	uuuu uuuu
MFI0	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
INTC2	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
PB	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PBC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PCPU	---- -000	---- -000	---- -000	---- -000	---- -uuu
PC	---- -111	---- -111	---- -111	---- -111	---- -uuu
PCC	---- -111	---- -111	---- -111	---- -111	---- -uuu
PDPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PE	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PF	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAFS	0000 -000	0000 -000	0000 -000	0000 -000	uuuu -uuu
PBFS	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
PDFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
SFS	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDH	---- --00	---- --00	---- --00	---- --00	---- --uu
CTMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMAH	---- --00	---- --00	---- --00	---- --00	---- --uu
EEA	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
EED	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMC0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
PTMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMDH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMAH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTMRPL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMRPH	---- --00	---- --00	---- --00	---- --00	---- --uu
SCC	000- 0000	000- 0000	000- 0000	000- 0000	uuu- uuuu
HIRCC	---- 0001	---- 0001	---- 0001	---- 0001	---- uuuu
HXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu
LXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
LVDC	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu
LCDC0	0-0- 0000	0-0- 0000	0-0- 0000	0-0- 0000	u-u- uuuu
LCDC1	---- -000	---- -000	---- -000	---- -000	---- -uuu
SIMC0	111- 0000	111- 0000	111- 0000	111- 0000	uuu- uuuu
SIMC0	111- 0000	111- 0000	111- 0000	111- 0000	uuu- uuuu
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
SIMA	0000 000-	0000 000-	0000 000-	0000 000-	uuuu uuu-
SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SIMTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC0	0111 0000	0111 0000	0111 0000	0111 0000	uuuu uuuu
FC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FARL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FARH	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
FD0L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD0H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu

HT69F350

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---x xxxx	---u uuuu	---u uuuu	---u uuuu	---u uuuu
STATUS	xx00 xxxx	uuuu uuuu	uu01 uuuu	xx1u uuuu	uu11 uuuu
IAR2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
MP2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SMOD1	---- -x00	---- -x00	---- -x00	---- -x00	---- -uuu
LVRC	0110 0110	0110 0110	0110 0110	0110 0110	uuuu uuuu
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
INTC2	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	---- -000	---- -000	---- -000	---- -000	---- -uuu
PC	---- -111	---- -111	---- -111	---- -111	---- -uuu
PCC	---- -111	---- -111	---- -111	---- -111	---- -uuu
PDPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PE	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PF	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PG	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHPU	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
PH	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PHC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PAFS	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBFS	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PDFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
PFFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHFS	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
SFSR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDH	---- --00	---- --00	---- --00	---- --00	---- --uu
CTMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMAH	---- --00	---- --00	---- --00	---- --00	---- --uu
EEA	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu
EED	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
PTM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0AH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM0RPL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0RPH	---- --00	---- --00	---- --00	---- --00	---- --uu
STMC0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
STMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMRP	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SCC	000- 0000	000- 0000	000- 0000	000- 0000	uuu- uuuu
HIRCC	---- 0001	---- 0001	---- 0001	---- 0001	---- uuuu
HXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu
LXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu
LVDC	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu
LCDC0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	u-u- uuuu
LCDC1	---- -000	---- -000	---- -000	---- -000	---- -uuu
SIMC0	111- 0000	111- 0000	111- 0000	111- 0000	uuu- uuuu
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
SIMA	0000 000-	0000 000-	0000 000-	0000 000-	uuuu uuu-

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SIMTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC0	0111 0000	0111 0000	0111 0000	0111 0000	uuuu uuuu
FC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC2	---- ---0	---- ---0	---- ---0	---- ---0	---- ---u
FARL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FARH	---0 0000	---0 0000	---0 0000	---0 0000	---u uuuu
FD0L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD0H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
PTM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1AH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM1RPL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1RPH	---- --00	---- --00	---- --00	---- --00	---- --uu
EEC	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu

HT69F360

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
STATUS	xx00 xxxx	uuuu uuuu	uu01 uuuu	xx1u uuuu	uu11 uuuu
PBP	---- ---0	---- ---0	---- ---0	---- ---0	---- ---u
IAR2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
MP2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SMOD1	---- -x00	---- -x00	---- -x00	---- -x00	---- -uuu
LVRC	0110 0110	0110 0110	0110 0110	0110 0110	uuuu uuuu
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI2	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
INTC2	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
PC	-111 1111	-111 1111	-111 1111	-111 1111	-uuu uuuu
PCC	-111 1111	-111 1111	-111 1111	-111 1111	-uuu uuuu
PDPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PE	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PF	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PG	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PH	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAFS	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBFS	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PDFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
PHFS	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
SFSR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMDH	---- --00	---- --00	---- --00	---- --00	---- --uu
CTMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMAH	---- --00	---- --00	---- --00	---- --00	---- --uu
EEA	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
EED	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
USR	0000 1011	0000 1011	0000 1011	0000 1011	uuuu uuuu
UCR1	0000 00x0	0000 00x0	0000 00x0	0000 00x0	uuuu uuuu
UCR2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
BRG	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TXR_RXR	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
PTM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
PTM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0AH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM0RPL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0RPH	---- --00	---- --00	---- --00	---- --00	---- --uu
SFSR1	---- -000	---- -000	---- -000	---- -000	---- -uuu
STMC0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
STMC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
STM RP	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SCC	000- 0000	000- 0000	000- 0000	000- 0000	uuu- uuuu
HIRCC	---- 0001	---- 0001	---- 0001	---- 0001	---- uuuu
HXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu
LXTC	---- -000	---- -000	---- -000	---- -000	---- -uuu
LVDC	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu
LCDC0	0-0- 0000	0-0- 0000	0-0- 0000	0-0- 0000	u-u- uuuu
LCDC1	---- -000	---- -000	---- -000	---- -000	---- -uuu
SIMC0	111- 0000	111- 0000	111- 0000	111- 0000	uuu- uuuu
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu

寄存器	上电复位	RES 或 LVR 复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (正常模式)	WDT 溢出 (HALT)
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
SIMA	0000 000-	0000 000-	0000 000-	0000 000-	uuuu uu-
SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
SIMTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC0	0111 0000	0111 0000	0111 0000	0111 0000	uuuu uuuu
FC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC2	---- --0	---- --0	---- --0	---- --0	---- --u
FARL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FARH	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu
FD0L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD0H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3L	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---
PTM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1AH	---- --00	---- --00	---- --00	---- --00	---- --uu
PTM1RPL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1RPH	---- --00	---- --00	---- --00	---- --00	---- --uu
EEC	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu

HT67F370

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常运行)	WDT 溢出 (空闲/休眠)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1L	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP1H	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBHP	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu
STATUS	xx00 xxxx	uuuu uuuu	xx1u uuuu	uu1l uuuu
PBP	---- --00	---- --00	---- --00	---- --uu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常运行)	WDT 溢出 (空闲 / 休眠)
IAR2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP2L	0000 0000	0000 0000	0000 0000	uuuu uuuu
MP2H	0000 0000	0000 0000	0000 0000	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
SMOD1	---- -x00	---- -x00	---- -x00	---- -uuu
LVRC	0110 0110	0110 0110	0110 0110	uuuu uuuu
MFI0	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI2	--00 --00	--00 --00	-00 --00	--uu --uu
INTC2	-000 -000	-000 -000	-000 -000	-uuu -uuu
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	-000 0000	-000 0000	-000 0000	-uuu uuuu
PC	-111 1111	-111 1111	-111 1111	-uuu uuuu
PCC	-111 1111	-111 1111	-111 1111	-uuu 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	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	0000 0000	0000 0000	0000 0000	uuuu uuuu
PG	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHPU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PH	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHC	1111 1111	1111 1111	1111 1111	1111 1111
PAFS	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBFS	0000 0000	0000 0000	0000 0000	uuuu uuuu
PCFS	-000 0---	-000 0---	-000 0---	-uuu u---

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常运行)	WDT 溢出 (空闲 / 休眠)
PDFS	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEFS	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFFS	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGFS	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHFS	1111 1111	1111 1111	1111 1111	uuuu uuuu
SFSR	0000 0000	0000 0000	0000 0000	uuuu uuuu
CTMC0	0000 0000	0000 0000	0000 0000	Uuuu uuuu
CTMC1	0000 0000	0000 0000	0000 0000	Uuuu uuuu
CTMDL	0000 0000	0000 0000	0000 0000	Uuuu uuuu
CTMDH	---- --00	---- --00	---- --00	---- --uu
CTMAL	0000 0000	0000 0000	0000 0000	Uuuu uuuu
CTMAH	---- --00	---- --00	---- --00	---- --uu
EEA	0000 0000	0000 0000	0000 0000	uuuu uuuu
EED	0000 0000	0000 0000	0000 0000	uuuu uuuu
USR	0000 1011	0000 1011	0000 1011	uuuu uuuu
UCR1	0000 00x0	0000 00x0	0000 00x0	uuuu uuuu
UCR2	0000 0000	0000 0000	0000 0000	uuuu uuuu
BRG	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TXR_RXR	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
PTM0C0	0000 0---	0000 0---	0000 0---	uuuu u---
PTM0C1	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0DH	---- --00	---- --00	---- --00	---- --uu
PTM0AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0AH	---- --00	---- --00	---- --00	---- --uu
PTM0RPL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM0RPH	---- --00	---- --00	---- --00	---- --uu
SFSR1	--00 0000	--00 0000	--00 0000	--uu uuuu
STMC0	0000 0---	0000 0---	0000 0---	uuuu u---
STMC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDH	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAH	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMRP	0000 0000	0000 0000	0000 0000	uuuu uuuu
SADC2	0--0 0000	0--0 0000	0--0 0000	u--u uuuu
SCC	000- 0000	000- 0000	000- 0000	uuu- uuuu
HIRCC	---- 0001	---- 0001	---- 0001	---- uuuu
HXTC	---- -000	---- -000	---- -000	---- -uuu
LXTC	---- -000	---- -000	---- -000	---- -uuu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常运行)	WDT 溢出 (空闲/休眠)
LVDC	--00 0000	--00 0000	--00 0000	--uu uuuu
INTEG	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	Uuuu uuuu
LCDC0	0-0- 0000	0-0- 0000	0-0- 0000	u-u- uuuu
LCDC1	---- -000	---- -000	---- -000	---- -uuu
SIMC0	111- 0000	111- 0000	111- 0000	uuu- uuuu
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
SIMTOC	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC0	0111 0000	0111 0000	0111 0000	uuuu uuuu
FC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
FC2	---- ---0	---- ---0	---- ---0	---- ---u
FARL	0000 0000	0000 0000	0000 0000	uuuu uuuu
FARH	--00 0000	--00 0000	--00 0000	--uu uuuu
FD0L	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD0H	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1L	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD1H	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2L	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD2H	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3L	0000 0000	0000 0000	0000 0000	uuuu uuuu
FD3H	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTMIC0	0000 0---	0000 0---	0000 0---	uuuu u---
PTMIC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1DH	---- --00	---- --00	---- --00	---- --uu
PTM1AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1AH	---- --00	---- --00	---- --00	---- --uu
PTM1RPL	0000 0000	0000 0000	0000 0000	uuuu uuuu
PTM1RPH	---- --00	---- --00	---- --00	---- --uu
SADOL	xxxx ----	xxxx ----	xxxx ----	uuuu ---- (ADRFS=0)
				uuuu uuuu (ADRFS=1)
SADOH	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRFS=0)
				---- uuuu (ADRFS=1)
SADC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
SADC1	0000 -000	0000 -000	0000 -000	uuuu -uuu

寄存器	上电复位	RES 复位 (正常运行)	WDT 溢出 (正常运行)	WDT 溢出 (空闲/休眠)
EEC	---- 0000	---- 0000	---- 0000	---- uuuu

注：“-”表示未定义
“u”表示不改变
“x”表示未知

输入 / 输出端口

Holtek 单片机的输入 / 输出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~PH 双向输入 / 输出。这些 I/O 端口映射到 RAM 数据存储器中的特定地址上，如特殊功能数据存储器表所示。所有 I/O 口可用于输入输出操作。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]”时 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。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	—	—	—	—	PBPU3	PBPU2	PBPU1	PBPU0
PB	—	—	—	—	PB3	PB2	PB1	PB0
PBC	—	—	—	—	PBC3	PBC2	PBC1	PBC0
PCPU	—	—	—	—	—	PCPU2	PCPU1	PCPU0
PC	—	—	—	—	—	PC2	PC1	PC0
PCC	—	—	—	—	—	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0

I/O 逻辑功能寄存器列表 – HT69F340

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	—	—	—	—	—	PCPU2	PCPU1	PCPU0
PC	—	—	—	—	—	PC2	PC1	PC0
PCC	—	—	—	—	—	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0
PGPU	PGPU7	PGPU6	PGPU5	PGPU4	PGPU3	PGPU2	PGPU1	PGPU0
PG	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
PGC	PGC7	PGC6	PGC5	PGC4	PGC3	PGC2	PGC1	PGC0
PHPU	—	—	—	—	PHPU3	PHPU2	PHPU1	PHPU0
PH	—	—	—	—	PH3	PH2	PH1	PH0
PHC	—	—	—	—	PHC3	PHC2	PHC1	PHC0

I/O 逻辑功能寄存器列表 – HT69F350

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	—	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	—	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PCC	—	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0
PGPU	PGPU7	PGPU6	PGPU5	PGPU4	PGPU3	PGPU2	PGPU1	PGPU0
PG	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
PGC	PGC7	PGC6	PGC5	PGC4	PGC3	PGC2	PGC1	PGC0
PHPU	PHPU7	PHPU6	PHPU5	PHPU4	PHPU3	PHPU2	PHPU1	PHPU0
PH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
PHC	PHC7	PHC6	PHC5	PHC4	PHC3	PHC2	PHC1	PHC0

I/O 逻辑功能寄存器列表 – HT69F360/HT67F370

“—”：未定义，读为“0”

PAWUn: 唤醒功能控制位

- 0: 除能
- 1: 使能

PAn/PBn/PCn/PDn/PEn/PFn/PGn/PHn: I/O 数据位

- 0: 数据 0
- 1: 数据 1

PACn/PBCn/PCCn/PDCn/PECn/PFCn/PGCn/PHCn: I/O 类型选择

- 0: 输出
- 1: 输入

PAPUn/PBPUn/PCPUn/PDPUn/PEPUn/PFPUn/PGPUn/PHPUn: 上拉功能控制

- 0: 除能
- 1: 使能

上拉电阻

许多产品应用在端口处于输入状态时，通常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~PHPU 来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。

该系列单片机提供了两种电阻选择，即正常电压模式下的 60kΩ 或低电压模式下的 15kΩ。电阻的选择通过 LVDC 寄存器中的 LVPU 位来执行。

LVPU 位	上拉电阻
0	60kΩ (正常模式)
1	15kΩ (低电压模式)

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	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
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 I/O 端口 PA bit 7~bit 0 唤醒控制位
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~PHC，用来控制输入 / 输出配置。从而每个 I/O 引脚都可以在软件控制下，动态地设置为带或不带上拉电阻。I/O 端口的每个引脚都映射到其相关端口控制寄存器中的一个位。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 CMOS 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。

但应注意，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。

引脚共用功能

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。此外，这些引脚功能可以通过应用程序控制一系列寄存器进行设定。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAFS	PAFS7	PAFS6	PAFS5	PAFS4	PAFS3	PAFS2	PAFS1	PAFS0
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	PBFS3	PBFS2	PBFS1	PBFS0
PCFS	—	PCFS6	PCFS5	PCFS4	PCFS3	—	—	—
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
PGFS	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
PHFS	PHFS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
SFSR	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
SFSR1	—	—	PA4FS	PB4FS	PB5FS	PB7FS	PB6FS	SFS8

引脚共用选择寄存器列表 – HT67F370

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAFS	PAFS7	PAFS6	PAFS5	PAFS4	—	PAFS2	PAFS1	PAFS0
PBFS	—	—	—	—	PBFS3	PBFS2	PBFS1	PBFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
SFS	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0

引脚共用功能选择寄存器列表 – HT69F340

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAFS	PAFS7	PAFS6	PAFS5	PAFS4	PAFS3	PAFS2	PAFS1	PAFS0
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	PBFS3	PBFS2	PBFS1	PBFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
PGFS	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
PHFS	—	—	—	—	PHFS3	PHFS2	PHFS1	PHFS0
SFSR	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0

引脚共用功能选择寄存器列表 – HT69F350

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAFS	PAFS7	PAFS6	PAFS5	PAFS4	PAFS3	PAFS2	PAFS1	PAFS0
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	PBFS3	PBFS2	PBFS1	PBFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
PGFS	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
PHFS	PHFS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
SFSR	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
SFSR1	—	—	—	—	—	PB7FS	PB6FS	SFS8

引脚共用功能选择寄存器列表 – HT69F360

● PAFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PAFS7	PAFS6	PAFS5	PAFS4	PAFS3	PAFS2	PAFS1	PAFS0
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 **PAFS7~PAFS6:** PA6 引脚共用功能选择

00: PA6
01: AN7
10: CTP
11: SCS

Bit 5~4 **PAFS5~PAFS4:** PA5 引脚共用功能选择

00: PA5
01: AN6
10: STP
11: SCK/SCL

Bit 3~2 **PAFS3~PAFS2:** PA3 引脚共用功能选择

00: PA3
01: AN4
10: STP
11: SDI/SDA

Bit 1~0 **PAFS1~PAFS0:** PA1 引脚共用功能选择

00: PA1
01: AN3
10: CTP
11: SDO

● PAFS 寄存器 – HT69F340

Bit	7	6	5	4	3	2	1	0
Name	PAFS7	PAFS6	PAFS5	PAFS4	—	PAFS2	PAFS1	PAFS0
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 7~6 **PAFS7~PAFS6:** PA6 引脚共用功能选择
 00: I/O
 01: I/O
 10: CTP_0
 11: SCS

Bit 5~4 **PAFS5~PAFS4:** PA5 引脚共用功能选择
 00: I/O
 01: I/O
 10: PTP_2
 11: SCK/SCL

Bit 3 未定义，读为 "0"

Bit 2 **PAFS2:** PA3 引脚共用功能选择
 0: I/O
 1: SDI/SDA

Bit 1~0 **PAFS1~PAFS0:** PA1 引脚共用功能选择
 00: I/O
 01: I/O
 10: CTP_1
 11: SDO

● PAFS 寄存器 – HT69F350/HT69F360

Bit	7	6	5	4	3	2	1	0
Name	PAFS7	PAFS6	PAFS5	PAFS4	PAFS3	PAFS2	PAFS1	PAFS0
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 **PAFS7~PAFS6:** PA6 引脚共用功能选择
 00: I/O
 01: I/O
 10: CTP_0
 11: SCS

Bit 5~4 **PAFS5~PAFS4:** PA5 引脚共用功能选择
 00: I/O
 01: I/O
 10: STP_1
 11: SCK/SCL

Bit 3~2 **PAFS3~PAFS2:** PA3 引脚共用功能选择
 00: I/O
 01: I/O
 10: STP_0
 11: SDI/SDA

Bit 1~0 **PAFS1~PAFS0:** PA1 引脚共用功能选择
 00: I/O
 01: I/O
 10: CTP_1
 11: SDO

● PBFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PBFS7	PBFS6	PBFS5	PBFS4	PBFS3	PBFS2	PBFS1	PBFS0
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 **PBFS7:** PB7 引脚共用功能选择, 该位结合 SFSR1 寄存器中 PB7FS 位
[PBFS7:PB7FS]=

00: PB7
01: AN2
10: PTP1B
11: TX

Bit 6 **PBFS6:** PB6 引脚共用功能选择, 该位结合 SFSR1 寄存器中 PB6FS 位
[PBFS6:PB6FS]=

00: PB6
01: AN1
10: PTP1
11: RX

Bit 5 **PBFS5:** PB5 引脚共用功能选择, 该位结合 SFSR1 寄存器中 PB5FS 位
[PB5FS:PBFS5]=

00: PB5
01: PTP0B
10: VREF
11: AN0

Bit 4 **PBFS4:** PB4 引脚共用功能选择, 该位结合 SFSR1 寄存器中 PB4FS 位
[PB4FS:PBFS4]=

00: PB4
01: PTP0
10: VREF1
11: PB4

Bit 3~2 **PBFS3~PBFS2:** PB3 或 PB2 引脚共用功能选择

00: PB3 或 PB2
01: PB3 或 PB2
10: PB3 或 PB2
11: XT2 或 XT1

Bit 1~0 **PBFS1~PBFS0:** PB1 或 PB0 引脚共用功能选择

00: PB1 或 PB0
01: PB1 或 PB0
10: PB1 或 PB0
11: OSC2 或 OSC1

● PBFS 寄存器 – HT69F340

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PBFS3	PBFS2	PBFS1	PBFS0
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 3~2 **PBFS3~PBFS2:** PB3/PB2 引脚共用功能选择

00: I/O
01: I/O
10: I/O
11: XT2/XT1

Bit 1~0 **PBFS1~PBFS0:** PB1/PB0 引脚共用功能选择
00: I/O
01: I/O
10: I/O
11: OSC2/OSC1

● **PBFS 寄存器 – HT69F350/HT69F360**

Bit	7	6	5	4	3	2	1	0
Name	PBFS7	PBFS6	PBFS5	PBFS4	PBFS3	PBFS2	PBFS1	PBFS0
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 **PBFS7:** PB7 引脚共用功能选择
0: I/O
1: PTPB_1

Bit 6 **PBFS6:** PB6 引脚共用功能选择
0: I/O
1: PTP_1

Bit 5 **PBFS5:** PB5 引脚共用功能选择
0: I/O
1: PTPB_0

Bit 4 **PBFS4:** PB4 引脚共用功能选择
0: I/O
1: PTP_0

Bit 3~2 **PBFS3~PBFS2:** PB3/PB2 引脚共用功能选择
00: I/O
01: I/O
10: I/O
11: XT2/XT1

Bit 1~0 **PBFS1~PBFS0:** PB1/PB0 引脚共用功能选择
00: I/O
01: I/O
10: I/O
11: OSC2/OSC1

● **PCFS 寄存器 – HT67F370**

Bit	7	6	5	4	3	2	1	0
Name	—	PCFS6	PCFS5	PCFS4	PCFS3	—	—	—
R/W	—	R/W	R/W	R/W	R/W	—	—	—
POR	—	0	0	0	0	—	—	—

Bit 7 未定义，读为“0”

Bit 6 **PCFS6:** PC6 引脚共用功能选择
0: PC6
1: AN11

Bit 5 **PCFS5:** PC5 引脚共用功能选择
0: PC5
1: AN10

Bit 4 **PCFS4:** PC4 引脚共用功能选择
0: PC4
1: AN9

Bit 3 **PCFS3:** PC3 引脚共用功能选择
 0: PC3
 1: AN8
Bit 2~0 未定义，读为“0”

● **PDFS 寄存器 – HT67F370**

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
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 **PDFS7:** PD7 引脚共用功能选择
 0: PD7
 1: SEG7/PTP1B
Bit 6 **PDFS6:** PD6 引脚共用功能选择
 0: PD6
 1: SEG6/PTP1
Bit 5 **PDFS5:** PD5 引脚共用功能选择
 0: PD5
 1: SEG5/PTP0B
Bit 4 **PDFS4:** PD4 引脚共用功能选择
 0: PD4
 1: SEG4/PTP0
Bit 3 **PDFS3:** PD3 引脚共用功能选择
 0: PD3/PTCK0
 1: SEG3
Bit 2 **PDFS2:** PD2 引脚共用功能选择
 0: PD2/STCK
 1: SEG2
Bit 1 **PDFS1:** PD1 引脚共用功能选择
 0: PD1/CTCK
 1: SEG1
Bit 0 **PDFS0:** PD0 引脚共用功能选择
 0: PD0/INT1/PTCK1
 1: SEG0

● **PDFS 寄存器 – HT69F340**

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
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 **PDFS7:** PD7 引脚共用功能选择
 0: I/O
 1: SEG7/PTPB_1
Bit 6 **PDFS6:** PD6 引脚共用功能选择
 0: I/O
 1: SEG6/PTP_1
Bit 5 **PDFS5:** PD5 引脚共用功能选择
 0: I/O
 1: SEG5/PTPB_0
Bit 4 **PDFS4:** PD4 引脚共用功能选择
 0: I/O
 1: SEG4/PTP_0

- Bit 3 **PDFS3:** PD3 引脚共用功能选择
0: I/O
1: SEG3
- Bit 2 **PDFS2:** PD2 引脚共用功能选择
0: I/O
1: SEG2/CTP_2
- Bit 1 **PDFS1:** PD1 引脚共用功能选择
0: I/O
1: SEG1
- Bit 0 **PDFS0:** PD0 引脚共用功能选择
0: I/O
1: SEG0

● **PDFS 寄存器 – HT69F350**

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
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 **PDFS7:** PD7 引脚共用功能选择
0: I/O
1: SEG7/PTPB_1
- Bit 6 **PDFS6:** PD6 引脚共用功能选择
0: I/O
1: SEG6/PTP_1
- Bit 5 **PDFS5:** PD5 引脚共用功能选择
0: I/O
1: SEG5/PTPB_0
- Bit 4 **PDFS4:** PD4 引脚共用功能选择
0: I/O
1: SEG4/PTP_0
- Bit 3 **PDFS3:** PD3 引脚共用功能选择
0: I/O
1: SEG3
- Bit 2 **PDFS2:** PD2 引脚共用功能选择
0: I/O
1: SEG2
- Bit 1 **PDFS1:** PD1 引脚共用功能选择
0: I/O
1: SEG1
- Bit 0 **PDFS0:** PD0 引脚共用功能选择
0: I/O
1: SEG0

● PDFS 寄存器 – HT69F360

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
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 **PDFS7:** PD7 引脚共用功能选择
0: I/O
1: SEG7/PTP1B

Bit 6 **PDFS6:** PD6 引脚共用功能选择
0: I/O
1: SEG6/PTP1

Bit 5 **PDFS5:** PD5 引脚共用功能选择
0: I/O
1: SEG5/PTP0B

Bit 4 **PDFS4:** PD4 引脚共用功能选择
0: I/O
1: SEG4/PTP0

Bit 3 **PDFS3:** PD3 引脚共用功能选择
0: I/O
1: SEG3

Bit 2 **PDFS2:** PD2 引脚共用功能选择
0: I/O
1: SEG2

Bit 1 **PDFS1:** PD1 引脚共用功能选择
0: I/O
1: SEG1

Bit 0 **PDFS0:** PD0 引脚共用功能选择
0: I/O
1: SEG0

● PEFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
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 **PEFS7:** PE7 引脚共用功能选择
0: PE7
1: SEG15

Bit 6 **PEFS6:** PE6 引脚共用功能选择
0: PE6
1: SEG14

Bit 5 **PEFS5:** PE5 引脚共用功能选择
0: PE5
1: SEG13

Bit 4 **PEFS4:** PE4 引脚共用功能选择
0: PE4
1: SEG12

Bit 3 **PEFS3:** PE3 引脚共用功能选择
0: PE3
1: SEG11

- Bit 2 **PEFS2:** PE2 引脚共用功能选择
0: PE2
1: SEG10
- Bit 1 **PEFS1:** PE1 引脚共用功能选择
0: PE1
1: SEG9
- Bit 0 **PEFS0:** PE0 引脚共用功能选择
0: PE0
1: SEG8

● **PEFS 寄存器 – HT69F340/HT69F350/HT69F360**

Bit	7	6	5	4	3	2	1	0
Name	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
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 **PEFS7:** PE7 引脚共用功能选择
0: I/O
1: SEG15
- Bit 6 **PEFS6:** PE6 引脚共用功能选择
0: I/O
1: SEG14
- Bit 5 **PEFS5:** PE5 引脚共用功能选择
0: I/O
1: SEG13
- Bit 4 **PEFS4:** PE4 引脚共用功能选择
0: I/O
1: SEG12
- Bit 3 **PEFS3:** PE3 引脚共用功能选择
0: I/O
1: SEG11
- Bit 2 **PEFS2:** PE2 引脚共用功能选择
0: I/O
1: SEG10
- Bit 1 **PEFS1:** PE1 引脚共用功能选择
0: I/O
1: SEG9
- Bit 0 **PEFS0:** PE0 引脚共用功能选择
0: I/O
1: SEG8

● PFFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
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 **PFFS7:** PF7 引脚共用功能选择
0: PF7
1: SEG23
- Bit 6 **PFFS6:** PF6 引脚共用功能选择
0: PF6
1: SEG22
- Bit 5 **PFFS5:** PF5 引脚共用功能选择
0: PF5
1: SEG21
- Bit 4 **PFFS4:** PF4 引脚共用功能选择
0: PF4
1: SEG20
- Bit 3 **PFFS3:** PF3 引脚共用功能选择
0: PF3
1: SEG19
- Bit 2 **PFFS2:** PF2 引脚共用功能选择
0: PF2
1: SEG18
- Bit 1 **PFFS1:** PF1 引脚共用功能选择
0: PF1
1: SEG17
- Bit 0 **PFFS0:** PF0 引脚共用功能选择
0: PF0
1: SEG16

● PFFS 寄存器 – HT69F340/HT69F350/HT69F360

Bit	7	6	5	4	3	2	1	0
Name	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
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 **PFFS7:** PF7 引脚共用功能选择
0: I/O
1: SEG23
- Bit 6 **PFFS6:** PF6 引脚共用功能选择
0: I/O
1: SEG22
- Bit 5 **PFFS5:** PE5 引脚共用功能选择
0: I/O
1: SEG21
- Bit 4 **PFFS4:** PF4 引脚共用功能选择
0: I/O
1: SEG20
- Bit 3 **PFFS3:** PF3 引脚共用功能选择
0: I/O
1: SEG19

- Bit 2 **PFFS2:** PF2 引脚共用功能选择
0: I/O
1: SEG18
- Bit 1 **PFFS1:** PF1 引脚共用功能选择
0: I/O
1: SEG17
- Bit 0 **PFFS0:** PF0 引脚共用功能选择
0: I/O
1: SEG16

● **PGFS 寄存器 – HT69F350/HT69F360**

Bit	7	6	5	4	3	2	1	0
Name	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
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 **PGFS7:** PG7 引脚共用功能选择
0: I/O
1: SEG31
- Bit 6 **PGFS6:** PG6 引脚共用功能选择
0: I/O
1: SEG30
- Bit 5 **PGFS5:** PG5 引脚共用功能选择
0: I/O
1: SEG29
- Bit 4 **PGFS4:** PG4 引脚共用功能选择
0: I/O
1: SEG28
- Bit 3 **PGFS3:** PG3 引脚共用功能选择
0: I/O
1: SEG27
- Bit 2 **PGFS2:** PG2 引脚共用功能选择
0: I/O
1: SEG26
- Bit 1 **PGFS1:** PG1 引脚共用功能选择
0: I/O
1: SEG25
- Bit 0 **PGFS0:** PG0 引脚共用功能选择
0: I/O
1: SEG24

● PHFS 寄存器 – HT69F350

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PHFS3	PHFS2	PHFS1	PHFS0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

Bit 7~4 未定义，读为“0”

Bit 3 **PHFS3**: PH3 引脚共用功能选择
0: I/O
1: SEG35

Bit 2 **PHFS2**: PH2 引脚共用功能选择
0: I/O
1: SEG34

Bit 1 **PHFS1**: PH1 引脚共用功能选择
0: I/O
1: SEG33

Bit 0 **PHFS0**: PH0 引脚共用功能选择
0: I/O
1: SEG32

● PHFS 寄存器 – HT69F360

Bit	7	6	5	4	3	2	1	0
Name	PHFS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
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 **PHFS7**: PH7 引脚共用功能选择
0: I/O
1: SEG39

Bit 6 **PHFS6**: PH6 引脚共用功能选择
0: I/O
1: SEG38

Bit 5 **PHFS5**: PH5 引脚共用功能选择
0: I/O
1: SEG37

Bit 4 **PHFS4**: PH4 引脚共用功能选择
0: I/O
1: SEG36

Bit 3 **PHFS3**: PH3 引脚共用功能选择
0: I/O
1: SEG35

Bit 2 **PHFS2**: PH2 引脚共用功能选择
0: I/O
1: SEG34

Bit 1 **PHFS1**: PH1 引脚共用功能选择
0: I/O
1: SEG33

Bit 0 **PHFS0**: PH0 引脚共用功能选择
0: I/O
1: SEG32

● PGFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
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 **PGFS7:** PG7 引脚共用功能选择
0: PG7
1: SEG31
- Bit 6 **PGFS6:** PG6 引脚共用功能选择
0: PG6
1: SEG30
- Bit 5 **PGFS5:** PG5 引脚共用功能选择
0: PG5
1: SEG29
- Bit 4 **PGFS4:** PG4 引脚共用功能选择
0: PG4
1: SEG28
- Bit 3 **PGFS3:** PG3 引脚共用功能选择
0: PG3
1: SEG27
- Bit 2 **PGFS2:** PG2 引脚共用功能选择
0: PG2
1: SEG26
- Bit 1 **PGFS1:** PG1 引脚共用功能选择
0: PG1
1: SEG25
- Bit 0 **PGFS0:** PG0 引脚共用功能选择
0: PG0
1: SEG24

● PHFS 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PHFS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
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 **PHFS7:** PH7 引脚共用功能选择
0: PH7
1: SEG39
- Bit 6 **PHFS6:** PH6 引脚共用功能选择
0: PH6
1: SEG38
- Bit 5 **PHFS5:** PH5 引脚共用功能选择
0: PH5
1: SEG37
- Bit 4 **PHFS4:** PH4 引脚共用功能选择
0: PH4
1: SEG36
- Bit 3 **PHFS3:** PH3 引脚共用功能选择
0: PH3
1: SEG35

- Bit 2 **PHFS2:** PH2 引脚共用功能选择
0: PH2
1: SEG34
- Bit 1 **PHFS1:** PH1 引脚共用功能选择
0: PH1
1: SEG33
- Bit 0 **PHFS0:** PH0 引脚共用功能选择
0: PH0
1: SEG32

● **SFSR 寄存器 – HT67F370**

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
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 **SFS7:** STCK 源选择
0: PA2
1: PD2
- Bit 6 **SFS6:** PTCK0 源选择
0: PA0
1: PD3
- Bit 5 **SFS5:** CTCK 源选择
0: PA2
1: PD1
- Bit 4 **SFS4:** INT1 源选择
0: PA0
1: PD0
- Bit 3 **SFS3:** PD7 特殊功能选择
0: SEG7
1: PTP1B
- Bit 2 **SFS2:** PD6 特殊功能选择
0: SEG6
1: PTP1
- Bit 1 **SFS1:** PD5 特殊功能选择
0: SEG5
1: PTP0B
- Bit 0 **SFS0:** PD4 特殊功能选择
0: SEG4
1: PTP0

● SFS 寄存器 – HT69F340

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
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 **SFS7:** PD2 特殊功能选择
0: SEG2
1: CTP_2
- Bit 6 **SFS6:** PTCK 源选择位
0: PA3
1: PD3
- Bit 5 **SFS5:** CTCK 源选择位
0: PA2
1: PD1
- Bit 4 **SFS4:** INT1 源选择位
0: PA0
1: PD0
- Bit 3 **SFS3:** PD7 特殊功能选择
0: SEG7
1: PTPB_1
- Bit 2 **SFS2:** PD6 特殊功能选择
0: SEG6
1: PTP_1
- Bit 1 **SFS1:** PD5 特殊功能选择
0: SEG5
1: PTPB_0
- Bit 0 **SFS0:** PD4 特殊功能选择
0: SEG4
1: PTP_0

● SFSR 寄存器 – HT69F350

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
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 **SFS7:** STCK 源选择位
0: PA2
1: PD2
- Bit 6 **SFS6:** PTCK 源选择位
0: PA0
1: PD3
- Bit 5 **SFS5:** CTCK 源选择位
0: PA2
1: PD1
- Bit 4 **SFS4:** INT1 源选择位
0: PA0
1: PD0
- Bit 3 **SFS3:** PD7 特殊功能选择位
0: SEG7
1: PTPB_1

- Bit 2 **SFS2:** PD6 特殊功能选择位
0: SEG6
1: PTP_1
- Bit 1 **SFS1:** PD5 特殊功能选择位
0: SEG5
1: PTPB_0
- Bit 0 **SFS0:** PD4 特殊功能选择位
0: SEG4
1: PTP_0

● **SFSR 寄存器 – HT69F360**

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
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 **SFS7:** STCK 源选择位
0: PA2
1: PD2
- Bit 6 **SFS6:** PTCK 源选择位
0: PA0
1: PD3
- Bit 5 **SFS5:** CTCK 源选择位
0: PA2
1: PD1
- Bit 4 **SFS4:** INT1 源选择位
0: PA0
1: PD0
- Bit 3 **SFS3:** PD7 特殊功能选择位
0: SEG7
1: PTP1B
- Bit 2 **SFS2:** PD6 特殊功能选择位
0: SEG6
1: PTP1
- Bit 1 **SFS1:** PD5 特殊功能选择位
0: SEG5
1: PTP0B
- Bit 0 **SFS0:** PD4 特殊功能选择位
0: SEG4
1: PTP0

● **SFSR1 寄存器 – HT67F370**

Bit	7	6	5	4	3	2	1	0
Name	—	—	PA4FS	PB4FS	PB5FS	PB7FS	PB6FS	SFS8
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 **PA4FS:** PA4 引脚共用功能选择
0: PA4/INT0/PTCK1
1: AN5

- Bit 4 **PB4FS:** PB4 引脚共用功能选择，该位结合 PBFS 寄存器中 PBFS4 位
[PB4FS:PBFS4]=
00: PB4
01: PTP0
10: VREFI
11: PB4
- Bit 3 **PB5FS:** PB5 引脚共用功能选择，该位结合 PBFS 寄存器中 PBFS5 位
[PB5FS:PBFS5]=
00: PB5
01: PTP0B
10: VREF
11: AN0
- Bit 2 **PB7FS:** PB7 引脚共用功能选择，该位结合 PBFS 寄存器中 PBFS7 位
[PBFS7:PB7FS]=
00: PB7
01: AN2
10: PTP1B
11: TX
- Bit 1 **PB6FS:** PB6 引脚共用功能选择，该位结合 PBFS 寄存器中 PBFS6 位
[PBFS6:PB6FS]=
00: PB6
01: AN1
10: PTP1
11: RX
- Bit 0 **SFS8:** PTCK1 源选择
0: PA4
1: PD0

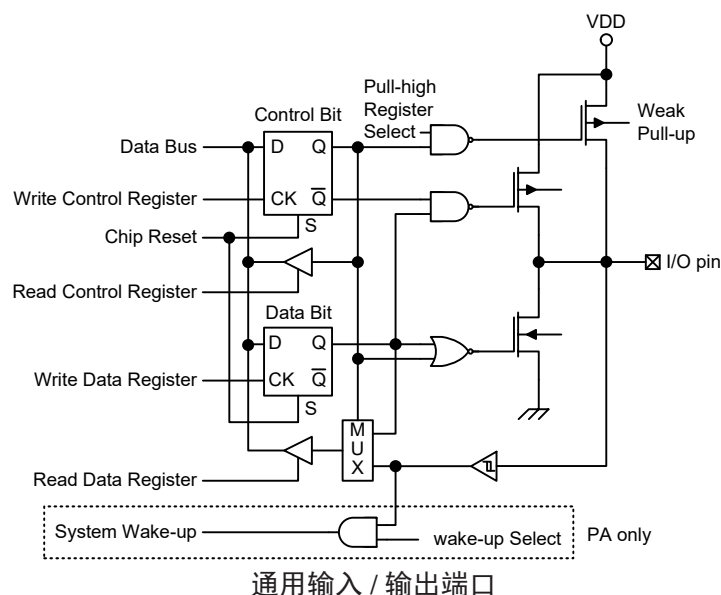
● SFSR1 寄存器 – HT69F360

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PB7FS	PB6FS	SFS8
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

- Bit 7~3 未定义，读为“0”
- Bit 2 **PB7FS:** PB7 引脚共用功能选择
0: PTP1B
1: UART TX
- Bit 1 **PB6FS:** PB6 引脚共用功能选择
0: PTP1
1: UART RX
- Bit 0 **SFS8:** PTCK1 源选择位
0: PA4
1: PD0

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。



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，也就是说所有输入 / 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。该系列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，标准型和周期型定时器章节。

简介

该系列单片机包含 2 个或 4 个 TM，取决于所选单片机类型。HT69F340 单片机包含一个 10-bit 的简易型 TM (CTM) 和一个 10-bit 的周期型 TM (PTM)。HT69F350 单片机包含一个 10-bit 的简易型 TM (CTM)，一个 16-bit 的标准型 TM (STM) 和一个 10-bit 的周期型 TM (PTM)。而 HT69F360/HT67F370 单片机则含有一个 10-bit 的简易型 TM (CTM)，一个 16-bit 的标准型 TM (STM) 和两个 10-bit 的周期型 TM (PTM0 和 PTM1)。虽然性质相似，但不同 TM 特性复杂度不同。本章介绍简易型，标准型和周期型 TM 的共性，更多详细资料见后面各相应章节。三种类型 TM 的特性和区别见下表。

TM 功能	CTM	STM	PTM
定时 / 计数器	√	√	√
捕捉输入	—	√	√
比较匹配输出	√	√	√
PWM 通道数	√	√	√
单脉冲输出	—	√	√
PWM 对齐方式	边沿对齐	边沿对齐	边沿对齐
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

单片机型号	CTM	STM	PTM
HT69F340	10-bit CTM	—	10-bit PTM
HT69F350		16-bit STM	10-bit PTM
HT69F360/HT67F370			10-bit PTM0 10-bit PTM1

TM 类型参考

TM 操作

这三种不同类型的 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 xTnCK2~xTnCK0 位可选择所需的时钟源，其中 x 代表 C、S 或 P 类型，n 代表具体 TM 编号。该

时钟源来自系统时钟 f_{SYS} 或内部高速时钟 f_H 或 f_{SUB} 时钟源或外部 $xTCKn$ 引脚的分频比。 $xTCKn$ 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，标准型和周期型 TM 各有两个内部中断，即内部比较器 A 和比较器 P，当发生比较匹配时比较器将产生一个 TM 中断。中断产生时将清除计数器，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 $xTCKn$ 。TM 输入引脚作为 TM 的时钟源输入脚，通过设置 $xTMnCO$ 寄存器中的 $xTnCK2\sim xTnCK0$ 位进行选择。外部时钟源可通过该引脚来驱动内部 TM。外部 TM 输入引脚与其他功能共用一个引脚，可利用 $xTnCK2\sim xTnCK0$ 位将其连接到内部 TM。TM 输入引脚可以选择上升沿或下降沿。 $xTCKn$ 引脚还可用作 $xTMn$ 单脉冲模式的外部触发引脚。

每个 TM 有一个或多个输出引脚 $xTPn$ 和 $PTPnB$ 。 $PTPnB$ 信号为 $PTPn$ 输出的反相信号。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 $xTPn$ 输出引脚也被 TM 用来产生 PWM 输出波形。当 TM 工作在捕捉输入模式下， $xTPn$ 引脚将作为输入脚。由于 $xTPn$ 引脚与其它功能共用，TM 输出功能需要通过寄存器先被设置。寄存器中的单个位决定了其相关引脚是用作外部 TM 输出引脚还是其他功能。

有些 TM 输出引脚的名字有个“_n”的后缀。引脚名中包含了后缀“_1”说明这些引脚为 TM 的多个输出引脚。这使得 TM 可以产生互补输出对，利用 I/O 寄存器数据列表进行选择。

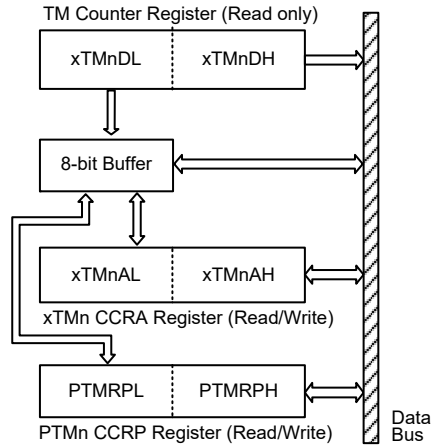
单片机型号	CTM	STM	PTM0	PTM1
HT69F340	CTP_0, CTP_1, CTP_2	—	PTP_0, PTP_1, PTP_2 PTPB_0, PTPB_1	—
HT69F350	CTP_0, CTP_1	STP_0, STP_1	PTP_0, PTP_1 PTPB_0, PTPB_1	—
HT69F360/ HT67F370	CTP_0, CTP_1	STP_0, STP_1	PTP0, PTP0B	PTP1, PTP1B

TM 输出引脚

编程注意事项

10-bit 或 16-bit 的 TM 计数寄存器和捕捉 / 比较寄存器 CCRA 和 CCRP 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。

由于 CCRA 和 CCRP 寄存器按照下图方式执行，具体存取这些寄存器对的方式如上所述，建议使用“MOV”指令，通过以下步骤访问 CCRA 和 CCRP 低字节，命名为 $xTMnAL$ 或 $PTMnRPL$ 。若不采用以下步骤访问 CCRA 和 CCRP 将导致不可预期的结果。



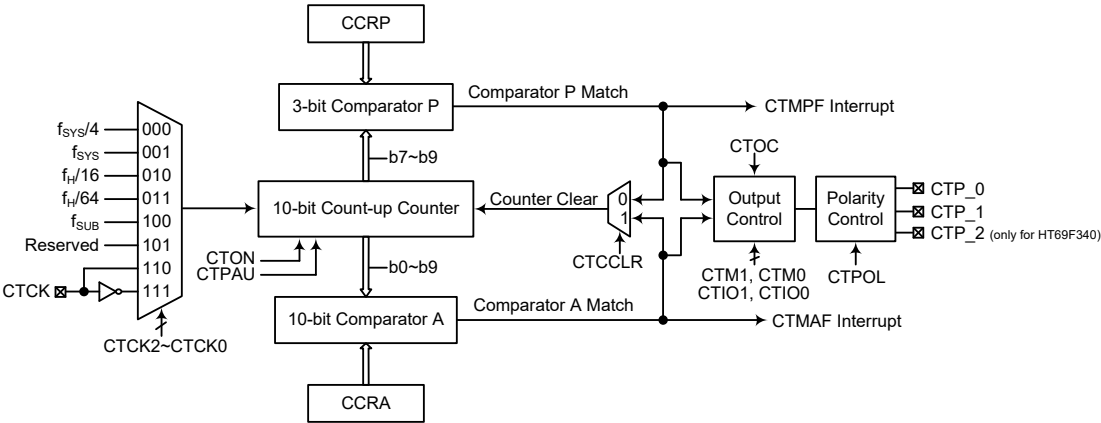
读写流程如下步骤所示：

- 写数据至 CCRA 或 CCRP
 - ◆ 步骤 1. 写数据至低字节寄存器 xTMnAL 或 PTMnRPL
 - 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 xTMnAH 或 PTMnRPH
 - 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 CCRA 或 CCRP 中读取数据
 - ◆ 步骤 1. 从高字节寄存器 xTMnDH、xTMnAH 或 PTMnRPH 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. 从低字节寄存器 xTMnDL、xTMnAL 或 PTMnRPL 中读取数据
 - 注意，此时读取 8-bit 缓存器中的数据。

简易型 TM – CTM

尽管简易型 TM 是三种 TM 类型中最简单的形式，它仍包括三种工作模式，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两个或三个外部输出脚。

单片机	TM 类型	TM 输入引脚	TM 输出引脚
HT69F340	10-bit CTM	CTCK	CTP_0, CTP_1, CTP_2
HT69F350	10-bit CTM	CTCK	CTP_0, CTP_1
HT69F360/ HT67F370	10-bit CTM	CTCK	CTP_0, CTP_1



注：CTM 外部引脚与其它功能共用引脚，因此在使用 CTM 之前应该合理配置相关引脚共用功能选择寄存器以确保使能 CTM 引脚功能。对于 CTCK 引脚还需设置相应的端口控制寄存器，将该引脚设置为输入口。

简易型 TM 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-bit 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-bit 计数器值的唯一方法是使 CTON 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，通常情况下会产生 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内部寄存器来实现的。

简易型 TM 寄存器介绍

简易型 TM 的所有操作由一系列寄存器控制。包含一对只读寄存器用来存放 10-bit 计数器的值，一对读 / 写寄存器存放 10-bit 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。

寄存器名称	位							
	7	6	5	4	3	2	1	0
CTMC0	CTPAU	CTCK2	CTCK1	CTCK0	CTON	CTRP2	CTRP1	CTRP0
CTMC1	CTM1	CTM0	CTIO1	CTIO0	CTOC	CTPOL	CTDPX	CTCCLR
CTMDL	D7	D6	D5	D4	D3	D2	D1	D0
CTMDH	—	—	—	—	—	—	D9	D8
CTMAL	D7	D6	D5	D4	D3	D2	D1	D0
CTMAH	—	—	—	—	—	—	D9	D8

简易型 TM 寄存器列表

• CTMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTPAU	CTCK2	CTCK1	CTCK0	CTON	CTRP2	CTRP1	CTRP0
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 CTPAU:** CTM 计数器暂停控制位
0: 运行
1: 暂停
通过设置此位为高可暂停计数器。此位清零可恢复正常计数器操作。在暂停情况下 TM 将保持上电并继续产生功耗。当此位由低到高转换，计数器将保留其剩余值，当其电平再变为低时从此位开始继续计数。
- Bit 6~4 CTCK2~CTCK0:** CTM 计数器时钟选择位
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: 保留
110: CTCK 上升沿时钟
111: CTCK 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_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。
- Bit 3 CTON:** CTM 计数器 On/Off 控制位
0: Off
1: On
此位控制 TM 的总开关功能。设置此位为高可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 减少功耗。当此位经由低到高转换时，内部计数器将复位清零，当此位由高到低转换时，内部计数器将保持其剩余值直到此位再次转变为高。
若 TM 处于比较匹配输出模式，当 CTON 位经由低到高转换时，TM 输出脚将复位至 CTOC 位指定的初始值。
- Bit 2~0 CTRP2~CTRP0:** CTM CCRP 3 位寄存器，与 CTM 计数器 bit9~bit7 比较
比较器 P 匹配周期
000: 1024 个 CTM 时钟
001: 128 个 CTM 时钟
010: 256 个 CTM 时钟
011: 384 个 CTM 时钟
100: 512 个 CTM 时钟
101: 640 个 CTM 时钟

110: 768 个 CTM 时钟
111: 896 个 CTM 时钟

此三位设定内部 CCRP 3 位寄存器的值，然后与内部计数器的高三位进行比较。如果 CTCCLR 位设定为 0 时，选中该比较结果清除内部计数器。CTCCLR 位设为 0，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，会使得计数器在最大值溢出。

● CTMC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTM1	CTM0	CTIO1	CTIO0	CTOC	CTPOL	CTDPX	CTCCLR
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 **CTM1~CTM0:** CTM 操作模式选择位

- 00: 比较匹配输出模式
- 01: 未定义
- 10: PWM 模式
- 11: 定时 / 计数器模式

这两个位为 TM 设置所需的操作模式。为保证操作的可靠性，在此位段发生任何改变之前需先关闭 TM。在定时 / 计数器模式下，TM 输出引脚控制必须除能。

Bit 5~4 **CTIO1~CTIO0:** CTM 外部引脚功能选择位

比较匹配输出模式

- 00: 未改变
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 未定义

定时 / 计数器模式

未使用

这两位用于确定在达到某些条件时 TM 外部引脚状态应如何改变。这些位的功能选择取决于当下 TM 的运行模式。

在比较匹配输出模式下，CTIO1 和 CTIO0 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，输出将不会改变。TM 输出脚的初始值通过 CTOC 位设置取得。注意，由 CTIO1 和 CTIO0 位得到的输出电平必须与通过 CTOC 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 CTON 位由低到高电平的转换可复位至初始值。

在 PWM 模式，CTIO1 和 CTIO0 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 TM 关闭时改变 CTIO1 和 CTIO0 位的值是很有必要的。若在 TM 运行时改变 CTIO1 和 CTIO0 的值，PWM 输出的值是无法预料的。

Bit 3 **CTOC:** CTM 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

此位为 TM 输出脚的输出控制位。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，比较匹配发生时其决定 TM 输出脚的逻辑电平值。在 PWM 模式时，其决定 PWM 信号是有效还是低有效。

Bit2 **CTPOL:** CTM 输出极性控制位

0: 同相

1: 反相

此位控制 TM 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 **CTDPX:** CTM 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 **CTCCLR:** 选择 CTM 计数器清零条件位

0: CTM 比较器 P 匹配

1: CTM 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于清除内部计数器。CTCCLR 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 的位都被清除为 0 时才能生效。CTCCLR 位在 PWM 模式时未使用。

• CTMDL 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 **D7~D0:** CTM 计数器低字节寄存器 bit 7~bit 0

CTM 10-bit 计数器 bit 7~bit 0

• CTMDH 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 **D9~D8:** CTM 计数器高字节寄存器 bit 1~bit 0

CTM 10-bit 计数器 bit 9~bit 8

• CTMAL 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 **D7~D0:** CTM CCRA 低字节寄存器 bit 7~bit 0

CTM 10-bit CCRA bit 7~bit 0

● CTMAH 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 **D9~D8**: CTM CCRA 高字节寄存器 bit 1~bit 0
CTM 10-bit CCRA bit 9~bit 8

简易型 TM 工作模式

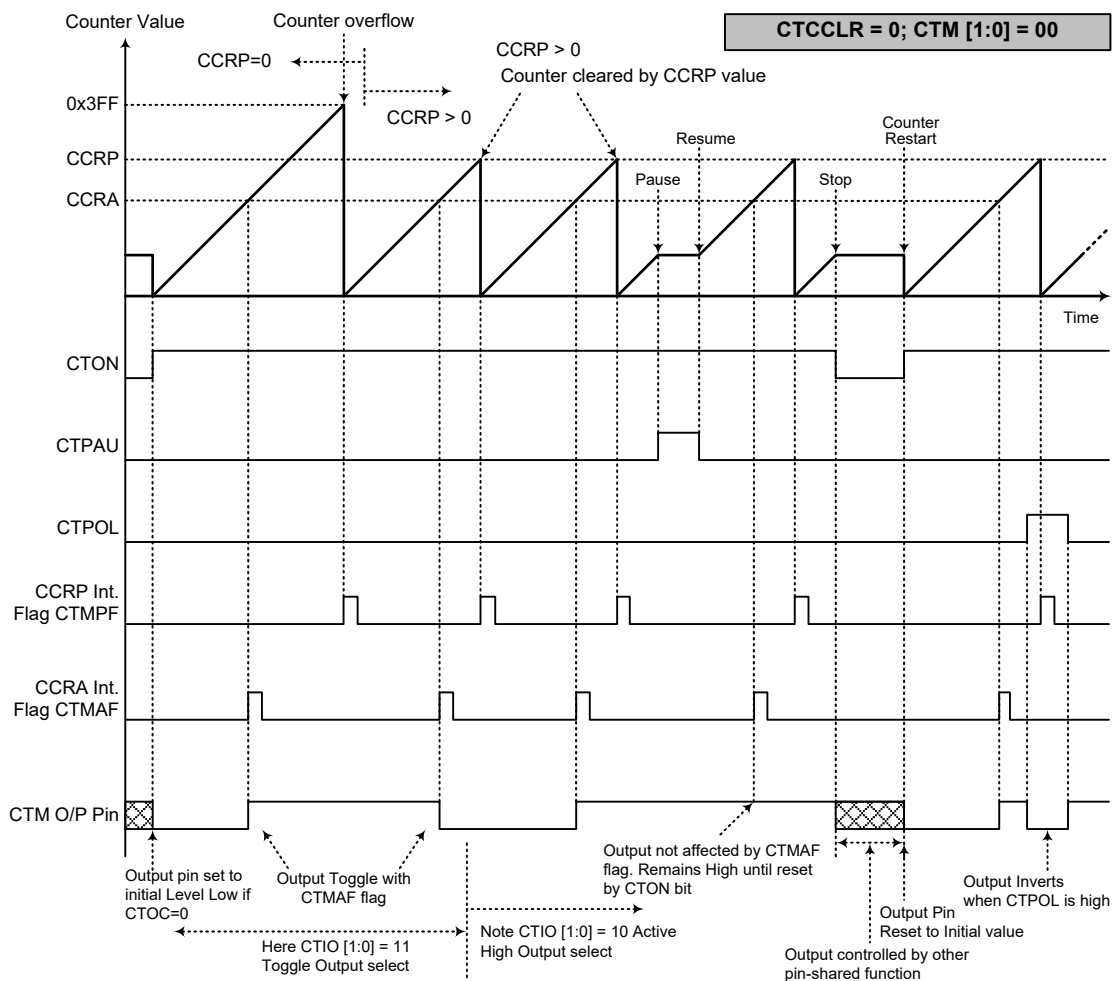
简易型 TM 有三种工作模式，即比较匹配输出模式，PWM 输出模式或定时 / 计数器模式。通过设置 CTMC1 寄存器的 CTM1 和 CTM0 位选择任意工作模式。

比较匹配输出模式

要工作在此模式，CTMC1 寄存器中的 CTM1 和 CTM0 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 CTCCLR 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 CTMAF 和 CTMPF 将分别置起。

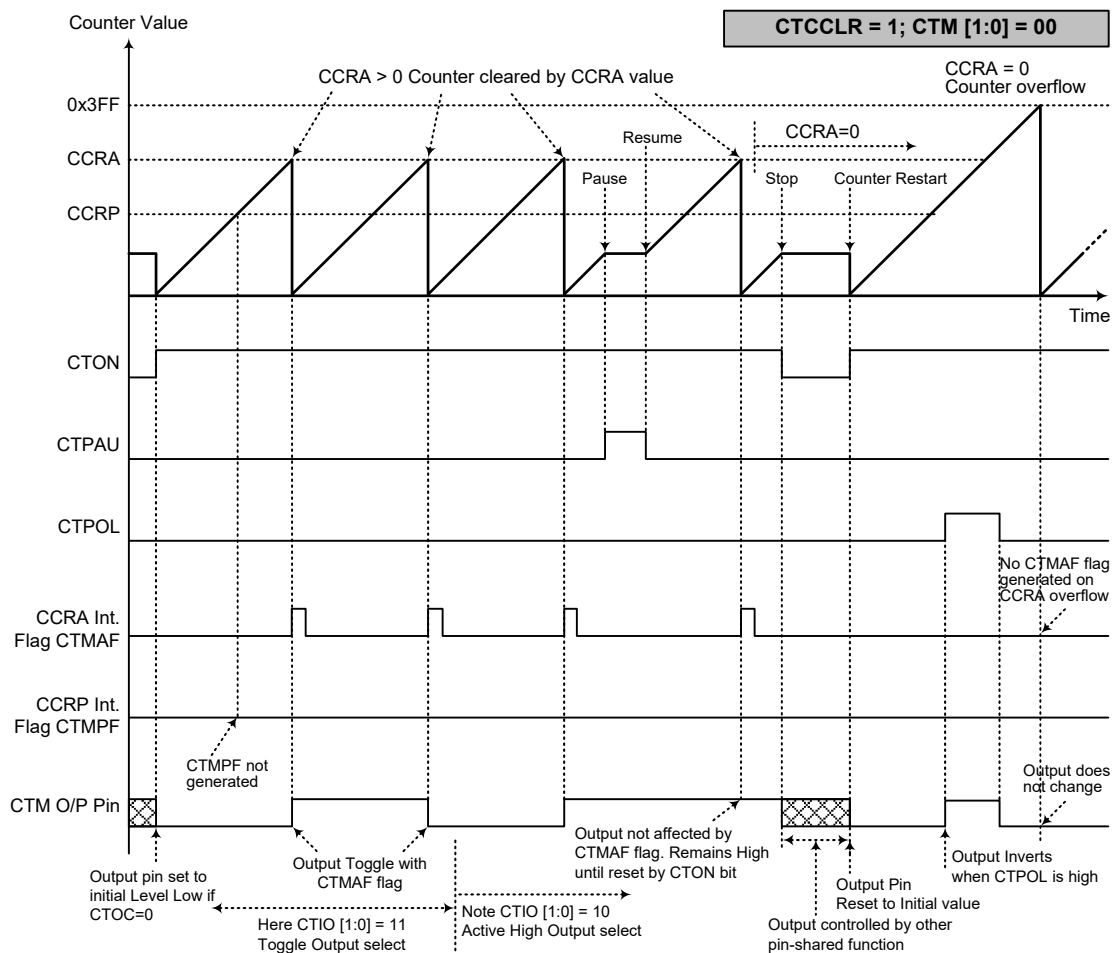
如果 CTMC1 寄存器的 CTCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 CTMAF 中断请求标志产生。所以当 CTCCLR 为高时，不产生 CTMPF 中断请求标志。如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 CTMAF 请求标志。

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 CTMAF 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 CTMPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 CTMC1 寄存器中 CTIO1 和 CTIO0 位决定。当比较器 A 比较匹配发生时，CTIO1 和 CTIO0 位决定 TM 输出脚输出高，低或翻转当前状态。在 CTON 位由低到高后，TM 输出脚初始状态为 CTC 位所指定的电平。注意，若 CTIO1 和 CTIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – CTCCLR=0

- 注：1. CTCCLR=0 时，比较器 P 匹配将清除计数器。
2. TM 输出引脚仅由 CTMAF 标志位控制
3. 输出引脚通过 CTON 位上升沿复位为初始值



比较匹配输出模式 – CTCCLR=1

- 注：1. CTCCLR=1 时，比较器 A 匹配将清除计数器
2. TM 输出引脚控制仅由 CTMAF 标志位控制
3. 输出引脚通过 CTON 上升沿复位至初始值
4. 当 CTCCLR=1 时不产生 CTMPF 标志位

定时 / 计数器模式

要工作在此模式，CTMC1 寄存器中的 CTM1 和 CTM0 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帮助理解此功能。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

要工作在此模式，CTMC1 寄存器中的 CTM1 和 CTM0 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控，其波形的选择就较为灵活。在 PWM 模式中，CTCCLR 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 CTMC1 寄存器的 CTD PX 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。CTMC1 寄存器中的 CTOC 位决定 PWM 波形的极性，CTIO1 和 CTIO0 位使能 PWM 输出或将 TM 输出脚置为逻辑高或逻辑低。CTPOL 位对 PWM 输出波形的极性取反。

● 10-bit CTM, PWM 模式, 边沿对其模式, CTD PX=0

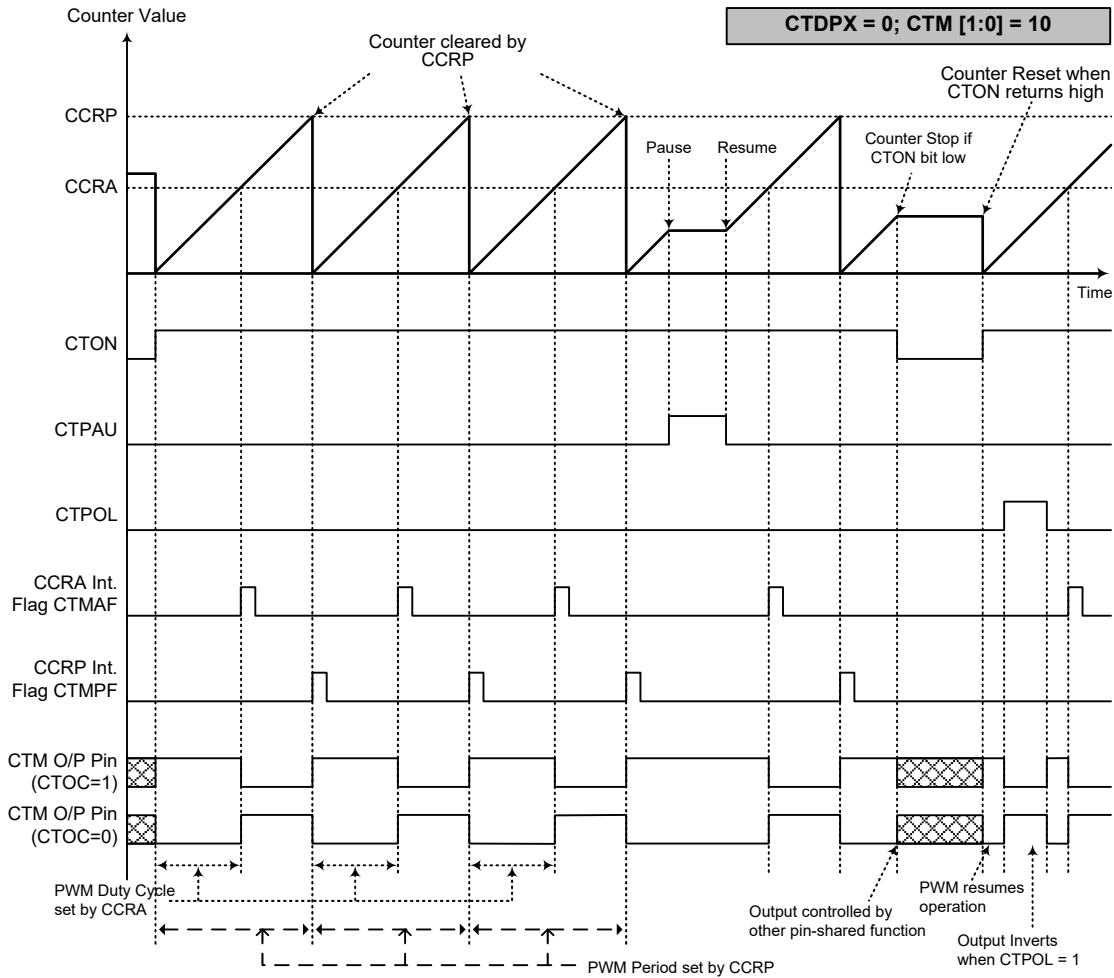
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

若 $f_{SYS}=16\text{MHz}$, CTM 时钟源选择 $f_{SYS}/4$, CCRP=100b, CCRA=128,
CTM PWM 输出频率 = $(f_{SYS}/4)/512=f_{SYS}/2048=7.8125\text{kHz}$, 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 CTM, PWM 模式, 边沿对其模式, CTD PX=1

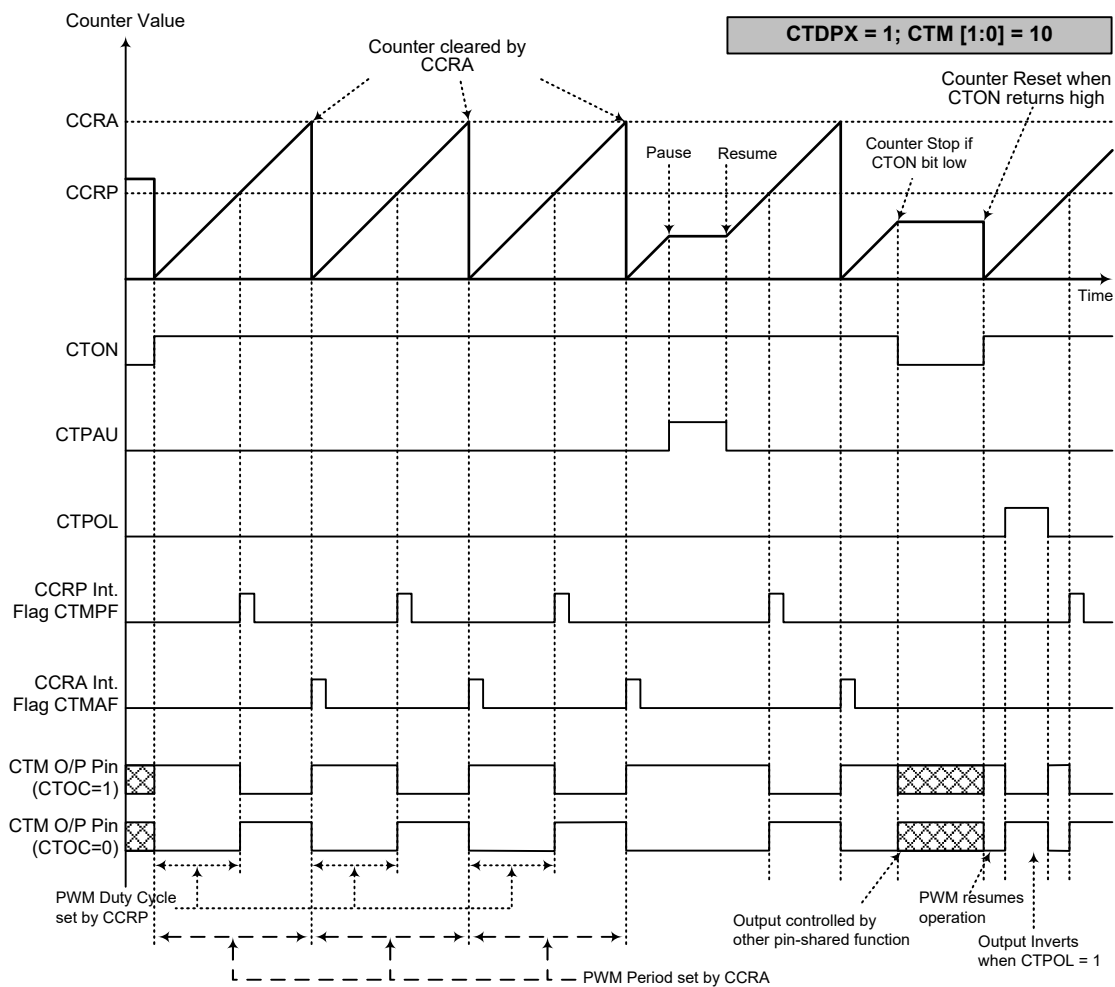
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

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 输出模式 – CTD PX=0

- 注：1. 这里的 CTD PX=0 – 计数器由 CCRP 清除
2. 计数器清除设置 PWM 周期
3. 即使在 CTIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. CTCCLR 位对 PWM 操作没有影响



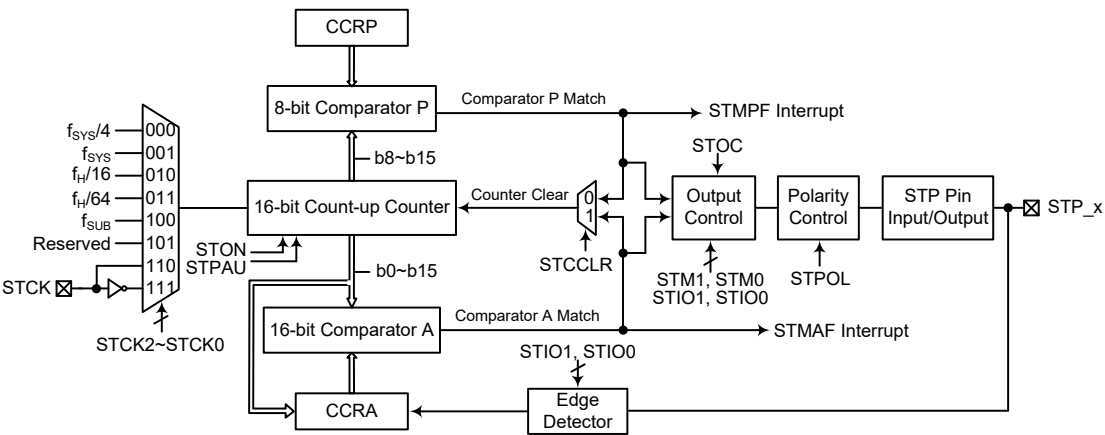
PWM 输出模式 – $CTDPX=1$

- 注：1. 这里的 $CTDPX=1$ – 计数器由 $CCRA$ 清除
2. 计数器清除设置 PWM 周期
3. 即使在 $CTIO[1:0]=00$ 或 01 时，内部 PWM 功能继续运行
4. $CTCCLR$ 位对 PWM 操作无影响

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 可驱动两个外部输出脚。

单片机	TM 类型	TM 输入引脚	TM 输出引脚
HT69F350	16-bit STM	STCK	STP_0, STP_1
HT69F360/ HT67F370	16-bit STM	STCK	STP_0, STP_1



注：STM 外部引脚与其它功能共用引脚，因此在使用 STM 之前应该合理配置相关引脚共用功能选择寄存器以确保使能 STM 引脚功能。对于 STCK 引脚还需设置相应的端口控制寄存器，将该引脚设置为输入口。

标准型 TM 框图

标准型 TM 操作

标准型 TM 的核心是一个由用户选择的内部或外部时钟源驱动的 16-bit 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 8 位宽度，与计数器的高 8 位比较；而 CCRA 是 16 位的，与计数器的所有位比较。

通过应用程序改变 16 位计数器值的唯一方法是使 STON 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 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 16-bit 计数器的值，一对读 / 写寄存器存放 16-bit 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 8-bit CCRP 的值。

寄存器名称	位							
	7	6	5	4	3	2	1	0
STMC0	STPAU	STCK2	STCK1	STCK0	STON	—	—	—
STMC1	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
STMDL	D7	D6	D5	D4	D3	D2	D1	D0
STMDH	D15	D14	D13	D12	D11	D10	D9	D8
STMAL	D7	D6	D5	D4	D3	D2	D1	D0
STMAH	D15	D14	D13	D12	D11	D10	D9	D8
STMRP	D7	D6	D5	D4	D3	D2	D1	D0

16 位标准型 TM 寄存器列表

● STMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	STPAU	STCK2	STCK1	STCK0	STON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **STPAU**: STM 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，STM 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 **STCK2~STCK0**: 选择 STM 计数时钟位

000: $f_{SYS}/4$

001: f_{SYS}

010: $f_H/16$

011: $f_H/64$

100: f_{SUB}

101: 保留位

110: STCK 上升沿时钟

111: STCK 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_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **STON**: STM 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变高。

若 TM 处于比较匹配输出模式时，当 STON 位经由低到高转换时，STM 输出脚将复位至 STOC 位指定的初始值。

Bit 2~0 未定义，读为“0”

● STMC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
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 **STM1~STM0**: 选择 STM 工作模式位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 输出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 STM1 和 STM0 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 **STIO1~STIO0**: 选择 STM 外部引脚输出功能位

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 输出模式 / 单脉冲输出模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 STP 上升沿输入捕捉
- 01: 在 STP 下降沿输入捕捉
- 10: 在 STP 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 外部引脚如何改变状态。这两位值的选择决定 TM 运行在何种模式下。

在比较匹配输出模式下, STIO1 和 STIO0 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 STOC 位设置取得。注意, 由 STIO1 和 STIO0 位得到的输出电平必须与通过 STOC 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 STON 位由低到高电平的转换复位至初始值。

在 PWM 模式, STIO1 和 STIO0 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 TM 关闭时改变 STIO1 和 STIO0 位的值是很有必要的。若在 TM 运行时改变 STIO1 和 STIO0 的值, PWM 输出的值是无法预料的。

Bit 3 **STOC**: STM 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 输出模式 / 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 **STPOL**: STM 输出极性控制位
0: 同相
1: 反相
此位控制 TM 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 STM 处于定时 / 计数器模式时其不受影响。
- Bit 1 **STDPX**: STM 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 **STCCLR**: 选择 STM 计数器清零条件位
0: 比较器 P 匹配
1: 比较器 A 匹配
此位用于选择清除计数器的方法。标准型 STM 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。STCCLR 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。STCCLR 位在 PWM 输出、单脉冲或输入捕捉模式时未使用。

• STMDL 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 **D7~D0**: STM 计数器低字节寄存器 bit 7 ~ bit 0
STM 16-bit 计数器 bit 7 ~ bit 0

• STMDH 寄存器

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 **D15~D8**: STM 计数器高字节寄存器 bit 7 ~ bit 0
STM 16-bit 计数器 bit 15 ~ bit 8

• STMAL 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 **D7~D0**: STM CCRA 低字节寄存器 bit 7 ~ bit 0
STM 16-bit CCRA bit 7 ~ bit 0

• STMAH 寄存器

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 **D15~D8:** STM CCRA 低字节寄存器 bit 7 ~ bit 0
STM 16-bit CCRA bit15 ~ bit 8

• STMRP 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 **STMRP:** STM CCRP 高字节寄存器 bit 7~bit 0
STM CCRP 8-bit 寄存器，与 STM 计数器 bit-15~bit 8 相比较。
比较器 P 匹配周期
0: 65536 个 STM 时钟
1~255: $256 \times (1 \sim 255)$ 个 STM 时钟
这八位用于设定内部 CCRP 8-bit 寄存器的值，然后将其与内部计数器高八位进行比较。当 STCCLR 位设为 0 时，可用比较结果清除计数器。STCCLR 位设为 0，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高八位比较，比较结果是 256 时钟周期的倍数。CCRP 被清零时，会使得计数器在最大值溢出。

标准型 TM 工作模式

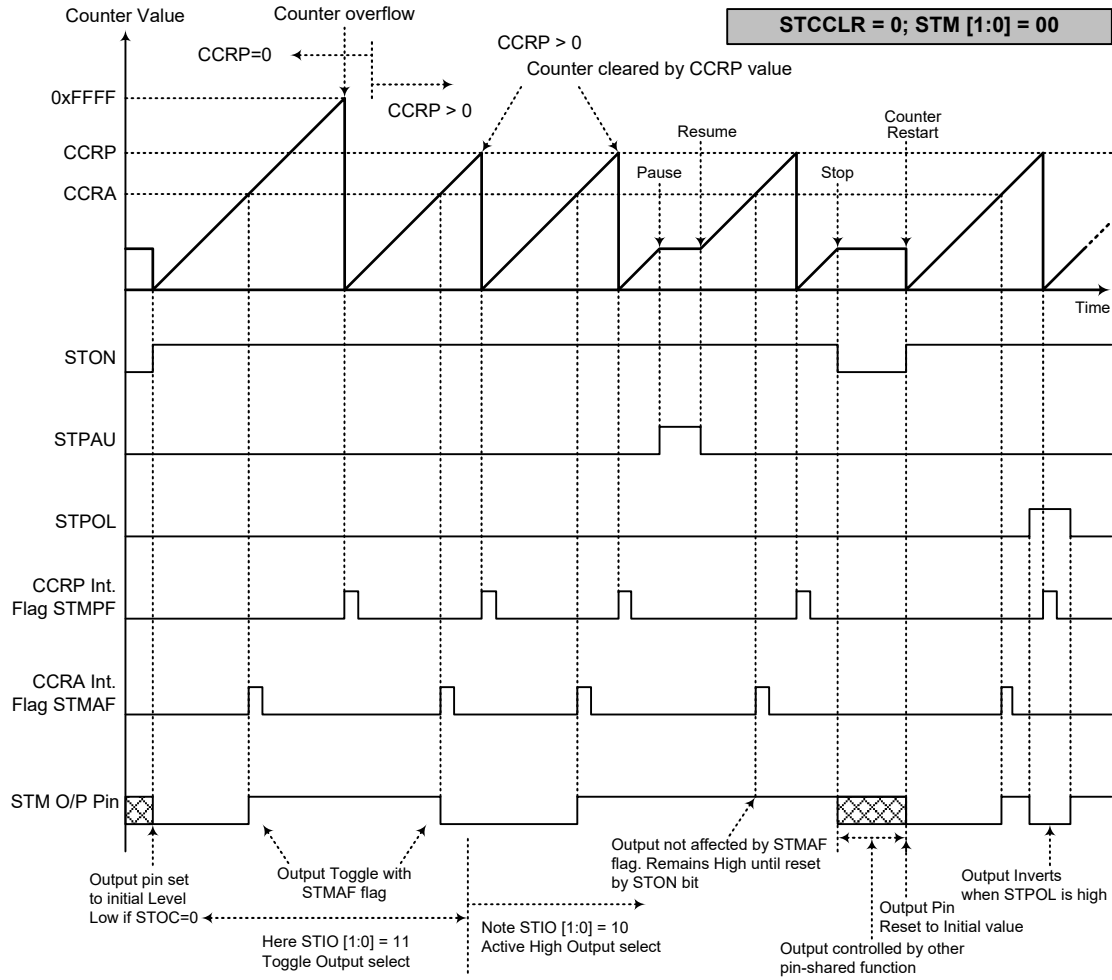
标准型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时 / 计数器模式。通过设置 STMC1 寄存器的 STM1 和 STM0 位选择任意模式。

比较匹配输出模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 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 STCCLR 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 STMAF 和 STMPF 将分别置位。

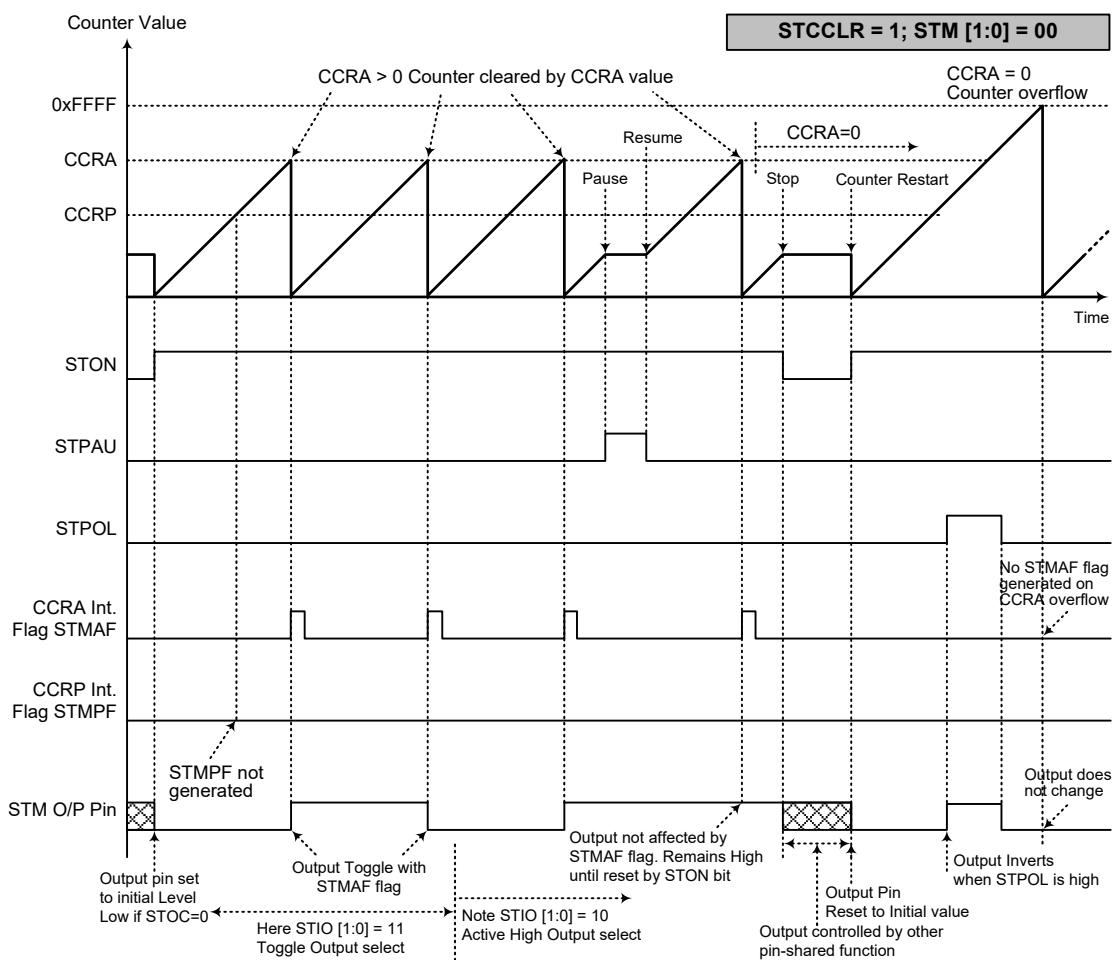
如果 STMC1 寄存器的 STCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅产生 STMAF 中断请求标志。所以当 STCCLR 为高时，不会产生 STMPF 中断请求标志。在比较匹配输出模式下，CCRA 不能设为“0”。

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 STMAF 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 STMPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 STMC1 寄存器中 STIO1 和 STIO0 位决定。当比较器 A 比较匹配发生时，STIO1 和 STIO0 位决定 TM 输出脚输出高，低或翻转当前状态。在 STON 位由低到高后，TM 输出脚初始状态为 STOC 位所指定的电平。注意，若 STIO1 和 STIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – STCCLR=0

- 注：1. STCCLR=0，比较器 P 匹配将清除计数器
2. TM 输出引脚仅由 STMAF 标志位控制
3. 输出引脚通过 STON 位上升沿复位至初始状态



比较匹配输出模式 – STCCLR=1

- 注：1. STCCLR=1 时，比较器 A 匹配将清除计数器
2. TM 输出引脚由 STMAF 标志位控制
3. 输出引脚通过 STON 上升沿复位至初始状态。
4. 当 STCCLR=1 时，不产生 STMPF。

定时 / 计数器模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 应设置为 11。定时 / 计数器模式操作方式与比较匹配输出模式相同，产生相同的中断标志位。唯一不同的就是在定时 / 计数器模式中未使用 TM 输出引脚。因此，在该模式下此引脚可用作普通 I/O 脚或其他功能。

PWM 输出模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”，且 STIO1 和 STIO0 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，STCCLR 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 STMC1 寄存器的 STDPX 位。

因此，PWM 波形的频率和占空比受 CCRA 和 CCRP 寄存器中的值控制。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。STMC1 寄存器中的 STOC 位决定 PWM 波形的极性，STIO1 和 STIO0 位使能 PWM 输出或将 TM 输出脚置为逻辑高或逻辑低。STnPOL 位对 PWM 输出波形的极性取反。

● 16-bit STM, PWM 模式，边沿对其模式，STDPX=0

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

若 $f_{SYS}=16\text{MHz}$ ，TM 时钟源为 $f_{SYS}/4$ ，CCRP=2 和 CCRA=128，

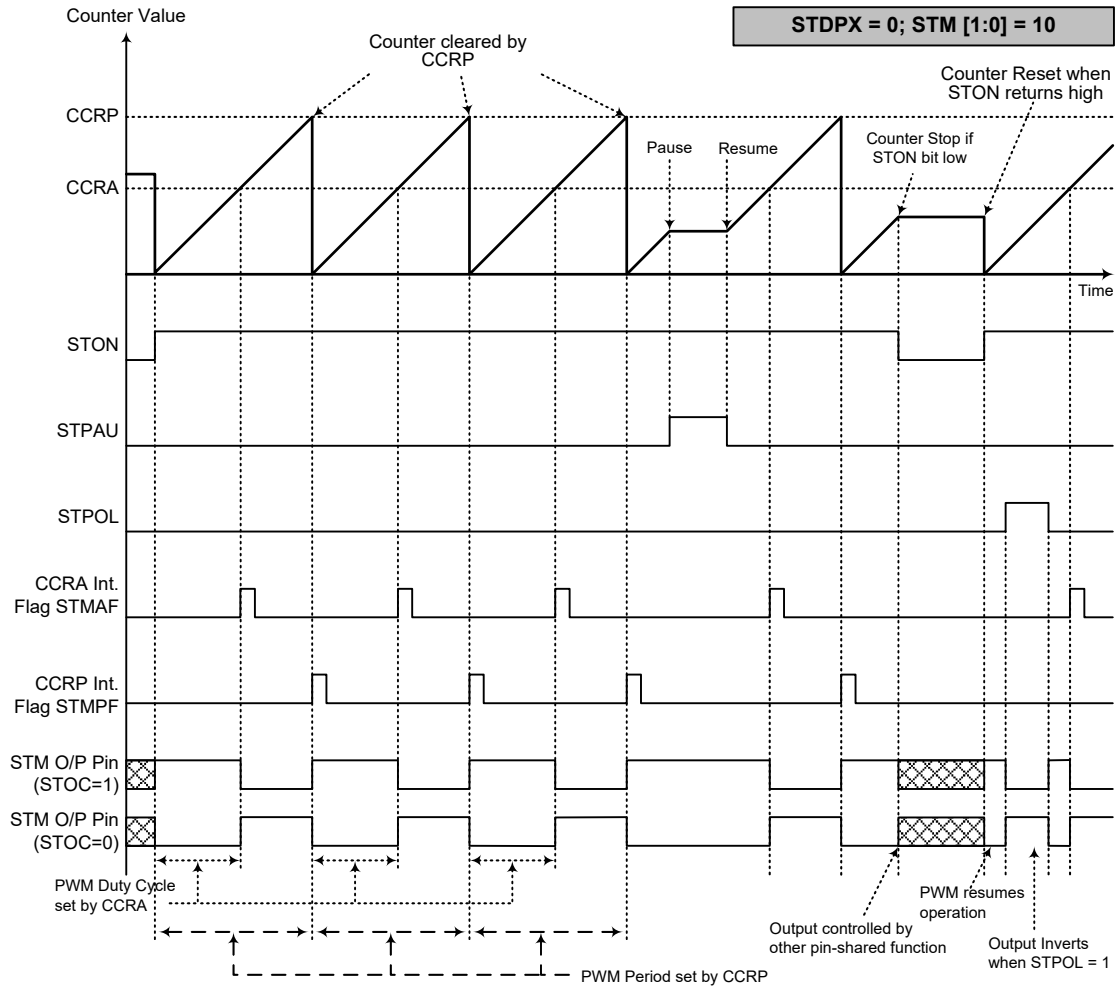
STM PWM 输出频率 = $(f_{SYS}/4)/512=f_{SYS}/2048=7.8125\text{kHz}$ ，周期 = $128/512=25\%$ 。

若 CCRA 所定义的 Duty 值等于或大于 Period 值，则 PWM 输出占空比为 100%。

● 16-bit STM, PWM 模式，边沿对其模式，STDPX=1

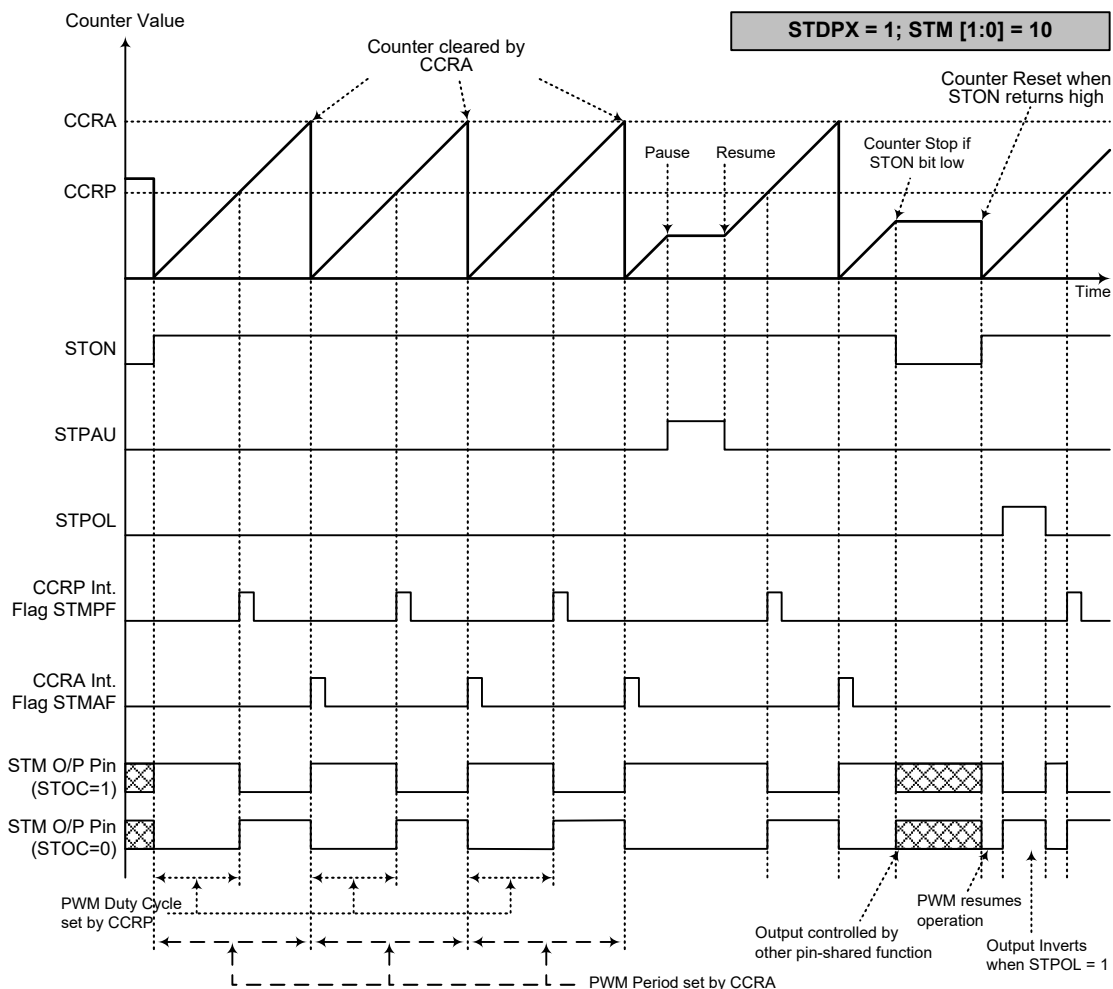
CCRP	1~255	0
Period	CCRA	
Duty	CCRP×256	65536

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 输出模式 – STDPX=0

- 注：1. STDPX=0, CCRP 清除计数器
2. 计数器清零设置 PWM 周期
3. 当 STIO[1:0]=000000 或 01 时，内部 PWM 功能继续运行
4. STCCLR 位对 PWM 操作无影响



PWM 输出模式 – STDPX=1

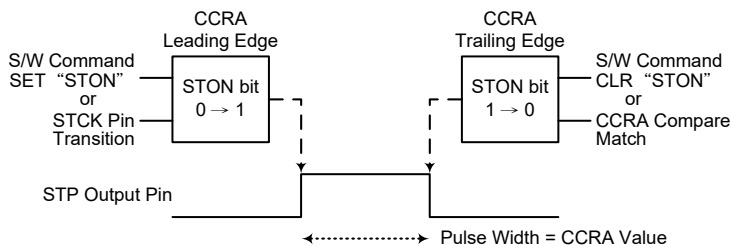
- 注：1. STDPX=1, CCRA 清除计数器
2. 计数器清零设置 PWM 周期
3. 当 STIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. STCCLR 位对 PWM 操作无影响

单脉冲模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”，同时 STIO1 和 STIO0 位需要设置为“11”。正如模式名所言，单脉冲输出模式，在 TM 输出脚将产生一个单脉冲输出。

脉冲输出可以通过应用程序控制 STON 位由低到高的转变来触发。而处于单脉冲模式时，STON 位可利用 STCK 脚自动由低转变为高，进而开始单脉冲输出。

当 STON 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 STON 位保持高电平。通过应用程序使 STON 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



- 注：1. 通过 CCRA 匹配停止计数器
2. CCRP 未使用
3. 通过设置 STCK 引脚或设置 STON 位为高来触发脉冲
4. 在单脉冲模式下，STIO[1:0] 必须设置为“11”，且不能改变。

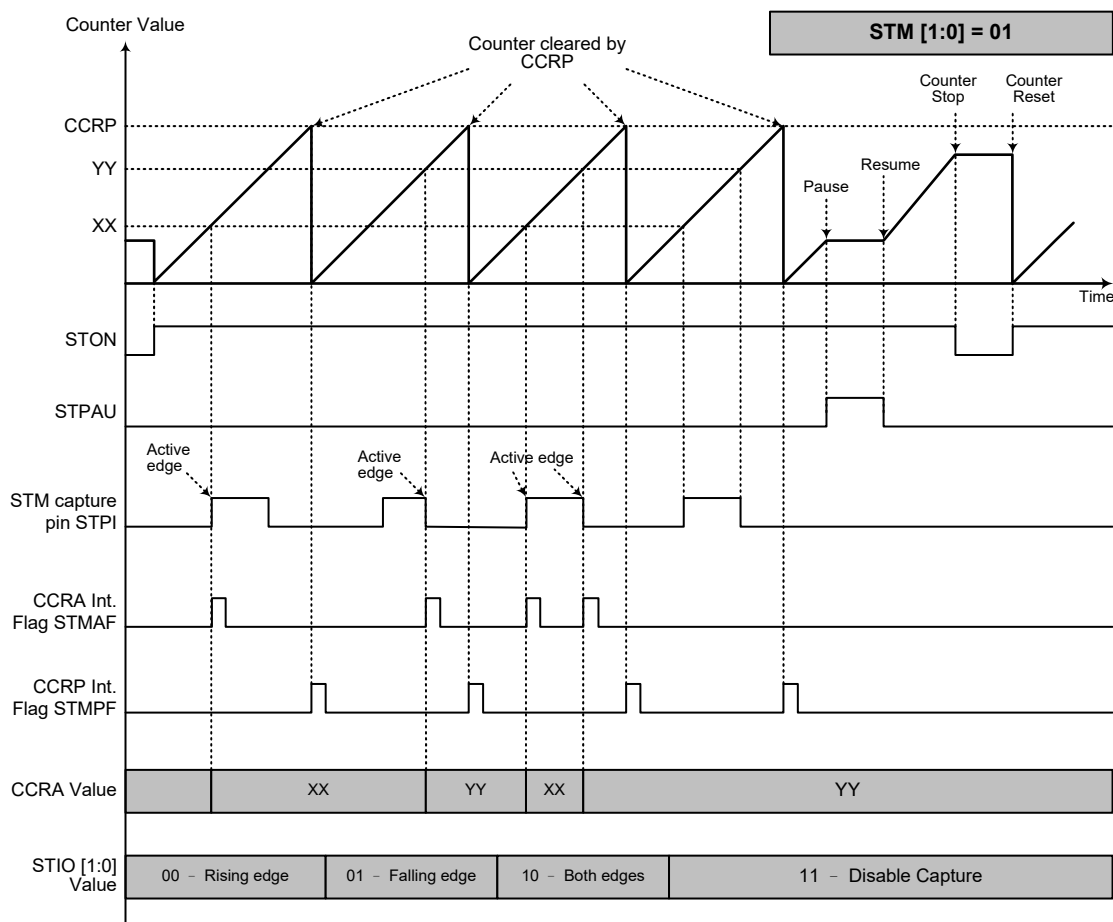
但比较器 A 的比较匹配也会自动清除 STON 位，从而产生单脉冲输出下降沿。此时 CCRA 的值可用于控制脉冲宽度。比较器 A 的比较匹配也能产生一个 TM 中断信号。当计数器重新启动，STON 位从低到高转换时，计数器将被复位回 0。在单脉冲模式下，CCRP，STCCLR 位和 STDPX 位未使用。

捕捉输入模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 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。STP 脚上的外部信号，通过设置 STMC1 寄存器的 STIO1 和 STIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。通过应用程序将 STON 位由低到高转变时，计数器启动。

当 STP 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。无论 STP 引脚发生哪种边沿转换，计数器将继续工作直到 STON 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 STIO1 和 STIO0 位选择 STP 引脚为上升沿，下降沿或双沿有效。如果 STIO1 和 STIO0 位都设置为高，无论 STP 引脚发生哪种边沿转换都不会产生捕捉操作，但计数器将会继续运行。

有几点注意事项须留意。如果 STCK 用作捕捉输入源，则不能将其选作 STM 的时钟源。如果捕捉脉宽小于 2 个定时器时钟周期，则可能会被硬件忽略。当计数器的值被有效捕捉边沿锁存到 CCRA 或 CCRB 寄存器后，再过 0.5 个定时器时钟周期，STMAF 标志位将被置高。从接收到有效捕捉边沿，到开始将计数器值锁存到 CCRA 或 CCRB 寄存器的动作，这之间的延迟时间小于 1.5 个定时器时钟周期。STCCLR 和 STDPX 位在此模式中未使用。



捕捉输入模式

- 注：1. STM[1:0]=01，有效边沿通过 STIO[1:0] 位设置
2. TM 捕捉输入脚有效边沿将计数器值传送到 CCRA 中
3. STCCLR 位未使用
4. 无输出功能 – STOC 和 STPOL 位未使用
5. CCRP 决定计数器的值，且当 CCRP 等于 0 时计数器有一个最大计数值

周期型 TM – PTM

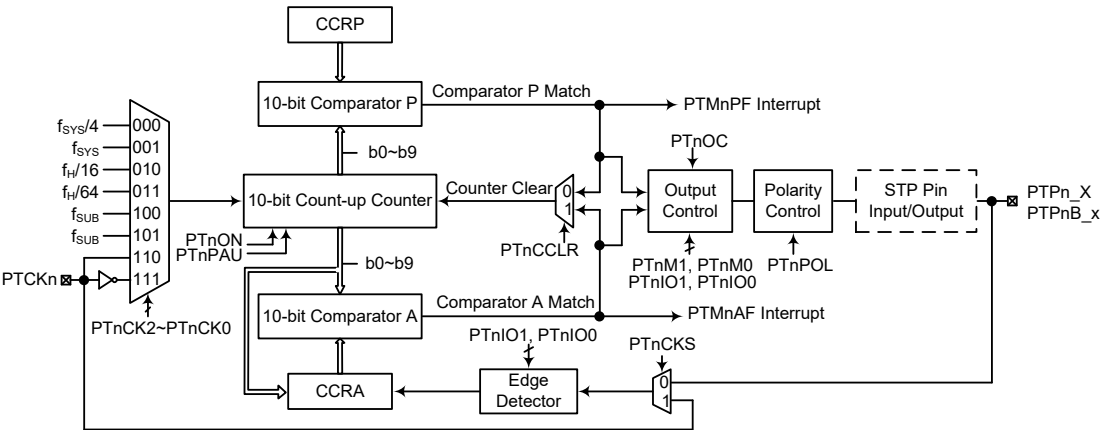
周期型 TM 包括 5 种工作模式，即比较匹配输出，定时 / 事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。周期型 TM 也由一个外部输入脚控制并驱动两个到五个外部输出脚。

单片机型号	TM 类型	TM 输入引脚	TM 输出引脚
HT69F340	10-bit PTM	PTCK	PTP_0, PTP_1, PTP_2 PTPB_0, PTPB_1
HT69F350	10-bit PTM	PTCK	PTP_0, PTP_1 PTPB_0, PTPB_1
HT69F360/ HT67F370	10-bit PTM0 10-bit PTM1	PTCK0 PTCK1	PTP0, PTP0B PTP1, PTP1B

周期型 TM 操作

周期型 TM 多达两个，都是 10 位宽度。其核心为 10-bit 向上计数器，由用户选择的内部或外部时钟源驱动。它还包括两个内部比较器即比较器 A 和比较器 P。这三个比较器将计数器的值与 CCRA 和 CCRP 寄存器中的值进行比较。CCRP 的宽度为 10 位。

通过应用程序改变 10-bit 计数器值的唯一方法是使 PTnON 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。周期型 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内部寄存器来实现的。



周期型 TM 方框图 (n 仅对 HT69F360/HT67F370 有效, n=0, 1)

周期型 TM 寄存器介绍

周期型 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-bit 内部计数器的值，两对读 / 写寄存器存放 10-bit CCRA 和 CCRP 的值。剩下两个控制寄存器用来设置不同的操作和控制模式。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PTMnC0	PTnPAU	PTnCK2	PTnCK1	PTnCK0	PTnON	—	—	—
PTMnC1	PTnM1	PTnM0	PTnIO1	PTnIO0	PTnOC	PTnPOL	PTnCKS	PTnCCLR
PTMnDL	D7	D6	D5	D4	D3	D2	D1	D0
PTMnDH	—	—	—	—	—	—	D9	D8
PTMnAL	D7	D6	D5	D4	D3	D2	D1	D0
PTMnAH	—	—	—	—	—	—	D9	D8
PTMnRPL	D7	D6	D5	D4	D3	D2	D1	D0
PTMnRPH	—	—	—	—	—	—	D9	D8

10 位周期型 TM 寄存器列表

• PTMnC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PTnPAU	PTnCK2	PTnCK1	PTnCK0	PTnON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **PTnPAU**: PTMn 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 **PTnCK2~PTnCK0**: 选择 PTMn 计数时钟位

000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: f_{SUB}
110: PTCKn 上升沿时钟
111: PTCKn 下降沿时钟

此三位用于选择 TM 的时钟源。外部引脚时钟源可被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **PTnON**: PTMn 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 PTM 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转高。若 TM 处于比较匹配输出模式时，当 PTnON 位经由低到高转换时，TM 输出脚将复位至 PTnOC 位指定的初始值。

Bit 2~0 未定义，读为“0”

● PTMnC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PTnM1	PTnM0	PTnIO1	PTnIO0	PTnOC	PTnPOL	PTnCKS	PTnCCLR
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 **PTnM1~PTnM0**: PTMn 工作模式选择位

- 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

这两位为 TM 设置所需工作模式。为确保操作可靠，应在 PTnM1 和 PTnM0 位发生任何改变之前关闭 TM。在定时 / 计数器模式中，TM 输出引脚状态未定义。

Bit 5~4 **PTnIO1~PTnIO0**: PTPn 外部引脚功能选择位

比较匹配输出模式

- 00: 无改变
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 模式 / 单脉冲输出模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 PTPn 或 PTCKn 上升沿输入捕捉
- 01: 在 PTPn 或 PTCKn 下降沿输入捕捉
- 10: 在 PTPn 或 PTCKn 双沿输入捕捉
- 11: 输入捕捉除能

定时 / 计数器模式

未使用

这两位用于确定在达到一定条件时 TM 外部引脚如何改变状态。此位段所选择的功能取决于 TM 当下的工作模式。在比较匹配输出模式中，当比较器 A 发生比较匹配时，PTnIO1~PTnIO0 决定了 TM 输出脚的状态应如何改变。当比较器 A 发生比较匹配时，TM 输出脚可以设置为切换高，切换低或翻转当前状态。当这两位都为零，则输出将不会发生改变。TM 输出脚的初始值应通过 PTMnC1 寄存器中的 PTnOC 位进行设置。注意，由 PTnIO1 和 PTnIO0 位所确定的输出电平必须与利用 PTnOC 位所设置的初始值不同，否则在发生比较匹配时 TM 输出脚上将不发生改变。在 TM 输出脚改变状态后，可通过改变 PTnON 位的电平从低变高将其复位至初始值。

Bit 3 **PTnOC**: PTPn 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

此位为 TM 输出脚的输出控制位。其操作取决于 TM 被使用在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 / 计数器模式下则不受此位影响。在比较匹配输出模式中，此位决定了在比较匹配发生之前 TM 输出脚的逻辑电平。在 PWM 模式中，此位决定了 PWM 信号是为高有效还是低有效。

Bit 2 **PTnPOL**: PTPn 输出极性控制位

- 0: 同相
- 1: 反相

此位控制 PTPn 输出引脚的极性。当此位置高，TM 输出脚将被反相，而此位为零时则为同相。若 TM 处于定时 / 计数器模式则对其无影响。

- Bit 1 **PTnCKS**: 输入捕捉触发源选择位
0: 捕捉输入模式的外部时钟源来自 PTPn
1: 捕捉输入模式的外部时钟源来自 PTCKn

- Bit 0 **PTnCCLR**: PTMn 计数器清除条件选择位
0: PTMn 比较器 P 匹配
1: PTMn 比较器 A 匹配

此位用于选择清除计数器的方法。记住周期型 TM 包含两个比较器，即比较器 A 和比较器 P，两者中的任意一个都可以清除内部计数器。当 PTnCCLR 位置高，在比较器 A 发生比较匹配时计数器将被清除。当此位为低，计数器将在比较器 P 发生比较匹配或是计数器溢出时被清除。只有在 CCRP 位都被清零时才可执行计数器溢出清除方式。PTnCCLR 位在 PWM，单脉冲或输入捕捉模式中未使用。

• PTMnDL 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 **D7~D0**: PTMn 计数器低字节寄存器 bit 7~bit 0
PTMn 10 位计数器 bit 7~bit 0

• PTMnDH 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 **D9~D8**: PTMn 计数器高字节寄存器 bit 1~bit 0
PTMn 10 位计数器 bit 9~bit 8

• PTMnAL 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 **D7~D0**: PTMn CCRA 低字节寄存器 bit 7~bit 0
PTMn 10 位 CCRA bit 7~bit 0

• PTMnAH 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 **D9~D8**: PTMn CCRA 高字节寄存器 bit 1~bit 0
PTMn 10 位 CCRA bit 9~bit 8

● PTMnRPL 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 **D7~D0:** PTMn CCRP 寄存器 bit 7~bit 0
PTMn 10 位 CCRP bit 7~bit 0

● PTMnRPL 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 **D9~D8:** PTMn CCRP 寄存器 bit 1~bit 0
PTMn 10 位 CCRP bit 9~bit 8

周期型 TM 工作模式

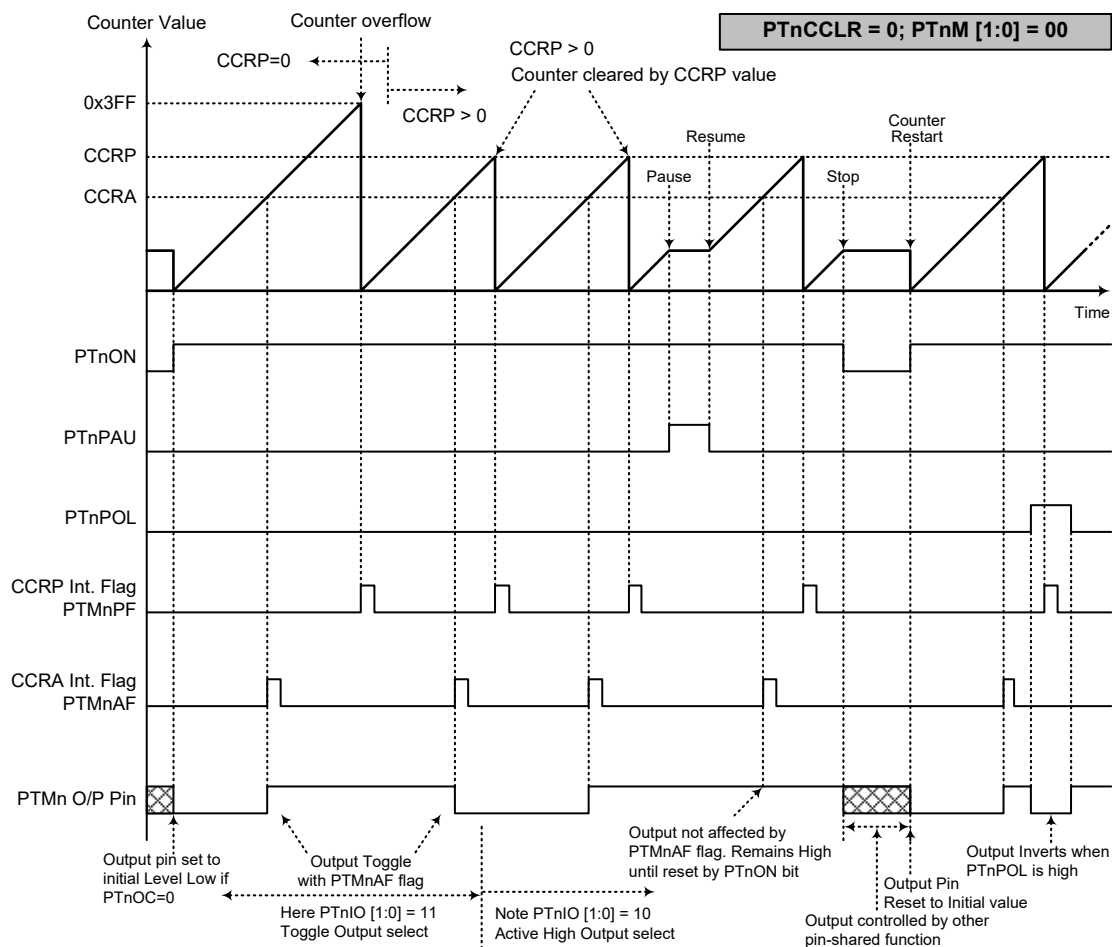
周期型 TM 有五种工作模式，即比较匹配输出模式，PWM 输出模式，单脉冲输出模式，捕捉输入模式或定时 / 计数器模式。通过设置 PTMnC1 寄存器的 PTnM1 和 PTnM0 位选择任意模式。

比较输出模式

要工作在此模式，PTMnC1 寄存器的 PTnM1 和 PTnM0 位的值应设为“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 PTnCCLR 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 PTMnAF 和 PTMnPF 将分别置起。

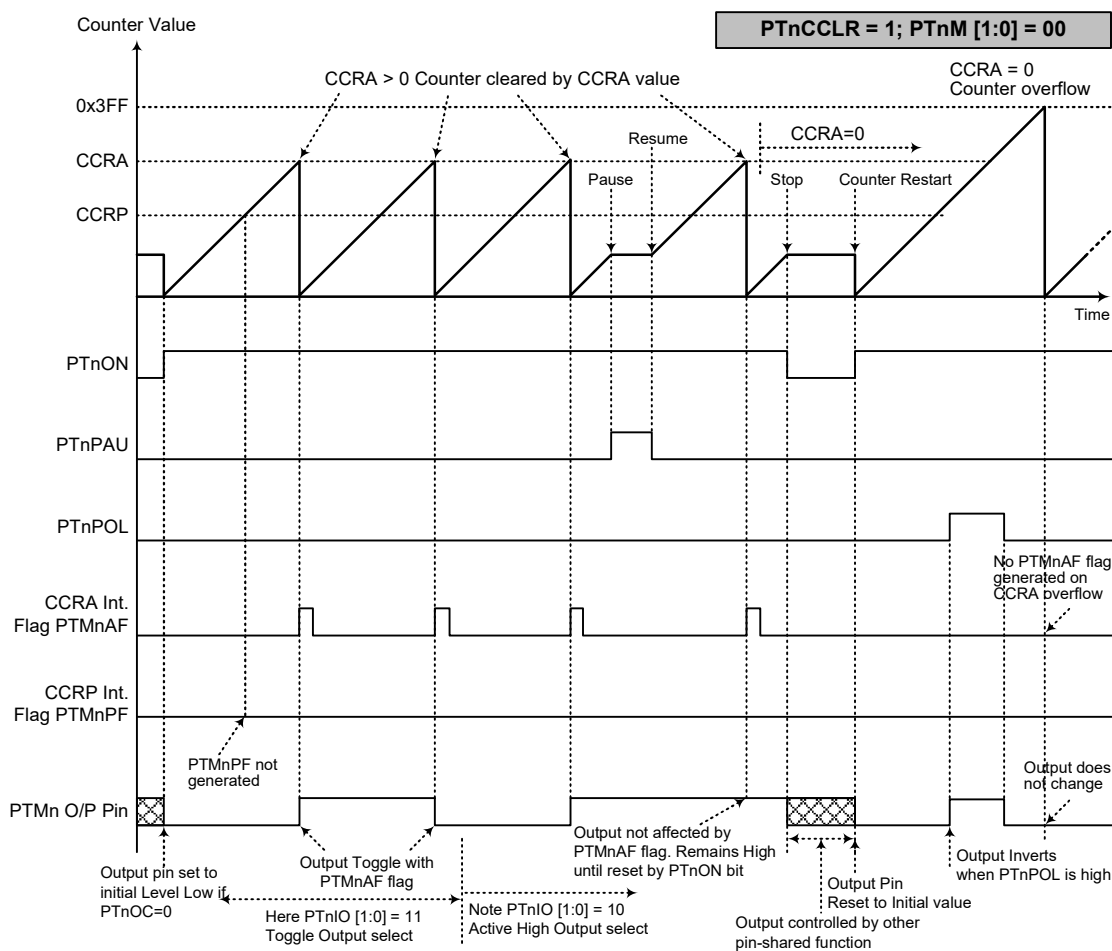
如果 PTMnC1 寄存器的 PTnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 PTMnAF 中断请求标志产生。所以当 PTnCCLR 为高时，不会产生 PTMnPF 中断请求标志。在比较匹配输出模式中，CCRA 不能被设置为“0”。

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 PTMnAF 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 PTMnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 PTMnC1 寄存器中 PTnIO1 和 PTnIO0 位决定。当比较器 A 比较匹配发生时，利用 PTnIO1 和 PTnIO0 位决定 TM 输出脚输出高，低或翻转当前状态。在 PTON 位由低到高后，TM 输出脚初始状态为 PTOC 位所指定的电平。注意，若 PTnIO1 和 PTnIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – PTnCCLR=0

- 注:
1. PTnCCLR=0, 比较器 P 匹配将清除计数器
 2. TM 输出脚仅由 PTMnAF 标志位控制
 3. 输出脚在 PTnON 位上升沿复位至其初始状态
 4. n=0, 1



比较匹配输出模式 – PTnCCLR=1

- 注：1. PTnCCLR=1，比较器 A 匹配将清除计数器
2. TM 输出脚仅由 PTMnAF 标志位控制
3. 输出脚在 PTnON 位上升沿复位至其初始状态
4. 当 PTnCCLR=1，不产生 PTMnPF 标志位
5. n=0, 1

定时 / 计数器模式

要工作在此模式，PTMnC1 寄存器的 PTnM1 和 PTnM0 位的值需设为“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

要工作在此模式，PTMnC1 寄存器的 PTnM1 和 PTnM0 位的值需设置为“10”，且 PTnIO1 和 PTnIO0 位的值也需设置为“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，PTnCCLR 位对 PWM 周期无效。CCRA 和 CCRP 寄存器共同产生 PWM 波形，一个用来清除内部计数器从而控制 PWM 波形频率，另一个则用于控制占空比。因此，PWM 波形频率和占空比可由 CCRA 和 CCRP 寄存器中的值来控制。

当比较器 A 或比较器 P 发生比较匹配时，将产生 CCRA 和 CCRP 的中断标志位。PTMnC1 寄存器的 PTnOC 位用于选择 PWM 波形所需极性，而 PTnIO1 和 PTnIO0 位用于使能 PWM 输出或将 TM 输出脚强制固定为高或低电平。PTnPOL 位用于取反 PWM 输出波形的极性。

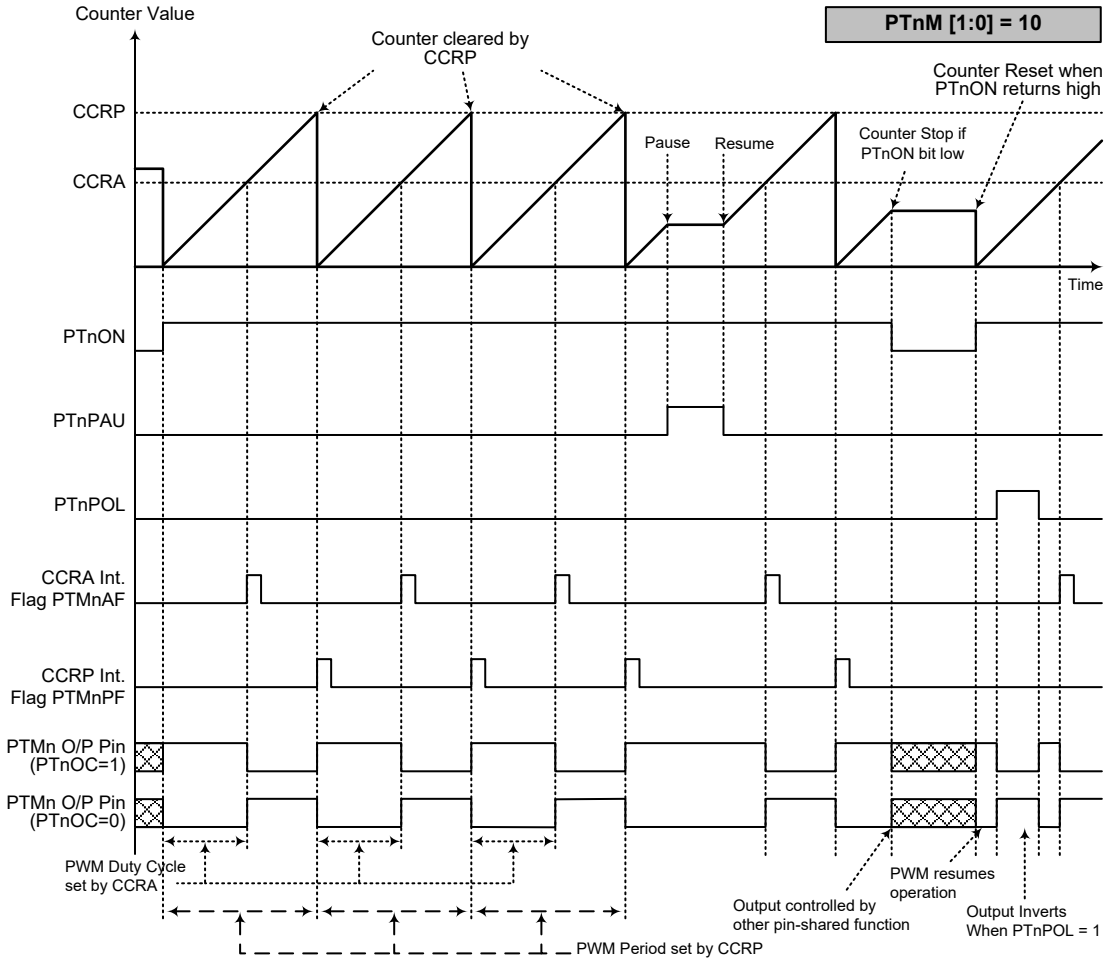
● 10-bit PWM 模式，边沿对其模式

CCRP	0	1~1023
Period	1024	1~1023
Duty	CCRA	

若 $f_{SYS}=16\text{MHz}$ ，TM 时钟源选择 $f_{SYS}/4$ ，CCRP=512，CCRA=128。

PTM PWM 输出频率 = $(f_{SYS}/4)/512=f_{SYS}/2048=708125\text{kHz}$ ，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%。



PWM 输出模式

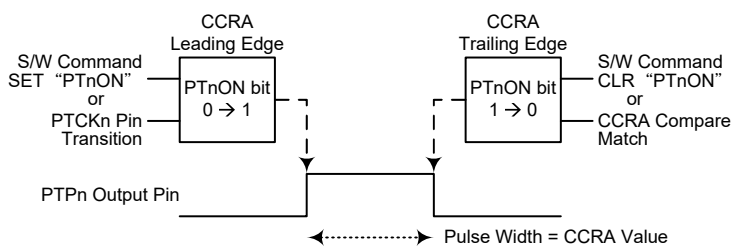
- 注：1. 计数器清除，设置 PWM 周期
2. 即使在 PTnIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
3. PTnCCLR 位对 PWM 操作无影响
4. n=0, 1

单脉冲输出模式

要在此模式，PTMnC1 寄存器的 PTnM1 和 PTnM0 位的值需设置为“10”，且 PTnIO1 和 PTnIO0 位的值也需设置为“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 PTnON 位由低到高的转变来触发脉冲前沿输出。但在单脉冲模式中，PTnON 位可利用外部 PTCKn 脚而自动由低转变为高，进而启动单脉冲输出。当 PTnON 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 PTnON 位保持高电平。通过应用程序使 PTnON 位清零或比较器 A 比较匹配发生时，产生脉冲下降沿。

而比较器 A 比较匹配发生时，会自动清除 PTnON 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，会产生 TM 中断。PTnON 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 PTnCCLR 位未使用。



单脉冲产生示意图 (n=0, 1)

捕捉输入模式

要工作在此模式，PTMnC1 寄存器的 PTnM1 和 PTnM0 位的值需设置为“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。PTPn 和 PTCKn 引脚上的外部信号可通过设置 PTMnC1 寄存器的 PTnCKS 位进行选择。输入引脚有效边沿可为上升沿，下降沿或双沿有效。有效边沿转换类型可通过设置 PTMnC1 寄存器的 PTnIO1 和 PTnIO0 位进行选择。当利用应用程序启动 PTnON 位由低到高转变时，计数器开始运行。

当 PTPn 和 PTCKn 引脚出现所需边沿转换时，计数器当前值被锁存到 CCRA 寄存器，并产生 TM 中断。无论 PTPn 和 PTCKn 引脚发生什么事件，计数器继续工作直到 PTnON 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。计数 CCRP 溢出中断信号的值可以测量长脉宽。如果 PTnIO1 和 PTnIO0 位都设为高，则无论 PTPn 和 PTCKn 引脚发生什么事件都不会产生捕捉操作，但必须注意，计数器将继续运行。

有几点注意事项须留意。如果 PTCKn 用作捕捉输入源，则不能将其选作 PTMn 的时钟源。如果捕捉脉宽小于 2 个定时器时钟周期，则可能会被硬件忽略。当计数器的值被有效捕捉边沿锁存到 CCRA 或 CCRB 寄存器后，再过 0.5 个定时器时钟周期，PTMnAF 标志位将被置高。从接收到有效捕捉边沿，到开始将计数器值锁存到 CCRA 或 CCRB 寄存器的动作，这之间的延迟时间小于 1.5 个定时器时钟周期。PTnCCLR, PTnOC 和 PTnPOL 位在此模式中未使用。

A/D 转换器 – HT67F370

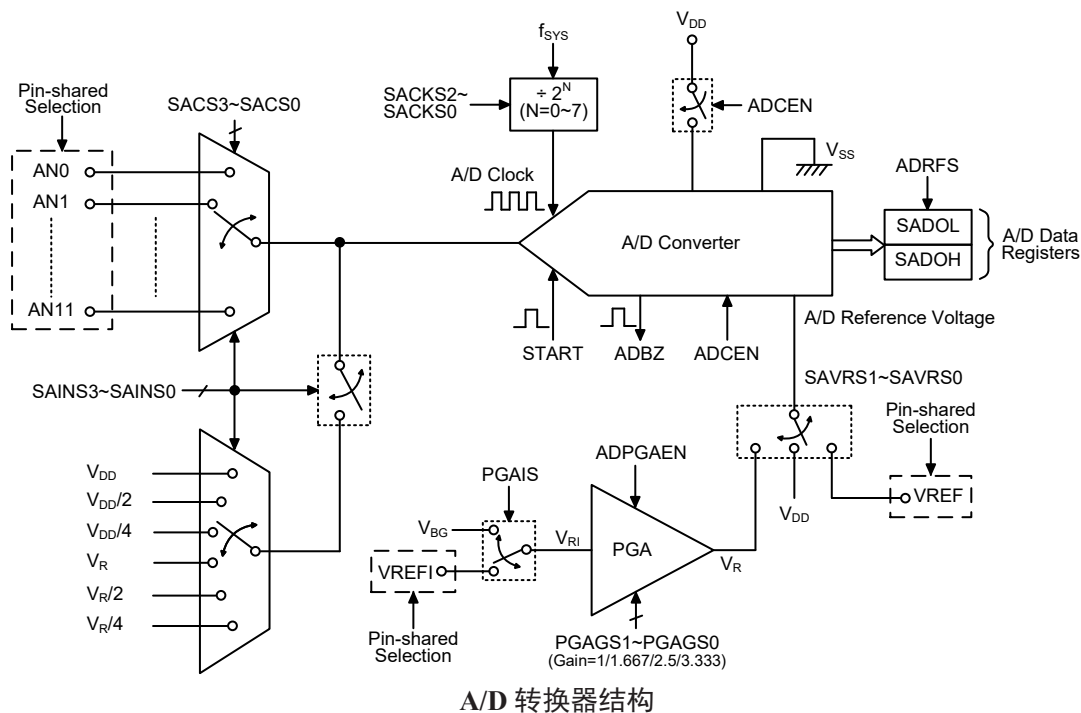
对于大多数电子系统而言，处理现实世界的模拟信号是共同的需求。为了完全由单片机来处理这些信号，首先需要通过 A/D 转换器将模拟信号转换成数字信号。将 A/D 转换器电路集成入单片机，可有效的减少外部器件，随之而来，具有降低成本和减少器件空间需求的优势。

A/D 转换器简介

HT67F370 包含一个多通道的 A/D 转换器，它们可以直接接入外部模拟信号 (来自传感器或其它控制信号) 或内部模拟信号并直接将这些信号转换成 12 位的数字量。选择转换外部或内部模拟信号由 SAINS3~SAINS0 位和 SACS3~SACS0 位共同控制。若需要转换外部模拟信号时，需要先正确设置相关的引脚共用功能选择寄存器，再通过 SAINS3~SAINS0 位和 SACS3~SACS0 位选择所需转换的通道。注意，若需要转换内部信号，除了 SAINS3~SAINS0 位和 SACS3~SACS0 位外，一些引脚共用控制位也需合理设置。关于 A/D 输入信号的详细描述请参考“A/D 转换器控制寄存器”和“A/D 转换器输入信号”两节内容。

外部输入通道	内部信号	A/D 通道选择位
12: AN0~AN11	6: V_{DD} , $V_{DD}/2$, $V_{DD}/4$, V_R , $V_R/2$, $V_R/4$	SAINS3~SAINS0, SACS3~SACS0

下面框图显示了 A/D 转换器内部结构及其相关寄存器。



A/D 转换器寄存器介绍

A/D 转换器的所有工作由五个寄存器控制。一对只读寄存器来存放 12-bit A/D 转换数据的值，剩下的三个控制寄存器设置 A/D 转换器的操作和控制功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SADOL (ADRF5=0)	D3	D2	D1	D0	—	—	—	—
SADOL (ADRF5=1)	D7	D6	D5	D4	D3	D2	D1	D0
SADOH (ADRF5=0)	D11	D10	D9	D8	D7	D6	D5	D4
SADOH (ADRF5=1)	—	—	—	—	D11	D10	D9	D8
SADC0	START	ADBZ	ADCEN	ADRF5	SACS3	SACS2	SACS1	SACS0
SADC1	SAINS3	SAINS2	SAINS1	SAINS0	—	SACKS2	SACKS1	SACKS0
SADC2	ADPGAEN	—	—	PGAIS	SAVRS1	SAVRS0	PGAGS1	PGAGS0

A/D 转换器寄存器列表

A/D 转换器数据寄存器 – SADOL, SADOH

对于具有 12 位 A/D 转换器的芯片，需要两个数据寄存器存放转换结果，一个高字节寄存器 SADOH 和一个低字节寄存器 SADOL。在 A/D 转换完毕后，单片机可以直接读取这些寄存器以获得转换结果。由于寄存器只使用了 16 位中的 12 位，其数据存储格式由 SADC0 寄存器的 ADRFS 位控制，如下表所示。D0~D11 是 A/D 转换数据结果位。未使用的位读为“0”。当 A/D 转换器除能时，数据寄存器的值将保持不变。

ADRF5	SADOH								SADOL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

A/D 转换器数据寄存器

A/D 转换器控制寄存器 – SADC0, SADC1, SADC2

寄存器 SADC0~SADC2 用来控制 A/D 转换器的功能和操作。这些 8 位的寄存器定义包括选择连接至内部 A/D 转换器的模拟通道，数字化数据格式，A/D 时钟源，并控制 A/D 转换器的开启和监视 A/D 转换器的忙碌状态。由于每个单片机只包含一个实际的模数转换电路，因此这些外部和内部模拟信号中的每一个都需要分别被发送到转换器。SADC0 寄存器中的 SACS3~SACS0 位用于选择哪个外部模拟输入通道被连接到内部 A/D 转换器。SADC1 寄存器中的 SAINS3~SAINS0 位用于选择外部模拟输入通道或内部模拟信号被连接到内部 A/D 转换器。A/D 转换器还包含了一个可编程增益放大器 PGA 用于产生 A/D 转换器内部参考电压。PGA 的所有操作由 SADC2 寄存器控制。

引脚共用功能选择寄存器的相关位用来定义 I/O 端口中的哪些引脚为 A/D 转换器的模拟输入，哪些引脚不作为 A/D 转换输入。当引脚作为 A/D 输入时，其原来的 I/O 或其它引脚共用功能消失，此外，其内部上拉电阻也将自动断开。

● SADC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	START	ADBZ	ADCEN	ADRF5	SACS3	SACS2	SACS1	SACS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **START**: 启动 A/D 转换位
0→1→0: 启动
此位用于启动 A/D 转换过程。通常此位为低，但如果设为高再被清零，将启动 A/D 转换过程。当该位设置为高，A/D 转换器将被复位。
- Bit 6 **ADBZ**: A/D 转换忙碌标志位
0: A/D 转换结束或未开始转换
1: A/D 转换中
此位用于表明 A/D 转换过程是否完成。当 START 位由低变为高再变为低时，ADBZ 位为高，表明 A/D 转换已开始。A/D 转换结束后，此位被清零。
- Bit 5 **ADCEN**: A/D 转换器使能 / 除能控制位
0: 除能
1: 使能
此位控制 A/D 内部功能。该位被置高将使能 A/D 转换器。如果该位设为低将关闭 A/D 转换器以降低功耗。当 A/D 转换器除能时，A/D 数据寄存器 SADOH 和 SADOL 的内容将保持不变。
- Bit 4 **ADRF5**: A/D 转换数据格式选择位
0: A/D 转换数据格式 → SADOH=D[11:4]; SADOL=D[3:0]
1: A/D 转换数据格式 → SADOH=D[11:8]; SADOL=D[7:0]
此位控制存放在两个 A/D 数据寄存器中的 12 位 A/D 转换结果的格式。细节方面请参考 A/D 数据寄存器章节。
- Bit 3~0 **SACS3~SACS0**: A/D 外部模拟通道输入选择位
0000: AN0
0001: AN1
:
:
1010: AN10
1011: AN11
1100~1111: 无通道，若选中时输入浮空

● SADC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SAINS3	SAINS2	SAINS1	SAINS0	—	SACKS2	SACKS1	SACKS0
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 7~4 **SAINS3~SAINS0**: A/D 输入信号选择位
0000: 外部信号 – 外部模拟通道输入
0001: 内部信号 – 内部 A/D 转换器电源电压 V_{DD}
0010: 内部信号 – 内部 A/D 转换器电源电压 $V_{DD}/2$
0011: 内部信号 – 内部 A/D 转换器电源电压 $V_{DD}/4$
0100: 外部信号 – 外部模拟通道输入
0101: 内部信号 – PGA 输出电压 V_R
0110: 内部信号 – PGA 输出电压 $V_R/2$
0111: 内部信号 – PGA 输出电压 $V_R/4$
1000~1011: 保留，需接地
1100~1111: 外部信号 – 外部模拟通道输入
当同时选择内部模拟信号和外部模拟信号时，无论 SACS3~SACS0 为何值，外

部通道输入信号都会自动关闭。此举预防了外部通道输入与内部模拟信号连接从而导致的不可预期的后果。

- Bit 3 未定义，读为“0”
- Bit 2~0 **SACKS2~SACKS0**: A/D 时钟源选择位
- 000: f_{SYS}
 - 001: $f_{SYS}/2$
 - 010: $f_{SYS}/4$
 - 011: $f_{SYS}/8$
 - 100: $f_{SYS}/16$
 - 101: $f_{SYS}/32$
 - 110: $f_{SYS}/64$
 - 111: $f_{SYS}/128$

这三位用于选择 A/D 转换器的时钟源。

● SADC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ADPGAEN	—	—	PGAIS	SAVRS1	SAVRS0	PGAGS1	PGAGS0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	—	—	0	0	0	0	0

- Bit 7 **ADPGAEN**: PGA 功能使能 / 除能控制位
- 0: 除能
 - 1: 使能
- 当 PGA 输出电压 V_R 被选择作为 A/D 转换器输入信号或参考电压时，应将此位置高以使能 PGA 功能。否则应将此位清零以除能 PGA 功能来减小功耗。
- Bit 6~5 未定义，读为“0”
- Bit 4 **PGAIS**: PGA 输入 (V_{RI}) 选择位
- 0: 外部 VREFI 引脚
 - 1: 内部独立参考电压 V_{BG}
- 当同时选择外部 VREFI 引脚电压和内部独立参考电压 V_{BG} 作为 PGA 输入时，硬件只会选择内部 V_{BG} 。
- Bit 3~2 **SAVRS1~SAVRS0**: A/D 转换器参考电压选择位
- 00: 内部 A/D 转换器电源电压 V_{DD}
 - 01: VREF 引脚
 - 1x: 内部 PGA 输出电压 V_R
- 这几位用于选择 A/D 转换器参考电压。当同时选择内部 A/D 转换器电源或内部 PGA 输出电压和外部 VREF 引脚输入电压作为参考电压时，硬件只会选择内部参考电压。
- Bit 1~0 **PGAGS1~PGAGS0**: PGA 增益 (Gain) 选择位
- 00: Gain=1
 - 01: Gain=1.667; 若设置 PGAIS=1, 使 $V_{RI}=V_{BG}$, 则 PGA 输出 $V_R=2V$
 - 10: Gain=2.5; 若设置 PGAIS=1, 使 $V_{RI}=V_{BG}$, 则 PGA 输出 $V_R=3V$
 - 11: Gain=3.333; 若设置 PGAIS=1, 使 $V_{RI}=V_{BG}$, 则 PGA 输出 $V_R=4V$

A/D 转换器操作

SADC0 寄存器中的 START 位，用于打开 A/D 转换器。当单片机设置此位从逻辑低到逻辑高，然后再到逻辑低，就会开始一个模数转换周期。

SADC0 寄存器中的 ADBZ 位用于表明模数转换过程是否正在进行。A/D 转换成功启动后，ADBZ 位会被单片机自动置为“1”。在转换周期结束后，ADBZ 位会自动置为“0”。此外，也会置位中断控制寄存器内相应的 A/D 中断请求标志位，如果中断使能，就会产生对应的内部中断信号。A/D 内部中断信号将引导程序跳转到相应的 A/D 内部中断地址。如果 A/D 内部中断被禁止，可以让

单片机轮询 SADC0 寄存器中的 ADBZ 位，检查此位是否被清除，作为另一种侦测 A/D 转换周期结束的方法。

A/D 转换器的时钟源为系统时钟 f_{SYS} 或其分频，而分频系数由 SADC1 寄存器中的 SACKS2~SACKS0 位决定。虽然 A/D 时钟源是由系统时钟 f_{SYS} 和 SACKS2~SACKS0 位决定，但可选择的最大 A/D 时钟源则有一些限制。由于若工作电压在 $1.8V \leq V_{DD} < 2.0V$ 时允许的 A/D 时钟周期 t_{ADCK} 的范围为 $2\mu s \sim 10\mu s$ ；若工作电压在 $2.0V \leq V_{DD} \leq 5.5V$ 时允许的 A/D 时钟周期 t_{ADCK} 的范围为 $0.5\mu s \sim 10\mu s$ ，所以选择系统时钟速度时必须小心。例如，如果系统时钟速度为 8MHz 时，若工作电压为 $2.0V \leq V_{DD} \leq 5.5V$ ，SACKS2~SACKS0 位不能设为“000”、“001”或“111”；若工作电压为 $1.8V \leq V_{DD} < 2.0$ ，SACKS2~SACKS0 位不能设为“000~011”或“111”。必须保证设置的 A/D 转换时钟周期不小于时钟周期的最小值或不大于时钟周期的最大值，否则将会产生不准确的 A/D 转换值。使用者可以参考下面的表格，被标上“*”或“#”的数值是不允许的。

f_{SYS}	A/D 时钟周期 (t_{ADCK})							
	SACKS [2:0]=000 (f_{SYS})	SACKS [2:0]=001 ($f_{SYS}/2$)	SACKS [2:0]=010 ($f_{SYS}/4$)	SACKS [2:0]=011 ($f_{SYS}/8$)	SACKS [2:0]=100 ($f_{SYS}/16$)	SACKS [2:0]=101 ($f_{SYS}/32$)	SACKS [2:0]=110 ($f_{SYS}/64$)	SACKS [2:0]=111 ($f_{SYS}/128$)
1MHz	$1\mu s$ #	$2\mu s$	$4\mu s$	$8\mu s$	$16\mu s$ **	$32\mu s$ **	$64\mu s$ **	$128\mu s$ **
2MHz	$500ns$ #	$1\mu s$ #	$2\mu s$	$4\mu s$	$8\mu s$	$16\mu s$ **	$32\mu s$ **	$64\mu s$ **
4MHz	$250ns$ **	$500ns$ #	$1\mu s$ #	$2\mu s$	$4\mu s$	$8\mu s$	$16\mu s$ **	$32\mu s$ **
8MHz	$125ns$ **	$250ns$ **	$500ns$ #	$1\mu s$ #	$2\mu s$	$4\mu s$	$8\mu s$	$16\mu s$ **
12MHz	$83ns$ **	$167ns$ **	$333ns$ **	$667ns$ #	$1.33\mu s$ #	$2.67\mu s$	$5.33\mu s$	$10.67\mu s$ **
16MHz	$62.5ns$ **	$125ns$ **	$250ns$ **	$500ns$ #	$1\mu s$ #	$2\mu s$	$4\mu s$	$8\mu s$

注：若 $1.8V \leq V_{DD} < 2.0V$ 参考“#”标示；若 $2.0V \leq V_{DD} \leq 5.5V$ 参考“*”标示。

A/D 时钟周期范例

SADC0 寄存器中的 ADCEN 位用于控制 A/D 转换电路电源的开启和关闭。该位必须置高以开启 A/D 转换器电源。当设置 ADCEN 位为高开启 A/D 转换器内部电路时，在 A/D 转换成功开启前需一段延时。即使通过相关引脚共用控制位选择无引脚作为 A/D 输入，如果 ADCEN 设为“1”，那么仍然会产生功耗。因此在功耗敏感的应用中，当未使用 A/D 转换器功能时，建议设置 ADCEN 为低以减少功耗。

A/D 转换器参考电压

A/D 转换器参考电压可以来自内部 A/D 正电源电压引脚 VDD、外部参考源引脚 VREF 或内部 PGA 输出电压 V_R ，可通过 SAVRS1~SAVRS0 位来选择。当 SAVRS1~SAVRS0 位为“00”时，A/D 转换器参考电压来自 VDD 引脚。当 SAVRS1~SAVRS0 位为“01”时，A/D 转换器参考电压来自 VREF 引脚。否则为其它值时 A/D 转换器参考电压来自 PGA 输出电压 V_R 。由于 VREF 引脚与其它功能共用，当选择 VREF 引脚作为参考电压源时，需合理设置相关引脚共用控制位选择 VREF 引脚功能且除能其它共用引脚功能。然而，如果应用程序选择外部参考电压引脚 VREF 和内部参考电压 V_{DD} 或 V_R 作为 A/D 转换器参考电压输入，硬件只会选择内部参考电压。模拟输入值一定不能超过所选的参考电压值 V_{REF} 。

A/D 转换器还有一个 VREFI 引脚，作为 PGA 输入，可用作 A/D 参考电压。若要选择这个引脚作为 PGA 输入信号，SADC2 寄存器中的 PGAIS 位必须清零且相关引脚共用控制位需合理设置。然后，PGA 输入也可由内部独立参考电

压 V_{BG} 提供。如果应用程序同时选择外部 VREFI 引脚电压和内部电压 V_{BG} 作为 PGA 输入，硬件只会选择内部电压 V_{BG} 。

SAVRS[1:0]	参考电压	说明
00	V_{DD}	内部 A/D 转换器电源电压
01	VREF 引脚	外部 A/D 转换器参考电压引脚 VREF
1x	V_R	内部 A/D 转换器 PGA 输出电压

A/D 转换器参考电压选择

A/D 转换器输入信号

所有的 A/D 模拟输入引脚都与 I/O 口及其它功能共用。使用 PxFS 和 SFSR1 寄存器中的相应位，可以将它们设置为 A/D 转换器模拟输入脚或其它共用功能。如果对应的引脚作为 A/D 转换输入，那么它原来的引脚功能将除能。通过这种方式，引脚的功能可由程序来控制，灵活地切换引脚功能。如果将引脚设为 A/D 输入，则通过寄存器编程设置的所有上拉电阻会自动断开。请注意，端口控制寄存器不需要为使能 A/D 输入而先设定为输入模式，当 A/D 输入功能选择位使能 A/D 输入时，端口控制寄存器的状态将被重置。

若 SAINS3~SAINS0 位为“0000”、“0100”或“1100~1111”，则选择转换外部模拟输入信号，具体通道编号由 SACS3~SACS0 位决定。若 SAINS3~SAINS0 位为“0001~0011”，则选择转换内部 V_{DD} 电压及其分压比为 1，1/2 或 1/4，若 SAINS3~SAINS0 位为“0101~0111”，则选择转换内部 PGA 输出电压及其分压比为 1，1/2 或 1/4。注意，当应用程序同时选择外部信号 (AN0~AN11) 和内部信号 (V_{DD} ， $V_{DD}/2$ ， $V_{DD}/4$ ， V_R ， $V_R/2$ 或 $V_R/4$) 作为 A/D 转换器输入信号时，硬件只会选择内部信号，外部模拟信号会自动关闭。

SAINS[3:0]	SACS[3:0]	输入信号	说明
0000, 0100, 1100~1111	0000~1011	AN0~AN11	外部模拟通道输入
	1100~1111	—	无通道，输入浮空
0001	xxxx	V_{DD}	内部 A/D 转换器电源电压
0010	xxxx	$V_{DD}/2$	内部 A/D 转换器电源电压 / 2
0011	xxxx	$V_{DD}/4$	内部 A/D 转换器电源电压 / 4
0101	xxxx	V_R	内部 A/D 转换器 PGA 输出电压
0110	xxxx	$V_R/2$	内部 A/D 转换器 PGA 输出电压 / 2
0111	xxxx	$V_R/4$	内部 A/D 转换器 PGA 输出电压 / 4
1000~1011	xxxx	—	保留，接地

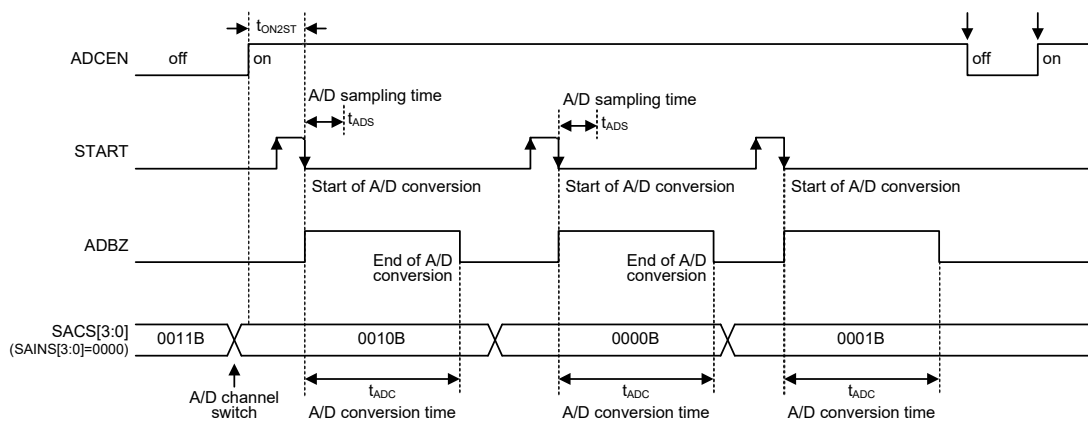
A/D 转换器输入信号选择

A/D 转换率和时序图

一个完整的 A/D 转换包含两部分，数据采样和数据转换。数据采样时间定义为 t_{ADS} ，需 4 个 A/D 时钟周期，而数据转换需 12 个 A/D 时钟周期。所以一个完整的 A/D 转换时间 t_{ADC} 一共需要 16 个 A/D 时钟周期。

最大 A/D 转换率 = $1 / (A/D \text{ 时钟周期} / 16)$

下列时序图表示模数转换过程中不同阶段的图形与时序。由应用程序控制开始 A/D 转换过程后，单片机的内部硬件就会开始进行转换，在这个过程中，程序可以继续其它功能。A/D 转换时间为 16 个 t_{ADCK} ， t_{ADCK} 为 A/D 时钟周期。



A/D 转换时序 – 外部通道输入

A/D 转换步骤概述

下面概述实现 A/D 转换过程的各个步骤。

- 步骤 1
通过 SADC1 寄存器中的 SACKS [2:0] 位，选择所需的 A/D 转换器转换时钟。
- 步骤 2
将 SADC0 寄存器中的 ADCEN 位置高使能 A/D。
- 步骤 3
通过配置 SADC1 寄存器的 SAINS3~SAINS0 位，选择连接至内部 A/D 转换器的信号。
若选择外部通道输入，接着执行步骤 4。
若选择内部模拟信号，接着执行步骤 5。
- 步骤 4
若已通过 SAINS3~SAINS0 位选择 A/D 输入信号来自外部通道输入，接着应设置相关的引脚共用控制位将该引脚规划为 A/D 输入引脚。通过设置 SACS3~SACS0 位选择哪个外部通道接至 A/D 转换器。接着执行步骤 6。
- 步骤 5
若已通过 SAINS3~SAINS0 位选择 A/D 输入信号来自内部模拟信号，无论 SACS3~SACS0 为何值，外部通道输入都会自动断开。接着执行步骤 6。
- 步骤 6
通过 SADC2 寄存器中的 SAVRS1~SAVRS0 位选择参考电压。如果选择 PGA 输出电压作为 A/D 转换器参考电压，必须先使能 PGA 功能，再设置 SADC2 寄存器中的 PGAIS 位选择 PGA 的输入源。
- 步骤 7
设置 SADC0 寄存器中的 ADRFS 位选择 A/D 转换器输出数据格式。
- 步骤 8
如果要使用中断，则中断控制寄存器需要正确地设置，以确保 A/D 中断功能是激活的。总中断控制位 EMI 需要置位为“1”，以及 A/D 转换器中断位 ADE 也需要置位为“1”。
- 步骤 9
现在可以通过设置 SADC0 寄存器中的 START 位从“0”到“1”再回到“0”，开始模数转换的过程。

● 步骤 10

如果 A/D 转换正在进行中，ADBZ 位会被置为逻辑高。A/D 转换完成后，ADBZ 位会被置为逻辑低，并可从 SADOH 和 SADOL 寄存器中读取输出数据。

注：若使用轮询 SADC0 寄存器中 ADBZ 位的状态的方法来检查转换过程是否结束时，则中断使能的步骤可以省略。

编程注意事项

在编程时，如果 A/D 转换器未使用，通过设置 SADC0 寄存器中的 ADCEN 为低，关闭 A/D 内部电路以减少电源功耗。此时，不考虑输入脚的模拟电压，内部 A/D 转换器电路不产生功耗。如果 A/D 转换器输入脚用作普通 I/O 脚，必须特别注意，输入电压为无效逻辑电平也可能增加功耗。

A/D 转换功能

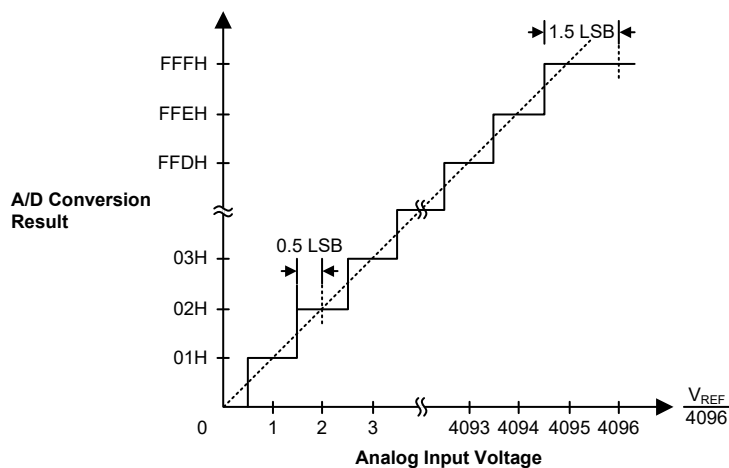
该单片机含有一个 12-bit A/D 转换器，它们转换的最大值可达 0FFFH。由于模拟输入最大值等于实际 A/D 转换器参考电压 V_{REF} ，因此每一位可表示 $V_{REF}/4096$ 的模拟输入值。

$$1 \text{ LSB} = V_{REF} \div 4096$$

通过下面的等式可估算 A/D 转换器输入电压值：

$$\text{A/D 输入电压} = \text{A/D 数字输出值} \times (V_{REF} \div 4096)$$

下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。除了数字化数值 0，其后的数字化数值会在精确点之前的 0.5 LSB 处改变，而数字化数值的最大值将在 V_{REF} 之前的 1.5 LSB 处改变。应注意的是，这里的 V_{REF} 电压指代的是通过 SAVRS 位选择的实际 A/D 转换器参考电压。



理想的 A/D 转换功能

A/D 程序范例

下面两个范例程序用来说明怎样使用 A/D 转换。第一个范例是轮询 SADC0 寄存器中的 ADBZ 位来判断 A/D 转换是否完成；第二个范例则使用中断的方式判断。

范例 1：使用轮询 ADBZ 的方式来检测转换结束

```
clr ADE                ; disable ADC interrupt
mov a, 03H
```

```

mov SADC1,a          ; select fsys/8 as A/D clock
set ADCEN
set PB5FS
set PBFS5           ; setup [PB5FS:PBFS5] to configure pin AN0
mov a,20h
mov SADC0,a
mov a,00h
mov SADC2,a         ; enable and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START           ; high pulse on start bit to initiate conversion
set START           ; reset A/D
clr START           ; start A/D
polling_EOC:
sz ADBZ             ; poll the SADC0 register ADBZ bit to detect end
                    ; of A/D conversion
jmp polling_EOC     ; continue polling
mov a,SADOL          ; read low byte conversion result value
mov SADOL_buffer,a  ; save result to user defined register
mov a,SADOH          ; read high byte conversion result value
mov SADOH_buffer,a  ; save result to user defined register
:
:
jmp start_conversion ; start next A/D conversion

```

范例 2：使用中断的方式来检测转换结束

```

clr ADE             ; disable ADC interrupt
mov a,03H
mov SADC1,a         ; select fsys/8 as A/D clock
set ADCEN
set PB5FS
set PBFS5           ; setup [PB5FS:PBFS5] to configure pin AN0
mov a,20h
mov SADC0,a
mov a,00h
mov SADC2,a         ; enable and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START           ; high pulse on START bit to initiate conversion
set START           ; reset A/D
clr START           ; start A/D
clr ADF             ; clear ADC interrupt request flag
set ADE             ; enable ADC interrupt
set EMI             ; enable global interrupt
:
:
; ADC interrupt service routine
ADC_ISR:
mov acc_stack,a     ; save ACC to user defined memory
mov a,STATUS
mov status_stack,a  ; save STATUS to user defined memory
:
:
mov a,SADOL          ; read low byte conversion result value
mov SADOL_buffer,a  ; save result to user defined register

```

```
mov a, SADOH                ; read high byte conversion result value
mov SADOH_buffer, a        ; save result to user defined register
:
:
EXIT_INT_ISR:
mov a, status_stack
mov STATUS, a              ; restore STATUS from user defined memory
mov a, acc_stack           ; restore ACC from user defined memory
reti
```

串行接口模块 – SIM

该系列单片机内有一个串行接口模块，包括两种易与外部设备硬件通信的串行接口：四线 SPI 或两线 I²C 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，所以要先通过相关的引脚共用功能选择位选择使用 SIM 功能引脚。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 位来选择使用 SPI 还是 I²C 通信接口。当 SIM 功能使能，可通过上拉电阻控制寄存器选择与输入 / 输出共用的 SIM 脚的上拉电阻，其相应的引脚则作为 SIM 的输入引脚。

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同步串行数据接口，这个协议可以简化与外部硬件通信的编程要求。

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 $\overline{SCS}$ 。若主机需要控制多个从机，可使用输入 / 输出引脚选择从机。

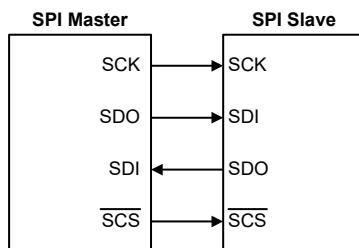
SPI 接口操作

SPI 接口是一个全双工同步串行数据传输器。SPI 接口的四线为：SDI、SDO、SCK 和 $\overline{SCS}$ 。SDI 和 SDO 是串行数据的输入和输出线。SCK 是串行时钟线，是从机的选择线。由于 SPI 的接口引脚与普通 I/O 口和 I²C 的功能脚共用，必须通过配置引脚共用功能选择位并设定 SIMC0 和 SIMC2 寄存器的对应位，来使能 SPI 接口。在所需 SPI 配置设定后，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启动，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。

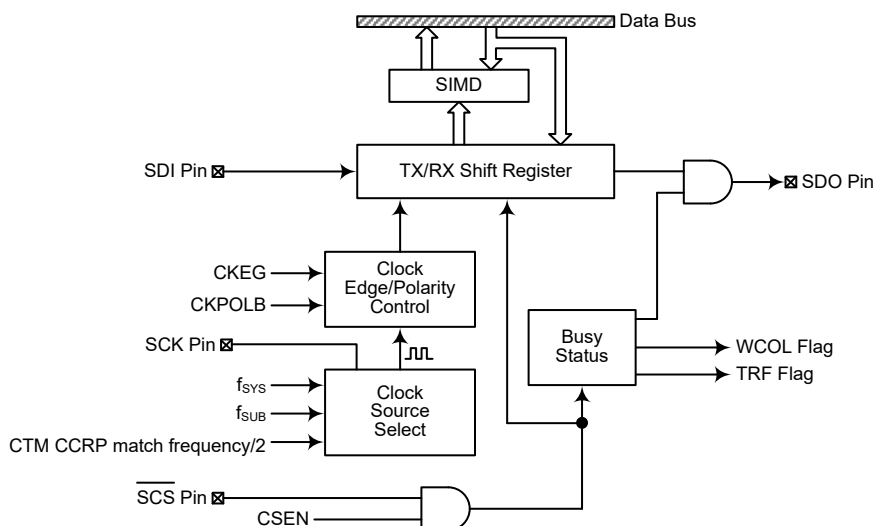
该系列单片机的 SPI 功能具有以下特点：

- 全双工同步数据传输
- 主从模式
- 最低有效位优先或最高有效位优先的数据传输模式
- 传输完成标志位
- 时钟源上升沿或下降沿有效

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。



SPI 主 / 从机连接方式



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 和两个控制寄存器 SIMC0 和 SIMC2。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	—	SIMDEB1	SIMDEB0	SIMEN	SIMICF
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMC2	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF

SPI 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 I²C 功能所共用。在单片机将数据写入到 SPI 总线中之前，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	x	x	x	x	x	x	x	x

“x”：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 I²C 接口功能中的寄存器 SIMA 是同一个寄存器。SPI 功能不会用到寄存器 SIMC1，SIMC1 只适用于 I²C 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。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	—	SIMDEB1	SIMDEB0	SIMEN	SIMICF
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_{sub}
- 100: SPI 主机模式; SPI 时钟为 CTM CCRP 匹配频率 / 2
- 101: SPI 从机模式
- 110: I²C 从机模式
- 111: 非 SIM 功能

这几位用于设置 SIM 功能的所有工作模式。此位段控制 SPI 的主从模式选择和 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 CTM。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 未定义，读为“0”

Bit 3 ~ 2 **SIMDEB1~SIMDEB0**: I²C 去抖时间选择位

- 00: 无去抖时间
- 01: 2 个系统时钟去抖时间
- 1x: 4 个系统时钟去抖时间

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。当此位清零时，SIM 接口除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 脚将失去 SPI 或 I²C 功能，SIM 工作电流减小到最小值。此位为高时，SIM 接口使能。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 控制寄存器中的内容将保持之前的设定，因此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，将保持原先设定，因此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 **SIMICF**: SIM 未完成标志位

- 0: 未发生 SIM 未完成现象
- 1: 发生 SIM 未完成现象

SIMICF 由 SCS 引脚所确定。当 SCS 引脚设为“1”，将清除 SPI 计数器，同时产生中断。若从设备未完成数据接收，则未完成标志位 SIMICF 置“1”。

● SIMC2 寄存器

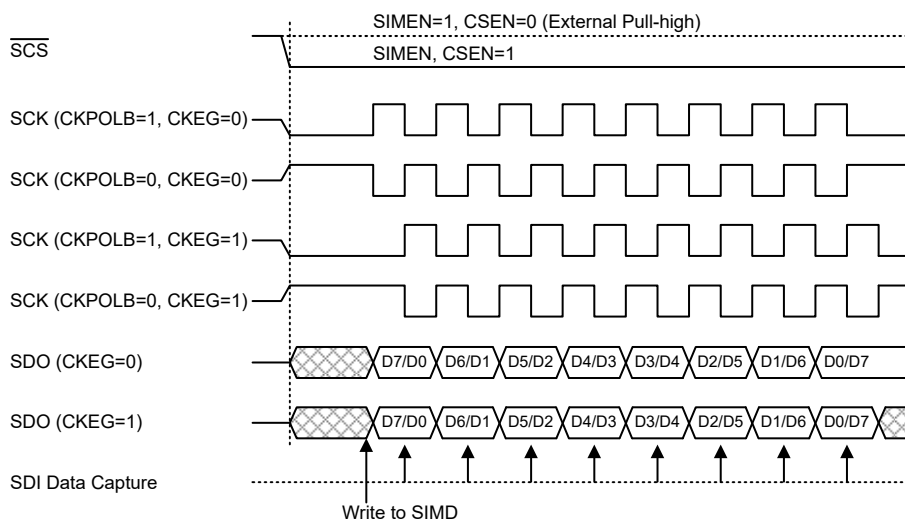
Bit	7	6	5	4	3	2	1	0
Name	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- 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, $\overline{SCS}$ 除能并工作在 I/O 引脚或其他功能。此位为高时, $\overline{SCS}$ 使能并作为选择脚。
- 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。
- 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。

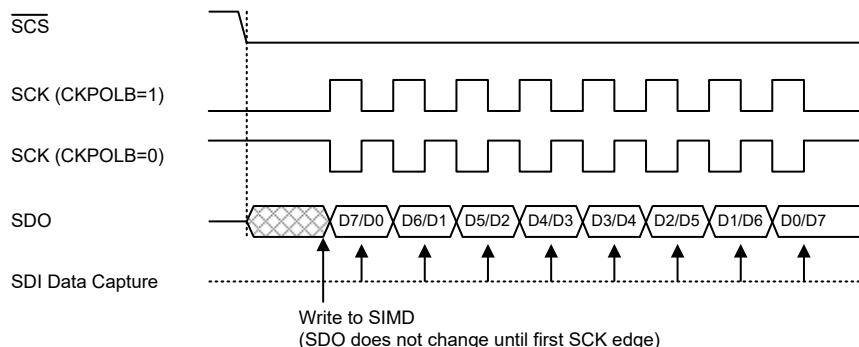
SPI 通信

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, 收到来自主机的时钟信号之后, 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 $\overline{SCS}$ 信号以能使

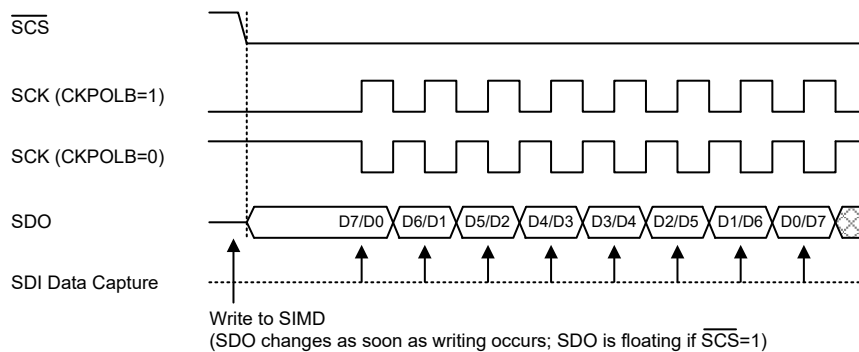
从机，从机的数据传输功能也应在与 $\overline{SCS}$ 信号相关的适当时候准备就绪，这由 CKPOLB 和 CKEG 位决定。所附时序图表明了 CKPOLB 和 CKEG 位各种设置情况下从机数据与 $\overline{SCS}$ 信号的关系。即使在单片机处于空闲模式，SPI 功能仍将继续执行。



SPI 主机模式时序



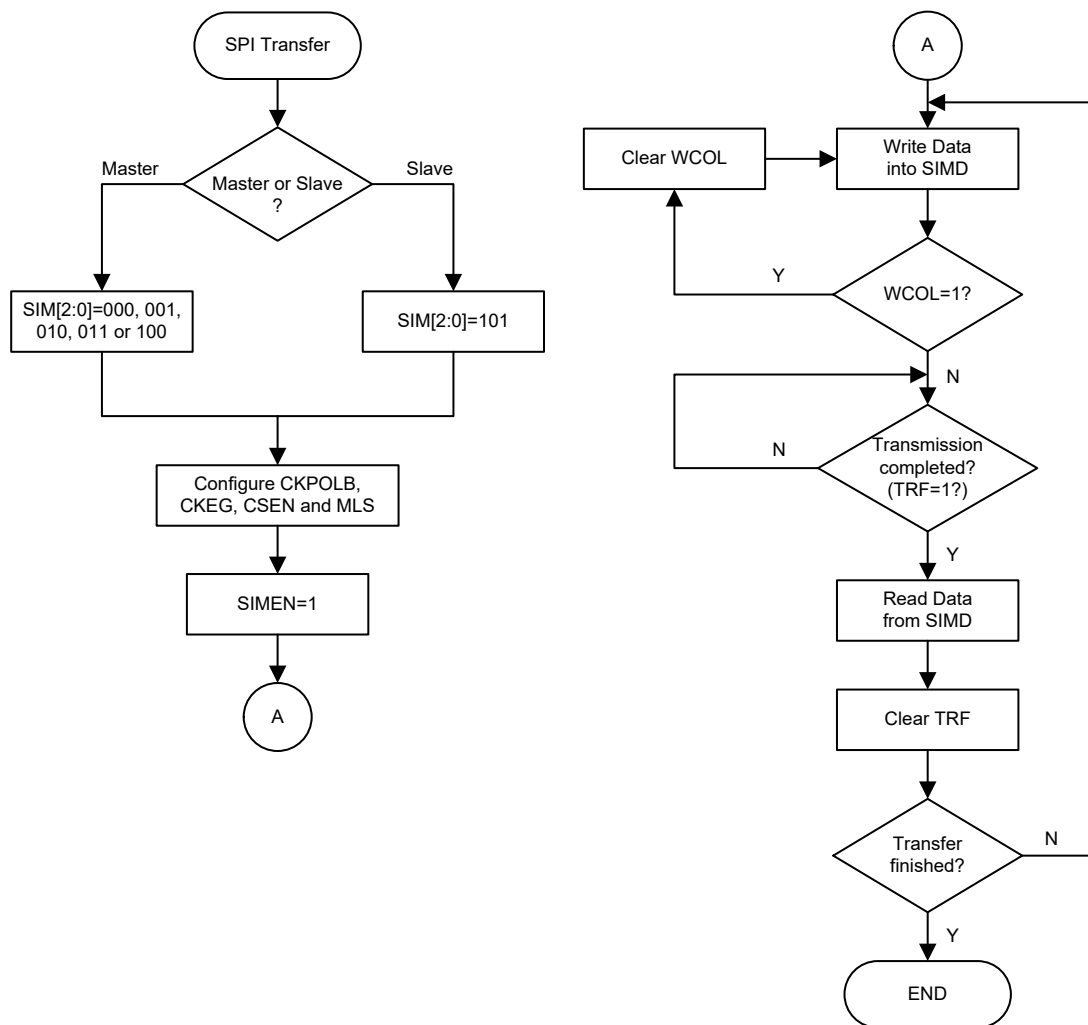
SPI 从机模式时序 – CKEG=0



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

注：对于 SPI 从机模式。若 SIMEN=1 且 CSEN=0，则无论 $\overline{SCS}$ 为何种电平，SPI 都为使能状态。

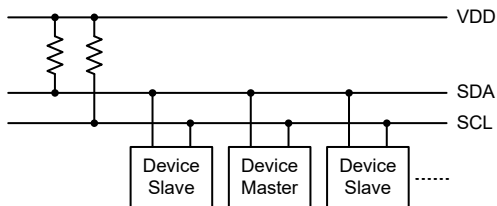
SPI 从机模式时序 – CKEG=1



SPI 传输控制流程图

I²C 接口

I²C 可以和传感器，EEPROM 内存等外部硬件设备进行通信。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，使之在很多的场合中大受欢迎。

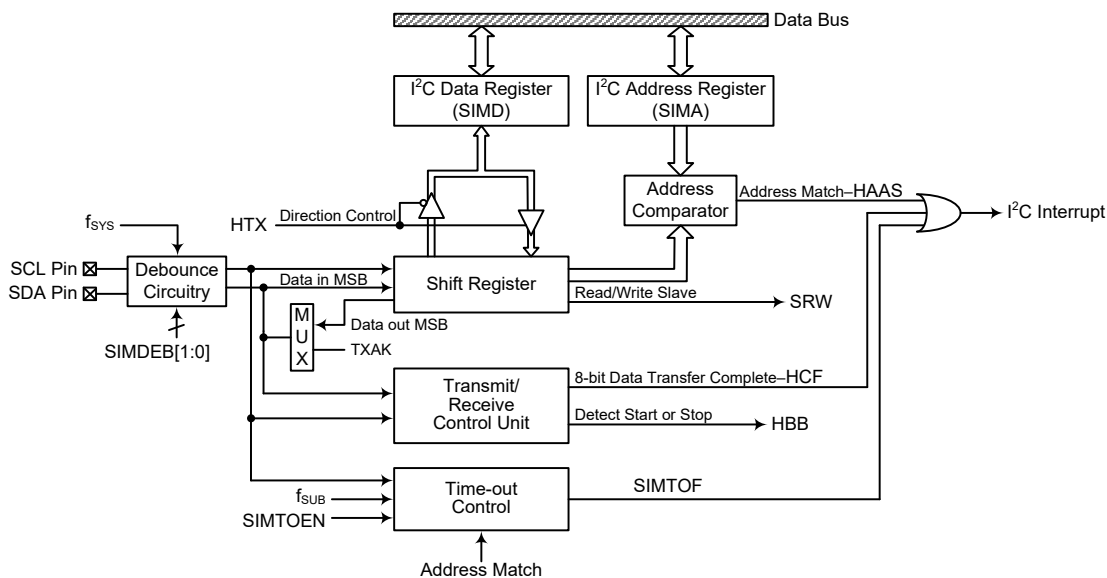


I²C 主机从机总线连接图

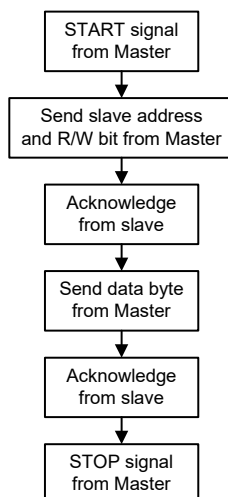
I²C 接口操作

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。即使 I²C 设备为有效状态，SCL/SDA 脚的上拉控制功能引脚共用仍可用，且通过配置其相应的上拉控制寄存器可以控制相关的内部上拉寄存器。



I²C 方框图



I²C 寄存器

I²C 总线有四个控制寄存器 SIMC0、SIMC1、SIMA 和 SIMTOC，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，在单片机将数据写入 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 中读取这个数据。I²C 总线上的所有传输或接收到的数据都必须通过 SIMD 寄存器。

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 溢出控制功能会用到寄存器 SIMC0 中的 SIMEN 位和 SIM2~SIM0 位段。SIMTOC 寄存器用于 I²C 溢出控制功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	—	SIMDEB1	SIMDEB0	SIMEN	SIMICF
SIMC1	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMA	A6	A5	A4	A3	A2	A1	A0	—
SIMTOC	SIMTOEN	SIMTOF	SIMTOS5	SIMTOS4	SIMTOS3	SIMTOS2	SIMTOS1	SIMTOS0

I²C 寄存器列表

• SIMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	—	SIMDEB1	SIMDEB0	SIMEN	SIMICF
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 $f_{\text{SYS}}/4$

001: SPI 主机模式; SPI 时钟为 $f_{\text{SYS}}/16$

010: SPI 主机模式; SPI 时钟为 $f_{\text{SYS}}/64$

011: SPI 主机模式; SPI 时钟为 f_{SUB}

100: SPI 主机模式; SPI 时钟为 CTM CCRP 匹配频率 / 2

101: SPI 从机模式

110: I²C 从机模式

111: 无 SIM 功能

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 CTM。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 未定义，读为“0”

Bit 3~2 **SIMDEB1~SIMDEB0**: I²C 去抖时间选择位

00: 无去抖时间

01: 2 个系统时钟去抖时间

1x: 4 个系统时钟去抖时间

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 脚将失去 SPI 或 I²C 功能，SIM 工作电流减小到最小值。此位为“1”时，SIM 接口使能。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 **SIMICF**: SIM 未完成标志位

0: 未发生 SIM 未完成现象

1: 发生 SIM 未完成现象

此位由 $\overline{\text{SCS}}$ 引脚所确定。当 $\overline{\text{SCS}}$ 引脚设为“1”，将清除 SPI 计数器，同时产生中断。若从机未完成数据接收，则未完成标志位 SIMICF 置“1”。

● 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**: I²C 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

HCF 是数据传输标志位。数据正在传输时该位为低。当 8 位数据传输完成，此位为高并产生一个中断。

Bit 6 **HAAS**: I²C 总线地址匹配标志位

0: 地址不匹配

1: 地址匹配

HAAS 标志位为地址匹配标志位。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**: I²C 总线忙标志位

0: I²C 总线闲

1: I²C 总线忙

HBB 标志位为 I²C 忙标志位。当检测到 START 信号时 I²C 忙，此位变为高电平。当检测到 STOP 信号时 I²C 总线停止，该位变为低电平。

Bit 4 **HTX**: I²C 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**: I²C 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

TXAK 位为发送确认标志位。从机设备接收 8 位数据之后会将该位在第九个时

钟传到总线上。如果从机设备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**: I²C 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 I²C 从机读写位。决定主机是否希望传输或接收来自 I²C 总线的数据。当传输地址和从机的地址相匹配时，HAAS 位会被设置为高，从机设备将检测 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，此时从机设备处于传输模式。当 SRW 位为“0”时，主机往总线上写数据，从机设备处于接收模式以读取该数据。

Bit 1 **IAMWU**: I²C 地址匹配唤醒控制位
0: 除能
1: 使能—唤醒后必须由应用程序清零

此位应设置为“1”使能 I²C 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 I²C 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**: I²C 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收到一个正确的确认位。如果从机设备处于发送状态，则从机设备将会检查 RXAK 位来判断主机接收方是否愿意继续接收下一个字节。因此直到 RXAK 为“1”时，从机传输方停止发送数据。这时，传输方将释放 SDA 线，主机发出停止信号释放 I²C 总线。

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 I²C 总线中时，要传输的数据应存在 SIMD 中。I²C 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 I²C 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	x	x	x	x	x	x	x	x

“x”：未知

• SIMA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	A6	A5	A4	A3	A2	A1	A0	—
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 7~1 **A6~A0**: I²C 从机地址位
A6~A0 是从机地址 bit 6 ~ bit 0。

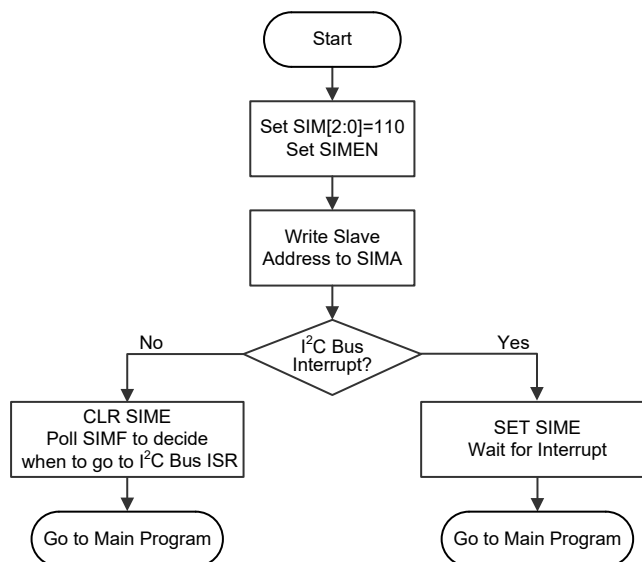
SIMA 寄存器也可供 SPI 接口使用，但此时其名称为 SIMC2。SIMA 寄存器用于存放从机设备的 7 位从机地址。SIMA 寄存器中的 bit7~bit1 定义了设备的从机地址。Bit 0 未定义。当连接到 I²C 总线的主机设备发送一个地址，与 SIMA 寄存器中的从机地址相匹配，则从机地址将被选中。注意 SIMA 寄存器与 SPI 接口使用的 SIMC2 寄存器地址是一样的。

Bit 0 未定义位
此位可通过软件程序进行读写。

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 位和 I2CTOF 位，以判断 I²C 总线中断是来自从机地址匹配，还是来自 8 位数据传输完毕或是来自 I²C 超时。在数据传输中，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 位为“110”和 SIMEN 位为“1”，以能使 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
设置中断控制寄存器中的 SIME 位，以能使 SIM 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到 I²C 总线上的主机产生而非从机设备。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。SCL 线保持为高电平时，SDA 线上发生从高到低的电平变化，此时即产生起始信号。

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 位和 I2CTOF 位，以判断 I²C 总线中断是来自从机地址匹配，还是来自 8 位数据传输完毕。若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 寄存器中执行一个空读的动作以释放 SCL 线。

I²C 总线读 / 写信号

SIMC1 寄存器的 SRW 位定义了主机是要从 I²C 总线上读取数据还是要将数据写到 I²C 总线上。从机设备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。

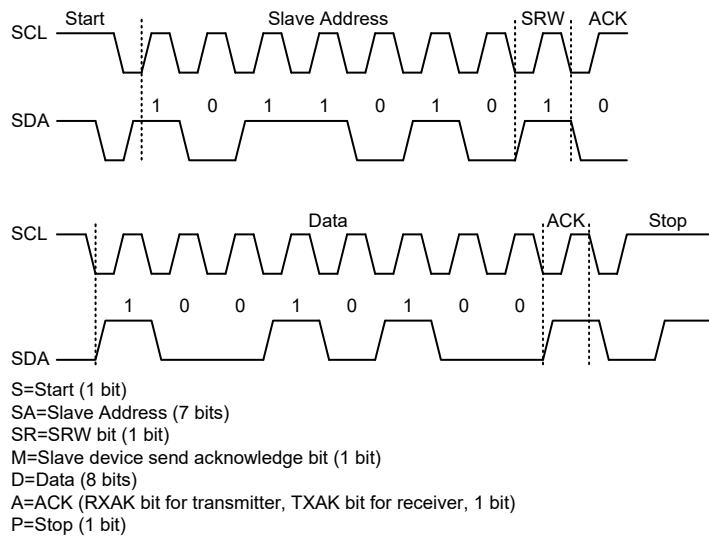
I²C 总线从机地址确认信号

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，单片机从机设备须设置成接收方，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。如果从机发送方没接收到主机接收方发出的应答信号，从机发送方将释放 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。

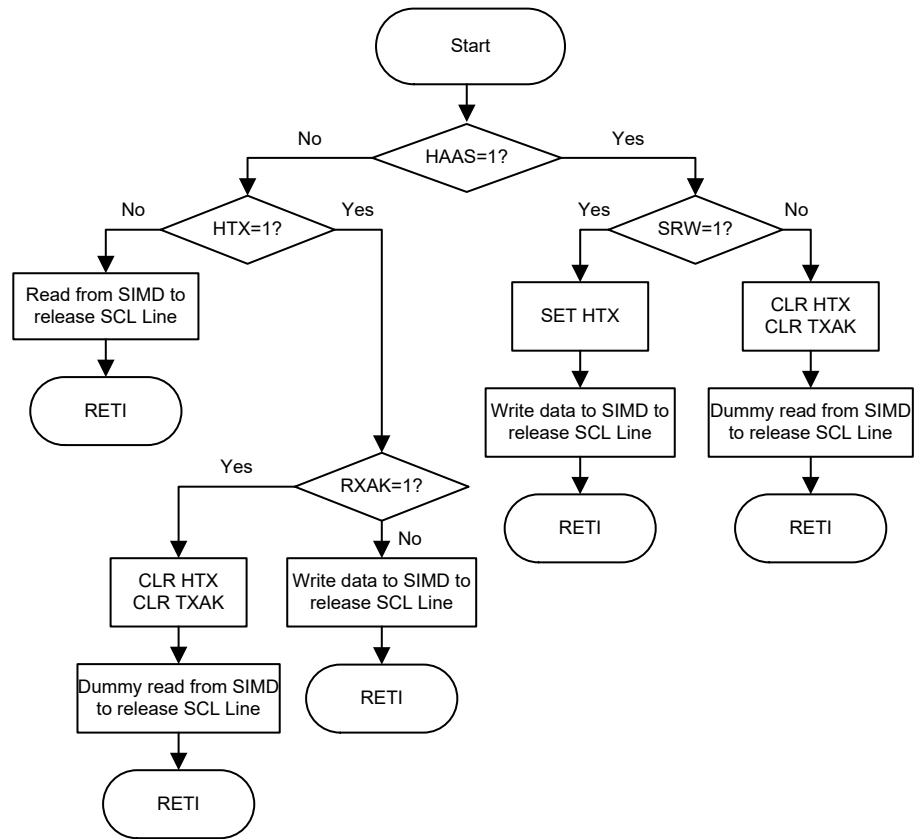
当接收器想要继续接收下一个数据时，必须在第 9 个时钟发出应答信号 (TXAK)。被设为发送方的从机将检测寄存器 SIMC1 中的 RXAK 位以判断是否传输下一个字节的数据，如果单片机不传输下一个字节，那么它将释放 SDA 线并等待接收主机的停止信号。



S	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 寄存器中空读数据以释放 SCL 线。

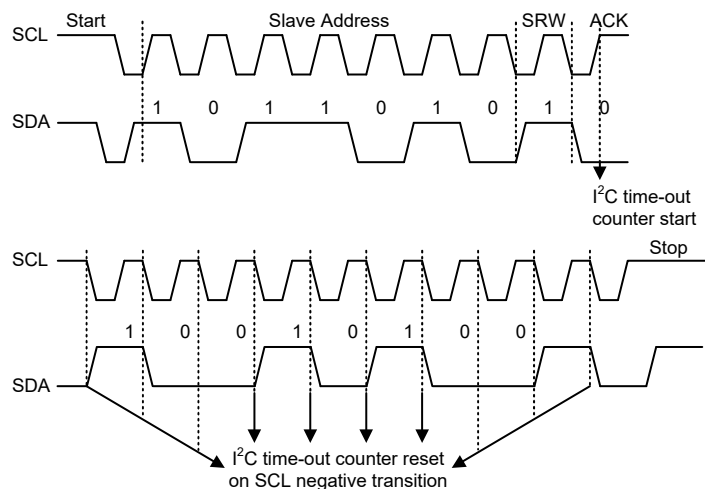
I²C 通信时序图



I²C 总线 ISR 流程图

I²C 超时控制

超时功能可减少 I²C 接收错误的时钟源而引起的锁死问题。如果连接到 I²C 总线的时钟源经过一段时间还未接收到，则在一定的超时周期后，I²C 电路和寄存器将复位。超时计数器在 I²C 总线“START”和“地址匹配”条件下开始计数，且在 SCL 下降沿清零。在下一个 SCL 下降沿到来之前，如果超时时间大于 I2CTOC 寄存器指定的超时周期，则超时发生。I²C “STOP”条件发生时超时功能停止。



I²C 超时时序图

当 I²C 超时计数器溢出时，计数器将停止计数，SIMTOEN 位被清零，且 SIMTOF 位被置高以表明超时计数器中断发生。超时计数器中断使用的也是 I²C 中断向量。当 I²C 超时发生时，I²C 内部电路会被复位，寄存器也将发生如下复位情况。

寄存器	I ² C 超时发生后
SIMD, SIMA, SIMC0	无变化
SIMC1	复位到 POR 状态

超时后的 I²C 寄存器状态

SIMTOF 标志位由应用程序清零。共有 64 个溢出周期，可通过 SIMTOC 寄存器的 SIMTOS 位段进行选择。溢出周期可通过公式计算： $((1 \sim 64) \times (32/f_{SUB}))$ 。由此可得溢出周期范围为 1ms~64ms。

• SIMTOC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIMTOEN	SIMTOF	SIMTOS5	SIMTOS4	ISIMTOS3	SIMTOS2	SIMTOS1	SIMTOS0
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 **SIMTOEN**: I²C 超时控制位

0: 除能
1: 使能

Bit 6 **SIMTOF**: I²C 超时标志位

0: 超时未发生
1: 超时发生

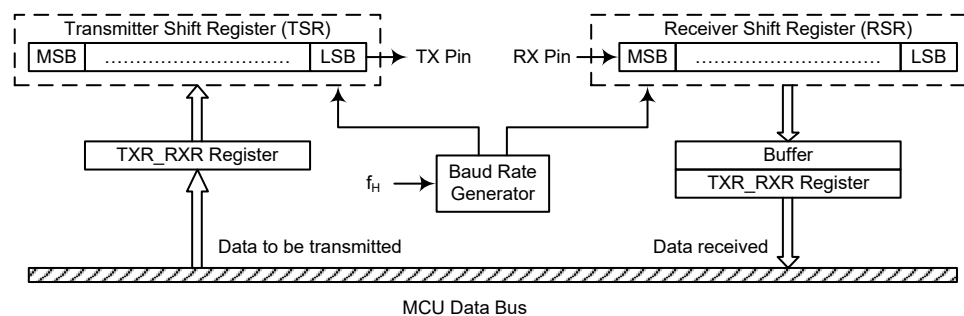
Bit 5~0 **SIMTOS5~SIMTOS0:** I²C 超时时间选择位
I²C 超时时钟源是 $f_{SUB}/32$
I²C 超时时间计算公式: $(SIMTOS[5:0]+1) \times (32/f_{SUB})$

UART 接口

HT69F360/HT67F370 单片机内建了全双工异步串行通讯 UART 接口, 可以很方便地与其它具有串行口的外部设备进行通信。UART 具有许多功能特性, 且可在发送或接收串行数据时, 将数据组成一个 8 位或 9 位的数据帧, 一帧一帧地进行传输。它还具有检测数据覆盖或帧错误等功能。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。

内建的 UART 功能包含以下特性:

- 全双工通用异步接收器和发送器 (UART) 通信
- 8 或 9 位字符长度
- 奇校验, 偶校验或无校验
- 1 位或 2 位停止位
- 8 位预分频波特率发生器
- 奇偶校验, 帧, 噪声和溢出错误检测
- 支持地址检测中断 (最后一位 = 1)
- 独立的发送和接收使能
- 2-byte FIFO 接收数据缓存器
- RX 引脚唤醒功能
- 发送和接收中断
- 中断可由下列条件触发:
 - ◆ 发送器为空
 - ◆ 发送器空闲
 - ◆ 接收完成
 - ◆ 接收器溢出
 - ◆ 地址模式



UART 数据传输方案

UART 外部接口

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。TX 和 RX 分别为 UART 发送脚和接收脚, 与 I/O 口或其它功能共用引脚。在使用 UART 功能前, 应先通过相应的引脚共用功能选择寄存器, 选择 TX 和 RX 引

脚功能。当 UARTEN 和 TXEN/RXEN 位置高时，将自动设置这些 I/O 脚或其它共用功能脚作为发送输出和接收输入。此时，用作发送输出的引脚其内部上拉电阻会被除能，而用作接收输入的引脚其内部上拉电阻由相应的上拉电阻控制位控制。当 UARTEN、TXEN 或 RXEN 位清零除能 TX 或 RX 引脚功能后，TX 或 RX 引脚将处于浮空状态。这时 TX 或 RX 引脚是否连接内部上拉电阻是由相应的 I/O 上拉电阻控制位决定的。

UART 数据传输方案

下图显示了 UART 的整体数据传输结构安排。通过应用程序将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所控制的速率下将 TSR 寄存器中数据一位位地移到 TX 引脚上，低位在前。只有 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。

要接收的数据在波特率发生器所控制的速率下，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，该寄存器用于数据发送和数据传输。

UART 状态和控制寄存器

与 UART 功能相关的有五个寄存器——控制 UART 模块整体功能的 USR、UCR1 和 UCR2 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/RXR。

寄存器名称	位							
	7	6	5	4	3	2	1	0
USR	PERR	NF	FERR	OERR	RIDLE	RXIF	TIDLE	TXIF
UCR1	UARTEN	BNO	PREN	PRT	STOPS	TXBRK	RX8	TX8
UCR2	TXEN	RXEN	BRGH	ADDEN	WAKE	RIE	TIIE	TEIE
TXR/RXR	D7	D6	D5	D4	D3	D2	D1	D0
BRG	D7	D6	D5	D4	D3	D2	D1	D0

UART 寄存器概要

● USR 寄存器

USR 寄存器为 UART 的状态寄存器，可通过程序读取以确定 UART 当前状态。此寄存器中所有标志位为只读。各个标志位的详细说明如下：

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。此只读标志位 PERR=0，表示奇偶校验正确；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。若此只读标志位 NF=0, 表明没有受到噪声干扰; 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, 至少能读取一个以上的字符。 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 寄存器中时, 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。当 RXIF 位为高, 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: 发送空闲标志位 0: 正在发送数据 1: 发送器空闲 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。当 TIDLE 位为高, 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 寄存器再写 TXR 寄存器将清除 TXIF。应注意当 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	x	0

“x”：未知

- Bit 7 UARTEN: UART 功能使能位**
 0: UART 除能，TX 和 RX 脚为 I/O 或其他引脚共用功能
 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或其他引脚共用功能；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
 此位用于选择数据长度格式，可选择长度为 8 位或 9 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。最高有效位置换成一个奇偶校验位。
- Bit 4 PRT: 奇偶校验选择位**
 0: 偶校验
 1: 奇校验
 此位为奇偶校验类型选择位。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 数据传输格式中的第 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	R/W
POR	0	0	0	0	0	0	0	0

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 引脚用作普通的输入 / 输出引脚或其他引脚共用功能。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 引脚用作普通的输入输出端口或其他引脚共用功能。

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 引脚用作普通输入 / 输出引脚或其他引脚共用功能。若 RXEN=1 且 UARTEN=1, 则接收将被使能, RX 引脚将由 UART 来控制。在数据接收时清除 RXEN 将中止数据接收且复位接收器, 此时 RX 引脚用作普通输入 / 输出引脚或其他引脚共用功能。

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 时对应 RX7) 或第 9 位 (BON=1 时对应 RX8) 为高, 那么接到的是地址而非数据。若相应的中断使能, 则每次接收到的数据的地址位 (根据 BNO 的值来确定是第 8 或是第 9 位) 置位时都将产生中断请求。若地址检测功能使能, 所接收的地址位为 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 UART 功能使能位
0: RX 脚下降沿唤醒 UART 功能除能
1: RX 脚下降沿唤醒 UART 功能使能
此位用于控制 RX 引脚下降沿时是否唤醒 UART 功能。此位仅当 UART 时钟源关闭时有效。若 UART 时钟源还开启, 则 RX 引脚唤醒 UART 功能无效。若此位置高且 UART 时钟关闭, 当 RX 引脚发生下降沿时会产生 UART 唤醒请求。若相应的中断使能, 将产生 RX 引脚唤醒 UART 的中断, 以告知单片机使其通过应用程序开启 UART 时钟源, 从而唤醒 UART 功能。否则, 若此位为低, 即使 RX 引脚发生下降沿也无法恢复 UART 功能。

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。

● TXR/RXR 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	x	x	x	x	x	x	x	x

“x”：未知

Bit 7~0 **D7~D0**: UART 发送 / 接收数据位 Bit 7~Bit 0

● BRG 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	x	x	x	x	x	x	x	x

“x”：未知

Bit 7~0 **D7~D0**: 波特率值

通过对 UCR2 寄存器的 BRGH 位进行编程选择上述相关规格以及 BRG 寄存器中的所需值可设定所需波特率。

注：若 BRGH=0，Baud rate= $f_{\text{SYS}}/[64 \times (N+1)]$

若 BRGH=1，Baud rate= $f_{\text{SYS}}/[16 \times (N+1)]$

波特率发生器

UART 自身具有一个波特率发生器，通过它可以设定串行数据通信速率。波特率由一个独立的内部 8 位计数器所控制，其周期取决于两个因素。第一个因素为波特率寄存器 BRG 中的值，第二个因素则是控制寄存器 UCR2 中的 BRGH 位的值。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。下列波特率计算公式中所用 BRG 寄存器的值 N 决定分频因数。注意 BRG 寄存器中的 N 为十进制数，其范围是 0 到 255。

UCR2 BRGH Bit	0	1
Baud Rate (BR)	$f_{\text{SYS}}/[64 (N+1)]$	$f_{\text{SYS}}/[16 (N+1)]$

通过对 UCR2 寄存器的 BRGH 位进行编程选择上述相关公式以及 BRG 寄存器中的所需值可设定所需波特率。注意，由于实际波特率值取决于 BRG 寄存器中的离散值 N，在实际值与理论值之间会出现相关误差。下面举例说明如何计算 BRG 寄存器 N 值和误差值。

波特率和误差计算

系统选用 4MHz 时钟频率且 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。

根据上表，波特率 $BR = f_{SYS} / [64 (N+1)]$

转换后的公式 $N = f_{SYS} / [(BR \times 64)] - 1$

带入参数 $N = [4000000 / (4800 \times 64)] - 1 = 12.0208$

取最接近的值，十进制 12 写入 BRG 寄存器，实际波特率如下

$BR = 4000000 / [64 \times (12+1)] = 4808$

因此，误差 = $(4808 - 4800) / 4800 = 0.16\%$

下面表格给出 BRGH 取不同值时的实际波特率和误差。

波特率 K/BPS	Baud Rates for BRGH=0		
	$f_{SYS}=16\text{MHz}$		
	BRG	Kbaud	误差 (%)
0.3	—	—	—
1.2	207	1.202	0.16
2.4	103	2.404	0.16
4.8	51	4.808	0.16
9.6	25	9.615	0.16
19.2	12	19.231	0.16
38.4	6	35.714	-6.99
57.6	3	62.5	8.51
115.2	1	125	8.51
250	0	250	0

BRGH=0 时的波特率和误差值

波特率 K/BPS	Baud Rates for BRGH=1		
	$f_{SYS}=16\text{MHz}$		
	BRG	Kbaud	误差 (%)
0.3	—	—	—
1.2	—	—	—
2.4	—	—	—
4.8	207	4.808	0.16
9.6	103	9.615	0.16
19.2	51	19.231	0.16
38.4	25	38.462	0.16
57.6	16	58.824	2.12
115.2	8	111.11	-3.55
250	3	250	0

BRGH=1 时的波特率和误差

UART 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 位起始位，8 位或 9 位数据位和 1 位或者 2 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，无校验，1 位停止位组成，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 等位设定。用于数据发送和接收的波特率由一个内部的 8 位波特率发生器产生，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 都置位，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，使其可作为普通输入 / 输出或其他引脚共用功能。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，另外误差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 UCR1 寄存器中的 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。

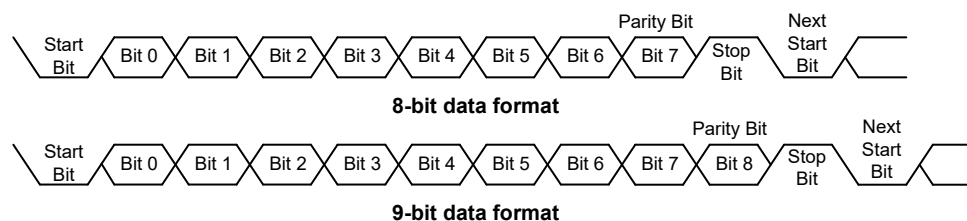
数据位、奇偶校验和停止位选择

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等多种因素组成。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

下图为 8 位和 9 位数据格式的发送和接收波形。



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。应注意的是 TSR 不像其它寄存器一样直接映射到数据存储，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 寄存器没有加载数据且波特率没有设置一个移位时钟源，发送器将不会工作。先写 TXR 寄存器再置高 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 引脚可用作普通输入 / 输出引脚或其他引脚复用功能。

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。应注意的是如果选择 9 位数据传输格式，最高位 MSB 存储在 UCR1 寄存器的 TX8 中。发送器的触发可由如下步骤完成：

- 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，选择所需波特率。
- 置高 TXEN，确保 TX 作为 UART 的发送端引脚。
- 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。
- 要继续发送数据可重复上述动作。

需要注意的是，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：

1. 读取 USR 寄存器
2. 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 标志位会产生中断。

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。在发送停止位或暂停帧后，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：

1. 读取 USR 寄存器
2. 写 TXR 寄存器

注意，清除 TXIF 和 TIDLE 软件执行次序相同

发送暂停

若 TXBRK=1 保持时间超过 $[(BRG+1) \times t_{th}]$ 且 TIDLE=1，下一帧将会发送暂停字。它是由一个起始位、 $13 \times N$ ($N=1, 2, \dots$) 位逻辑 0 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 外部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 寄存器中加载到为空的接收数据寄存器 RXR 中。RX 引脚上的每一位数据会被采样三次以判断其逻辑状态。应注意 RSR 不像其它寄存器一样映射在数据存储区，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，连续地从外部 RX 引脚进入。在读取模式中，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。注意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。接收器的启动可由如下步骤完成：

- 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，选择所需波特率。
- 置高 RXEN，确保 RX 引脚作为 UART 的接收端引脚。

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 寄存器中存在有效数据时，USR 寄存器中的 RXIF 位将会置位，至少一个字符可被读取。
- 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：

1. 读取 USR 寄存器
2. 读取 RXR 寄存器

接收暂停

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、数据位和无效停止位将会置位 FERR 标志，且在下一起始位前必须检测到有效的停止位。接收器不会将线上的暂停状态当成下一个起始位。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。注意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，即在起始位和停止位之间，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 UART 如何处理。

溢出错误 – OERR 标志位

RXR 寄存器是一个两层的 FIFO 缓冲器，它能在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，通过溢出错误标志位 OERR 显示。

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。

先读取 USR 寄存器再读取 RXR 寄存器可将 OERR 清零。

噪声干扰 – NF 标志位

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 寄存器可将 NF 清零。

帧错误 – FERR 标志位

如果检测到 0 而不是停止位，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 标志位

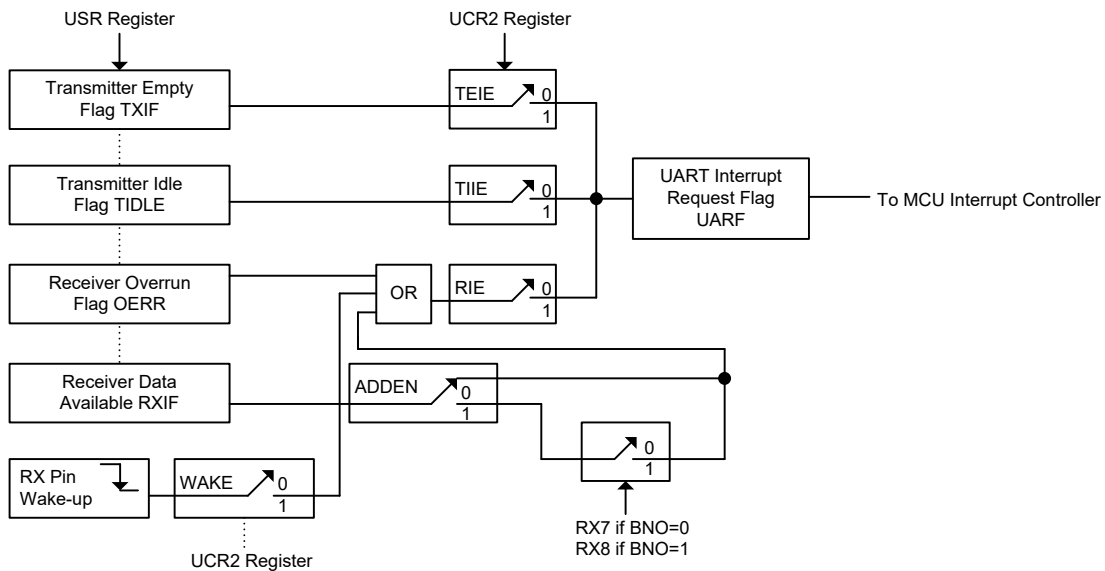
若接收到数据出现奇偶校验错误，USR 寄存器中只读标志 PERR 置位。只有使能了奇偶校验，选择了校验类型，此标志位才有效。它同数据一起存储在缓冲器中，可被任何复位清除。注意，FERR 和 PERR 与相应的数据一起存储于缓冲器中，在读取数据之前必须先访问错误标志位。

UART 中断结构

以下几种情况将产生 UART 中断，即发送寄存器为空、发送器空闲、接收器数据有效、接收器溢出、地址检测和 RX 引脚唤醒都会产生中断。在出现上述情况时，将产生一个低脉冲以引起单片机注意。当其中任何一种情况发生时，若其对应的中断控制位使能、整体 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。这其中的四种情况在 USR 寄存器中有相关的标志位，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两个接收器中断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 使其功能，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 UCR 寄存器中的 WAKE 和 RIE 位被置位，由 RX 引脚上的下降沿唤醒单片机时将会产生 RX 唤醒中断。此时将有一定的延时周期，即系统启动延时，以供振荡器重启并在系统恢复正常操作之前达到稳定。

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，以确定屏蔽或同意 UART 模块的中断请求。



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 使能，则在数据有效时只有在接收到的数据最高位为 1 才会产生中断，注意中断使能位 MFE.URE 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一位为高才会产生中断。若 ADDEN 除能，每次置位 RXIF 都会产生接收有效数据中断，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，为保证操作的正确必须将奇偶检验使能位清零，除能奇偶校验。

ADDEN	9th Bit (BNO=1) 8th Bit (BNO=0)	产生 UART 中断
0	0	√
	1	√
1	0	×
	1	√

ADDEN 位功能表

UART 暂停和唤醒

当 f_{sys} 关闭，UART 功能停止。此时该模块的所有时钟源关闭。若 f_{sys} 在数据传输进行中关闭，发送将停止直到 UART 模块时钟再次使能。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 MCU 进入暂停模式，注意 USR、UCR1、UCR2、接收 / 发送寄存器以及 BRG 寄存器都不会受到影响。建议在 MCU 进入暂停模式前先确保数据发送或接收已完成。

UART 功能中包括了 RX 引脚的唤醒功能，由 UCR2 寄存器中 WAKE 位控制使能或除能。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 引脚的下降沿可将单片机从暂停模式中唤醒。应注意唤醒后系统需要一定的系统时钟周期才能正常工作，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 UART 中断使能控制位 UARE 也必须置位；若这两个控制位没有被置位，那么，单片机将可以被唤醒但不会产生中断。注意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。

中断

中断是单片机一个重要功能。当一个外部事件或内部功能如定时器模块要求单片机注意时，其响应中断将强制主程序暂时中止，使得单片机可以直接注意到其需求。此系列单片机提供多个外部中断和内部中断，外部中断由外部 INT0~INT1 引脚信号触发，而内部中断由多种内部功能如 TM、LVD、EEPROM、SIM、UART、A/D 转换器和时基等产生。

中断寄存器

所有的中断控制，即在单片机发生一些情况时设定的请求标志位和通过应用程序设置的中断使能位，由位于特殊功能数据存储器中的一系列寄存器控制，如下表所示。寄存器数目取决于所选单片机，一般分为三类。第一类为 INTC0~INTC2 寄存器，设置基本中断；第二类为 MFI0~MFI2 寄存器，设置多功能中断；最后一类是一个 INTEG 寄存器，设置外部中断触发边沿类型。

每个寄存器包含几个控制单个寄存器使能或除能的使能位和表明出现中断请求的中断标志位。其命名规格遵循特定模式。首先是中断类型简写，然后是中断的数目 (可选)，接着是表示使能 / 除能位的字母 “E” 或表示请求标志位的字母 “F”。

功能	使能位	请求标志	注释
总中断	EMI	—	—
INTn 引脚	INTnE	INTnF	n=0~1
时基	TBnE	TBnF	n=0~1
多功能	MFnE	MFnF	n=0~2
SIM	SIME	SIMF	—
UART (HT69F360/HT67F370)	URE	URF	—
A/D 转换器 (HT67F370)	ADE	ADF	—
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
CTM	CTMPE	CTMPF	—
	CTMAE	CTMAF	—
STM	STMPE	STMPF	—
	STMAE	STMAF	—
PTM (HT69F340/HT69F350)	PTMPE	PTMPF	—
	PTMAE	PTMAF	
PTM (HT69F360/HT67F370)	PTMnPE	PTMnPF	n=0~1
	PTMnAE	PTMnAF	

中断寄存器位命名规范

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
INTC2	—	—	—	SIMF	—	—	—	SIME
MF10	—	—	CTMAF	CTMPF	—	—	CTMAE	CTMPE
MF11	—	—	PTMAF	PTMPF	—	—	PTMAE	PTMPE
MF12	—	—	DEF	LVF	—	—	DEE	LVE

中断寄存器列表 – HT69F340

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
INTC2	—	—	—	SIMF	—	—	—	SIME
MF10	STMAF	STMPF	CTMAF	CTMPF	STMAE	STMPE	CTMAE	CTMPE
MF11	—	—	PTMAF	PTMPF	—	—	PTMAE	PTMPE
MF12	—	—	DEF	LVF	—	—	DEE	LVE

中断寄存器列表 – HT69F350

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
INTC2	—	—	UARTF	SIMF	—	—	UARTE	SIME
MF10	STMAF	STMPF	CTMAF	CTMPF	STMAE	STMPE	CTMAE	CTMPE
MF11	PTM1AF	PTM1PF	PTM0AF	PTM0PF	PTM1AE	PTM1PE	PTM0AE	PTM0PE
MF12	—	—	DEF	LVF	—	—	DEE	LVE

中断寄存器列表 – HT69F360

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
INTC2	—	ADF	URF	SIMF	—	ADE	URE	SIME
MF10	STMAF	STMPF	CTMAF	CTMPF	STMAE	STMPE	CTMAE	CTMPE
MF11	PTM1AF	PTM1PF	PTM0AF	PTM0PF	PTM1AE	PTM1PE	PTM0AE	PTM0PE
MF12	—	—	DEF	LVF	—	—	DEE	LVE

中断寄存器列表 – HT67F370

• INTEG 寄存器

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 未定义，读为“0”

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

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

• INTC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	TB0F	INT1F	INT0F	TB0E	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 未定义，读为“0”

Bit 6 **TB0F**: 时基 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 **TB0E**: 时基 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

● INTC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
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 **MF2F**: 多功能中断 2 请求标志位
0: 无请求
1: 中断请求
- 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 **TB1F**: 时基 1 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **MF2E**: 多功能中断 2 控制位
0: 除能
1: 使能
- 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 **TB1E**: 时基 1 中断控制位
0: 除能
1: 使能

● INTC2 寄存器 – HT67F370

Bit	7	6	5	4	3	2	1	0
Name	—	ADF	URF	SIMF	—	ADE	URE	SIME
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 **ADF**: A/D 转换器中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 **URF**: UART 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 未定义，读为“0”
- Bit 2 **ADE**: A/D 转换器控制位
0: 除能
1: 使能

- Bit 1 **URE:** UART 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

● **INTC2 寄存器 – HT69F340/HT69F350**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	SIMF	—	—	—	SIME
R/W	—	—	—	R/W	—	—	—	R/W
POR	—	—	—	0	—	—	—	0

- Bit 7~5 未定义，读为 “0”
- Bit 4 **SIMF:** 串行接口模块请求标志位
0: 无请求
1: 中断请求
- Bit 3~1 未定义，读为 “0”
- Bit 0 **SIME:** 串行接口模块中断控制位
0: 除能
1: 使能

● **INTC2 寄存器 – HT69F360**

Bit	7	6	5	4	3	2	1	0
Name	—	—	UARTF	SIMF	—	—	UARTE	SIME
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义，读为 “0”
- Bit 5 **UARTF:** UART 请求标志位
0: 无请求
1: 中断请求
- Bit 4 **SIMF:** 串行接口模块请求标志位
0: 无请求
1: 中断请求
- Bit 3~2 未定义，读为 “0”
- Bit 1 **UARTE:** UART 中断控制位
0: 除能
1: 使能
- Bit 0 **SIME:** 串行接口模块中断控制位
0: 除能
1: 使能

● MF10 寄存器 – HT69F340

Bit	7	6	5	4	3	2	1	0
Name	—	—	CTMAF	CTMPF	—	—	CTMAE	CTMPE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 未定义，读为 "0"

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

Bit 4 **CTMPF**: CTM 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 未定义，读为 "0"

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

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

● MF10 寄存器 – HT69F350/HT69F360/HT67F370

Bit	7	6	5	4	3	2	1	0
Name	STMAF	STMPF	CTMAF	CTMPF	STMAE	STMPE	CTMAE	CTMPE
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 **STMAF**: STM 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求

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

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

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

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

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

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

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

● MFI1 寄存器 – HT69F340/HT69F350

Bit	7	6	5	4	3	2	1	0
Name	—	—	PTMAF	PTMPF	—	—	PTMAE	PTMPE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 未定义，读为“0”

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

Bit 4 **PTMPF**: PTM 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 未定义，读为“0”

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

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

● MFI1 寄存器 – HT69F360/HT67F370

Bit	7	6	5	4	3	2	1	0
Name	PTM1AF	PTM1PF	PTM0AF	PTM0PF	PTM1AE	PTM1PE	PTM0AE	PTM0PE
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 **PTM1AF**: PTM1 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求

Bit 6 **PTM1PF**: PTM1 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求

Bit 5 **PTM0AF**: PTM0 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求

Bit 4 **PTM0PF**: PTM0 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求

Bit 3 **PTM1AE**: PTM1 比较器 A 匹配中断控制位
0: 除能
1: 使能

Bit 2 **PTM1PE**: PTM1 比较器 P 匹配中断控制位
0: 除能
1: 使能

Bit 1 **PTM0AE**: PTM0 比较器 A 匹配中断控制位
0: 除能
1: 使能

Bit 0 **PTM0PE**: PTM0 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	—	—	DEE	LVE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义，读为“0”
- Bit 5 **DEF**: 数据 EEPROM 中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **LVF**: LVD 中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 3~2 未定义，读为“0”
- Bit 1 **DEE**: 数据 EEPROM 中断控制位
 0: 除能
 1: 使能
- Bit 0 **LVE**: LVD 中断控制位
 0: 除能
 1: 使能

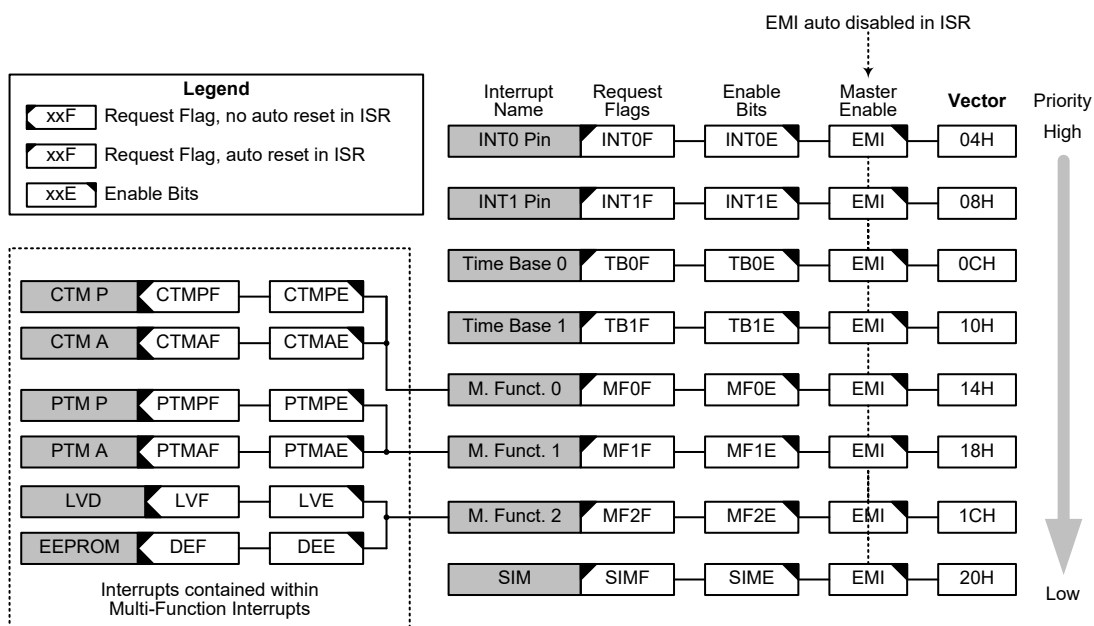
中断操作

当发生中断事件，如 TM 比较器 P 或比较器 A 匹配，相关的中断请求标志位将被置位。请求标志位能否使程序跳转到相关中断服务取决于中断使能位的状态。若使能位置高，程序将跳转到相应的终端服务程序中。反之，若使能位为零，即使设置了请求标志位也不会产生中断，程序也不会跳转到相应中断服务程序。若总中断使能位清零，则所有中断除能。

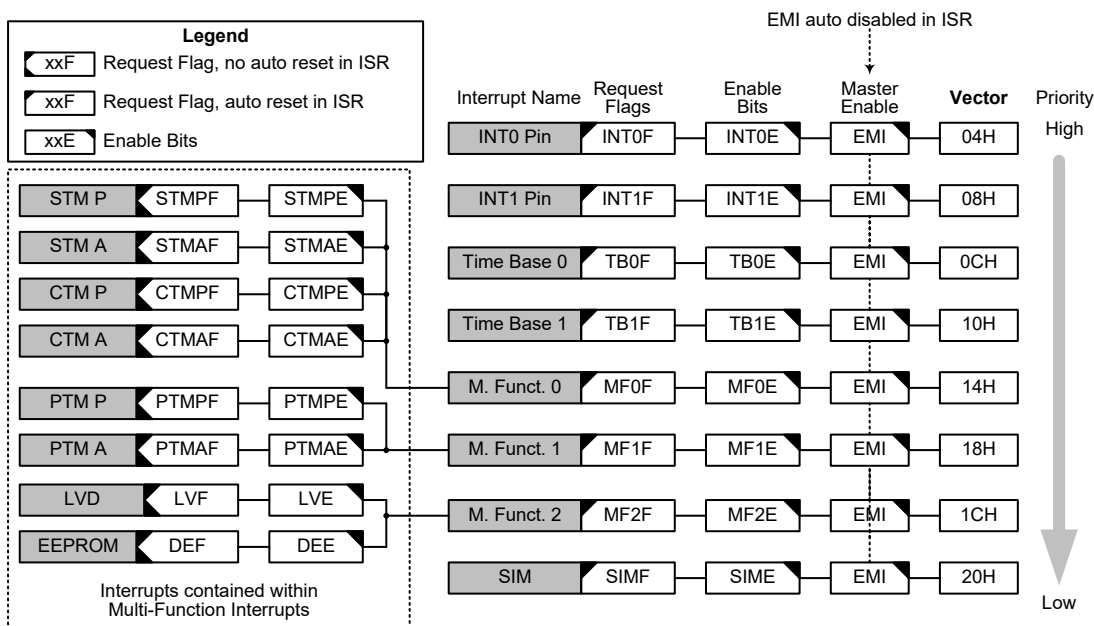
当中断产生，存储下一条执行指令地址的程序计数器 PC 将被压入堆栈。此时将相应中断向量值作为新地址加载至 PC。单片机将从该中断向量中取得下一条指令，一般为“JMP”指令，程序将跳转到中断服务程序。这部分的代码可控制相应的中断。中断服务程序必须以“RETI”指令结束，从堆栈中重新获取原始 PC 地址，使得单片机能从中断发生的地方继续开始正常操作。

各个中断使能位以及相应的请求标志位，以优先级的顺序如下图所示。其中几个中断源有自己的中断向量，但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，而其他中断都将被屏蔽，防止发生进一步的中断嵌套。但在此期间若出现其他中断请求，则中断无法立即响应，但会记录下请求标志位。

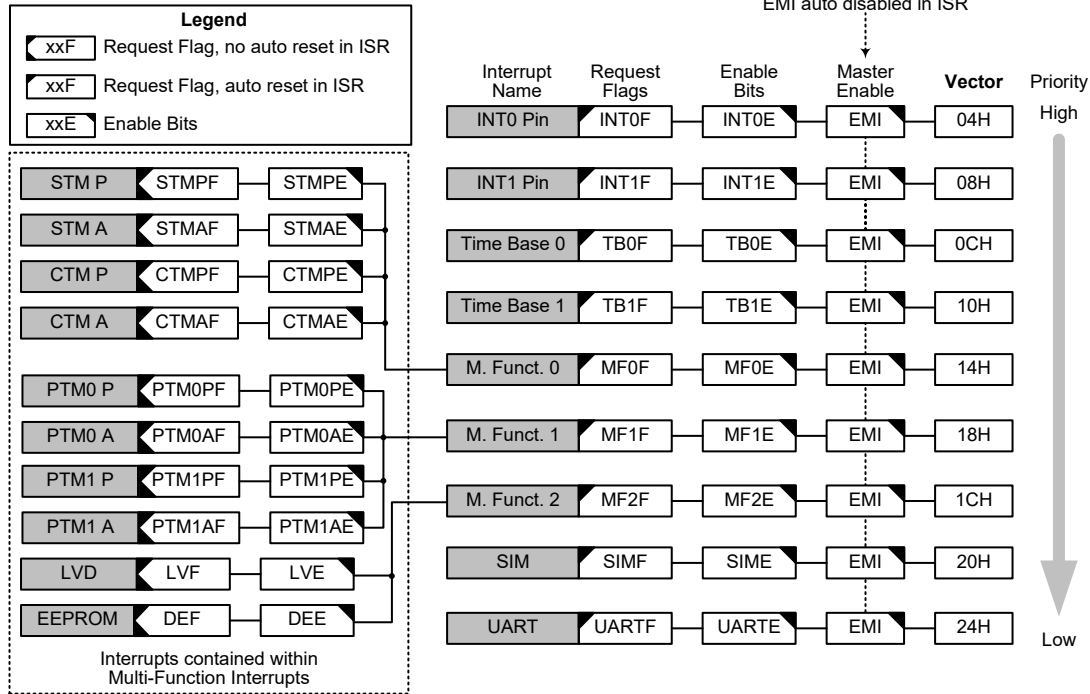
如果某个中断服务子程序正在执行时，有另一个中断要求立即响应，那么 EMI 位应在程序进入中断子程序后置位，以允许此中断嵌套。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到 SP 减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，执行优先级如下流程图所示。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应将相应的标志置起。



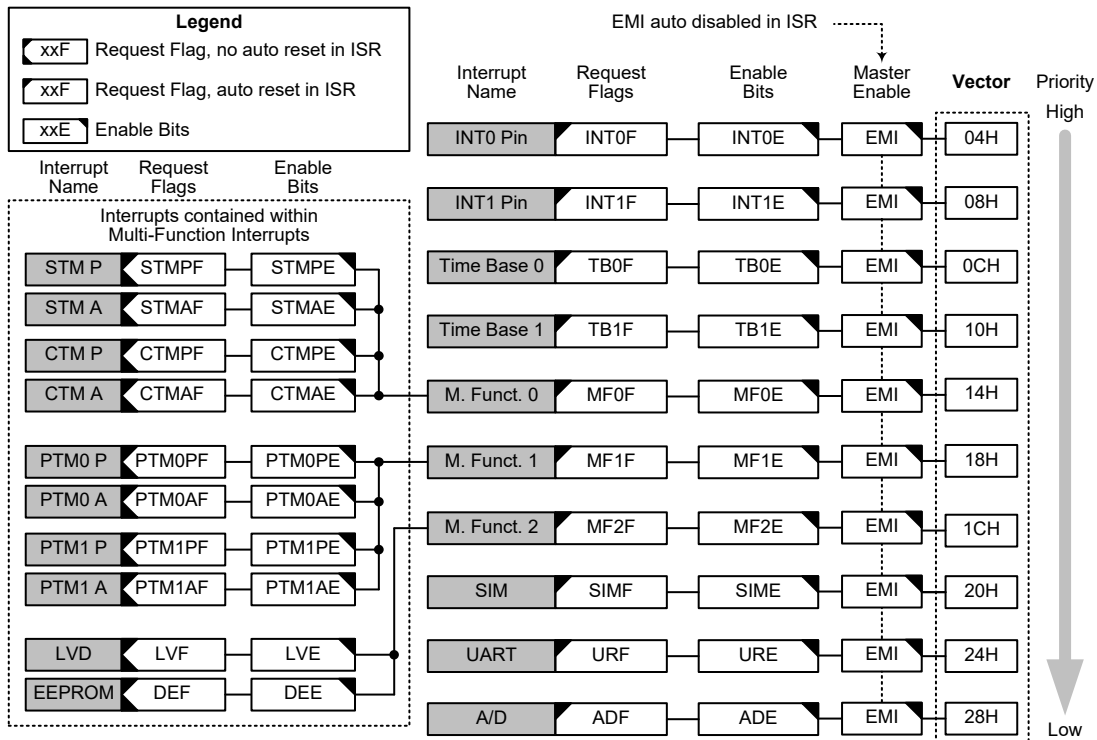
中断结构 – HT69F340



中断结构 – HT69F350



中断结构 – HT69F360



中断结构 – HT67F370

外部中断

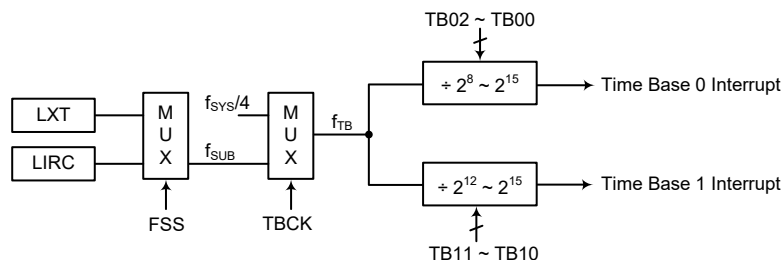
通过 INT0~INT1 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，外部中断引脚的状态发生变化，外部中断请求标志 INT0F~INT1F 被置位时外部中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和外部中断使能位 INT0E~INT1E 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。由于外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，则此引脚只作为外部中断脚使用。此时该引脚必须通过设置端口控制寄存器的相关共用引脚选择位，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态出现所需转变，将调用外部中断向量子程序。当响应外部中断服务子程序时，外部中断请求标志位 INT0F~INT1F 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，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产生外部中断。注意 INTEG 也可以用来除能外部中断功能。

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。当出现这种信号时，其中断请求标志位 TBnF 将置位。当总中断使能位 EMI 和时基使能位 TBnE 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 TBnF 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。其时钟源 f_{TB} 来自内部时钟源 f_{SUB} 或 $f_{SYS}/4$ 。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。控制时基中断周期的时钟源可通过 TBC 寄存器中的相应位选择。



时基中断

• TBC 寄存器

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

Bit 7 **TBON**: TB0 和 TB1 控制位
0: 除能
1: 使能

Bit 6	TBCK: f_{TB} 时钟选择位 0: f_{SUB} 1: $f_{SYS}/4$
Bit 5~4	TB11~TB10: 时基 1 溢出周期选择位 00: $2^{12}/f_{TB}$ 01: $2^{13}/f_{TB}$ 10: $2^{14}/f_{TB}$ 11: $2^{15}/f_{TB}$
Bit 3	未定义, 读为 “0”
Bit 2~0	TB02~TB00: 时基 0 溢出周期选择位 000: $2^8/f_{TB}$ 001: $2^9/f_{TB}$ 010: $2^{10}/f_{TB}$ 011: $2^{11}/f_{TB}$ 100: $2^{12}/f_{TB}$ 101: $2^{13}/f_{TB}$ 110: $2^{14}/f_{TB}$ 111: $2^{15}/f_{TB}$

多功能中断

此系列单片机中有多达三种多功能中断。与其它独立中断不同, 它没有独立源, 而是由其它现有的中断源构成, 即 TM 中断, EEPROM 中断和 LVD 中断。

当多功能中断中任何一种中断请求标志位 MF0F~MF2F 被置位, 将出现多功能中断请求。而当其所包含功能产生中断请求标志位将置位多功能中断标志位。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, EEPROM 中断和 LVD 中断, 不会自动复位, 必须由应用程序手动复位。

EEPROM 中断

EEPROM 中断也属于多功能中断。当写周期结束, 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

低压检测 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、标准型和周期型 TM 各有两个中断，都属于多功能中断。所有的 TM 中断都包含有两个中断请求标志位 xTMnPF 和 xTMnPF 及两个使能位 xTMnPE 和 xTMnAE。当 TM 比较器 P 或比较器 A 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 MFnF 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程序执行。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。

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 接口完成接收或发送，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 需先被置位。当中断使能，堆栈未满且以上情况发生时，可跳转至相应 SIM 中断向量程序执行。当串行接口中断响应，中断请求标志位 SIMF 将自动复位，EMI 也将自动清零以除能其它中断。

UART 中断

UART 模块中，发送器为空、发送器空闲、接收器数据有效、接收器溢出、地址检测和 RX 引脚唤醒都会触发产生 UART 中断。当出现以上任意一种情况时，将产生一个低脉冲以引起单片机的注意。若要程序跳转到相应中断向量地址，总中断控制位 EMI 和 UART 中断使能位 UARTE 需先被设置。当中断使能，堆栈未满且以上任一情况发生时，可跳转至 UART 中断向量程序执行。当 UART 中断响应时，其中断标志位 UARTE 将被自动清零。EMI 也将自动清零以除能其他中断。但 USR 寄存器标志位只有在 UART 执行特定动作后才可被清零，详见 UART 章节。

A/D 转换器中断

A/D 转换器中断由 A/D 转换动作的结束来控制。当 A/D 转换器中断请求标志 ADF 被置位，即 A/D 转换过程完成时，中断请求发生。若要跳转到相应中断向量地址，总中断控制位 EMI 和 A/D 转换器中断使能位 ADE 需先被置位。当中断使能，堆栈未满且 A/D 转换动作结束时，可跳转至 A/D 转换器中断向量程序执行。当 A/D 转换器中断响应时，相应的中断请求标志位 ADF 会自动清零且 EMI 位会被清零以除能其它中断。

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

通过除能相关中断使能位，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 MFnF 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”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累加器，状态寄存器或其它的寄存器的内容，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，保留 EMI 位 0 状态，除能进一步中断。

低电压检测 – LVD

此系列单片机都具有低电压检测功能，即 LVD。该功能使能单片机监测电源电压 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，电池供电电压随着其使用时间的增长逐渐减少，此功能可提前产生一个电池低电压警告信号。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。其中的 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	LVPU	VLVD2	VLVD1	VLVD0
R/W	—	—	R	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”

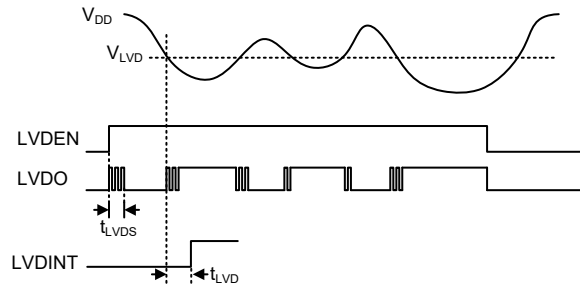
Bit 5 **LVDO**: LVD 输出标志位
0: 无低压检测
1: 低压检测

Bit 4 **LVDEN**: 低压检测控制位
0: 除能
1: 使能

Bit 3	LVPU: I/O 端口上拉电阻控制位 0: 电压为 5V, 所有引脚上拉电阻为 60k Ω (正常模式) 1: 电压为 5V, 所有引脚上拉电阻为 15k Ω (低压模式)
Bit 2~0	VLVD2~VLVD0: LVD 电压选择位 000: 1.8V 001: 2.0V 010: 2.4V 011: 2.7V 100: 3.0V 101: 3.3V 110: 3.6V 111: 4.0V

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 1.8V~4.0V。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。当单片机处于休眠模式, 低电压检测将自动除能。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。只有在低压情况出现置位 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, 在单片机进入休眠模式前应将 LVF 标志置为高。

LCD 驱动

对于设计中带有 LCD 功能的大批量应用，选择定制而非较昂贵的基于字符的显示方式可以有效地降低成本。然而，驱动此类定制的显示器需要振幅及时间可变的 COM 和 SEG 信号，且需要很多特殊的考虑以正确地操作 LCD。此系列单片机有内部 LCD 信号产生电路及多种配置选项，可以自动地产生时间与振幅可变的信号直接驱动 LCD，与用户 LCD 的接口连接也相当容易。

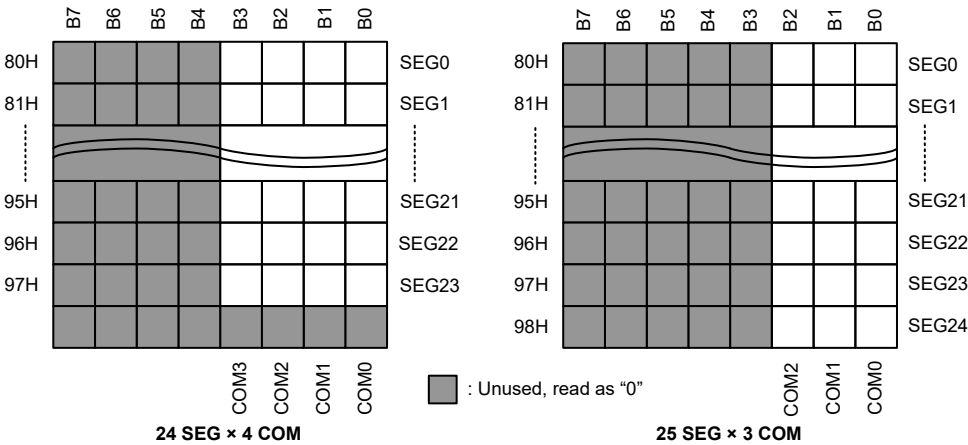
单片机型号	占空比	驱动输出	偏压	偏压类型	波形类型
HT69F340	1/4 或 1/3	24×4 或 25×3	1/3 或 1/2	C 或 R	A 或 B
HT69F350	1/4 或 1/3	36×4 或 37×3	1/3 或 1/2	C 或 R	A 或 B
HT69F360/ HT67F370	1/4 或 1/3	48×4 或 49×3	1/3 或 1/2	C 或 R	A 或 B

LCD 显示存储器

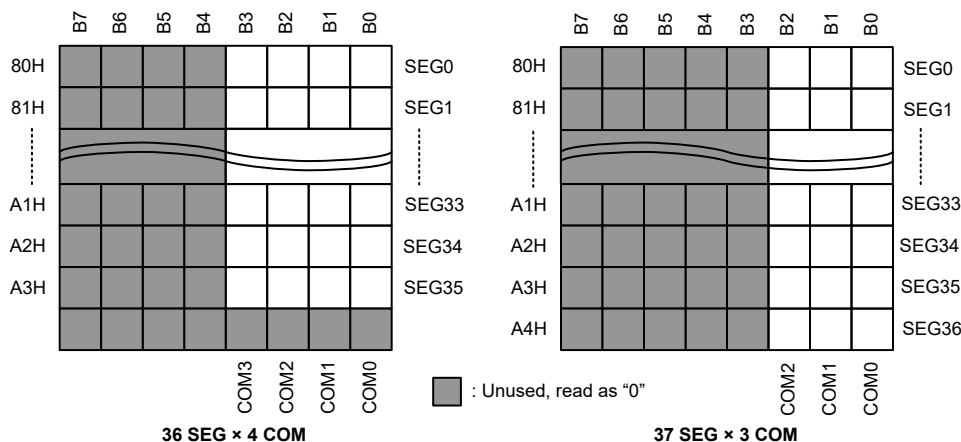
数据存储器中有一部分区域是专门为 LCD 的显示数据而保留，即 LCD 存储。单片机内部显示驱动电路会自动读取任何写入此处的数据并据此产生 LCD 驱动信号。因此任何写入 LCD 存储器的数据，会立即映射到连接单片机的 LCD 显示器上。

由于 LCD 存储器地址与通用数据存储器地址重叠，因此 LCD 数据存储于独立的数据存储区 Sector 1 中。数据存储器 Sector 的选择是通过数据存储器中的一个特殊功能寄存器来完成的，即存储器指针高字节寄存器 MP1H 或 MP2H。当要读取 LCD 存储器时，首先要将 MP1H 或 MP2H 的值设为“01H”来选择对 Sector 1 操作。此后，用户可以通过存储器指针低字节 MP1L 或 MP2L 使用间接寻址方式来对存储区进行操作。选择了 Sector 1 之后，使用 MP1L 或 MP2L 可以对以“80H”作为起始地址的存储区操作，就可以直接对显示存储区进行读或者写的操作了。直接寻址显示存储区可由相关扩展指令实现。

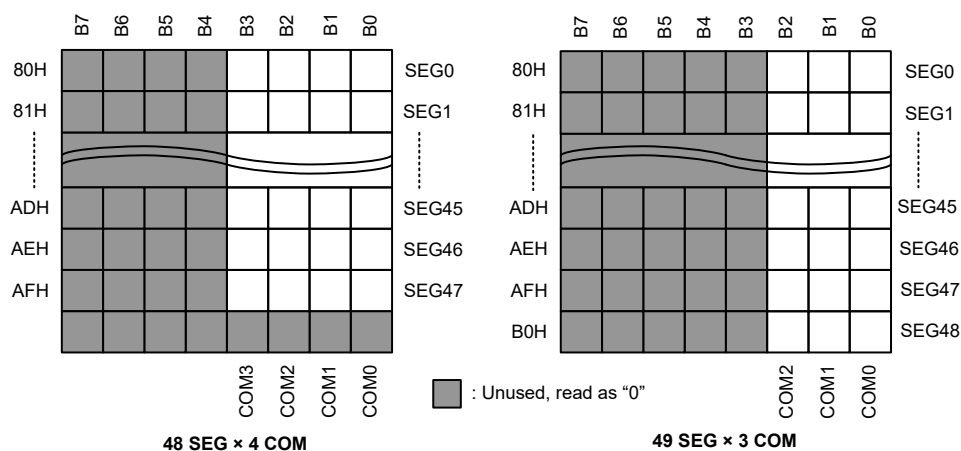
所附 LCD 存储器映射图显示了此系列单片机内部 LCD 存储器与 COM、SEG 端的显示映射。该系列单片机中小容量存储的 LCD 存储器映射可由下图推出。



LCD 存储映射 – HT69F340



LCD 存储映射 – HT69F350



LCD 存储映射 – HT69F360/HT67F370

LCD 时钟源

LCD 时钟是由内部时钟源 f_{SUB} 通过内部分频电路进行 8 分频获得。通过对 SCC 寄存器中的 FSS 位进行配置， f_{SUB} 内部时钟源可由 LIRC 或 LXT 振荡器提供。该方法用于产生理想的频率为 4kHz 的 LCD 时钟，以获得更好的 LCD 显示效果。

LCD 寄存器

LCD 控制寄存器 LCDC0 和 LCDC1 位于数据存储区，用于设置 LCD 驱动器的各种特性。这些寄存器的各个位可用来设置如 LCD 波形类型、占空比类型、偏压占空比、偏压电阻选择、R/C 类型以及 LCD 的使能 / 除能和 LCD 电源选择。寄存器 LCDC0 中的 LCDEN 位只有当单片机工作于正常模式、低速模式或空闲模式时才可以控制 LCD 的使能与除能，如果单片机处于休眠模式，则显示将一直处于除能状态。寄存器 LCDC0 中的 RSEL1~RSEL0 位用于选择内部偏压电阻来提供 LCD 适当的偏压电流。在应用中，选择匹配的 LCD 面板也可以降低偏压电流。此外，LCDC0 寄存器中的 TYPE 位是用于选择 A 型或 B 型的 LCD 控制信号。显示引脚 SEGn 的输出功能是被用作 segment 驱动、I/O 口或是其他功能则由引脚共用功能选择寄存器决定。

寄存器名称	位							
	7	6	5	4	3	2	1	0
LCDC0	TYPE	—	DTYC	—	BIAS	RSEL1	RSEL0	LCDEN
LCDC1	—	—	—	—	—	RCT	LCDP1	LCDP0

● LCDC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TYPE	—	DTYC	—	BIAS	RSEL1	RSEL0	LCDEN
R/W	R/W	—	R/W	—	R/W	R/W	R/W	R/W
POR	0	—	0	—	0	0	0	0

Bit 7 **TYPE**: LCD 波形选择控制位

0: A 型

1: B 型

Bit 6 未定义, 读为 “0”

Bit 5 **DTYC**: LCD 占空比控制位

0: 1/3 占空比

1: 1/4 占空比

Bit 4 未定义, 读为 “0”

Bit 3 **BIAS**: LCD 偏压控制位

0: 1/2 偏压

1: 1/3 偏压

Bit 2~1 **RSEL1~RSEL0**: LCD 偏压电阻选择位

1/3 偏压

00: 600kΩ

01: 300kΩ

10: 100kΩ

11: 50kΩ

1/2 偏压

00: 400kΩ

01: 200kΩ

10: 67kΩ

11: 34kΩ

Bit 0 **LCDEN**: LCD 使能控制位

0: 除能

1: 使能

在正常, 低速或空闲模式下, LCD on/off 功能可由此位控制。在休眠模式下, LCD 保持关闭。

● LCDC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	RCT	LCDP1	LCDP0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 未定义, 读为 “0”

Bit 2 **RCT**: LCD R/C 类型控制位

0: R 型

1: C 型

Bit 1~0 **LCDP1~LCDP0: C 型 LCD 电源选择位**

00: 电源来自 VLCD/V1/V2

01: 电源来自 VC=DPN Vref (~1.04V)

10: 电源来自 VB=V_{DD}

11: 电源来自 VA=V_{DD}

若 LCDEN=1 且 RCT=1, 则 PC0~2 将分别为 V2, C1, C2 功能。

LCD 复位功能

LCD 内建复位功能, 即 LCDC0 寄存器中的 LCDEN 位反相后与 SLEEP 功能的或运算。当 LCDEN 在单片机进入休眠模式前置为“1”以使能 LCD 驱动功能, LCD 将在单片机进入休眠模式后复位。清零 LCDEN 也将复位 LCD 功能。

LCDEN	SLEEP 模式	复位 LCD
0	Off	√
0	On	√
1	Off	×
1	On	√

LCD 复位功能

LCD 驱动输出

LCD 驱动器提供的 COM 和 SEG 输出数目, 以及波形类型, R/C 类型, 偏压和占空比的选择, 取决于 LCD 控制位的设置。

由于 LCD 基本性质的缘故, 它们的像素点只能加上 AC 电压, 如果加上 DC 电压, 将会引起永久性的损害。因此 LCD 显示器的对比度由提供到每个像素的实际 RMS 电压控制, 这个值相当于 COM 引脚上的 RMS 电压值减去 SEG 引脚上的电压值。RMS 电压必须大于 LCD 的饱和电压, 以便能打开像素点, 但同时也要小于阈值电压, 以便能关闭像素点。

若要将 DC 电压限制为 0 且以尽量少的连接数来控制尽可能多的像素点, 需要产生时间振幅可变的信号供给 LCD 使用。这些时间与振幅都可变的信号由单片机内的 LCD 驱动电路自动产生。占空比决定使用 common 口的个数, 也称为底板或 COMs。1/4 占空比的 COM 口个数为 4, 因此该值定义了每个 LCD 信号帧内的时间片数。单片机提供两种类型的信号即 A 型和 B 型, 通过寄存器 LCDC0 中的 TYPE 位加以选择。B 型提供较低频率的信号, 然而, 较低的频率可能引起闪烁, 从而影响显示的清晰度。

LCD 电压源偏压

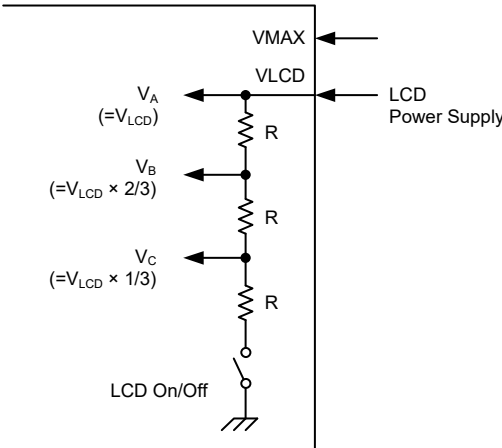
LCD 驱动器需要几种电压值以产生时间振幅可变的信号。该系列单片机可以通过软件控制 LCDC1 寄存器的 RCT 位来选择 R 型或 C 型的偏压类型。选择 C 型偏压将使能内部充电泵电路。

R 型偏压

对于 R 型偏压, 必须要在 VLCD 引脚上提供外部 LCD 电压源, 以产生内部偏压电压。这个外部电压源可以是单片机的电压源也可以是其它电压源。对于 R 型 1/2 偏压的结构, 要用到 V_{SS}、V_A 和 V_B 三种电压。V_A 等于 VLCD 引脚上的外部供电电压源。V_B 由单片机内部产生, 其值等于 V_{LCD}/2。对于 R 型 1/3 偏压的结构, 要用到 V_{SS}、V_A、V_B 和 V_C 四种电压值。V_A 等于 V_{LCD}, V_B 等于 V_{LCD}×2/3, V_C 等于 V_{LCD}×1/3。

对于 1/2 或 1/3 偏压，可利用 LCDC0 寄存器的 RSEL0 和 RSEL1 位来选择不同值的内部偏置电阻。内部偏置电阻和 VLCD 引脚上的电压共同决定了偏置电流。VLCD 引脚电压决定了 VMAX 引脚的连接方式。

若 VDD 电压大于 VLCD 引脚上的电压，则 VMAX 应连接到 VDD；否则，VMAX 引脚应连接到 VLCD 引脚。注意若使用 R 型偏压则无需连接外部电容或电阻。



R Type 1/3 Bias

R 型偏压电平

条件	VMAX 连接
$V_{DD} > V_{LCD}$	VMAX 连接至 VDD
其它	VMAX 连接至 VLCD

R 型偏压 VMAX 引脚连接方式

C 型偏压

对于 C 型偏压，通过 LCDC1 寄存器的 LCDP1~LCDP0 位来选择外部引脚或内部电源。LCD 电压加在引脚 VLCD、V1、V2 或内部电源上以产生内部偏置电压。

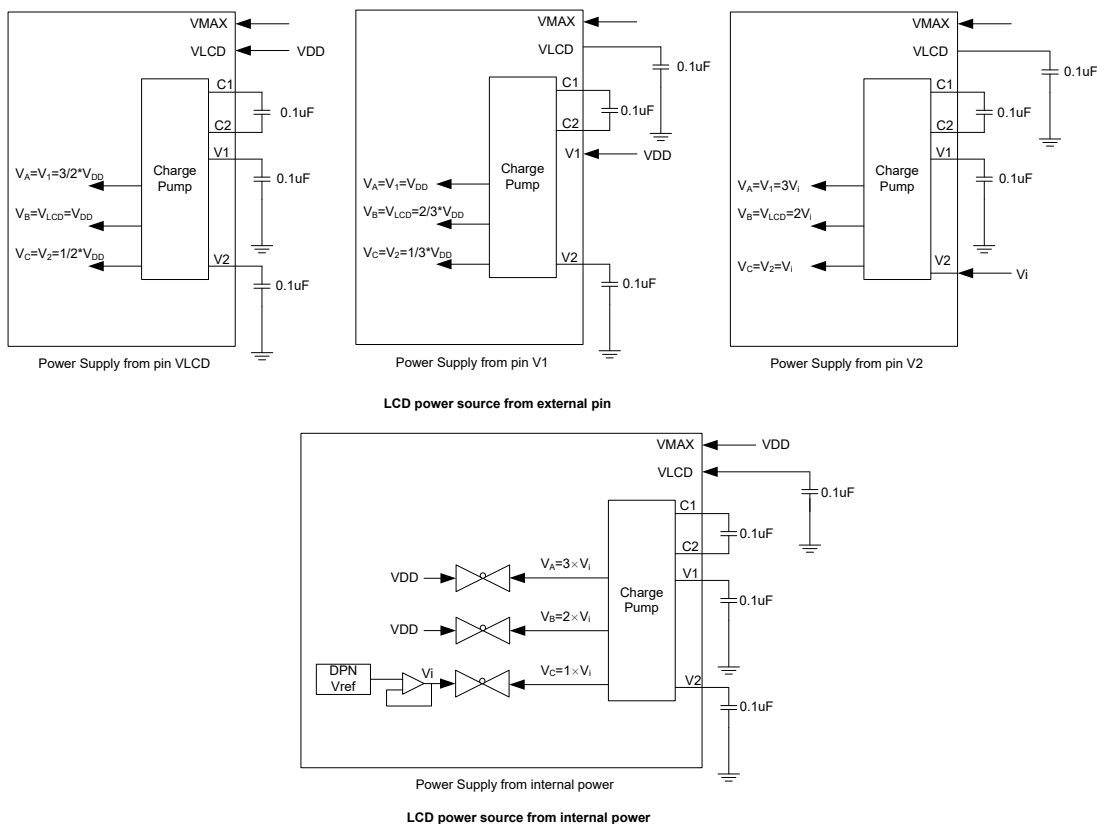
当电压源来自 VLCD 引脚，C 型偏压利用一个选择 1/3 偏压的内部充电泵电路可以产生大于 VLCD 引脚的电压。这一特性在单片机供电电压小于 LCD 所需供电电压的应用中非常有用。为产生所需电压值，应在 C1 和 C2 引脚之间连接充电泵电容。

该系列单片机为 LCD 电压源提供了一个内建损耗电路。当 LCDP 位被配置为 01、10 或 11 时，可以由 DPN Vref (~1.04V) 或内部电源 (VDD) 产生偏置电压。

对于 C 型 1/3 偏压选项，要用到 VSS、VA、VB 和 VC 四种电压值。当电源来自引脚 V1 或 VDD，即得最大偏压。引脚 VLCD 必须连接一个 0.1μf 电容到地。且充电泵将产生 VB 和 VC，VB 等于 VA×2/3，VC 等于 VA×1/3。

当电源来自引脚 VLCD 或 VDD，内部将产生 VA，其值等于 VLCD×1.5。VB 等于 VA×2/3 而 VC 等于 VA×1/3。

当电源来自引脚 V2 或 DPN Vref，内部将产生 VA，其值等于 VC×3。VB 等于 VC×2 而 VC 等于 V2 或 DPN Vref。



注：引脚 VMAX 必须连接到最大电压以使引脚防漏。

C 型偏压电压值

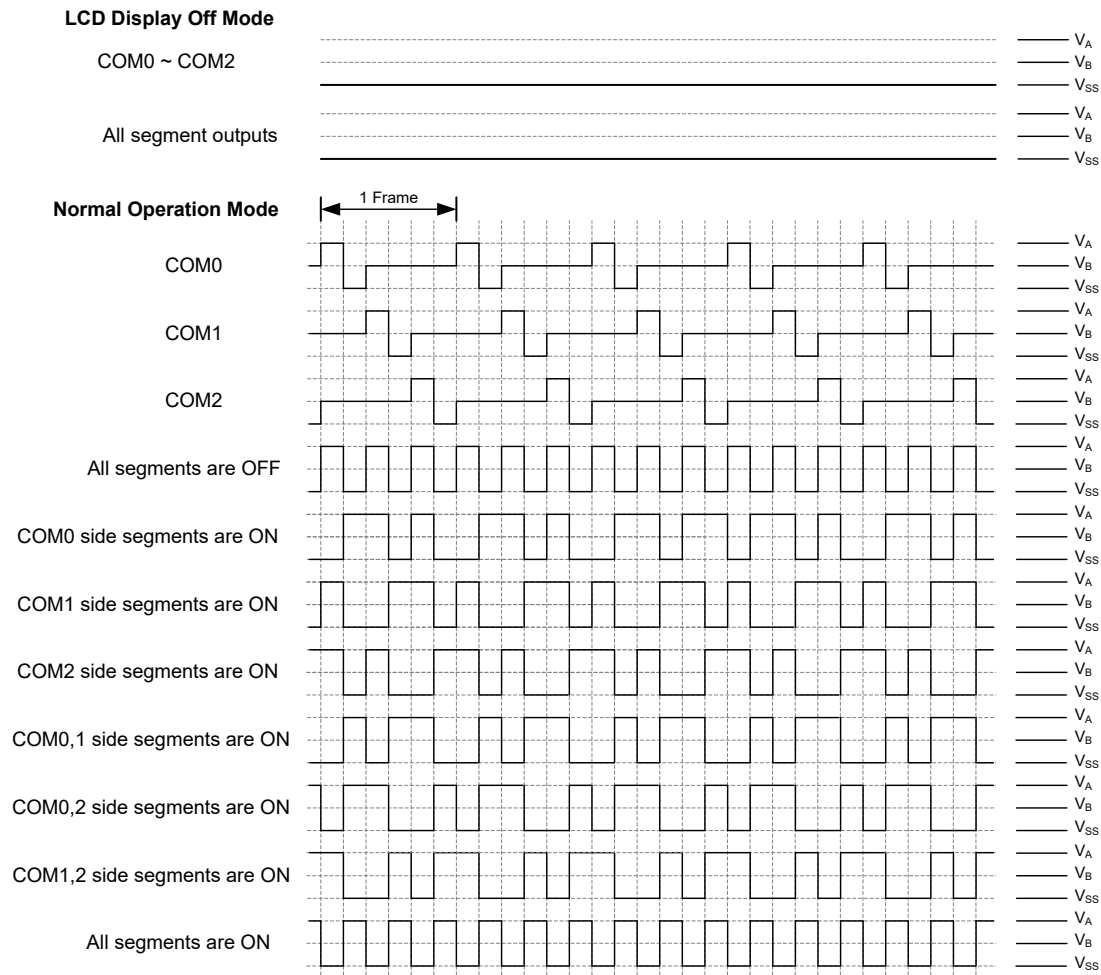
VMAX 引脚的连接方式取决于偏压和 VLCD 引脚电压，详情如下表所示。一定要确保这些充电泵所产生的内部电压不能超过 5.5V 的最大 VDD 电压。

偏压类型		VMAX 连接
1/3 偏压	$V_{DD} > V_{LCD} \times 1.5$	VMAX 连接至 VDD
	其它	VMAX 连接至 V1
1/2 偏压	$V_{DD} > V_{LCD}$	VMAX 连接至 VDD
	其它	VMAX 连接至 VLCD

C 型偏压 VMAX 引脚连接方式

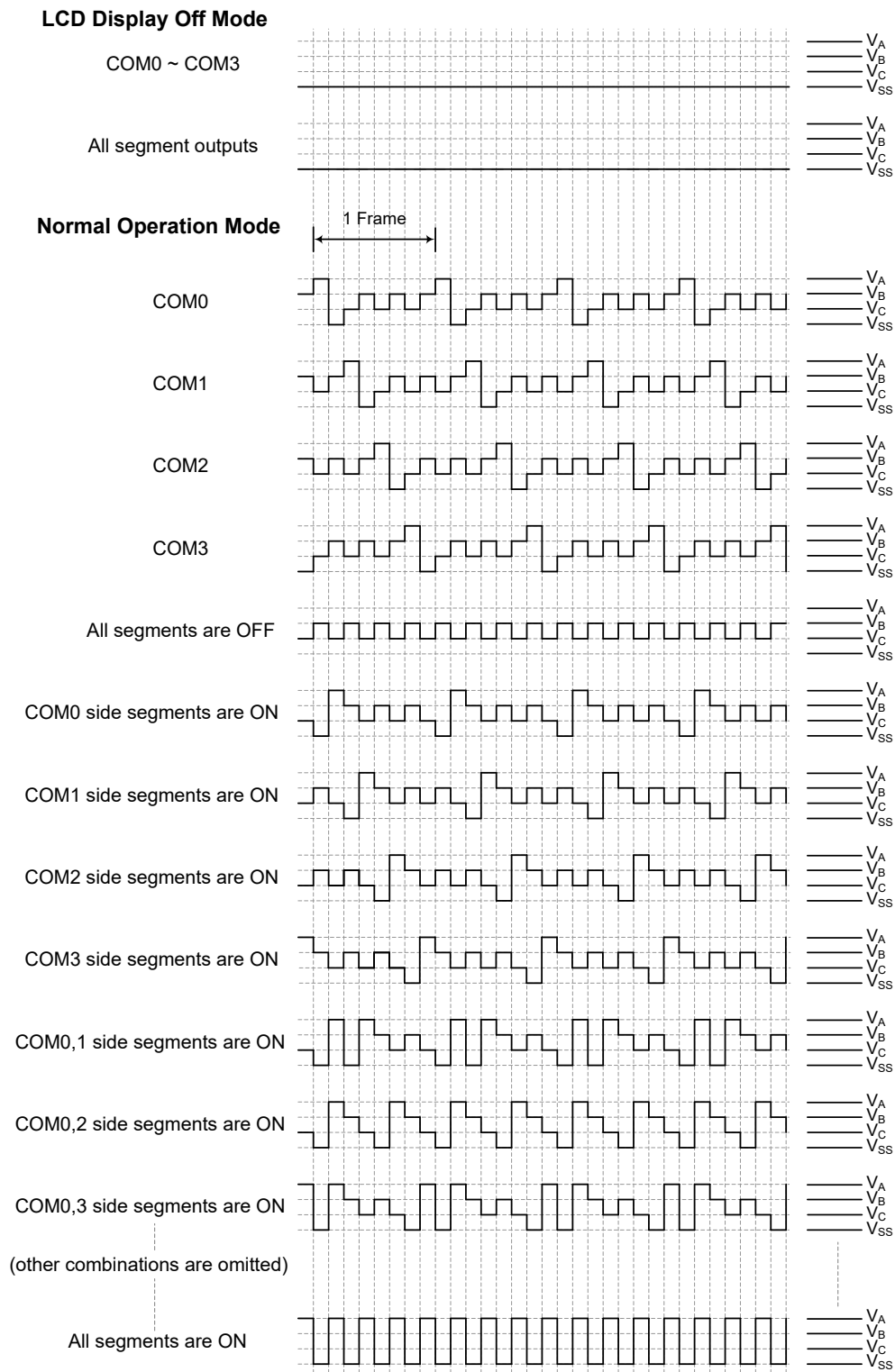
LCD 波形时序图

下面的时序图描绘了在各种占空比和偏压下单片机所产生的显示驱动信号。
LCD 具有大量不同的排列，这里只显示几种类型。



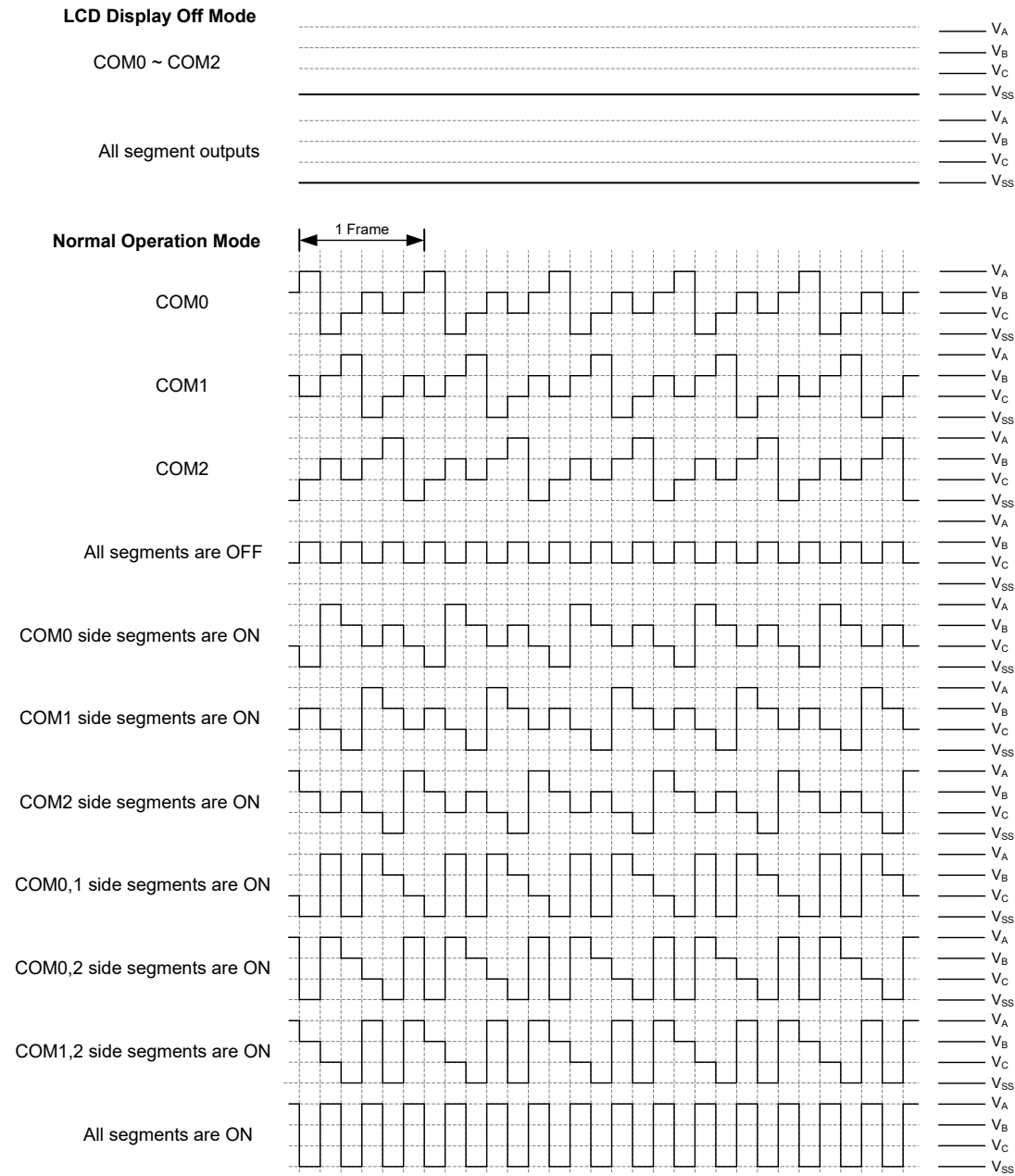
LCD 驱动输出 – A 型, 1/3Duty, 1/2Bias

注：对于 1/2 偏压，R 型和 C 型的 $V_A=V_{LCD}$ ， $V_B=V_{LCD} \times 1/2$



LCD 驱动输出 - A 型, 1/4Duty, 1/3Bias

注: 对于 1/3 R 型偏压, $V_A = V_{LCD}$, $V_B = V_{LCD} \times 2/3$, $V_C = V_{LCD} \times 1/3$
对于 1/3 C 型偏压, $V_A = V_{LCD} \times 1.5$, $V_B = V_{LCD}$, $V_C = V_{LCD} \times 1/2$

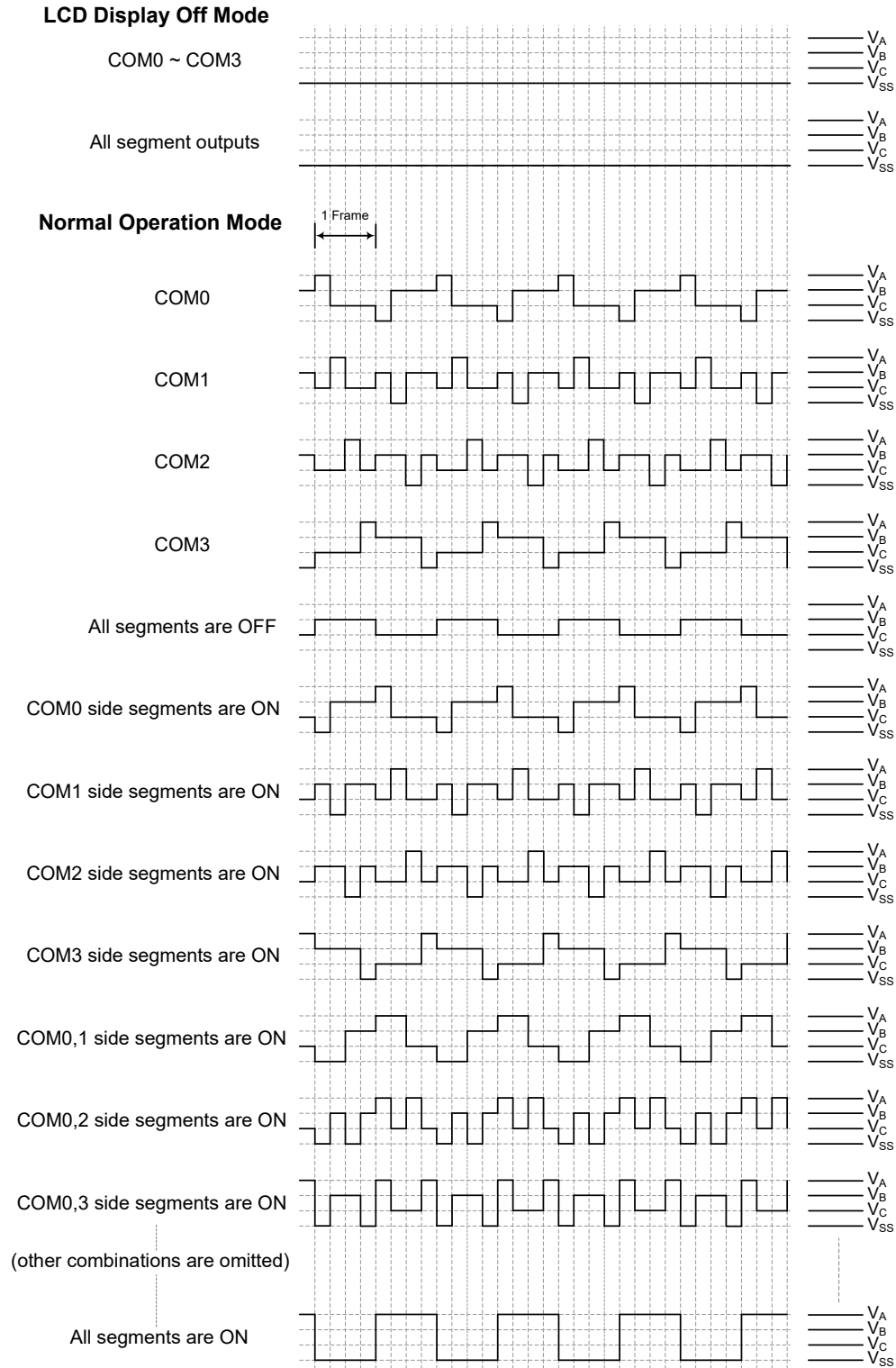


LCD 驱动输出 - A 型 - 1/3Duty, 1/3Bias

注：对于 1/3 R 型偏压， $V_A = V_{LCD}$ ， $V_B = V_{LCD} \times 2/3$ ， $V_C = V_{LCD} \times 1/3$
对于 1/3 C 型偏压， $V_A = V_{LCD} \times 1.5$ ， $V_B = V_{LCD}$ ， $V_C = V_{LCD} \times 1/2$



注: 对于 1/3 R 型偏压, $V_A=V_{LCD}$, $V_B=V_{LCD} \times 2/3$, $V_C=V_{LCD} \times 1/3$
 对于 1/3 C 型偏压, $V_A=V_{LCD} \times 1.5$, $V_B=V_{LCD}$, $V_C=V_{LCD} \times 1/2$



LCD 驱动输出 – B 型 – 1/4Duty, 1/3Bias

注：对于 1/3 R 型偏压， $V_A=V_{LCD}$ ， $V_B=V_{LCD} \times 2/3$ ， $V_C=V_{LCD} \times 1/3$
对于 1/3 C 型偏压， $V_A=V_{LCD} \times 1.5$ ， $V_B=V_{LCD}$ ， $V_C=V_{LCD} \times 1/2$

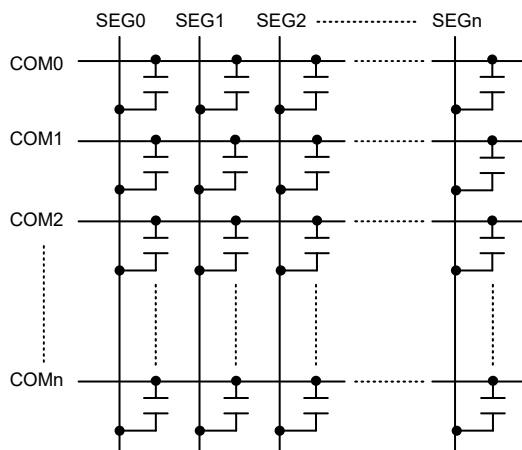
编程注意事项

LCD 编程时要注意几点，其中之一就是在单片机上电后，要保证 LCD 存储器正确地初始化。与通用数据存储器一样，在上电后，LCD 存储器的内容是未知的。由于 LCD 存储器的内容会映射到实际的 LCD，所以上电后，为获得正确的显示图形，初始化此存储器内容是非常重要的。

在实际应用中，必须要考虑 LCD 的实际容性负载。对于单片机来说，LCD 的像素点一般可以看作电容性的负载，要确保所连接的像素点不能过多。这点对于可以连接多个 LCD 像素点的 COM 口来说尤为重要。下图描绘了 LCD 的等效电路。

另外还有一个要注意的就是当单片机进入空闲模式或低速模式后所发生的变化。LCDC0 控制寄存器中的 LCDEN 控制位允许显示关闭以降低功耗。当此位被清零，就会停止产生显示的驱动信号，并处于一种低功耗的空白显示的状态。

要注意当上电复位后，LCDEN 位会被清零，显示功能关闭。



LCD 面板等效电路

配置选项

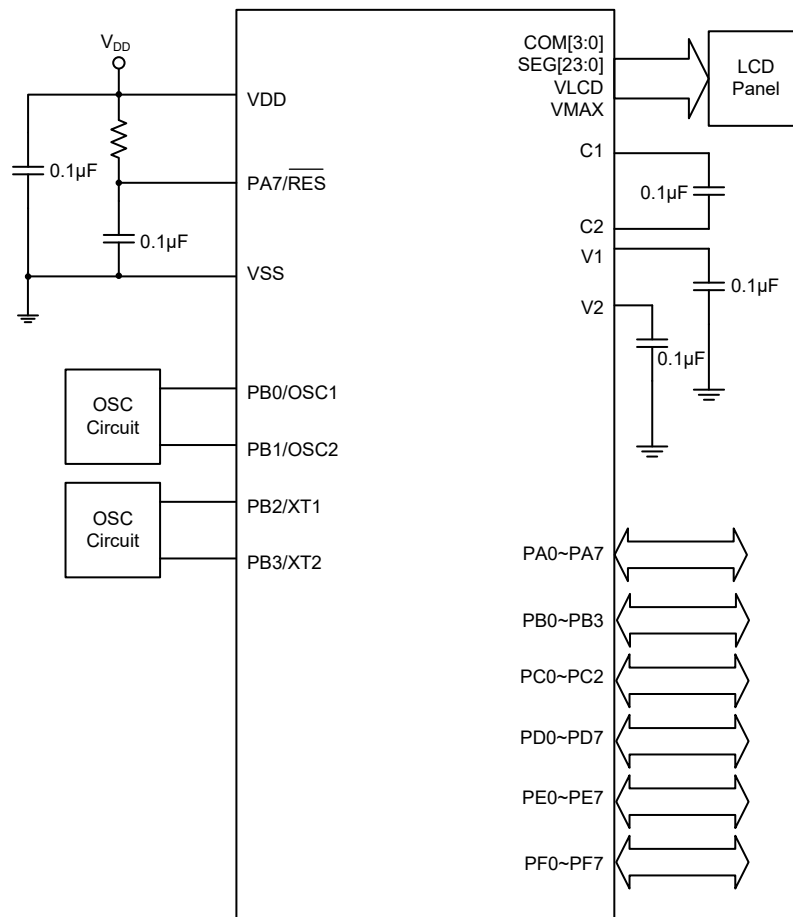
配置选项在烧写程序时写入芯片。通过 HT-IDE 的软件开发环境，使用者在开发过程中可以选择配置选项。当配置选项烧入单片机后，无法再通过应用程序修改。所有位必须按系统的需要定义，具体内容可参考下表：

序号	选项
1	PA7 或 $\overline{\text{RES}}$ 引脚选择 1. $\overline{\text{RES}}$ 引脚 2. PA7
振荡器选项	
2	HIRC 频率选择： 4MHz、8MHz 和 12MHz

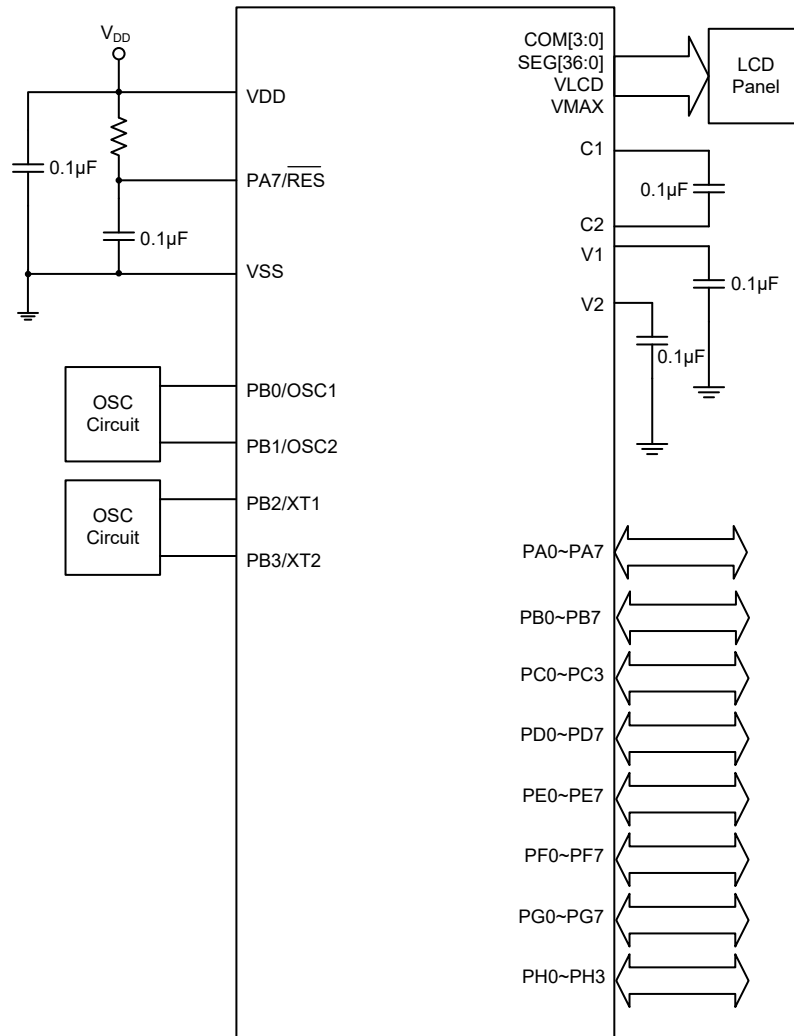
注：当 HIRC 配置选项已选定上表中的一个频率，HIRC1 和 HIRC0 位选择的频率应与其保持一致，以确保能够达到交流电气特性中标示的 HIRC 频率精度。

应用电路

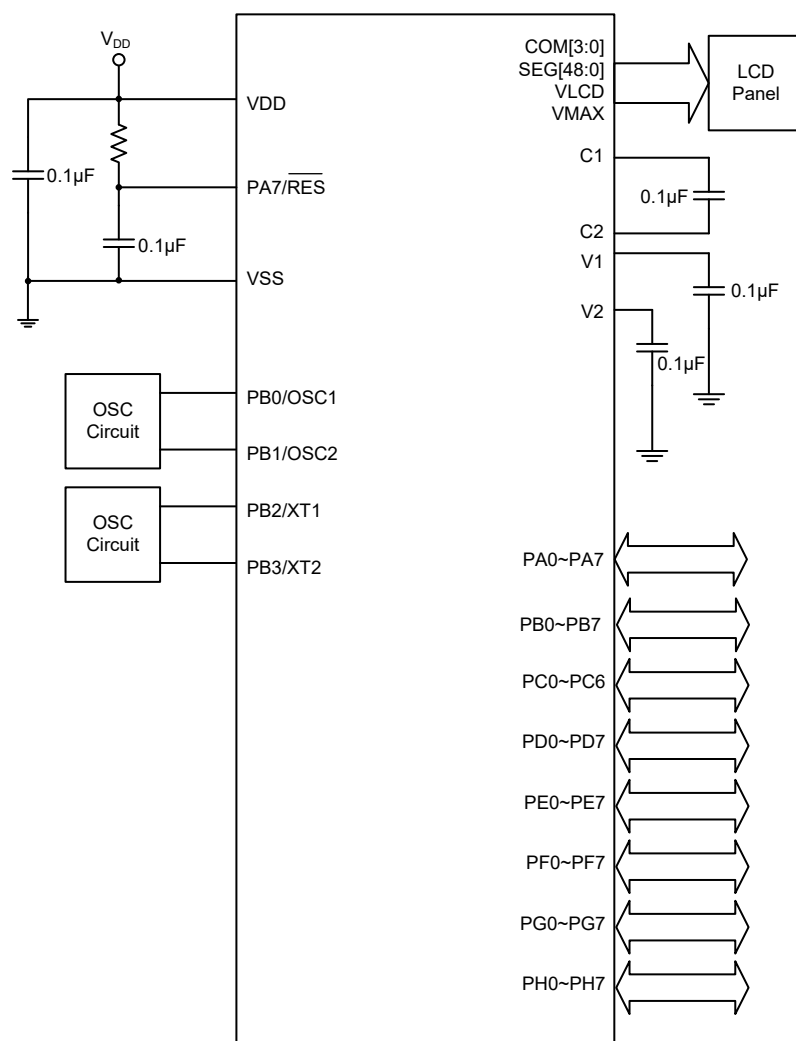
HT69F340



HT69F350



HT69F360/HT67F370



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它们的应用。

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 0.5 μ s 中执行完成，而分支或调用操作则将在 1 μ s 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 Holtek 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 Holtek 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 Holtek 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入口的 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，Holtek 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

当要操作的数据存储器位于数据存储器 Sector 0 时，下表说明了与数据存储器存取有关的指令。

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, SC
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, SC
ADD A, x	ACC 与立即数相加，结果放入 ACC	1	Z, C, AC, OV, SC
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, SC
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, SC
SUB A, x	ACC 与立即数相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
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, SC, CZ
SBC A, x	ACC 与立即数、进位标志相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
SBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z, C, AC, OV, SC, CZ
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, SC, CZ
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

助记符	说明	指令周期	影响标志位
移位			
RRA [m]	数据存储器右移一位，结果放入 ACC	1	无
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
SNZ [m]	如果数据存储器不为零，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 and TBLH	2 注	无
TABRDL [m]	读取最后页的 ROM 内容，并送至数据存储器 and TBLH	2 注	无
ITABRD [m]	读表指针 TBLP 自加，读取特定页的 ROM 内容，并送至数据存储器 and TBLH	2 注	无
ITABRDL [m]	读表指针 TBLP 自加，读取最后页的 ROM 内容，并送至数据存储器 and 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

助记符	说明	指令周期	影响标志位
CLR WDT	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。

扩展指令集

扩展指令用来提供更大范围的数据存储器寻址。当被存取的数据存储器位于 Sector 0 之外的任何数据存储器 Sector，扩展指令可直接存取数据存储器而无需使用间接寻址，此举不仅可节省 Flash 存储器空间的使用，同时可提高 CPU 执行效率。

助记符	说明	指令周期	影响标志位
算术运算			
LADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	2	Z, C, AC, OV, SC
LADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	2 ^注	Z, C, AC, OV, SC
LADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	2	Z, C, AC, OV, SC
LADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	2 ^注	Z, C, AC, OV, SC
LSUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	2	Z, C, AC, OV, SC, CZ
LSUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	2 ^注	Z, C, AC, OV, SC, CZ
LSBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	2	Z, C, AC, OV, SC, CZ
LSBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	2 ^注	Z, C, AC, OV, SC, CZ
LDAA [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	2 ^注	C
逻辑运算			
LAND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	2	Z
LOR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	2	Z
LXOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	2	Z
LANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	2 ^注	Z
LORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	2 ^注	Z
LXORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	2 ^注	Z
LCPL [m]	对数据存储器取反，结果放入数据存储器	2 ^注	Z
LCPLA [m]	对数据存储器取反，结果放入 ACC	2	Z
递增和递减			
LINCA [m]	递增数据存储器，结果放入 ACC	2	Z
LINC [m]	递增数据存储器，结果放入数据存储器	2 ^注	Z
LDECA [m]	递减数据存储器，结果放入 ACC	2	Z
LDEC [m]	递减数据存储器，结果放入数据存储器	2 ^注	Z
移位			
LRRA [m]	数据存储器右移一位，结果放入 ACC	2	无
LRR [m]	数据存储器右移一位，结果放入数据存储器	2 ^注	无
LRRCA [m]	带进位将数据存储器右移一位，结果放入 ACC	2	C
LRRC [m]	带进位将数据存储器右移一位，结果放入数据存储器	2 ^注	C
LRLA [m]	数据存储器左移一位，结果放入 ACC	2	无
LRL [m]	数据存储器左移一位，结果放入数据存储器	2 ^注	无
LRLCA [m]	带进位将数据存储器左移一位，结果放入 ACC	2	C
LRLC [m]	带进位将数据存储器左移一位，结果放入数据存储器	2 ^注	C
数据传送			
LMOV A,[m]	将数据存储器送至 ACC	2	无
LMOV [m],A	将 ACC 送至数据存储器	2 ^注	无

助记符	说明	指令周期	影响标志位
位运算			
LCLR [m].i	清除数据存储器的位	2 注	无
LSET [m].i	置位数据存储器的位	2 注	无
转移			
LSZ [m]	如果数据存储器为零，则跳过下一条指令	2 注	无
LSZA [m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	2 注	无
LSNZ [m]	如果数据存储器不为零，则跳过下一条指令	2 注	无
LSZ [m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	2 注	无
LSNZ [m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	2 注	无
LSIZ [m]	递增数据存储器，如果结果为零，则跳过下一条指令	2 注	无
LSDZ [m]	递减数据存储器，如果结果为零，则跳过下一条指令	2 注	无
LSIZA [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	2 注	无
LSDZA [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	2 注	无
查表			
LTABRD [m]	读取特定页的 ROM 内容，并送至数据存储器 and TBLH	3 注	无
LTABRDL [m]	读取最后页的 ROM 内容，并送至数据存储器 and TBLH	3 注	无
LITABRD [m]	读表指针 TBLP 自加，读取特定页的 ROM 内容，并送至数据存储器 and TBLH	3 注	无
LITABRDL [m]	读表指针 TBLP 自加，读取最后页的 ROM 内容，并送至数据存储器 and TBLH	3 注	无
其它指令			
LCLR [m]	清除数据存储器	2 注	无
LSET [m]	置位数据存储器	2 注	无
LSWAP [m]	交换数据存储器的高低字节，结果放入数据存储器	2 注	无
LSWAPA [m]	交换数据存储器的高低字节，结果放入 ACC	2	无

注：1. 对扩展跳转指令而言，如果比较的结果牵涉到跳转即需 3 个周期，如果没有发生跳转，则只需两个周期。
2. 任何扩展指令若要改变 PCL 的内容将需要 3 个周期来执行。

指令定义

ADC A, [m]	Add Data Memory to ACC with Carry
指令说明	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。
功能表示	$ACC \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C、SC
ADCM A, [m]	Add ACC to Data Memory with Carry
指令说明	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。
功能表示	$[m] \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C、SC
ADD A, [m]	Add Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C、SC
ADD A, x	Add immediate data to ACC
指令说明	将累加器和立即数相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + x$
影响标志位	OV、Z、AC、C、SC
ADDM A, [m]	Add ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C、SC
AND A, [m]	Logical AND Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC \text{ "AND" } [m]$
影响标志位	Z

AND A, x	Logical AND immediate data to ACC
指令说明	将累加器中的数据和立即数做逻辑与，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "AND" } x$
影响标志位	Z
ANDM A, [m]	Logical AND ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC \text{ "AND" } [m]$
影响标志位	Z
CALL addr	Subroutine call
指令说明	无条件地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继续执行程序，由于此指令需要额外的运算，所以为一个 2 周期的指令。
功能表示	$Stack \leftarrow Program\ Counter + 1$ $Program\ Counter \leftarrow addr$
影响标志位	无
CLR [m]	Clear Data Memory
指令说明	将指定数据存储器的内容清零。
功能表示	$[m] \leftarrow 00H$
影响标志位	无
CLR [m].i	Clear bit of Data Memory
指令说明	将指定数据存储器的第 i 位内容清零。
功能表示	$[m].i \leftarrow 0$
影响标志位	无
CLR WDT	Clear Watchdog Timer
指令说明	WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。
功能表示	WDT cleared $TO \ \& \ PDF \leftarrow 0$
影响标志位	TO、PDF

CPL [m]	Complement Data Memory
指令说明	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 0 变 1。
功能表示	$[m] \leftarrow \overline{[m]}$
影响标志位	Z
CPLA [m]	Complement Data Memory with result in ACC
指令说明	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 0 变 1，而结果被储存回累加器且数据存储器中的内容不变。
功能表示	$ACC \leftarrow \overline{[m]}$
影响标志位	Z
DAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC + 00H$ 或 $[m] \leftarrow ACC + 06H$ 或 $[m] \leftarrow ACC + 60H$ 或 $[m] \leftarrow ACC + 66H$
影响标志位	C
DEC [m]	Decrement Data Memory
指令说明	将指定数据存储器内容减 1。
功能表示	$[m] \leftarrow [m] - 1$
影响标志位	Z
DECA [m]	Decrement Data Memory with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] - 1$
影响标志位	Z

HALT	Enter power down mode
指令说明	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。
功能表示	$TO \leftarrow 0$ $PDF \leftarrow 1$
影响标志位	TO、PDF
INC [m]	Increment Data Memory
指令说明	将指定数据存储器的内容加 1。
功能表示	$[m] \leftarrow [m] + 1$
影响标志位	Z
INCA [m]	Increment Data Memory with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] + 1$
影响标志位	Z
JMP addr	Jump unconditionally
指令说明	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。
功能表示	Program Counter $\leftarrow$ addr
影响标志位	无
MOV A, [m]	Move Data Memory to ACC
指令说明	将指定数据存储器的内容复制到累加器。
功能表示	$ACC \leftarrow [m]$
影响标志位	无
MOV A, x	Move immediate data to ACC
指令说明	将 8 位立即数载入累加器。
功能表示	$ACC \leftarrow x$
影响标志位	无

MOV [m], A	Move ACC to Data Memory
指令说明	将累加器的内容复制到指定的数据存储器。
功能表示	$[m] \leftarrow \text{ACC}$
影响标志位	无
NOP	No operation
指令说明	空操作，接下来顺序执行下一条指令。
功能表示	无操作
影响标志位	无
ORA, [m]	Logical OR Data Memory to ACC
指令说明	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。
功能表示	$\text{ACC} \leftarrow \text{ACC} \text{ "OR" } [m]$
影响标志位	Z
ORA, x	Logical OR immediate data to ACC
指令说明	将累加器中的数据和立即数逻辑或，结果存放到累加器。
功能表示	$\text{ACC} \leftarrow \text{ACC} \text{ "OR" } x$
影响标志位	Z
ORM A, [m]	Logical OR ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow \text{ACC} \text{ "OR" } [m]$
影响标志位	Z
RET	Return from subroutine
指令说明	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。
功能表示	$\text{Program Counter} \leftarrow \text{Stack}$
影响标志位	无
RET A, x	Return from subroutine and load immediate data to ACC
指令说明	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。
功能表示	$\text{Program Counter} \leftarrow \text{Stack}$ $\text{ACC} \leftarrow x$
影响标志位	无

RETI	Return from interrupt
指令说明	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响应，则这个中断将在返回主程序之前被响应。
功能表示	Program Counter $\leftarrow$ Stack
影响标志位	EMI $\leftarrow$ 1 无
RL [m]	Rotate Data Memory left
指令说明	将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
功能表示	[m].(i+1) $\leftarrow$ [m].i (i=0~6) [m].0 $\leftarrow$ [m].7
影响标志位	无
RLA [m]	Rotate Data Memory left with result in ACC
指令说明	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。
功能表示	ACC.(i+1) $\leftarrow$ [m].i (i=0~6) ACC.0 $\leftarrow$ [m].7
影响标志位	无
RLC [m]	Rotate Data Memory Left through Carry
指令说明	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。
功能表示	[m].(i+1) $\leftarrow$ [m].i (i=0~6) [m].0 $\leftarrow$ C C $\leftarrow$ [m].7
影响标志位	C
RLC A [m]	Rotate Data Memory left through Carry with result in ACC
指令说明	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。
功能表示	ACC.(i+1) $\leftarrow$ [m].i (i=0~6) ACC.0 $\leftarrow$ C C $\leftarrow$ [m].7
影响标志位	C

RR [m]	Rotate Data Memory right
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow [m].0$
影响标志位	无
RRA [m]	Rotate Data Memory right with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow [m].0$
影响标志位	无
RRC [m]	Rotate Data Memory right through Carry
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
RRCA [m]	Rotate Data Memory right through Carry with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
SBC A, [m]	Subtract Data Memory from ACC with Carry
指令说明	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。
功能表示	$ACC \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ

SBC A, x	Subtract immediate data from ACC with Carry
指令说明	将累加器减去立即数以及进位标志的反，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。
功能表示	$ACC \leftarrow ACC - [x] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ
SBCM A, [m]	Subtract Data Memory from ACC with Carry and result in Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ
SDZ [m]	Skip if Decrement Data Memory is 0
指令说明	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。
功能表示	$[m] \leftarrow [m] - 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] - 1$ ，如果 $ACC=0$ 跳过下一条指令执行
影响标志位	无
SET [m]	Set Data Memory
指令说明	将指定数据存储器的每一位设置为 1。
功能表示	$[m] \leftarrow FFH$
影响标志位	无

SET [m].i	Set bit of Data Memory
指令说明	将指定数据存储器的第 i 位置位为 1。
功能表示	$[m].i \leftarrow 1$
影响标志位	无
SIZ [m]	Skip if increment Data Memory is 0
指令说明	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。
功能表示	$[m] \leftarrow [m] + 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
SIZA [m]	Skip if increment Data Memory is zero with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] + 1$ ，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 Data Memory is not 0
指令说明	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。
功能表示	如果 $[m].i \neq 0$ ，跳过下一条指令执行
影响标志位	无
SNZ [m]	Skip if Data Memory is not 0
指令说明	指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定存储器，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。
功能表示	如果 $[m] \neq 0$ ，跳过下一条指令执行
影响标志位	无

SUB A, [m]	Subtract Data Memory from ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ
SUBM A, [m]	Subtract Data Memory from ACC with result in Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ
SUB A, x	Subtract immediate Data from ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC - x$
影响标志位	OV、Z、AC、C、SC、CZ
SWAP [m]	Swap nibbles of Data Memory
指令说明	将指定数据存储器的低 4 位和高 4 位互相交换。
功能表示	$[m].3 \sim [m].0 \leftrightarrow [m].7 \sim [m].4$
影响标志位	无
SWAPA [m]	Swap nibbles of Data Memory with result in ACC
指令说明	将指定数据存储器的低 4 位与高 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。
功能表示	$ACC.3 \sim ACC.0 \leftarrow [m].7 \sim [m].4$ $ACC.7 \sim ACC.4 \leftarrow [m].3 \sim [m].0$
影响标志位	无

SZ [m]	Skip if Data Memory is 0
指令说明	指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。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。
功能表示	如果 [m]=0，跳过下一条指令执行
影响标志位	无
SZA [m]	Skip if Data Memory is 0 with data movement to ACC
指令说明	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。
功能表示	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 Data Memory is 0
指令说明	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。
功能表示	如果 [m].i=0，跳过下一条指令执行
影响标志位	无
TABRD [m]	Read table (specific page) to TBLH and Data Memory
指令说明	将表格指针 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定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]	Read table (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

ITABRD [m]	Increment table pointer low byte first and read table (specific page) to TBLH and data memory
指令说明	自加表格指针低字节 TBLP，将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定的数据存储器且将高字节移至 TBLH。
功能表示	$[m] \leftarrow \text{程序代码 (低字节)}$ $TBLH \leftarrow \text{程序代码 (高字节)}$
影响标志位	无
ITABRDL [m]	Increment table pointer low byte first and read table (last page) to TBLH and data memory
指令说明	自加表格指针低字节 TBLP，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	$[m] \leftarrow \text{程序代码 (低字节)}$ $TBLH \leftarrow \text{程序代码 (高字节)}$
影响标志位	无
XOR A, [m]	Logical XOR Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC \text{ "XOR" } [m]$
影响标志位	Z
XORM A, [m]	Logical XOR ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC \text{ "XOR" } [m]$
影响标志位	Z
XOR A, x	Logical XOR immediate data to ACC
指令说明	将累加器的数据与立即数逻辑异或，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "XOR" } x$
影响标志位	Z

扩展指令定义

扩展指令被用来直接存取存储在任何数据存储器 Sector 中的数据。

LADC A, [m] Add Data Memory to ACC with Carry

指令说明 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。

功能表示 $ACC \leftarrow ACC + [m] + C$

影响标志位 OV、Z、AC、C、SC

LADCM A, [m] Add ACC to Data Memory with Carry

指令说明 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。

功能表示 $[m] \leftarrow ACC + [m] + C$

影响标志位 OV、Z、AC、C、SC

LADD A, [m] Add Data Memory to ACC

指令说明 将指定的数据存储器内容和累加器内容相加，
结果存放到累加器。

功能表示 $ACC \leftarrow ACC + [m]$

影响标志位 OV、Z、AC、C、SC

LADDM A, [m] Add ACC to Data Memory

指令说明 将指定的数据存储器内容和累加器内容相加，
结果存放到指定的数据存储器。

功能表示 $[m] \leftarrow ACC + [m]$

影响标志位 OV、Z、AC、C、SC

LAND A, [m] Logical AND Data Memory to ACC

指令说明 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。

功能表示 $ACC \leftarrow ACC \text{ “AND” } [m]$

影响标志位 Z

LANDM A, [m] Logical AND ACC to Data Memory

指令说明 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。

功能表示 $[m] \leftarrow ACC \text{ “AND” } [m]$

影响标志位 Z

LCLR [m]	Clear Data Memory
指令说明	将指定数据存储器的内容清零。
功能表示	$[m] \leftarrow 00H$
影响标志位	无
LCLR [m].i	Clear bit of Data Memory
指令说明	将指定数据存储器的第 i 位内容清零。
功能表示	$[m].i \leftarrow 0$
影响标志位	无
LCPL [m]	Complement Data Memory
指令说明	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 0 变 1。
功能表示	$[m] \leftarrow \overline{[m]}$
影响标志位	Z
LCPLA [m]	Complement Data Memory with result in ACC
指令说明	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 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。
功能表示	$ACC \leftarrow \overline{[m]}$
影响标志位	Z
LDAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
指令说明	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 对低四位加“6”，否则低四位保持不变；如果高四位的 值大于“9”或 C=1，那么 BCD 调整就执行对高四位加“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。
功能表示	$[m] \leftarrow ACC + 00H$ 或 $[m] \leftarrow ACC + 06H$ 或 $[m] \leftarrow ACC + 60H$ 或 $[m] \leftarrow ACC + 66H$
影响标志位	C

LDEC [m]	Decrement Data Memory
指令说明	将指定数据存储器的内容减 1。
功能表示	$[m] \leftarrow [m] - 1$
影响标志位	Z
LDECA [m]	Decrement Data Memory with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] - 1$
影响标志位	Z
LINC [m]	Increment Data Memory
指令说明	将指定数据存储器的内容加 1。
功能表示	$[m] \leftarrow [m] + 1$
影响标志位	Z
LINCA [m]	Increment Data Memory with result in ACC
指令说明	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。
功能表示	$ACC \leftarrow [m] + 1$
影响标志位	Z
LMOV A, [m]	Move Data Memory to ACC
指令说明	将指定数据存储器的内容复制到累加器中。
功能表示	$ACC \leftarrow [m]$
影响标志位	无
LMOV [m], A	Move ACC to Data Memory
指令说明	将累加器的内容复制到指定数据存储器。
功能表示	$[m] \leftarrow ACC$
影响标志位	无
LOR A, [m]	Logical OR Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z

LORM A, [m]	Logical OR ACC to Data Memory
指令说明	将存在指定数据存储器中的数据 and 累加器逻辑或，结果放到数据存储器。
功能表示	$[m] \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z
LRL [m]	Rotate Data Memory left
指令说明	将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
功能表示	$[m].(i+1) \leftarrow [m].i \ (i=0\sim6)$ $[m].0 \leftarrow [m].7$
影响标志位	无
LRLA [m]	Rotate Data Memory left with result in ACC
指令说明	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。
功能表示	$ACC.(i+1) \leftarrow [m].i \ (i=0\sim6)$ $ACC.0 \leftarrow [m].7$
影响标志位	无
LRLC [m]	Rotate Data Memory Left through Carry
指令说明	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。
功能表示	$[m].(i+1) \leftarrow [m].i \ (i=0\sim6)$ $[m].0 \leftarrow C$ $C \leftarrow [m].7$
影响标志位	C
LRLC A [m]	Rotate Data Memory left through Carry with result in ACC
指令说明	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。
功能表示	$ACC.(i+1) \leftarrow [m].i \ (i=0\sim6)$ $ACC.0 \leftarrow C$ $C \leftarrow [m].7$
影响标志位	C

LRR [m]	Rotate Data Memory right
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow [m].0$
影响标志位	无
LRRA [m]	Rotate Data Memory right with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow [m].0$
影响标志位	无
LRRC [m]	Rotate Data Memory right through Carry
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
LRRC A [m]	Rotate Data Memory right through Carry with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
LSBC A, [m]	Subtract Data Memory from ACC with Carry
指令说明	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。
功能表示	$ACC \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ

LSBCM A, [m]	Subtract Data Memory from ACC with Carry and result in Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ
LSDZ [m]	Skip if Decrement Data Memory is 0
指令说明	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 3 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。
功能表示	$[m] \leftarrow [m] - 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
LSDZA [m]	Skip if decrement Data Memory is zero with result in ACC
指令说明	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 3 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。
功能表示	$ACC \leftarrow [m] - 1$ ，如果 $ACC=0$ 跳过下一条指令执行
影响标志位	无
LSET [m]	Set Data Memory
指令说明	将指定数据存储器的每一个位置位为 1。
功能表示	$[m] \leftarrow FFH$
影响标志位	无
LSET [m].i	Set bit of Data Memory
指令说明	将指定数据存储器的第 i 位置位为 1。
功能表示	$[m].i \leftarrow 1$
影响标志位	无

LSIZ [m]	Skip if increment Data Memory is 0
指令说明	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 3 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。
功能表示	$[m] \leftarrow [m] + 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
LSIZA [m]	Skip if increment Data Memory is zero with result in ACC
指令说明	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 3 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。
功能表示	$ACC \leftarrow [m] + 1$ ，如果 $ACC=0$ 跳过下一条指令执行
影响标志位	无
LSNZ [m].i	Skip if bit i of Data Memory is not 0
指令说明	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 3 个周期的指令。如果结果为 0，则程序继续执行下一条指令。
功能表示	如果 $[m].i \neq 0$ ，跳过下一条指令执行
影响标志位	无
LSNZ [m]	Skip if Data Memory is not 0
指令说明	指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定数据存储器，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 3 个周期的指令。如果结果为 0，则程序继续执行下一条指令。
功能表示	如果 $[m] \neq 0$ ，跳过下一条指令执行
影响标志位	无
LSUBA, [m]	Subtract Data Memory from ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ

LSUBM A, [m]	Subtract Data Memory from ACC with result in Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ
LSWAP [m]	Swap nibbles of Data Memory
指令说明	将指定数据存储器的低 4 位和高 4 位互相交换。
功能表示	$[m].3 \sim [m].0 \leftrightarrow [m].7 \sim [m].4$
影响标志位	无
LSWAPA [m]	Swap nibbles of Data Memory with result in ACC
指令说明	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。
功能表示	$ACC.3 \sim ACC.0 \leftarrow [m].7 \sim [m].4$ $ACC.7 \sim ACC.4 \leftarrow [m].3 \sim [m].0$
影响标志位	无
LSZ [m]	Skip if Data Memory is 0
指令说明	指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。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 3 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。
功能表示	如果 $[m]=0$ ，跳过下一条指令执行
影响标志位	无
LSZA [m]	Skip if Data Memory is 0 with data movement to ACC
指令说明	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 3 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。
功能表示	$ACC \leftarrow [m]$ ，如果 $[m]=0$ ，跳过下一条指令执行
影响标志位	无

LSZ [m].i	Skip if bit i of Data Memory is 0
指令说明	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 3 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。
功能表示	如果 [m].i=0，跳过下一条指令执行
影响标志位	无
LTABRD [m]	Move the ROM code (specific page) to TBLH and data memory
指令说明	将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定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
LTABRDL [m]	Read table (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
LITABRD [m]	Increment table pointer low byte first and read table (specific page) to TBLH and data memory
指令说明	自加表格指针低字节 TBLP，将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 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
LITABRDL [m]	Increment table pointer low byte first and read table (last page) to TBLH and data memory
指令说明	自加表格指针低字节 TBLP，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

LXOR A, [m]

指令说明

功能表示

影响标志位

Logical XOR Data Memory to ACC

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。 $ACC \leftarrow ACC \text{ "XOR" } [m]$

Z

LXORM A, [m]

指令说明

功能表示

影响标志位

Logical XOR ACC to Data Memory

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。 $[m] \leftarrow ACC \text{ "XOR" } [m]$

Z

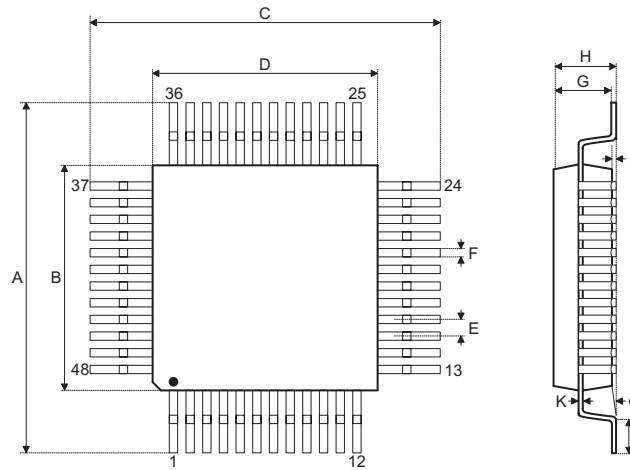
封装信息

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

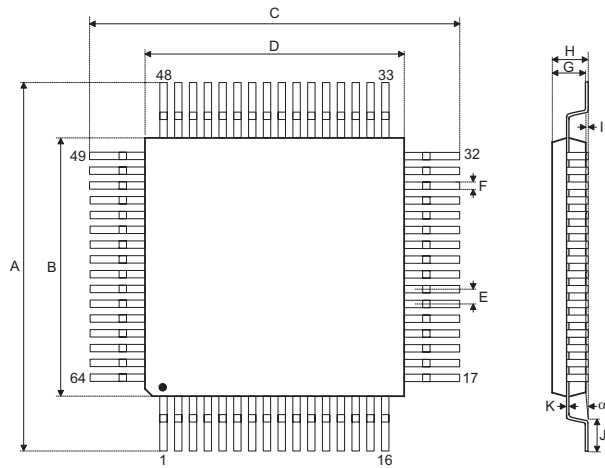
48-pin LQFP (7mm × 7mm) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
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)		
	最小值	典型值	最大值
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°

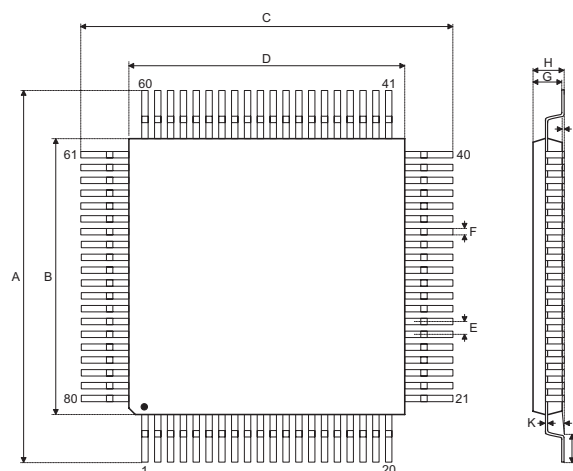
64-pin LQFP (7mm × 7mm) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	0.354 BSC		
B	0.276 BSC		
C	0.354 BSC		
D	0.276 BSC		
E	0.016 BSC		
F	0.005	0.007	0.009
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)		
	最小值	典型值	最大值
A	9.00 BSC		
B	7.00 BSC		
C	9.00 BSC		
D	7.00 BSC		
E	0.40 BSC		
F	0.13	0.18	0.23
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°

80-pin LQFP (10mm × 10mm) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	0.472 BSC		
B	0.394 BSC		
C	0.472 BSC		
D	0.394 BSC		
E	0.016 BSC		
F	0.005	0.007	0.009
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)		
	最小值	典型值	最大值
A	12.00 BSC		
B	10.00 BSC		
C	12.00 BSC		
D	10.00 BSC		
E	0.40 BSC		
F	0.13	0.18	0.23
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© 2025 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。