



内置 SPI 及 USB 接口的 A/D 型 Flash 单片机

HT66FB540/HT66FB542
HT66FB550/HT66FB560

版本: V1.90 日期: 2024-01-22

www.holtek.com

目录

特性	6
CPU 特性	6
周边特性	6
概述	7
选型表	8
方框图	8
引脚图	9
引脚说明	11
极限参数	24
直流电气特性	25
交流电气特性	27
LVD & LVR 电气特性	28
ADC 特性	29
比较器电气特性	29
上电复位 (AC+DC) 电气特性	30
系统结构	30
时序和流水线结构	30
程序计数器	31
堆栈	32
算术逻辑单元 – ALU	33
Flash 程序存储器	33
结构	33
特殊向量	33
查表	34
查表范例	34
在系统烧录 – ISP	35
在应用烧录 – IAP	41
在线烧录 – ICP	45
片上调试	45
数据存储器	46
结构	46
特殊功能寄存器	51
间接寻址寄存器 – IAR0, IAR1	51
存储器指针 – MP0, MP1	51
存储区指针 – BP	51
累加器 – ACC	53
程序计数器低字节寄存器 – PCL	53
表格寄存器 – TBLP, TBHP, TBLH	53
状态寄存器 – STATUS	53

振荡器	55
系统振荡器概述	55
系统时钟配置	55
外部晶体 / 陶瓷振荡器 – HXT	56
内部 PLL 频率发生器	57
内部 RC 振荡器 – HIRC	59
外部 32.768kHz 晶体振荡器 – LXT	59
内部 32kHz 振荡器 – LIRC	60
辅助振荡器	60
工作模式和系统时钟	61
系统时钟	61
系统工作模式	62
控制寄存器	63
快速唤醒	65
工作模式切换和唤醒	65
待机电流的注意事项	68
唤醒	68
编程注意事项	69
看门狗定时器	70
看门狗定时器时钟源	70
看门狗定时器控制寄存器	70
看门狗定时器操作	71
使用 WDTC 寄存器使能 / 除能看门狗定时器	71
复位和初始化	72
复位概述	72
复位功能	72
WDTC 寄存器软件复位	75
复位初始状态	75
输入 / 输出端口	90
上拉电阻	93
I/O 口唤醒	95
PA 口唤醒极性控制寄存器	97
输入 / 输出端口控制寄存器	97
I/O 口输出电流控制寄存器	99
I/O 口输出摆率控制寄存器	101
PA 口电源控制寄存器	102
输入 / 输出引脚结构	103
编程注意事项	104
定时器模块 – TM	105
简介	105
TM 操作	105
TM 时钟源	105
TM 中断	105
TM 外部引脚	106
TM 输入 / 输出引脚控制寄存器	106

编程注意事项	111
简易型 TM – CTM.....	113
简易型 TM 操作	113
简易型 TM 寄存器介绍	113
简易型 TM 工作模式	117
标准型 TM – STM	123
标准型 TM 操作	123
标准型 TM 寄存器介绍	124
标准型 TM 工作模式	130
A/D 转换器.....	140
A/D 简介	140
A/D 转换寄存器介绍	140
A/D 操作	147
A/D 输入引脚	148
A/D 转换步骤	148
编程注意事项	149
A/D 转换功能	149
A/D 转换应用范例	150
比较器	151
比较器操作	151
比较器寄存器	151
比较器中断	154
编程注意事项	154
串行接口模块 – SIM.....	154
SPI 接口	154
I ² C 接口	162
SPIA 串行接口模块 – SPIA	171
SPIA 接口操作.....	171
SPIA 寄存器.....	172
SPIA 通信.....	174
SPIA 使能 / 除能	176
SPIA 操作.....	177
错误侦测	178
外围时钟输出	178
外围时钟操作	178
中断	179
中断寄存器	179
中断操作	186
外部中断	189
比较器中断	189
USB 中断	189
时基中断 – HT66FB540/HT66FB550/HT66FB560	189
串行接口模块中断	190
串行外设接口中断	190

多功能中断	191
A/D 转换器中断	191
LVD 中断	191
TM 中断	191
中断唤醒功能	191
编程注意事项	192
低电压检测 – LVD	192
LVD 寄存器	192
LVD 操作	193
USB 接口	194
电源	194
USB 暂停模式唤醒和远程唤醒	194
USB 操作	195
USB 接口寄存器	196
配置选项	213
应用电路	214
指令集	215
简介	215
指令周期	215
数据的传送	215
算术运算	215
逻辑和移位运算	215
分支和控制转换	216
位运算	216
查表运算	216
其它运算	216
指令集概要	217
惯例	217
指令定义	219
封装信息	230
24-pin SSOP (150mil) 外形尺寸	231
28-pin SSOP (150mil) 外形尺寸	232
48-pin LQFP (7mm×7mm) 外形尺寸	233
64-pin LQFP (7mm×7mm) 外形尺寸	234

特性

CPU 特性

- 工作电压：
 - ◆ V_{DD} (单片机)：
 - $f_{SYS}=4\text{MHz}/6\text{MHz}$: 2.2V~5.5V
 - $f_{SYS}=12\text{MHz}$: 2.7V~5.5V
 - ◆ V_{DD} (USB 模式)：
 - $f_{SYS}=6\text{MHz}/12\text{MHz}$: 3.3V~5.5V
 - $f_{SYS}=16\text{MHz}$: 4.5V~5.5V
- 在 $V_{DD}=5\text{V}$, 系统频率为 16MHz 时, 指令周期为 0.25 μs
- 提供暂停和唤醒功能, 以降低功耗
- 4 种振荡模式:
 - 外部高频晶振 – HXT
 - 外部 32.765kHz 晶振 – LXT (仅适用于 HT66FB540/HT66FB550/HT66FB560)
 - 内部高频 RC – HIRC
 - 内部 32kHz RC – LIRC
- 多种工作模式: 正常模式、低速模式、空闲模式和休眠模式
- 所有指令都可在 1 个或 2 个指令周期内完成
- 查表指令
- 61 条功能强大的指令系统
- 多达 12 层堆栈
- 位操作指令

周边特性

- Flash 程序存储器: 4K×16~16K×16
- RAM 数据存储器: 256×8~1024×8
- 兼容 USB 2.0 全速模式
- 支持多达 8 个端点 (包括 EP0)
- 除 EP0 外的所有端点都支持中断传输和批量传输
- 除 EP0 外的所有端点都可以配置 8、16、32、64 个字节的 FIFO
- EP0 支持控制传输
- EP0 有 8 字节的 FIFO
- 支持 3.3V LDO 和内部带 1.5k Ω 上拉电阻的 UDP
- 准确度 0.25% 的内部 12MHz RC 振荡器, 可用于所有 USB 模式
- 看门狗定时器功能
- 多达 45 个双向输入 / 输出口
- 2 个与 I/O 口复用的外部中断输入
- 多个定时器模块用于时间测量、捕捉输入、比较匹配输出、PWM 输出及单脉冲输出
 - ◆ 2 个 10-bit 简易型定时器模块 – CTM

- ◆ 1 个 10-bit 标准型定时器模块 – STM
- ◆ 1 个 16-bit 标准型定时器模块 – STM
- 串行接口模块 – SIM，用于 SPI 或 I²C 通信
- 单个串行外设接口 – SPIA
- 最多两个比较器功能
- 双时基功能，可提供固定时间的中断信号
- 多达 16 通道 12 位分辨精度的 A/D 转换器
- 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
- 支持在应用烧录功能 – IAP
- 支持在系统烧录功能 – ISP
- 多种封装类型

概述

HT66FB5xx 系列单片机是一款带 USB 接口的 A/D 型具有 8 位高性能精简指令集的 Flash 单片机，专门为需要 USB 接口与外部模拟信号直接相连的产品而设计。该系列单片机具有一系列功能和特性，其 Flash 存储器可多次编程的特性给用户提供了较大的方便。存储器方面，还包含了一个 RAM 数据存储器用于应用程序数据存储。

在模拟特性方面，该系列单片机包含一个多通道 12 位 A/D 转换器和双比较器功能。还带有多个使用灵活的定时器模块，可提供定时功能、脉冲产生功能及 PWM 产生功能。内建完整的 SPI、I²C 和 USB 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。

该系列单片机提供了丰富的 HXT、LXT、HIRC 和 LIRC 振荡器功能选项，且内建完整的系统振荡器，无需外围元器件。其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。

外加时基功能、I/O 使用灵活等其它特性，使此系列单片机可以广泛应用于各种产品中，例如传感器信号处理、马达驱动、工业控制、消费类产品、子系统控制等方面。

该系列单片机配合 Holtek 开发系统和编程工具可提供高效的产品开发周期。

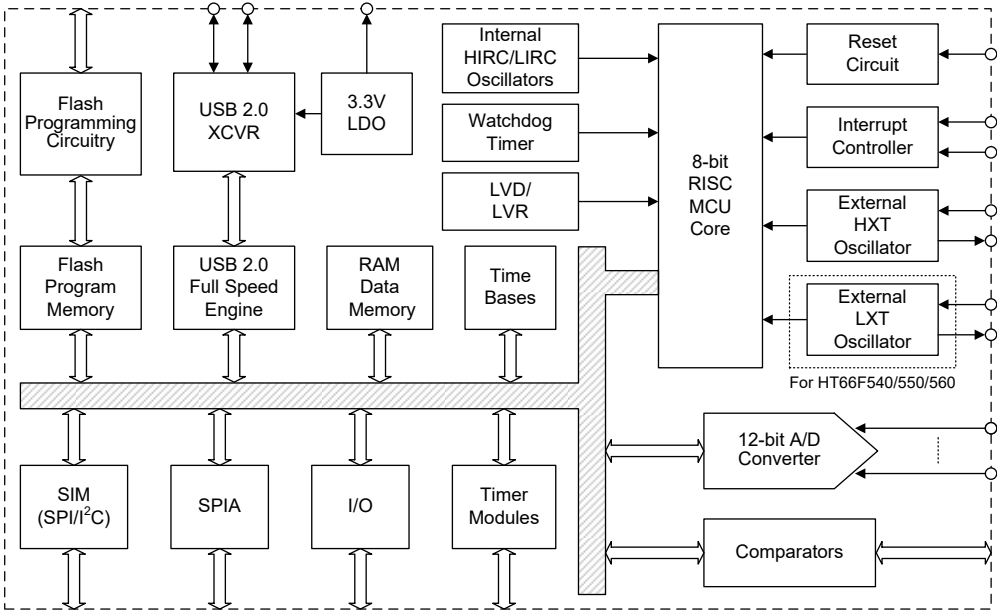
选型表

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

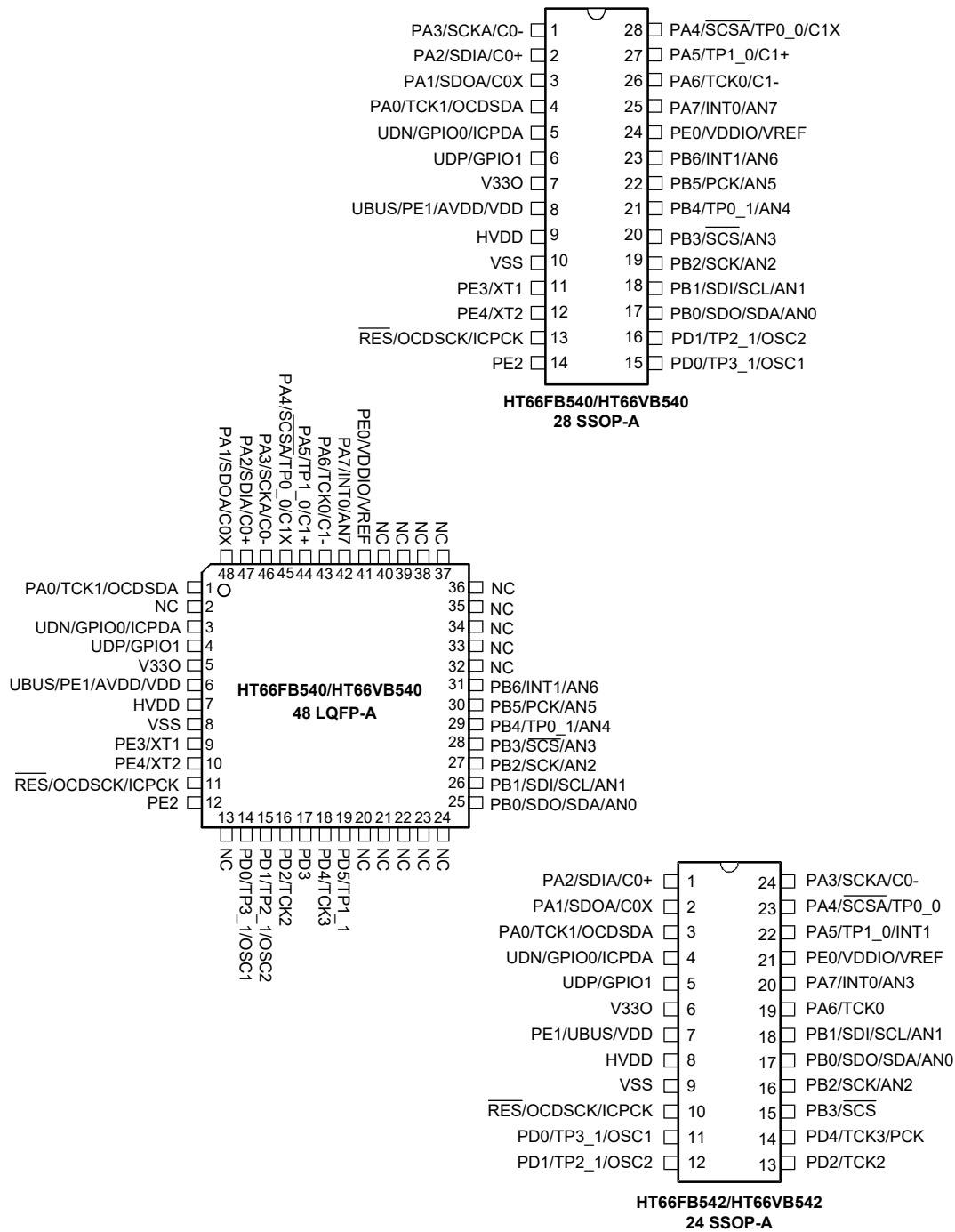
单片机型号	V _{DD}	ROM	RAM	I/O	外部中断	A/D
HT66FB540	2.2V~5.5V	4K×16	512×8	25	2	12-bit×8
HT66FB542	2.2V~5.5V	4K×16	256×8	17	2	12-bit×4
HT66FB550	2.2V~5.5V	8K×16	768×8	37	2	12-bit×16
HT66FB560	2.2V~5.5V	16K×16	1024×8	45	2	12-bit×16

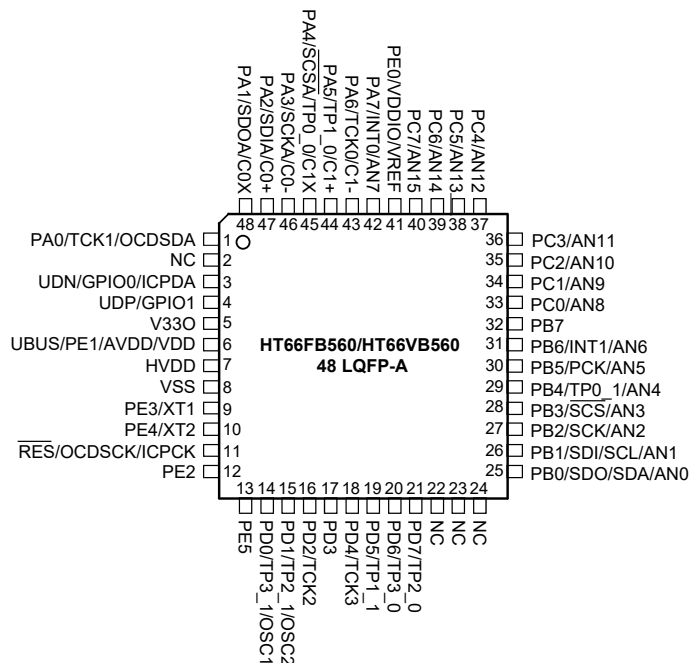
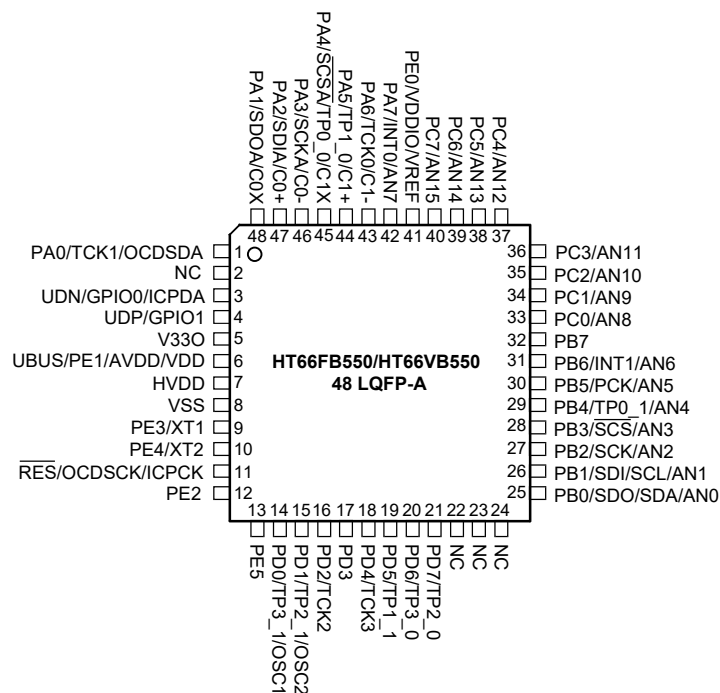
单片机型号	TM 模块	SIM (SPI/I ² C)	SPIA	LXT	比较器	堆栈	封装类型
HT66FB540	10-bit CTM×2 10-bit STM×1 16-bit STM×1	√	√	√	2	8	28SSOP 48LQFP
HT66FB542	10-bit CTM×2 10-bit STM×1 16-bit STM×1	√	√	—	1	8	24SSOP
HT66FB550	10-bit CTM×2 10-bit STM×1 16-bit STM×1	√	√	√	2	8	48LQFP
HT66FB560	10-bit CTM×2 10-bit STM×1 16-bit STM×1	√	√	√	2	12	48/64LQFP

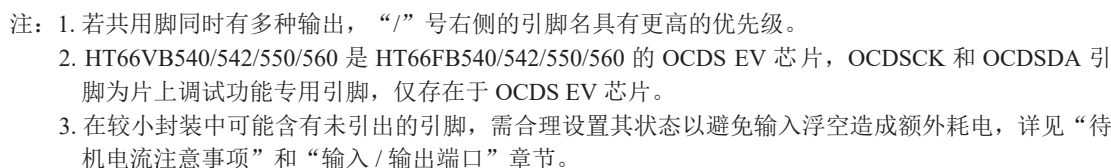
方框图



引脚图







该系列单片机的所有引脚都以它们的端口名称进行标注，例如 PA.0、PA.1 等，用于描述这些引脚的数字输入/输出功能。然而，这些引脚也与其它功能共用，如模数转换器，串行引脚等。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。

引脚名称	功能	OPT	I/T	O/T	说明
PA0/TCK1/OCSDSA	PA0	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK1	—	ST	—	TM1 输入
	OCSDSA	—	ST	CMOS	片上调试数据输入 / 输出，仅 OCDS EV 有
PA1/SDOA/C0X	PA1	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDOA	—	—	CMOS	SPIA 数据输出
	C0X	—	—	CMOS	比较器 0 输出

引脚名称	功能	OPT	I/T	O/T	说明
PA2/SDIA/C0+	PA2	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDIA	—	ST	—	SPIA 数据输入
	C0+	—	AN	—	比较器 0 正输入
PA3/SCKA/C0-	PA3	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCKA	—	ST	NMOS	SPIA 串行时钟
	C0-	—	AN	—	比较器 0 负输入
PA4/SCSA/TP0_0/ C1X	PA4	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCSA	—	ST	CMOS	SPIA 从机选择
	TP0_0	TMPC0	ST	CMOS	TM0 输入 / 输出
	C1X	—	—	CMOS	比较器 1 输出
PA5/TP1_0/C1+	PA5	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_0	TMPC0	ST	CMOS	TM1 输入 / 输出
	C1+	—	AN	—	比较器 1 正输入
PA6/TCK0/C1-	PA6	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK0	—	ST	—	TM0 输入
	C1-	—	AN	—	比较器 1 负输入
PA7/INT0/AN7	PA7	PAPU PAWU	ST	NMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	—	ST	—	外部中断 0
	AN7	ACER0	AN	—	A/D 通道 7
PB0/SDO/SDA/AN0	PB0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDO	—	—	CMOS	SPI 数据输出
	SDA	—	ST	NMOS	I ² C 数据
	AN0	ACER0	AN	—	A/D 通道 0
PB1/SDI/SCL/AN1	PB1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDI	—	ST	—	SPI 数据输入
	SCL	—	ST	NMOS	I ² C 时钟
	AN1	ACER0	AN	—	A/D 通道 1
PB2/SCK/AN2	PB2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCK	—	ST	CMOS	SPI 串行时钟
	AN2	ACER0	AN	—	A/D 通道 2
PB3/SCS/AN3	PB3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCS	—	ST	CMOS	SPI 从机选择
	AN3	ACER0	AN	—	A/D 通道 3

引脚名称	功能	OPT	I/T	O/T	说明
PB4/TP0_1/AN4	PB4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP0_1	TMPC0	ST	CMOS	TM0 输入 / 输出
	AN4	ACER0	AN	—	A/D 通道 4
PB5/PCK/AN5	PB5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PCK	—	—	CMOS	外围时钟输出
	AN5	ACER0	AN	—	A/D 通道 5
PB6/INT1/AN6	PB6	PXPU PXWU	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	—	ST	—	外部中断 1
	AN6	ACER0	AN	—	A/D 通道 6
PD0/TP3_1/OSC1	PD0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_1	TMPC1	ST	CMOS	TM3 输入 / 输出
	OSC1	CO	HXT	—	HXT 脚
PD1/TP2_1/OSC2	PD1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_1	TMPC1	ST	CMOS	TM2 输入 / 输出
	OSC2	CO	—	HXT	HXT 脚
PD2/TCK2	PD2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK2	—	ST	—	TM2 输入
PD3	PD3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PD4/TCK3	PD4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK3	TMPC1	ST	—	TM3 输入
PD5/TP1_1	PD5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_1	TMPC0	ST	CMOS	TM1 输入 / 输出
PE0/VDDIO/VREF	PE0	PXPU PXWU 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
	VDDIO	CO	PWR	—	PA 外部电源输入
	VREF	—	AN	—	ADC 参考电压输入
PE1/UBUS	PE1	—	ST	—	通用 I/O 口，仅输入
	UBUS	—	PWR	—	USB SIE VDD
PE2	PE2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PE3/XT1	PE3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT1	CO	LXT	—	LXT 脚

引脚名称	功能	OPT	I/T	O/T	说明
PE4/XT2	PE4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT2	CO	—	LXT	LXT 脚
RES/OCDSCK/ ICPCK	RES	—	ST	—	复位输入
	OCDSCK	—	ST	—	片上调试时钟输入，仅 OCDS EV 有
	ICPCK	—	ST	—	在线烧录时钟输入
UDN/GPIO0/ICPDA	UDN	—	ST	CMOS	USB UDN 口
	GPIO0	—	ST	CMOS	通用 I/O 口
	ICPDA	—	ST	CMOS	在线烧录数据 / 地址
UDP/GPIO1	UDP	—	ST	CMOS	USB UDP 口
	GPIO1	—	ST	CMOS	通用 I/O 口
VDD/AVDD	VDD/ AVDD	—	PWR	—	正电源
VSS/AVSS	VSS/ AVSS	—	PWR	—	负电源，接地
V33O	V33O	—	—	PWR	3.3V 稳压器输出
HVDD	HVDD	—	PWR	—	HIRC 正电源

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：施密特触发输入；

CMOS：CMOS 输出；

AN：模拟信号

HXT：高频晶体振荡器

LXT：低频晶体振荡器

此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

HT66FB542

引脚名称	功能	OPT	I/T	O/T	说明
PA0/TCK1/OCDSDA	PA0	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK1	—	ST	—	TM1 输入
	OCDSDA	—	ST	CMOS	片上调试数据输入 / 输出，仅 OCDS EV 有
PA1/SDOA/C0X	PA1	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDOA	—	—	CMOS	SPIA 数据输出
	C0X	—	—	CMOS	比较器 0 输出
PA2/SDIA/C0+	PA2	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDIA	—	ST	—	SPIA 数据输入
	C0+	—	AN	—	比较器 0 正输入
PA3/SCKA/C0-	PA3	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCKA	—	ST	NMOS	SPIA 串行时钟
	C0-	—	AN	—	比较器 0 负输入
PA4/ $\overline{\text{SCSA}}$ /TP0_0	PA4	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{SCSA}}$	—	ST	CMOS	SPIA 从机选择
	TP0_0	TMPC0	ST	CMOS	TM0 输入 / 输出
PA5/TP1_0/INT1	PA5	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_0	TMPC0	ST	CMOS	TM1 输入 / 输出
	INT1	—	ST	—	外部中断 1
PA6/TCK0	PA6	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK0	—	ST	—	TM0 输入
PA7/INT0/AN3	PA7	PAPU PAWU	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	—	ST	—	外部中断 0
	AN3	ACER0	AN	—	A/D 通道 3
PB0/SDO/SDA/AN0	PB0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDO	—	—	CMOS	SPI 数据输出
	SDA	—	ST	NMOS	I2C 数据
	AN0	ACER0	AN	—	A/D 通道 0
PB1/SDI/SCL/AN1	PB1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDI	—	ST	—	SPI 数据输入
	SCL	—	ST	NMOS	I2C 时钟
	AN1	ACER0	AN	—	A/D 通道 1

引脚名称	功能	OPT	I/T	O/T	说明
PB2/SCK/AN2	PB2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCK	—	ST	CMOS	SPI 串行时钟
	AN2	ACER0	AN	—	A/D 通道 2
PB3/SCS	PB3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCS	—	ST	CMOS	SPI 从机选择
PD0/TP3_1/OSC1	PD0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_1	TMPC1	ST	CMOS	TM3 输入 / 输出
	OSC1	CO	HXT	—	HXT 脚
PD1/TP2_1/OSC2	PD1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_1	TMPC1	ST	CMOS	TM2 输入 / 输出
	OSC2	CO	—	HXT	HXT 脚
PD2/TCK2	PD2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK2	—	ST	—	TM2 输入
PD4/TCK3/PCK	PD4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK3	TMPC1	ST	—	TM3 输入
	PCK	—	—	CMOS	外围时钟输出
PE0/VDDIO/VREF	PE0	PXPU PXWU 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
	VDDIO	CO	PWR	—	PA 外部电源输入
	VREF	—	AN	—	ADC 参考电压输入
PE1/UBUS/VDD	PE1	—	ST	—	通用 I/O 口，仅输入
	UBUS	—	PWR	—	USB SIE VDD
	VDD	—	PWR	—	正电源
RES/OCDSCK/ICPCK	RES	—	ST	—	复位输入
	OCDSCK	—	ST	—	片上调试时钟输入，仅 OCDS EV 有
	ICPCK	—	ST	—	在线烧录时钟输入
UDN/GPIO0/ICPDA	UDN	—	ST	CMOS	USB UDN 口
	GPIO0	—	ST	CMOS	通用 I/O 口
	ICPDA	—	ST	CMOS	在线烧录数据 / 地址
UDP/GPIO1	UDP	—	ST	CMOS	USB UDP 口
	GPIO1	—	ST	CMOS	通用 I/O 口
VSS	VSS	—	PWR	—	负电源，接地
V33O	V33O	—	—	PWR	3.3V 稳压器输出
HVDD	HVDD	—	PWR	—	HIRC 正电源

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：施密特触发输入； CMOS：CMOS 输出； AN：模拟信号
 HXT：高频晶体振荡器； LXT：低频晶体振荡器
 此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

HT66FB550

引脚名称	功能	OPT	I/T	O/T	说明
PA0/TCK1/OCDSDA	PA0	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK1	—	ST	—	TM1 输入
	OCDSDA	—	ST	CMOS	片上调试数据输入 / 输出，仅 OCDS EV 有
PA1/SDOA/C0X	PA1	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDOA	—	—	CMOS	SPIA 数据输出
	C0X	—	—	CMOS	比较器 0 输出
PA2/SDIA/C0+	PA2	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDIA	—	ST	—	SPIA 数据输入
	C0+	—	AN	—	比较器 0 正输入
PA3/SCKA/C0-	PA3	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCKA	—	ST	NMOS	SPIA 串行时钟
	C0-	—	AN	—	比较器 0 负输入
PA4/SCSA/TP0_0/ C1X	PA4	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCSA	—	ST	CMOS	SPIA 从机选择
	TP0_0	TMPC0	ST	CMOS	TM0 输入 / 输出
	C1X	—	—	CMOS	比较器 1 输出
PA5/TP1_0/C1+	PA5	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_0	TMPC0	ST	CMOS	TM1 输入 / 输出
	C1+	—	AN	—	比较器 1 正输入
PA6/TCK0/C1-	PA6	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK0	—	ST	—	TM0 输入
	C1-	—	AN	—	比较器 1 负输入
PA7/INT0/AN7	PA7	PAPU PAWU	ST	NMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	—	ST	—	外部中断 0
	AN7	ACER0	AN	—	A/D 通道 7
PB0/SDO/SDA/AN0	PB0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDO	—	—	CMOS	SPI 数据输出
	SDA	—	ST	NMOS	I ² C 数据
	AN0	ACER0	AN	—	A/D 通道 0

引脚名称	功能	OPT	I/T	O/T	说明
PB1/SDI/SCL/AN1	PB1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDI	—	ST	—	SPI 数据输入
	SCL	—	ST	NMOS	I ² C 时钟
	AN1	ACER0	AN	—	A/D 通道 1
PB2/SCK/AN2	PB2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCK	—	ST	CMOS	SPI 串行时钟
	AN2	ACER0	AN	—	A/D 通道 2
PB3/ $\overline{\text{SCS}}$ /AN3	PB3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{SCS}}$	—	ST	CMOS	SPI 从机选择
	AN3	ACER0	AN	—	A/D 通道 3
PB4/TP0_1/AN4	PB4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP0_1	TMPC0	ST	CMOS	TM0 输入 / 输出
	AN4	ACER0	AN	—	A/D 通道 4
PB5/PCK/AN5	PB5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PCK	—	—	CMOS	外围时钟输出
	AN5	ACER0	AN	—	A/D 通道 5
PB6/INT1/AN6	PB6	PXPU PXWU	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	—	ST	—	外部中断 1
	AN6	ACER0	AN	—	A/D 通道 6
PB7	PB7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PC0/AN8	PC0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN8	ACER1	AN	—	A/D 通道 8
PC1/AN9	PC1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN9	ACER1	AN	—	A/D 通道 9
PC2/AN10	PC2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN10	ACER1	AN	—	A/D 通道 10
PC3/AN11	PC3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN11	ACER1	AN	—	A/D 通道 11
PC4/AN12	PC4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN12	ACER1	AN	—	A/D 通道 12

引脚名称	功能	OPT	I/T	O/T	说明
PC5/AN13	PC5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN13	ACER1	AN	—	A/D 通道 13
PC6/AN14	PC6	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN14	ACER1	AN	—	A/D 通道 14
PC7/AN15	PC7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN15	ACER1	AN	—	A/D 通道 15
PD0/TP3_1/OSC1	PD0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_1	TMPC1	ST	CMOS	TM3 输入 / 输出
	OSC1	CO	HXT	—	HXT 脚
PD1/TP2_1/OSC2	PD1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_1	TMPC1	ST	CMOS	TM2 输入 / 输出
	OSC2	CO	—	HXT	HXT 脚
PD2/TCK2	PD2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK2	—	ST	—	TM2 输入
PD3	PD3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PD4/TCK3	PD4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK3	TMPC1	ST	—	TM3 输入
PD5/TP1_1	PD5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_1	TMPC0	ST	CMOS	TM1 输入 / 输出
PD6/TP3_0	PD6	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_0	TMPC1	ST	CMOS	TM3 输入 / 输出
PD7/TP2_0	PD7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_0	TMPC1	ST	CMOS	TM2 输入 / 输出
PE0/VDDIO/VREF	PE0	PXPU PXWU 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
	VDDIO	CO	PWR	—	PA 外部电源输入
	VREF	—	AN	—	ADC 参考电压输入
PE1/UBUS	PE1	—	ST	—	通用 I/O 口，仅输入
	UBUS	—	PWR	—	USB SIE VDD
PE2	PE2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能

引脚名称	功能	OPT	I/T	O/T	说明
PE3/XT1	PE3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT1	CO	LXT	—	LXT 脚
PE4/XT2	PE4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT2	CO	—	LXT	LXT 脚
PE5	PE5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
RES/OCDSCK/ ICPCK	RES	—	ST	—	复位输入
	OCDSCK	—	ST	—	片上调试时钟输入，仅 OCDS EV 有
	ICPCK	—	ST	—	在线烧录时钟输入
UDN/GPIO0/ICPDA	UDN	—	ST	CMOS	USB UDN 口
	GPIO0	—	ST	CMOS	通用 I/O 口
	ICPDA	—	ST	CMOS	在线烧录数据 / 地址
UDP/GPIO1	UDP	—	ST	CMOS	USB UDP 口
	GPIO1	—	ST	CMOS	通用 I/O 口
VDD/AVDD	VDD/ AVDD	—	PWR	—	正电源
VSS/AVSS	VSS/AVSS	—	PWR	—	负电源，接地
V33O	V33O	—	—	PWR	3.3V 稳压器输出
HVDD	HVDD	—	PWR	—	HIRC 正电源

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：施密特触发输入；

CMOS：CMOS 输出；

AN：模拟信号；

HXT：高频晶体振荡器；

LXT：低频晶体振荡器；

此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

HT66FB560

引脚名称	功能	OPT	I/T	O/T	说明
PA0/TCK1/OCDSDA	PA0	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK1	—	ST	—	TM1 输入
	OCDSDA	—	ST	CMOS	片上调试数据输入 / 输出，仅 OCDS EV 有
PA1/SDOA/C0X	PA1	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDOA	—	—	CMOS	SPIA 数据输出
	C0X	—	—	CMOS	比较器 0 输出
PA2/SDIA/C0+	PA2	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDIA	—	ST	—	SPIA 数据输入
	C0+	—	AN	—	比较器 0 正输入
PA3/SCKA/C0-	PA3	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCKA	—	ST	NMOS	SPIA 串行时钟
	C0-	—	AN	—	比较器 0 负输入
PA4/ $\overline{\text{SCSA}}$ /TP0_0/ C1X	PA4	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{SCSA}}$	—	ST	CMOS	SPIA 从机选择
	TP0_0	TMPC0	ST	CMOS	TM0 输入 / 输出
	C1X	—	—	CMOS	比较器 1 输出
PA5/TP1_0/C1+	PA5	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_0	TMPC0	ST	CMOS	TM1 输入 / 输出
	C1+	—	AN	—	比较器 1 正输入
PA6/TCK0/C1-	PA6	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK0	—	ST	—	TM0 输入
	C1-	—	AN	—	比较器 1 负输入
PA7/INT0/AN7	PA7	PAPU PAWU	ST	NMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	—	ST	—	外部中断 0
	AN7	ACER0	AN	—	A/D 通道 7
PB0/SDO/SDA/AN0	PB0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDO	—	—	CMOS	SPI 数据输出
	SDA	—	ST	NMOS	I ² C 数据
	AN0	ACER0	AN	—	A/D 通道 0

引脚名称	功能	OPT	I/T	O/T	说明
PB1/SDI/SCL/AN1	PB1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SDI	—	ST	—	SPI 数据输入
	SCL	—	ST	NMOS	PC 时钟
	AN1	ACER0	AN	—	A/D 通道 1
PB2/SCK/AN2	PB2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	SCK	—	ST	CMOS	SPI 串行时钟
	AN2	ACER0	AN	—	A/D 通道 2
PB3/ $\overline{\text{SCS}}$ /AN3	PB3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	$\overline{\text{SCS}}$	—	ST	CMOS	SPI 从机选择
	AN3	ACER0	AN	—	A/D 通道 3
PB4/TP0_1/AN4	PB4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP0_1	TMPC0	ST	CMOS	TM0 输入 / 输出
	AN4	ACER0	AN	—	A/D 通道 4
PB5/PCK/AN5	PB5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PCK	—	—	CMOS	外围时钟输出
	AN5	ACER0	AN	—	A/D 通道 5
PB6/INT1/AN6	PB6	PXPU PXWU	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	—	ST	—	外部中断 1
	AN6	ACER0	AN	—	A/D 通道 6
PB7	PB7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PC0/AN8	PC0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN8	ACER1	AN	—	A/D 通道 8
PC1/AN9	PC1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN9	ACER1	AN	—	A/D 通道 9
PC2/AN10	PC2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN10	ACER1	AN	—	A/D 通道 10
PC3/AN11	PC3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN11	ACER1	AN	—	A/D 通道 11
PC4/AN12	PC4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN12	ACER1	AN	—	A/D 通道 12

引脚名称	功能	OPT	I/T	O/T	说明
PC5/AN13	PC5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN13	ACER1	AN	—	A/D 通道 13
PC6/AN14	PC6	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN14	ACER1	AN	—	A/D 通道 14
PC7/AN15	PC7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	AN15	ACER1	AN	—	A/D 通道 15
PD0/TP3_1/OSC1	PD0	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_1	TMPC1	ST	CMOS	TM3 输入 / 输出
	OSC1	CO	HXT	—	HXT 脚
PD1/TP2_1/OSC2	PD1	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_1	TMPC1	ST	CMOS	TM2 输入 / 输出
	OSC2	CO	—	HXT	HXT 脚
PD2/TCK2	PD2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK2	—	ST	—	TM2 输入
PD3	PD3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PD4/TCK3	PD4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TCK3	TMPC1	ST	—	TM3 输入
PD5/TP1_1	PD5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP1_1	TMPC0	ST	CMOS	TM1 输入 / 输出
PD6/TP3_0	PD6	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP3_0	TMPC1	ST	CMOS	TM3 输入 / 输出
PD7/TP2_0	PD7	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	TP2_0	TMPC1	ST	CMOS	TM2 输入 / 输出
PE0/VDDIO/VREF	PE0	PXPU PXWU 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
	VDDIO	CO	PWR	—	PA 外部电源输入
	VREF	—	AN	—	ADC 参考电压输入
PE1/UBUS	PE1	—	ST	—	通用 I/O 口，仅输入
	UBUS	—	PWR	—	USB SIE VDD
PE2	PE2	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能

引脚名称	功能	OPT	I/T	O/T	说明
PE3/XT1	PE3	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT1	CO	LXT	—	LXT 脚
PE4/XT2	PE4	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	XT2	CO	—	LXT	LXT 脚
PE5	PE5	PXPU PXWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
PF0~PF7	PFn	PFPU PFWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
RES/OCDSCK/ ICPCK	RES	—	ST	—	复位输入
	OCDSCK	—	ST	—	片上调试时钟输入，仅 OCDS EV 有
	ICPCK	—	ST	—	在线烧录时钟输入
UDN/GPIO0/ICPDA	UDN	—	ST	CMOS	USB UDN 口
	GPIO0	—	ST	CMOS	通用 I/O 口
	ICPDA	—	ST	CMOS	在线烧录数据 / 地址
UDP/GPIO1	UDP	—	ST	CMOS	USB UDP 口
	GPIO1	—	ST	CMOS	通用 I/O 口
VDD/AVDD	VDD/ AVDD	—	PWR	—	正电源
VSS/AVSS	VSS/AVSS	—	PWR	—	负电源，接地
V33O	V33O	—	—	PWR	3.3V 稳压器输出
HVDD	HVDD	—	PWR	—	HIRC 正电源

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：施密特触发输入； CMOS：CMOS 输出； AN：模拟信号；
 HXT：高频晶体振荡器；
 LXT：低频晶体振荡器；
 此引脚功能表是针对大封装芯片而言的，对于小封装的芯片可能不具有上述引脚和功能。

极限参数

电源供应电压	$V_{SS}-0.3V \sim V_{SS}+6.0V$
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_{OH} 总电流	-100mA
I_{OL} 总电流	150mA
总功耗	500mW

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

直流电气特性

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{DD1}	工作电压 (晶振)	—	f _{sys} =4MHz	2.2	—	5.5	V
			f _{sys} =6MHz	2.2	—	5.5	V
			f _{sys} =8MHz	2.2	—	5.5	V
			f _{sys} =12MHz	2.7	—	5.5	V
			f _{sys} =16MHz	4.5	—	5.5	V
V _{DD2}	工作电压 (HIRC)	—	f _{sys} =12MHz	2.7	—	5.5	V
I _{DD1}	工作电流 (晶振, f _{sys} =f _H , f _s =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, f _H =4MHz	—	0.8	1.5	mA
		5V	ADC off, WDT 使能	—	1.8	4.0	mA
		3V	无负载, f _H =6MHz	—	1.0	2.0	mA
		5V	ADC off, WDT 使能	—	2.5	5.0	mA
		3V	无负载, f _H =8MHz	—	1.3	3.0	mA
		5V	ADC off, WDT 使能	—	3.0	5.5	mA
		3V	无负载, f _H =12MHz	—	2.0	4.0	mA
		5V	ADC off, WDT 使能	—	4.0	7.0	mA
I _{DD2}	工作电流 (HIRC, f _{sys} =f _H , f _s =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, f _H =12MHz	—	2.0	4.0	mA
		5V	ADC off, WDT 使能	—	4.0	7.0	mA
I _{DD3}	工作电流 (LXT, f _{sys} =f _L =f _{LXT} , f _s =f _{SUB} =f _{LXT})	3V	无负载, ADC off, WDT 使能,	—	40	80	μA
		5V	LVR 使能	—	70	150	μA
I _{DD4}	工作电流 (LIRC, f _{sys} =f _L =f _{LIRC} , f _s =f _{SUB} =f _{LIRC})	3V	无负载, ADC off, WDT 使能,	—	40	80	μA
		5V	LVR 使能, f _{LIRC} =32kHz	—	70	150	μA
I _{DD5}	工作电流 (HIRC, f _{sys} =f _H , f _s =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, f _H =12MHz, ADC off, WDT 使能, USB 使能,	—	5.5	10.0	mA
		5V	PLL on, V33O on	—	11	16	mA
I _{DD6}	工作电流 (晶振, f _{sys} =f _H , f _s =f _{SUB} =f _{LIRC})	5V	无负载, f _H =6MHz, ADC off, WDT 使能, USB 使能, PLL on, V33O on	—	10	15	mA
		5V	无负载, f _H =12MHz, ADC off, WDT 使能, USB 使能, PLL on, V33O on	—	11	16	mA
		5V	无负载, f _H =16MHz, ADC off, WDT 使能, USB 使能, PLL on, V33O on	—	12	17	mA
I _{STB1}	待机电流 (Idle 1) (晶振, f _{sys} =f _H , f _s =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, 系统 HALT, ADC off, WDT 使能,	—	0.8	1.5	mA
		5V	f _{sys} =OSC on (FSYSON=1)	—	1.5	3.0	mA
I _{STB2}	待机电流 (Idle 0) (晶振或 HIRC, f _{sys} =off, f _s =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, 系统 HALT ADC off, WDT 使能,	—	1.5	3.0	μA
		5V	f _{sys} =OSC off (FSYSON=0)	—	3.0	6.0	μA

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{STB3}	待机电流 (Idle 0) (LXT, f _{SYS} =off, f _S =f _{SUB} =f _{LXT})	3V	无负载, 系统 HALT ADC off, WDT 使能	—	2.0	4.0	μA
		5V		—	3.5	7.0	μA
I _{STB4}	待机电流 (Idle) (LIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LIRC})	3V	无负载, 系统 HALT ADC off, WDT 使能	—	1.5	3.0	μA
		5V		—	3.0	6.0	μA
I _{STB5}	待机电流 (Sleep 0) (晶振或 HIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, 系统 HALT ADC off, WDT 除能	—	0.1	1.0	μA
		5V		—	0.3	2.0	μA
I _{STB6}	待机电流 (Sleep 1) (晶振或 HIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LXT} 或 f _{LIRC})	3V	无负载, 系统 HALT ADC off, WDT 使能	—	1.5	3.0	μA
		5V		—	3.0	6.0	μA
I _{STB7}	待机电流 (Sleep 0) (晶振或 HIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LXT} 或 f _{LIRC})	—	无负载, 系统 HALT ADC off, WDT 除能, LVR 使能, LVDEN=1	—	60	90	μA
I _{SUS1}	暂停电流 (Sleep 0) (晶振或 HIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LXT} 或 f _{LIRC})	5V	无负载, 系统 HALT, ADC off, WDT 除能, USB 收发器和 3.3V 稳压器 on, clr SUSP2 (UCC.4)	—	360	420	μA
I _{SUS2}	暂停电流 (Sleep 0) (晶振或 HIRC, f _{SYS} =off, f _S =f _{SUB} =f _{LXT} 或 f _{LIRC})	5V	无负载, 系统 HALT, ADC off, WDT 除能, USB 收发器和 3.3V 稳压器 on, set SUSP2 (UCC.4)	—	240	320	μA
V _{IL1}	I/O, TCKn 和 INTn 的低 电平输入电压	—	—	0	—	0.2V _{DD}	V
V _{IH1}	I/O, TCKn 和 INTn 的高 电平输入电压	—	—	0.8V _{DD}	—	V _{DD}	V
V _{IL2}	低电平输入电压 (RES)	—	—	0	—	0.4V _{DD}	V
V _{IH2}	高电平输入电压 (RES)	—	—	0.9V _{DD}	—	V _{DD}	V
I _{OL}	I/O 口灌电流	3V	V _{OL} =0.1V _{DD} (PAOI 或 PXOI=1)	4	8	—	mA
		5V	V _{OL} =0.1V _{DD} (PAOI 或 PXOI=1)	10	20	—	mA
		5V	V _{OL} =0.1V _{DD} (PAOI 或 PXOI=0)	2	4	—	mA
I _{OH}	I/O 口源电流	3V	V _{OH} =0.9V _{DD} (PAOI 或 PXOI=1)	-2	-4	—	mA
		5V	V _{OH} =0.9V _{DD} (PAOI 或 PXOI=1)	-5	-10	—	mA
		5V	V _{OH} =0.9V _{DD} (PAOI 或 PXOI=0)	-2	-4	—	mA
V _{V330}	3.3V 稳压器输出电压	5V	V _{V330} =70mA	3.0	3.3	3.6	V
R _{UDP}	UDP 到 V330 上拉电阻	3.3V	—	-5%	1.5	+5%	kΩ
R _{PH}	输入 / 输出上拉电阻	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ
R _{PL}	UBUS 下拉电阻	5V	SUSP2=1, RUBUS=0	0.5	1	1.5	MΩ

注: HT66FB542 不存在 LXT 振荡器。

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 _{SYST}	系统时钟 (HXT)	2.2V~5.5V	—	2	—	4	MHz
		2.2V~5.5V		2	—	6	MHz
		2.2V~5.5V		2	—	8	MHz
		2.7V~5.5V		2	—	12	MHz
		4.5V~5.5V		2	—	16	MHz
f _{SYST2}	系统时钟 (HIRC)	2.2V~5.5V	非 USB 模式, Ta=25°C	-3%	12	+3%	MHz
		3.0V~5.5V	非 USB 模式, Ta=-40°C~85°C	-6%	12	+6%	MHz
		2.2V~5.5V	非 USB 模式, Ta=-40°C~85°C	-10%	12	+10%	MHz
		3.3V~5.5V	USB 模式	-0.25%	12	+0.25%	MHz
f _{SYST3}	系统时钟 (LXT)	—	—	—	32768	—	Hz
f _{LIRC}	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
f _{TIMER}	定时器输入频率 (TCKn)	2.2V~5.5V	—	2	—	8	MHz
		2.7V~5.5V		2	—	12	MHz
		4.5V~5.5V		2	—	16	MHz
t _{BGS}	VBG 打开稳定时间	—	—	10	—	—	ms
t _{TIMER}	TCKn 输入脉宽	—	—	0.3	—	—	μs
t _{RES}	外部复位低电压脉宽	—	—	10	—	—	μs
t _{INT}	中断脉冲宽度	—	—	10	—	—	μs
t _{SST}	系统启动延时周期 (从 HALT 中唤醒, f _{SYST} 在 HALT 模式下关闭, 低速模式 → 正常模式, 正常模式 → 低速模式)	—	f _{SYST} =HXT 或 LXT (低速模式 → 正常模式 (HXT), 正常模式 → 低速模式 (LXT))	1024	—	—	t _{SYST}
			f _{SYST} =HXT (从 HALT 中唤醒, f _{SYST} 在 HALT 模式下关闭)	1024	—	—	t _{SYST}
			f _{SYST} =HIRC	1024	—	—	t _{SYST}
			f _{SYST} =LIRC	2	—	—	t _{SYST}
	系统启动延时周期 (从 HALT 中唤醒, f _{SYST} 在 HALT 模式下开启)	—	—	2	—	—	t _{SYST}
t _{RSTD}	系统启动延时周期 (复位)	—	—	1024	—	—	t _{SYST}
	复位延时时间 (上电复位)	—	—	25	50	100	ms
	复位延时时间 (其它复位)	—	—	8.3	16.7	33.3	ms

注: HT66FB542 不存在 LXT 振荡器。

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 _{LVR1}	低电压复位电压	—	LVR 使能, 选择 2.1V	-5%	2.1	+5%	V
V _{LVR2}			LVR 使能, 选择 2.55V		2.55		V
V _{LVR3}			LVR 使能, 选择 3.15V		3.15		V
V _{LVR4}			LVR 使能, 选择 3.8V		3.8		V
V _{LVD1}	低电压检测电压	—	LVDEN=1, V _{LVD} =2.0V	-5%	2.0	+5%	V
V _{LVD2}			LVDEN=1, V _{LVD} =2.2V		2.2		V
V _{LVD3}			LVDEN=1, V _{LVD} =2.4V		2.4		V
V _{LVD4}			LVDEN=1, V _{LVD} =2.7V		2.7		V
V _{LVD5}			LVDEN=1, V _{LVD} =3.0V		3.0		V
V _{LVD6}			LVDEN=1, V _{LVD} =3.3V		3.3		V
V _{LVD7}			LVDEN=1, V _{LVD} =3.6V		3.6		V
V _{LVD8}			LVDEN=1, V _{LVD} =4.0V		4.0		V
V _{BG}	带隙参考缓冲电压	—	—	-3%	1.25	+3%	V
I _{BG}	使用 V _{BG} 的额外功耗	—	—	—	200	300	μA
I _{LVD}	使用 LVD/LVR 的额外功耗	3V	LVD 除能 → LVD 使能 (LVR 使能)	—	30	45	μA
		5V		—	60	90	μA
t _{LVR}	低电压复位脉宽	—	—	120	240	480	μs
t _{LVD}	低电压中断脉宽	—	—	20	45	90	μs
t _{SRESET}	软件复位脉宽	—	—	45	90	120	μs
t _{LVDs}	LVDO 稳定时间	—	LVR 使能, LVD off → on	15	—	—	μs

ADC 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				
AV _{DD}	ADC 工作电压	—	V _{REF} =AV _{DD}	2.7	—	5.5	V
V _{ADI}	ADC 输入电压	—	—	0	—	V _{REF}	V
V _{REF}	ADC 参考电压	—	AV _{DD} =3V	2.0	—	AV _{DD}	V
		—	AV _{DD} =5V	2.0	—	AV _{DD}	V
t _{ADC}	A/D 转换时间	2.7V~5.5V	12-bit ADC	16	—	20	t _{AD}
t _{AD}	A/D 时钟周期	2.7V~5.5V	—	0.5	—	10	μs
t _{ON2ST}	A/D 转换器 On-to-Start 时间	2.7V~5.5V	—	4	—	—	μs
DNL	A/D 非线性微分误差	3V/5V	V _{REF} =AV _{DD}	-3	—	+3	LSB
INL	A/D 非线性积分误差	3V/5V	V _{REF} =AV _{DD}	-4	—	+4	LSB
I _{ADC}	打开 A/D 增加的功耗 (仅 ADC 使能, 其它除能)	3V	无负载, t _{AD} =0.5μs	—	1.0	2.0	mA
		5V	无负载, t _{AD} =0.5μs	—	1.5	3.0	mA

比较器电气特性

Ta=25°C

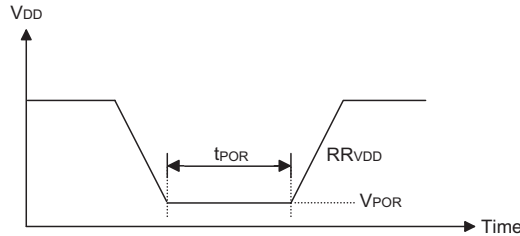
符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{CMP}	比较器工作电压	—	—	2.2	—	5.5	V
I _{CMP}	比较器工作电流	5V	—	—	—	200	μA
I _{CSTB}	比较器暂停模式工作电流	5V	比较器除能	—	—	0.1	μA
V _{CMPOS}	比较器输入失调电压	5V	—	-10	—	+10	mV
V _{HYS}	迟滞宽度	5V	—	20	40	60	mV
V _{CM}	比较器共模电压范围	—	—	V _{SS}	—	V _{DD} -1.4V	V
A _{OL}	比较器开环增益	—	—	60	80	—	dB
t _{PD}	比较器响应时间	3V	100mV 偏置 (注)	—	200	400	ns
		5V					

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

上电复位 (AC+DC) 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 _{VDD}	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

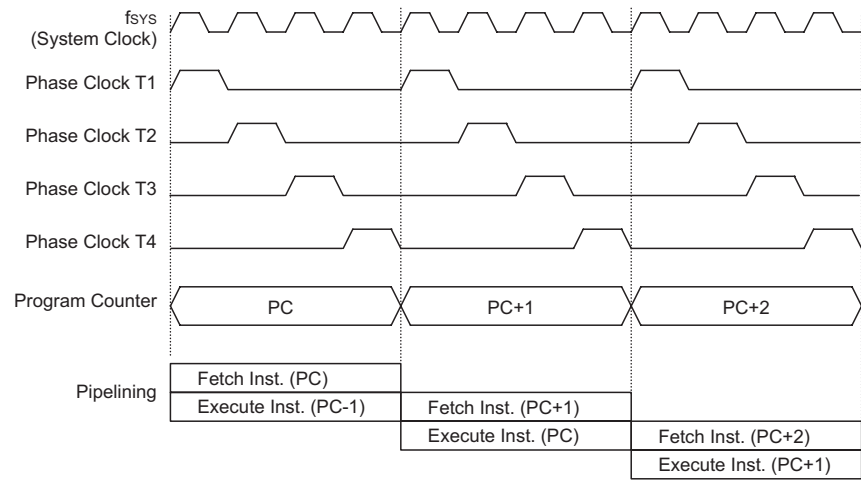


系统结构

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外，其它指令都能在一个指令周期内完成。8 位 ALU 参与指令集中所有的运算，它可完成算术运算、逻辑运算、移位、递增、递减和分支等功能，而内部的数据路径则是以通过累加器或 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 控制系统时，仅需要少数的外部器件。

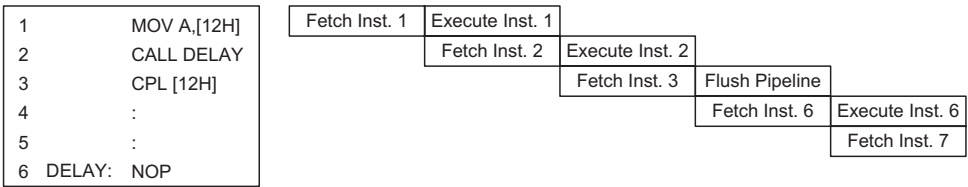
时序和流水线结构

主系统时钟由 HXT, LXT, HIRC 或者 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。注意，HT66FB542 不存在 LXT 振荡器。在 T1 时间，程序计数器自动加一并抓取一条新的指令。剩下的时间 T2~T4 完成译码和执行功能，因此，一个 T1~T4 时间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，如子程序的调用或者跳转，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地址。除了“JMP”和“CALL”指令需要跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完成以后自动加一。选择不同型号的单片机，程序寄存器的宽度会因程序存储器的容量的不同而不同。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 PMBP0 位来选择。然而只有较低的 8 位，即所谓的程序低字节寄存器 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、中断或者复位等，单片机通过加载所需要的位址到程序寄存器来控制程序，对于条件跳转指令，一旦条件符合，在当前指令执行时取得的下一条指令将会被舍弃，而由一个空指令周期来取代。

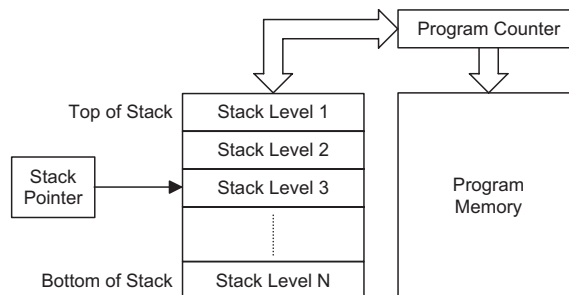
单片机型号	程序计数器	
	程序计数器高字节	PCL 寄存器
HT66FB540	PC11~PC8	PCL7~PCL0
HT66FB542	PC11~PC8	
HT66FB550	PC12~PC8	
HT66FB560	PMBP0, PC12~PC8	

程序计数器

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

堆栈

堆栈是一个特殊的存储器空间，用来保存程序计数器中的值。各单片机有不同的堆栈层数，堆栈寄存器既不是数据存储器的一部分，也不是程序存储器的一部分，而且它既不能读出，也不能写入。堆栈的使用是通过堆栈指针 SP 来指示的，堆栈指针也不能读出或写入。当发生子程序调用或中断响应时，程序计数器中的内容会被压入堆栈；在子程序调用结束或中断响应结束时，执行指令 RET 或 RETI，堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。



单片机型号	堆栈层数
HT66FB540/HT66FB542/HT66FB550	8
HT66FB560	12

如果堆栈已满，且有非屏蔽的中断发生，则只有中断请求标志位会被置位，而中断响应会被禁止，直到堆栈指针发生递减（执行 RET 或 RETI 指令），中断才会被响应。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 指令仍然可以执行，从而造成堆栈溢出。使用时应避免堆栈溢出的情况发生，因为这样会造成不可预期的程序分支指令的执行错误。

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份，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运算，并将结果储存在指定的寄存器，当 ALU 计算或操作时，可能导致进位、借位或其它状态的改变，而相关的状态寄存器会因此更新内容以显示这些改变，ALU 所提供的功能如下：

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

Flash 程序存储器

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

结构

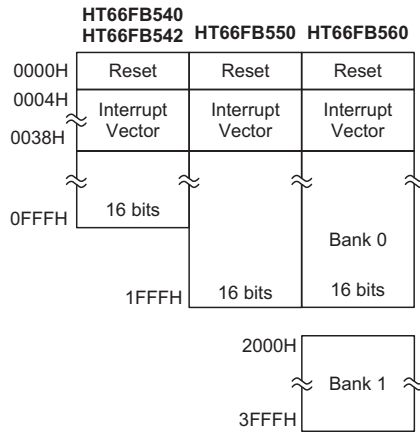
程序存储器的容量为 4K×16 位到 16K×16 位。程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口，数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。

单片机型号	ROM	Banks
HT66FB540/HT66FB542	4K×16	0
HT66FB550	8K×16	0
HT66FB560	16K×16	0, 1

HT66FB560 程序存储器分为两个 Bank，Bank 0 和 Bank 1。通过 BP 寄存器的第 5 位选择所需要的 Bank。

特殊向量

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



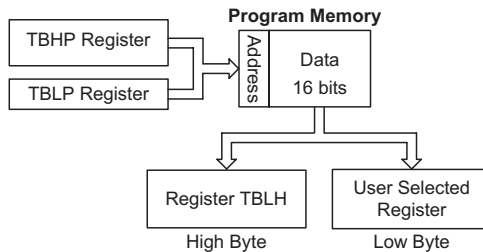
程序存储器结构

查表

程序存储器中的任何地址都可以定义为一个表格，以便存储固定的数据。使用查表指令时，查表指针需要先行设定，其方式是将表格的低位地址放在表格指针寄存器 TBLP 中。这个寄存器定义的是表格较低的 8 位地址。

在设定完表格指针后，表格数据可以使用“TABRD [m]”指令从程序所在的存储器的当前页或者存储器的最后一页中查表来读取。当这些指令执行时，程序存储器的表格的低字节，将会传输到用户所指定的数据存储器中。程序存储器表格的高字节，将会传输到特殊寄存器 TBLH 中。传输数据中任何未定义的字节将会读取为“0”。

下图为查表寻址 / 数据流程图：



查表范例

以下范例说明表格指针和表格数据如何被定义和执行。这个例子使用的表格数据用 ORG 伪指令储存在存储器的最后一页，HT66FB550 中 ORG 伪指令中的值为 1F00H，即 8K 程序存储器最后一页存储器的起始地址，而表格指针的初始值则为 06H，这可保证从数据表格读取的第一笔数据位于程序存储器地址 1F06H，即最后一页起始地址后的第六个地址。注意，假如“TABRD [m]”指令被使用，则表格指针指向当前页。在这个例子中，表格数据的高字节等于零，而当“TABRD [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

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

表格读取程序举例

```
tempreg1 db ?      ; temporary register #1
tempreg2 db ?      ; temporary register #2
:
:
mov a,06h           ; initialise low byte table pointer - note that
                     ; this address is referenced
mov tblp, a
mov a,1Fh           ; initialise high table pointer
mov tbhp, a
:
:
tabrd tempreg1      ; transfers value in table referenced by table pointer
                     ; data at program memory address "1F06H" transferred to
                     ; tempreg1 and TBLH
dec tblp            ; reduce value of table pointer by one
tabrd tempreg2      ; transfers value in table referenced by table pointer
                     ; data at program memory address "1F05H" transferred to
                     ; tempreg2 and TBLH
                     ; in this example the data "1AH" is transferred to
                     ; tempreg1 and data "0FH" to register tempreg2
:
:
org 1F00h           ; sets initial address of program memory
dc 000Ah, 000Bh, 000Ch, 000Dh, 000Eh, 000Fh, 001Ah, 001Bh
:
:
```

在系统烧录 – ISP

Flash 型程序存储器提供用户便利地对同一芯片进行程序的更新和修改。另外，Holtek 单片机提供 2 线 USB 接口的在系统烧录方式。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录，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。

芯片内部程序存储器可以通过 USB 接口在系统进行烧录，即使用 UDN 和 UDP 引脚。电源由 UBUS 引脚提供。芯片在系统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。Flash 程序存储器的读 / 写功能由一系列寄存器实现。

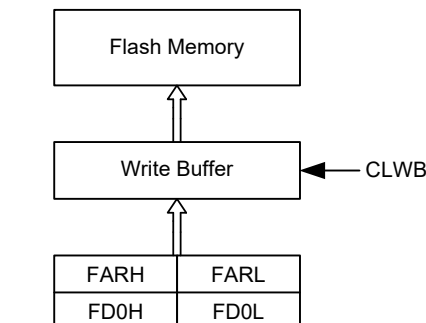
Flash 存储器读 / 写页面大小

Flash 存储器有两种页面大小，32 或 64 个字。当所选的 Flash 存储器大于 8K 字节，每个页面和缓存器将被分配 64 个字的页面大小。否则，分配 32 个字的页面大小。

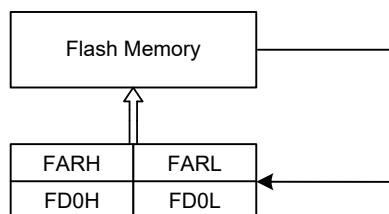
下图说明了读 / 写页面和缓存器的分配。写缓存器由 FRCR 寄存器中的 CLWB 位控制。CLWB 位置高，以使能清除写缓存器的操作，操作完成后，该位会由硬件清除。

写缓存器中的数据已满时 FWEN 位会被置高，当该位被置高，写缓存器内的数据将被写入到 Flash 程序存储器，FWT 位用来指示写操作，设置此位为高来检测写入操作是否完成，该位将由硬件清零。读字节可由地址来设定，FRDEN 用来使能读功能，FRD 用来指示读操作。当读操作完成后，该位由硬件清零。

单片机型号	页面大小 (字)	写缓存器 (字)
HT66FB540/HT66FB542 (4K×16)	32	32
HT66FB550 (8K×16)	32	32
HT66FB560 (16K×16)	64	64



向 FD0L/FD0H 写入一个字

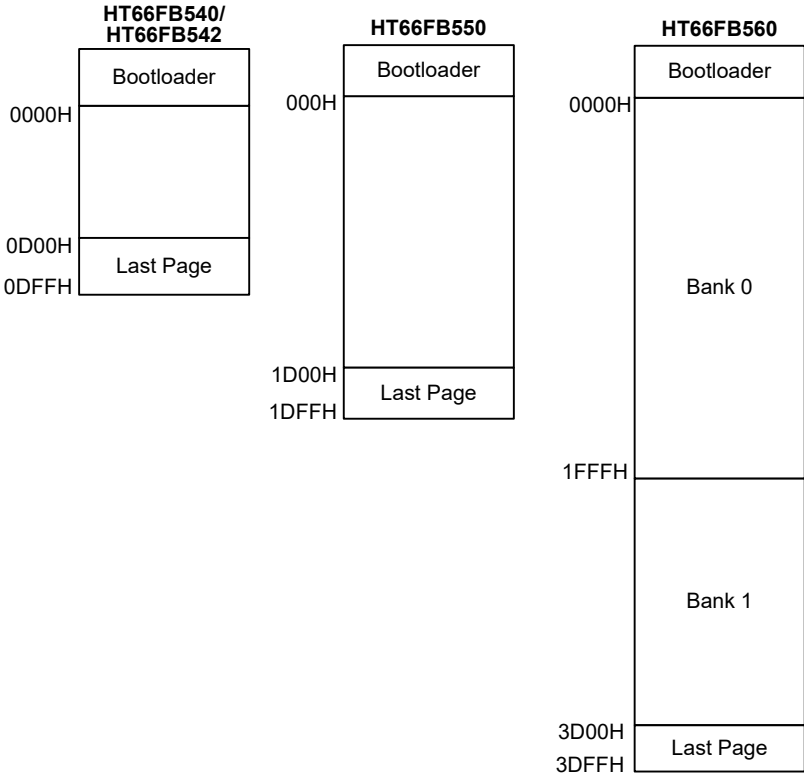


从 FD0L/FD0H 读取一个字

- 注：1. 写入数据到高字节，意味着高 / 低字节被写入到写缓存器，将使 Flash 存储器地址自动加 1，新地址将被加载到 FARH 和 FARL 寄存器。但是，用户通过在同一页面写入数据到 FARH 和 FARL 寄存器也可以写入新的地址，然后数据将被写入相应的地址。
2. 如果该地址已经达到的 Flash 存储器的上限，如 11111B (32 字) 或 111111b (64 字)。这时地址不会增加，将停止在该页面的最后一个地址，写入数据无效。
3. 这种情况下，用户必须再设定一个新的地址，以写入新的数据。
4. 如果数据是由写缓存器写入，写缓存器将在写操作就绪后 2ms 由硬件自动清除。
5. 首次使用或重新写入数据到写缓存器，用户可以使用清除缓存位 CLWB 来清除写入缓存器。

ISP Bootloader

该系列单片机提供 ISP Bootloader 功能来升级 Flash 存储器的软件。用户可以选择使用 Holtek IDE 工具提供的 ISP Bootloader 软件或创建自己的 Bootloader。若选择 Holtek 的 Bootloader 软件，将占用 0.5K 字的 Flash 存储。下图说明了带 Holtek Bootloader 软件的 Flash 存储结构。



带 Bootloader 的 Flash 存储器结构

Flash 程序存储器寄存器

Flash 程序存储器有两个地址寄存器，四个 16 位数据寄存器和两个控制寄存器。地址和数据高字节寄存器以及控制寄存器位于 Bank 1，其它寄存器位于 Bank 0。使用地址、数据和控制寄存器可以实现对 Flash 存储器的 16-bit 数据读写操作。这几个寄存器控制了内部 Flash 程序存储器总的操作，即地址寄存器 FARL 和 FARH，数据寄存器 FDnL 和 FDnH，控制寄存器 FCR 和 FRCR。FARL 和 FDnL 寄存器位于 Bank 0 中，它们能像其它特殊功能寄存器一样直接被访问。而 FDnH、FARH、FCR 和 FRCR 寄存器位于 Bank 1 中，不能直接被访问，只能通过 MP1 和 IAR1 进行间接读取或写入。

寄存器名称	位							
	7	6	5	4	3	2	1	0
FARL	D7	D6	D5	D4	D3	D2	D1	D0
FARH	—	—	—	—	D11	D10	D9	D8
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

寄存器名称	位							
	7	6	5	4	3	2	1	0
FCR	CFWEN	FMOD2	FMOD1	FMOD0	BWT	FWT	FRDEN	FRD
FRCR	—	—	—	FSWRST	—	—	—	CLWB

程序存储器寄存器列表 – HT66FB540/HT66FB542

寄存器名称	位							
	7	6	5	4	3	2	1	0
FARL	D7	D6	D5	D4	D3	D2	D1	D0
FARH	—	—	—	D12	D11	D10	D9	D8
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
FCR	CFWEN	FMOD2	FMOD1	FMOD0	BWT	FWT	FRDEN	FRD
FRCR	—	—	—	FSWRST	—	—	—	CLWB

程序存储器寄存器列表 – HT66FB550

寄存器名称	位							
	7	6	5	4	3	2	1	0
FARL	D7	D6	D5	D4	D3	D2	D1	D0
FARH	—	—	D13	D12	D11	D10	D9	D8
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
FCR	CFWEN	FMOD2	FMOD1	FMOD0	BWT	FWT	FRDEN	FRD
FRCR	—	—	—	FSWRST	—	—	—	CLWB

程序存储器寄存器列表 – HT66FB560

● FARL 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 Flash 程序存储器地址
Flash 程序存储器地址 bit 7~bit 0

● FARH 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	D11	D10	D9	D8
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	x	x	x	x

“x”：未知

Bit 7~4 保留，未使用
Bit 3~0 Flash 程序存储器地址
Flash 程序存储器地址 bit 11~bit 8

● FARH 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	D12	D11	D10	D9	D8
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	x	x	x	x	x

“x”：未知

Bit 7~5 保留，未使用
Bit 4~0 Flash 程序存储器地址
Flash 程序存储器地址 bit 12~bit 8

● FARH 寄存器 – HT66FB560

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

“x”：未知

Bit 7~6 保留，未使用
Bit 5~0 Flash 程序存储器地址
Flash 程序存储器地址 bit 13~bit 8

● FCR 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CFWEN	FMOD2	FMOD1	FMOD0	BWT	FWT	FRDEN	FRD
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 **CFWEN**: Flash 程序存储器写使能位 FWEN 控制位
0: 除能
1: 无效
该位用于控制 FWEN 位使能 / 除能。当该位由软件清零, Flash 程序存储器写使能位 FWEN 也会被清零。此位置高无效, 用户可以检查该位来确认 FWEN 位的状态。
- Bit 6~4 **FMOD2~FMOD0**: Flash 程序存储器工作模式控制位
000: 写存储器模式
001: 页擦除模式
010: 保留
011: 读存储器模式
100: 保留
101: 保留
110: FWEN (Flash 存储器写使能) 位控制模式
111: 保留
- Bit 3 **BWT**: 模式转换控制位
0: 模式转换周期结束
1: 激活模式转换周期
模式转换周期结束后, 硬件自动将此位清零。
- Bit 2 **FWT**: Flash 存储器写控制位
0: 写周期结束
1: 激活写周期
此位为 Flash 存储器写控制位, 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。
- Bit 1 **FRDEN**: Flash 存储器读使能位
0: 除能
1: 使能
此位为 Flash 存储器读使能位, 执行 Flash 存储器读操作之前需将此位置高。将此位清零则禁止 Flash 存储器读操作。
- Bit 0 **FRD**: Flash 存储器读控制位
0: 读周期结束
1: 激活读周期
此位为 Flash 存储器读控制位, 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 FRDEN 未先置高时, 此位置高无效。
注: 在同一条指令中 FWT、FRDEN 和 FRD 不能同时置为“1”。

● FRCR 寄存器

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

- Bit 7~5 未定义, 读为“0”
- Bit 4 **FSWRST**: 软件复位 MCU 控制位
必须清零
- Bit 3~1 未定义, 读为“0”

Bit 0 **CLWB:** Flash 程序存储器写缓存器清除控制位
0: 不开始写缓存器清除流程或写缓存器清除流程已完成
1: 开始写缓存器清除流程
该位用于控制 Flash 程序存储器写缓存器清除流程。

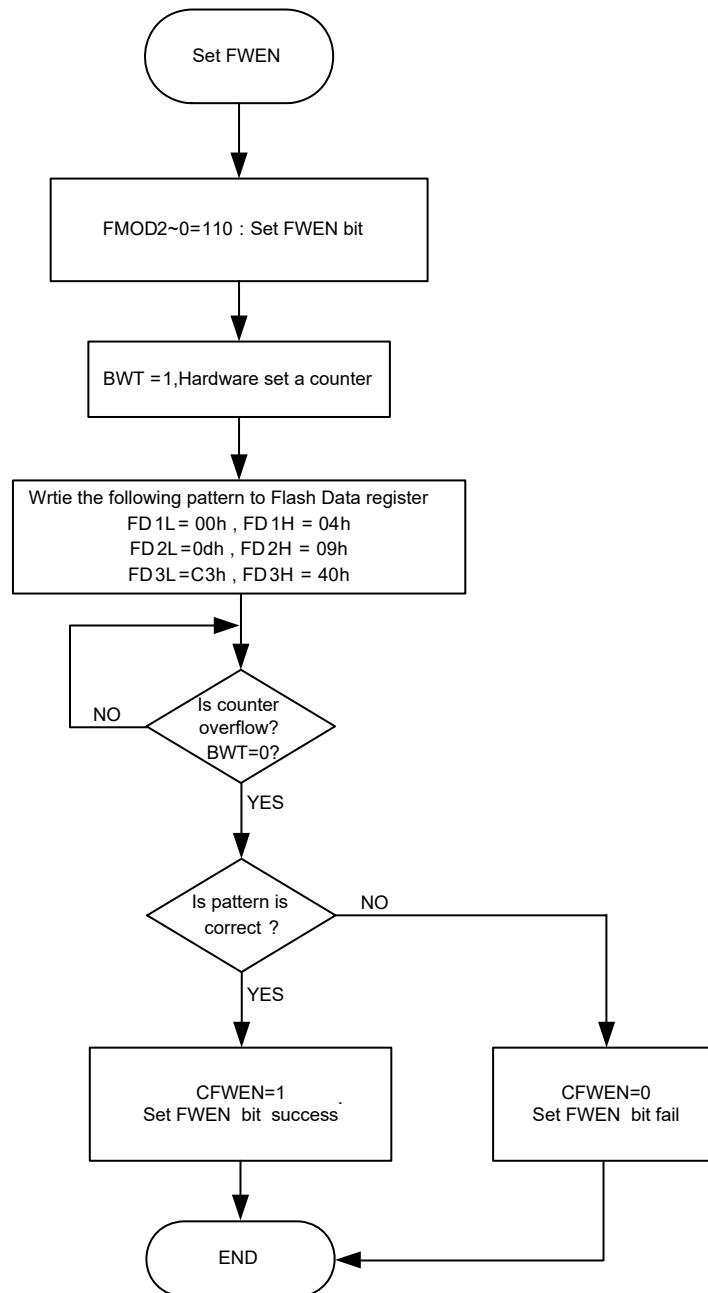
在应用烧录 – IAP

HT66FB5xx 系列单片机不仅提供 ISP 功能，而且还提供 IAP 功能，其 Flash 存储器可多次编程的特性给用户提供了较大的方便。IAP 功能的方便性是可以使用内部固件进行程序的更新，而无需外部烧录器或 PC。此外，IAP 接口通过 I/O 引脚可以设置为任何类型的通信协议，如 UART、USB。设计师可以指定 I/O 引脚与外部存储设备进行通信，包括更新程序。关于内部固件，用户可以选择 Holtek 提供的版本或创建自己的内部固件。以下章节说明了如何实现 IAP 固件的流程。

使能 Flash 写控制流程

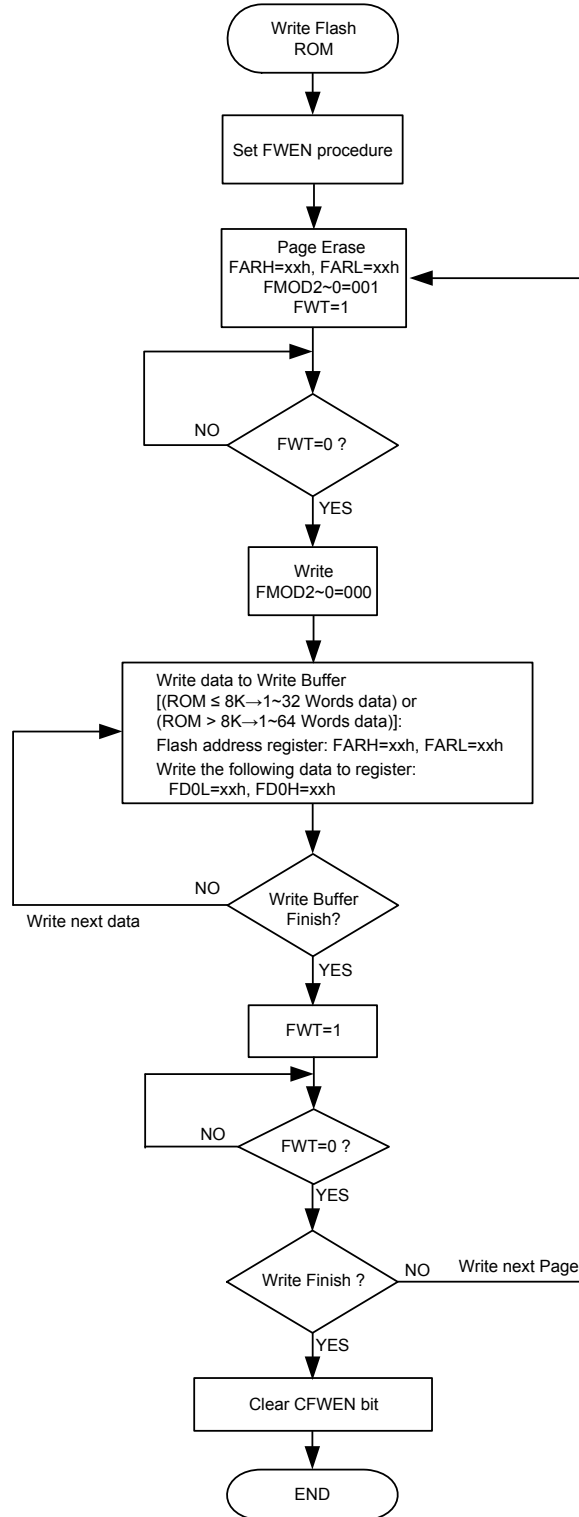
实现 IAP 固件的第一步是使能 Flash 写控制，包括以下步骤。

- 写入数据“110”到 FCR 寄存器中的 FMOD[2:0] 位，使能 Flash 写控制位 FWEN。
- 设置 FCR 寄存器中的 BWT 位为“1”。
- 单片机开始 1ms 计数。在此期间，用户应写入正确数据序列到 Flash 数据寄存器 FD1L~FD3L 和 FD1H~FD3H。
- 一旦 1ms 计数溢出或写入序列不正确，使能 Flash 写控制流程将无效，用户应重复上述步骤。
- 不管流程是否有效，单片机将自动清除 BWT 位。
- 使能 Flash 写序列数据 (00H、04H、0DH、09H、C3H、40H) 应写入 Flash 数据寄存器。
- 一旦使能 Flash 写操作，用户可以使用 Flash 控制寄存器更新 Flash 存储器。
- 用户只需通过清除 FCR 寄存器中的 CFWEN 位来除能 Flash 写流程，不必执行上述步骤。

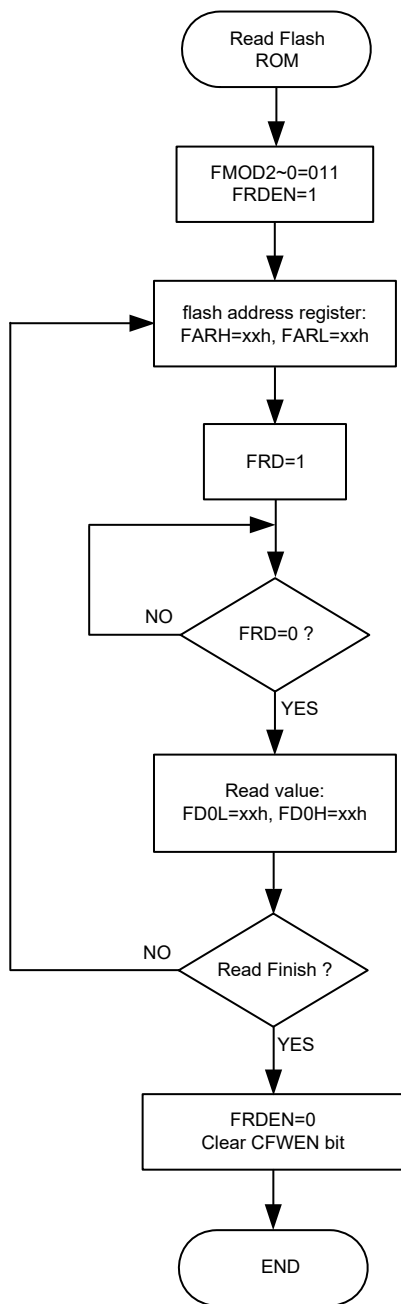


Flash 存储器写入和读取流程

下图为写入和读取 Flash 存储器流程图。



写入 Flash 存储器流程



读取 Flash 存储器流程

在线烧录 – ICP

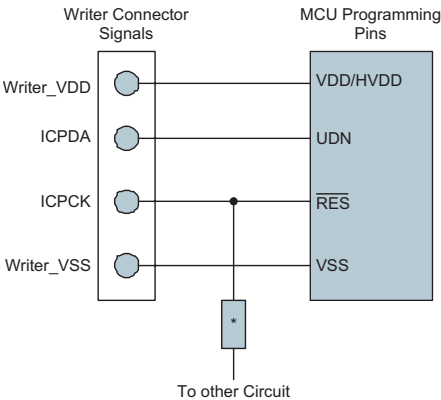
Flash 型程序存储器提供用户便利地对同一芯片进行程序的更新和修改。

另外，Holtek 单片机提供 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录，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 烧录器引脚名称	MCU 在线烧录引脚名称	功能
ICPDA	UDN	串行数据烧录
ICPCK	RES	时钟烧录
VDD	VDD/HVDD	电源
VSS	VSS	地

芯片内部程序存储器可以通过 4 线的接口在线进行烧录。其中一条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。芯片在线烧写的详细使用说明超出此文档的描述范围，将由专门的参考文献提供。

在烧录过程中，烧录器会控制 UDN 和 RES 脚进行数据和时钟烧录，用户必须确保这两个引脚没有连接至其它输出脚。



注：* 可能为电阻或电容。若为电阻则其值必须大于 300Ω，若为电容则其必须小于 1nF。

片上调试

EV 芯片 HT66VB5xx 用于 HT66FB5xx 系列单片机仿真。此 EV 芯片提供片上调试功能 (OCDS—On-Chip Debug Support) 用于开发过程中的 HT66FB5xx 系列单片机调试。除了片上调试功能方面，EV 芯片 HT66VB5xx 和 HT66FB5xx 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，从而实现 HT66VB5xx 对 HT66FB5xx 系列单片机的仿真。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 HT66VB5xx 进行调试时，HT66FB5xx 系列单片机 OCSDA 和 OCDSCK 引脚上的其它共用功能对 EV 芯片 HT66VB5xx 无效。由于这两个 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，请参考“Holtek e-Link for 8-bit MCU OCDS User's Guide”文件。

Holtek e-Link 引脚名称	EV IC 引脚名称	功能
OCSDA	OCSDA	片上调试串行数据 / 地址输入 / 输出
OCDSCK	OCDSCK	片上调试时钟输入
VDD	VDD/HVDD	电源
GND	VSS	地

数据存储器

数据存储器是内容可以更改的 8 位 RAM 内部存储器，用来存储临时数据。

结构

数据存储器分为两个部分，第一部分是特殊功能寄存器，这些寄存器有特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，而有些是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。

总的数据存储器被分为几个区，由所选的单片机决定的。大部分特殊功能数据寄存器均可在所有 Bank 被访问，处于 40H~46H 地址的 FRCR、FCR、FARH 和 FDnH 寄存器却只能在 Bank 1 中被访问到。切换不同区域可通过设置区域指针 (BP) 实现。所有单片机的数据存储器的起始地址都是“00H”。

单片机型号	容量	Banks
HT66FB540	512×8	0: 80H~FFH 1: 80H~FFH 2: 80H~FFH 3: 80H~FFH
HT66FB542	256×8	0: 80H~FFH 1: 80H~FFH
HT66FB550	768×8	0: 80H~FFH 1: 80H~FFH 2: 80H~FFH 3: 80H~FFH 4: 80H~FFH 5: 80H~FFH
HT66FB560	1024×8	0: 80H~FFH 1: 80H~FFH 2: 80H~FFH 3: 80H~FFH 4: 80H~FFH 5: 80H~FFH 6: 80H~FFH 7: 80H~FFH

Bank 0~3		Bank 0, 2~3		Bank 1
00H	IAR0	40H		FRCR
01H	MP0	41H		FCR
02H	IAR1	42H	FARL	FARH
03H	MP1	43H	FD0L	FD0H
04H	BP	44H	FD1L	FD1H
05H	ACC	45H	FD2L	FD2H
06H	PCL	46H	FD3L	FD3H
07H	TBLP	47H	TMPC0	
08H	TBLH	48H	TMPC1	
09H	TBHP	49H	TM0C0	
0AH	STATUS	4AH	TM0C1	
0BH	SMOD	4BH		
0CH	LVDC	4CH	TM0DL	
0DH	INTEG	4DH	TM0DH	
0EH	WDTC	4EH	TM0AL	
0FH	TBC	4FH	TM0AH	
10H	INTC0	50H	TM0RP	
11H	INTC1	51H		
12H	INTC2	52H	TM1C0	
13H	INTC3	53H	TM1C1	
14H	MF10	54H	TM1DL	
15H	MF11	55H	TM1DH	
16H	MF12	56H	TM1AL	
17H		57H	TM1AH	
18H	PAWU	58H	TM2C0	
19H	PAPU	59H	TM2C1	
1AH	PA	5AH	TM2DL	
1BH	PAC	5BH	TM2DH	
1CH	PADIR	5CH	TM2AL	
1DH	PAOI	5DH	TM2AH	
1EH	PSLEW	5EH	TM3C0	
1FH	PXWU	5FH	TM3C1	
20H	PXPU	60H	TM3DL	
21H	PXOI	61H	TM3DH	
22H	PB	62H	TM3AL	
23H	PBC	63H	TM3AH	
24H		64H	USB_STAT	
25H		65H	UINT	
26H	PD	66H	USC	
27H	PDC	67H	USR	
28H	PE	68H	UCC	
29H	PEC	69H	AWR	
2AH		6AH	STLI	
2BH		6BH	STLO	
2CH		6CH	SIES	
2DH		6DH	MISC	
2EH		6EH	UFIEN	
2FH	ADRL	6FH	UFOEN	
30H	ADRH	70H	UFC0	
31H	ADCR0	71H		
32H	ADCR1	72H	FIFO0	
33H	ACER0	73H	FIFO1	
34H		74H	FIFO2	
35H	CP0C	75H	FIFO3	
36H	CP1C	76H		
37H	I2CTOC	77H		
38H	SIMC0	78H		
39H	SIMC1	79H		
3AH	SIMD	7AH	CTRL	
3BH	SIMA/SIMC2	7BH	LVRC	
3CH	SPIAC0	7CH		
3DH	SPIAC1	7DH	PAPS0	
3EH	SPIAD	7EH	PAPS1	
3FH	SBSC	7FH	SYSC	

 : Unused, read as 00H

HT66FB540 特殊数据存储

Bank 0~1		Bank 0	Bank 1
00H	IAR0	40H	FRCR
01H	MP0	41H	FCR
02H	IAR1	42H	FARL
03H	MP1	43H	FD0L
04H	BP	44H	FD1L
05H	ACC	45H	FD2L
06H	PCL	46H	FD3L
07H	TBLP	47H	TMPC0
08H	TBLH	48H	TMPC1
09H	TBHP	49H	TM0C0
0AH	STATUS	4AH	TM0C1
0BH	SMOD	4BH	
0CH	LVDC	4CH	TM0DL
0DH	INTEG	4DH	TM0DH
0EH	WDTC	4EH	TM0AL
0FH		4FH	TM0AH
10H	INTC0	50H	TM0RP
11H	INTC1	51H	
12H	INTC2	52H	TM1C0
13H	INTC3	53H	TM1C1
14H	MFI0	54H	TM1DL
15H	MFI1	55H	TM1DH
16H	MFI2	56H	TM1AL
17H		57H	TM1AH
18H	PAWU	58H	TM2C0
19H	PAPU	59H	TM2C1
1AH	PA	5AH	TM2DL
1BH	PAC	5BH	TM2DH
1CH	PADIR	5CH	TM2AL
1DH	PAOI	5DH	TM2AH
1EH	PSLEW	5EH	TM3C0
1FH	PXWU	5FH	TM3C1
20H	PXPU	60H	TM3DL
21H	PXOI	61H	TM3DH
22H	PB	62H	TM3AL
23H	PBC	63H	TM3AH
24H		64H	USB_STAT
25H		65H	UINT
26H	PD	66H	USC
27H	PDC	67H	USR
28H	PE	68H	UCC
29H	PEC	69H	AWR
2AH		6AH	STLI
2BH		6BH	STLO
2CH		6CH	SIES
2DH		6DH	MISC
2EH		6EH	UFIE
2FH	ADRL	6FH	UFOEN
30H	ADRH	70H	UFC0
31H	ADCR0	71H	
32H	ADCR1	72H	FIFO0
33H	ACER0	73H	FIFO1
34H		74H	FIFO2
35H	CP0C	75H	FIFO3
36H		76H	
37H	I2CTOC	77H	
38H	SIMC0	78H	
39H	SIMC1	79H	
3AH	SIMD	7AH	CTRL
3BH	SIMA/SIMC2	7BH	LVRC
3CH	SPIAC0	7CH	
3DH	SPIAC1	7DH	PAPS0
3EH	SPIAD	7EH	PAPS1
3FH	SBSC	7FH	SYSC

□ : Unused, read as 00H

HT66FB542 特殊数据存储器

Bank 0~5		Bank 0, 2~5		Bank 1
00H	IAR0	40H		FRCR
01H	MP0	41H		FCR
02H	IAR1	42H	FARL	FARH
03H	MP1	43H	FD0L	FD0H
04H	BP	44H	FD1L	FD1H
05H	ACC	45H	FD2L	FD2H
06H	PCL	46H	FD3L	FD3H
07H	TBLP	47H	TMPC0	
08H	TBLH	48H	TMPC1	
09H	TBHP	49H	TM0C0	
0AH	STATUS	4AH	TM0C1	
0BH	SMOD	4BH		
0CH	LVDC	4CH	TM0DL	
0DH	INTEG	4DH	TM0DH	
0EH	WDTC	4EH	TM0AL	
0FH	TBC	4FH	TM0AH	
10H	INTC0	50H	TM0RP	
11H	INTC1	51H		
12H	INTC2	52H	TM1C0	
13H	INTC3	53H	TM1C1	
14H	MF10	54H	TM1DL	
15H	MF11	55H	TM1DH	
16H	MF12	56H	TM1AL	
17H		57H	TM1AH	
18H	PAWU	58H	TM2C0	
19H	PAPU	59H	TM2C1	
1AH	PA	5AH	TM2DL	
1BH	PAC	5BH	TM2DH	
1CH	PADIR	5CH	TM2AL	
1DH	PAOI	5DH	TM2AH	
1EH	PSLEW	5EH	TM3C0	
1FH	PXWU	5FH	TM3C1	
20H	PXPU	60H	TM3DL	
21H	PXOI	61H	TM3DH	
22H	PB	62H	TM3AL	
23H	PBC	63H	TM3AH	
24H	PC	64H	USB_STAT	
25H	PCC	65H	UINT	
26H	PD	66H	USC	
27H	PDC	67H	USR	
28H	PE	68H	UCC	
29H	PEC	69H	AWR	
2AH		6AH	STLI	
2BH		6BH	STLO	
2CH		6CH	SIES	
2DH		6DH	MISC	
2EH		6EH	UFIEH	
2FH	ADRL	6FH	UFOEN	
30H	ADRH	70H	UFC0	
31H	ADCR0	71H	UFC1	
32H	ADCR1	72H	FIFO0	
33H	ACER0	73H	FIFO1	
34H	ACER1	74H	FIFO2	
35H	CP0C	75H	FIFO3	
36H	CP1C	76H	FIFO4	
37H	I2CTOC	77H	FIFO5	
38H	SIMC0	78H		
39H	SIMC1	79H		
3AH	SIMD	7AH	CTRL	
3BH	SIMA/SIMC2	7BH	LVRC	
3CH	SPIAC0	7CH		
3DH	SPIAC1	7DH	PAPS0	
3EH	SPIAD	7EH	PAPS1	
3FH	SBSC	7FH	SYSC	

 : Unused, read as 00H

HT66FB550 特殊数据存储器

Bank 0~7		Bank 0, 2~7	Bank 1
00H	IAR0	40H	FRCR
01H	MP0	41H	FCR
02H	IAR1	42H	FARL FARH
03H	MP1	43H	FD0L FD0H
04H	BP	44H	FD1L FD1H
05H	ACC	45H	FD2L FD2H
06H	PCL	46H	FD3L FD3H
07H	TBLP	47H	TMPC0
08H	TBLH	48H	TMPC1
09H	TBHP	49H	TM0C0
0AH	STATUS	4AH	TM0C1
0BH	SMOD	4BH	
0CH	LVDC	4CH	TM0DL
0DH	INTEG	4DH	TM0DH
0EH	WDTC	4EH	TM0AL
0FH	TBC	4FH	TM0AH
10H	INTC0	50H	TM0RP
11H	INTC1	51H	
12H	INTC2	52H	TM1C0
13H	INTC3	53H	TM1C1
14H	MFI0	54H	TM1DL
15H	MFI1	55H	TM1DH
16H	MFI2	56H	TM1AL
17H		57H	TM1AH
18H	PAWU	58H	TM2C0
19H	PAPU	59H	TM2C1
1AH	PA	5AH	TM2DL
1BH	PAC	5BH	TM2DH
1CH	PADIR	5CH	TM2AL
1DH	PAOI	5DH	TM2AH
1EH	PSLEW	5EH	TM3C0
1FH	PXWU	5FH	TM3C1
20H	PXPU	60H	TM3DL
21H	PXOI	61H	TM3DH
22H	PB	62H	TM3AL
23H	PBC	63H	TM3AH
24H	PC	64H	USB_STAT
25H	PCC	65H	UINT
26H	PD	66H	USC
27H	PDC	67H	USR
28H	PE	68H	UCC
29H	PEC	69H	AWR
2AH	PF	6AH	STLI
2BH	PFC	6BH	STLO
2CH	PFPU	6CH	SIES
2DH	PFWU	6DH	MISC
2EH		6EH	UFIEH
2FH	ADRL	6FH	UFOEN
30H	ADRH	70H	UFC0
31H	ADCR0	71H	UFC1
32H	ADCR1	72H	FIFO0
33H	ACER0	73H	FIFO1
34H	ACER1	74H	FIFO2
35H	CP0C	75H	FIFO3
36H	CP1C	76H	FIFO4
37H	I2CTOC	77H	FIFO5
38H	SIMC0	78H	FIFO6
39H	SIMC1	79H	FIFO7
3AH	SIMD	7AH	CTRL
3BH	SIMA/SIMC2	7BH	LVRC
3CH	SPIAC0	7CH	
3DH	SPIAC1	7DH	PAPS0
3EH	SPIAD	7EH	PAPS1
3FH	SBSC	7FH	SYSC

□ : Unused, read as 00H

HT66FB560 特殊数据存储器

特殊功能寄存器

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

间接寻址寄存器 – IAR0, IAR1

间接寻址寄存器 IAR0 和 IAR1，位于数据存储区，并没有实际的物理地址。间接寻址方式是使用间接寻址寄存器或者存储器指针对数据操作，以取代定义在实际存储器地址的直接存储器寻址方式。在间接寻址寄存器上的任何动作，将对存储器指针 (MP0 或 MP1) 所指定的存储器地址产生对应的读 / 写操作。IAR0 和 MP0, IAR1 和 MP1 对数据存储区中数据的操作是成对出现的。间接寻址寄存器不是实际存在的，直接读取 IAR 寄存器将会返回 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, MP1

该系列单片机提供两个存储器指针，即 MP0 和 MP1。由于这些指针在数据存储区中能像普通的寄存器一样被写入和操作，因此提供了一个有效的间接寻址和数据追踪的方法。当对间接寻址寄存器进行任何操作时，单片机所指向的实际地址是由存储器指针所指定的地址。MP0, IAR0 用于访问 Bank 0，而 MP1 和 IAR1 可通过 BP 寄存器访问所有的 Bank。直接寻址仅可用于 Bank 0，其它所有 Bank 都可使用 MP1 和 IAR1 进行间接寻址。

以下范例说明如何清除一个具有 4 个 RAM 地址的区块，它们已经事先被定义成地址 adres1 到 adres4。

间接寻址程序范例

```
data. section 'data'
adres1      db?
adres2      db?
adres3      db?
adres4      db?
block       db?

code. section at 0 code
org 00h

start:
mov a,04h           ; setup size of block
mov block,a
mov a,offset adres1 ; Accumulator loaded with first RAM address
mov mp0,a          ; setup memory pointer with first RAM address
loop:
clr IAR0           ; clear the data at address defined by MP0
inc mp0            ; increment memory pointer
sdz block          ; check if last memory location has been cleared
jmp loop
continue:
```

在以上的例子中，没有提及具体的数据存储区地址。

存储区指针 – BP

数据存储区被分为几个部分，具体数目由所选择的单片机型号决定。可以通过设置存储区指针 (Bank Pointer) 值来访问不同的程序和数据存储区。BP 指针的第 5 位用于选择程序存储区 0 或 1，第 0~2 位用于选择数据存储区 0~7。

复位后，数据存储区会初始化到 Bank 0，但是在暂停模式下的 WDT 溢出复位，

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

由于程序存储器和数据存储器共用同一个 BP 寄存器，编程时务必注意。

单片机型号	位							
	7	6	5	4	3	2	1	0
HT66FB540	—	—	—	—	—	—	DMBP1	DMBP0
HT66FB542	—	—	—	—	—	—	—	DMBP0
HT66FB550	—	—	—	—	—	DMBP2	DMBP1	DMBP0
HT66FB560	—	—	PMBP0	—	—	DMBP2	DMBP1	DMBP0

存储区指针

● BP 寄存器 – HT66FB540

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

Bit 7~3 未定义，读为“0”

Bit 1~0 **DMBP1, DMBP0**: 数据存储区指针
00: Bank 0
01: Bank 1
10: Bank 2
11: Bank 3

● BP 寄存器 – HT66FB542

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

Bit 7~1 未定义，读为“0”

Bit 0 **DMBP0**: 数据存储区指针
0: Bank 0
1: Bank 1

● BP 寄存器 – HT66FB550

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

Bit 7~3 未定义，读为“0”

Bit 2~0 **DMBP2~DMBP0**: 数据存储区指针
000: Bank 0
001: Bank 1
010: Bank 2
011: Bank 3
100: Bank 4

101: Bank 5
110~111: 未定义

● BP 寄存器 – HT66FB560

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

Bit 7~6 未定义，读为“0”
Bit 5 **PMBP0**: 程序存储区指针
0: Bank 0, 程序存储器地址 0000H~1FFFH
1: Bank 1, 程序存储器地址 2000H~3FFFH
Bit 4~3 未定义，读为“0”
Bit 2~0 **DMBP2~DMBP0**: 数据存储区指针
000: Bank 0
001: Bank 1
010: Bank 2
011: Bank 3
100: Bank 4
101: Bank 5
110: Bank 6
111: Bank 7

累加器 – ACC

对于任何单片机来说，累加器是相当重要的，且与 ALU 所完成的运算有密切关系，所有的 ALU 得到的运算结果都将暂存在累加器中，如果没有累加器，ALU 必须在每次进行如加法、减法和移位等运算时，将结果写入数据存储器中，这样会造成程序编写和时间的负担。另外，数据传输通常涉及到累加器的临时储存功能，如在用户定义的寄存器和另一个寄存器之间，由于两者之间的不能直接传送数据，因此需要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，程序计数器的低字节被设置在数据存储器的特殊功能区域，程序员可对此寄存器进行操作，很容易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专用程序存储器某一地址，然而，由于寄存器只有 8 位的长度，因此只允许在本页的程序存储器中跳转。注意，使用这种运算时，会插入一个空指令周期。

表格寄存器 – TBLP, TBHP, TBLH

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

状态寄存器 – STATUS

这 8 位寄存器包括零标志位 (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)、和看门狗溢出标志位 (TO)。这些标志位同时记录单片机的状态数据和算术 / 逻辑运算。

除了 TO 和 PDF 标志位以外，状态寄存器的其它位像其它大多数寄存器一样可以被改变。但是任何数据写入状态寄存器将不会改变 TO 和 PDF 标志位。另外，执行不同指令操作后，与状态寄存器相关的运算将会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出、执行“CLR WDT”或“HALT”指令的影响。PDF 指令只会受执行“HALT”或“CLR WDT”指令或系统上电的影响。

Z、OV、AC 和 C 标志位通常反映最近的运算操作的状态。

另外，当进入一个中断程序或者执行子程序调用时状态寄存器将不会自动压入到堆栈中保存。假如状态寄存器的内容很重要，且中断子程序会改变状态寄存器的内容，则需要保存备份以备恢复。注意，状态寄存器的 0~3 位可以读取和写入。

● 状态寄存器

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

“x”：未知

Bit 7~6 未定义，读为“0”

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

Bit 4 **PDF**: 暂停标志位
0: 系统上电或执行“CLR WDT”指令
1: 执行“HALT”指令将会置位 PDF 位。

Bit 3 **OV**: 溢出标志位
0: 不发生溢出时
1: 当运算结果高两位的进位状态异或结果为 1 时

Bit 2 **Z**: 零标志位
0: 算数运算或者逻辑运算的结果不为零时
1: 算数运算或者逻辑运算的结果为零时

Bit 1 **AC**: 辅助进位标志位
0: 没有辅助进位时
1: 当低字节的加法造成进位或者高字节的减法没有造成借位时

Bit 0 **C**: 进位标志位
0: 没有进位时
1: 当加法造成进位或者减法没有造成借位时，同时移位指令也会影响 C 标志位。

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获得更多范围的功能。振荡器的灵活性使得在速度和功耗之间可以达到较佳的优化。振荡器选项是通过配置选项和寄存器来完成的。

系统振荡器概述

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

注意，HT66FB542 不存在 LXT 振荡器和时基功能，此章中所有 LXT 和时基相关的功能都不适用于 HT66FB542。

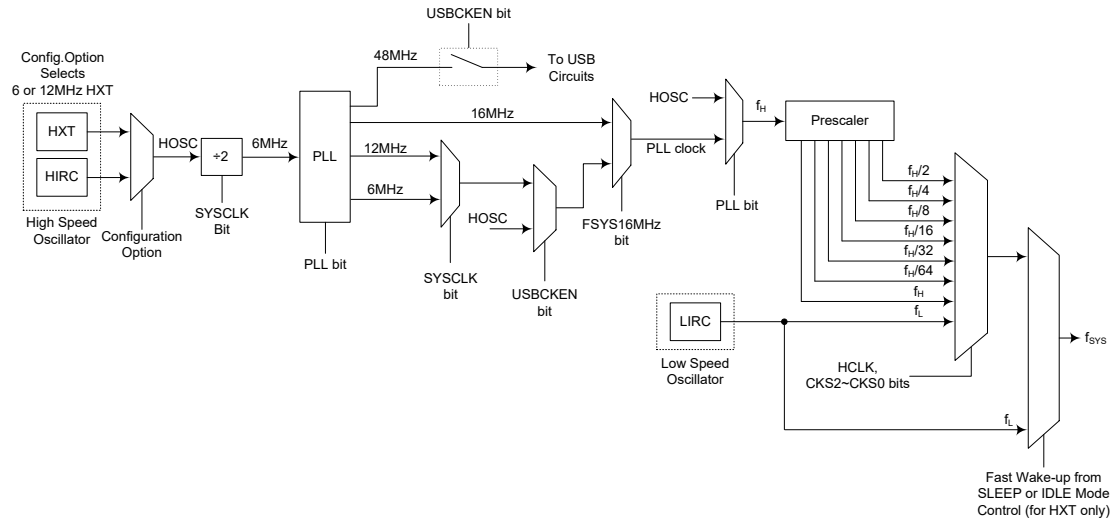
振荡类型	名称	频率范围	引脚
外部晶振	HXT	6MHz 或 12MHz	OSC1/OSC2
内部高速 RC	HIRC	12MHz	—
外部低速晶振	LXT	32.768kHz	XT1/XT2
内部低速 RC	LIRC	32kHz	—

注：对于 USB 应用，HXT 需要连接一个 6MHz 或 12MHz 晶振。

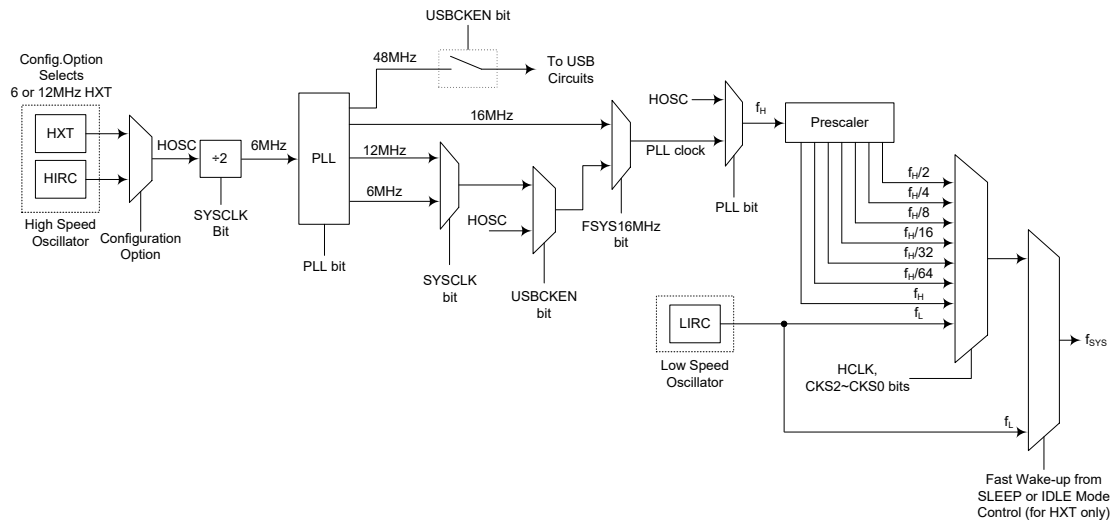
振荡器类型

系统时钟配置

此系列单片机有若干个系统振荡器，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，PLL 频率发生器和内部 12MHz RC 振荡器。两个低速振荡器包括外部 32768Hz 振荡器和内部 32kHz RC 振荡器。注意，HT66FB542 不存在 LXT 振荡器。使用高速或低速振荡器作为系统时钟的选择是通过设置 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位决定的，系统时钟可动态选择。高速或低速振荡器的实际时钟源经由配置选项选择。请注意，两个振荡器必须做出选择，即一个高速和一个低速振荡器。此外，内部 PLL 频率发生器的时钟源是外部晶体振荡器提供的，可以由软件控制位使能，以产生各种频率供 USB 接口和系统时钟使用。



系统时钟配置 – HT66FB542



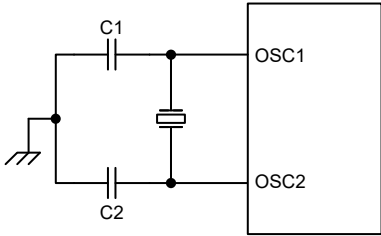
系统时钟配置 – HT66FB540/HT66FB550/HT66FB560

外部晶体 / 陶瓷振荡器 – HXT

外部高频晶体 / 陶瓷振荡器是一个高频振荡器。

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，而不需其它外部电容。为保证某些低频率的晶体振荡和陶瓷谐振器的振荡频率更精准，建议连接两个小容量电容 C1 和 C2 到 VSS，具体数值与客户选择的晶体 / 陶瓷晶振有关。对于 USB 应用，如果选择 HXT 振荡器，HXT 引脚必须连接一个 6MHz 或 12MHz 的晶振。

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他们之间的连线都应尽可能的接近单片机。



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 / 陶瓷振荡器 – HXT

晶体振荡器 C1 和 C2 值		
晶体频率	C1	C2
16MHz	0pF	0pF
12MHz	0pF	0pF
8MHz	0pF	0pF
6MHz	0pF	0pF
4MHz	0pF	0pF
1MHz	100pF	100pF
注意：C1 和 C2 数值仅作参考用		

注：对于 USB 应用，HXT 需要连接一个 6MHz 或 12MHz 晶振。

晶体振荡器电容推荐值

内部 PLL 频率发生器

内部 PLL 频率发生器是用来产生供 USB 接口和系统时钟使用的频率。PLL 发生器打开 / 关闭由 USC 寄存器中的 PLL 控制位设置。上电复位后，PLL 控制位被清零以打开 PLL 发生器。该 PLL 发生器将提供固定的 48MHz 工作频率给 USB，以及另一个 6MHz、12MHz 或 16MHz 频率给系统时钟源。系统频率由 UCC 寄存器中的 SYSCLK、Fsys16MHZ 和 USBCKEN 位决定。此外，通过这些控制位系统时钟可以选择 HXT。CLK_ADJ 位用来自动调整 PLL 时钟。

• SYSC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CLK_ADJ	USBdis	RUBUS	—	—	HFV	—	—
R/W	R/W	R/W	R/W	—	—	R/W	—	—
POR	0	0	0	—	—	0	—	—

- Bit 7 **CLK_ADJ:** PLL 时钟自动调整功能控制位
0: 除能
1: 使能
注意，如果用户选择 HIRC 作为系统时钟，该位需设为 1 来自动调整 PLL 频率。
- Bit 6 **USBdis:** USB SIE 控制位
USB 相关控制位，具体描述详见其它章节
- Bit 5 **RUBUS:** USB 引脚下拉电阻
USB 相关控制位，具体描述详见其它章节
- Bit 4~3 未定义，读为 “0”

- Bit 2 **HFV**: USB 模式高频电压控制位
此位必须设置为 1。
- Bit 1~0 未定义，读为“0”

• UCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	Rctrl	SYSCLK	Fsys16MHz	SUSP2	USBCKEN	—	EPS1	EPS0
R/W	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W
POR	0	0	0	x	0	—	x	x

“x”：未知

- Bit 7 **Rctrl**: UDP 和 UBUS 之间有无 7.5kΩ 电阻控制位
USB 相关控制位，具体描述详见其它章节
- Bit 6 **SYSCLK**: 系统时钟频率选择位
0: 12MHz
1: 6MHz
注：如果使用 6MHz 晶体振荡器或谐振器，该位应设置为 1。
如果使用 12MHz 晶体振荡器或谐振器，该位应设置为 0。
如果使用 12MHz HIRC，该位应设置为 0。
- Bit 5 **Fsys16MHz**: PLL 16MHz 输出控制位
0: HXT
1: PLL 16MHz
- Bit 4 **SUSP2**: 暂停模式下降低功耗控制位
USB 相关控制位，具体描述详见其它章节
- Bit 3 **USBCKEN**: USB 时钟使能控制位
0: 除能
1: 使能
- Bit 2 未定义，读为“0”
- Bit 1~0 **EPS1~EPS0**: 选择要访问端点的 FIFO
USB 相关控制位，具体描述详见其它章节

• USC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	URD	SELPS2	PLL	SELUSB	RESUME	URST	RMWK	SUSP
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R
POR	1	0	0	0	x	x	x	x

“x”：未知

- Bit 7 **URD**: USB 复位信号控制位
USB 相关控制位，具体描述详见其它章节
- Bit 6 **SELPS2**: 芯片在 PS2 模式下工作标志位
USB 相关控制位，具体描述详见其它章节
- Bit 5 **PLL**: PLL 控制位
0: 打开 PLL
1: 关闭 PLL
- Bit 4 **SELUSB**: 芯片在 USB 模式下工作标志位
USB 相关控制位，具体描述详见其它章节
- Bit 3 **RESUME**: USB 恢复标志位
USB 相关控制位，具体描述详见其它章节

- Bit 2 **URST**: USB 复位标志位
 USB 相关控制位，具体描述详见其它章节
- Bit 1 **RMWK**: USB 远程唤醒命令
 USB 相关控制位，具体描述详见其它章节
- Bit 0 **SUSP**: USB 暂停标志位
 USB 相关控制位，具体描述详见其它章节

下表说明了 PLL 的输出频率是由相关控制位选择的。高速系统时钟 f_H 有如下选择：

PLL	USBCEEN	Fsys16MHz	f_H
0	0	0	HOSC (HXT 或 HIRC)
0	0	1	f_{PLL} — 16MHz
0	1	0	f_{PLL} — 6MHz 或 12MHz，由 UCC 寄存器中的 SYSClk 位决定
0	1	1	f_{PLL} — 16MHz
1	x	x	HOSC (HXT 或 HIRC)

“x”：未知

内部 RC 振荡器 – HIRC

内部 RC 振荡器是一个集成的系统振荡器，不需其它外部器件。内部 RC 振荡器具有一个固定的频率：12MHz。虽然此振荡器配置成本较低，但是振荡频率会因 V_{DD} 、温度以及芯片制成工艺不同而改变，因此不适合用来作较严格计时或需要震荡频率精度较高的场合。在电源电压为 3.3V 或者 5V 及温度为 25°C 的条件下，12MHz 这三个固定频率的容差为 3% (非 USB 模式)。如果选择了该内部时钟，无需额外的引脚；PD0 和 PD1 可以作为通用 I/O 口使用。HIRC 有自己的电源引脚，HVDD。该引脚需连接到 VDD，并连接一个 0.1μF 电容到地。

外部 32.768kHz 晶体振荡器 – LXT

外部 32.768kHz 晶体振荡器是一个低速振荡器，可通过配置选项选择。注意，HT66FB542 不存在 LXT 振荡器。时钟频率固定为 32.768kHz，此时 XT1 和 XT2 间引脚必须连接 32.768kHz 的晶体振荡器。需要外部电阻和电容连接到 32.768kHz 晶振以帮助起振。对于那些要求精确频率的场合中，可能需要这些元件来对由制程产生的误差提供频率补偿。在系统上电期间，LXT 振荡器启动需要一定的延时。

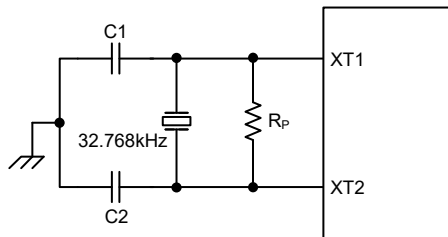
当系统进入空闲 / 休眠模式，系统时钟关闭以降低功耗。然而在某些应用，比如空闲 / 休眠模式下要保持内部定时器功能，必须提供额外的时钟，且与系统时钟无关。

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

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

- 若 LXT 振荡器未被用于任何时钟源，XT1/XT2 脚能被作为一般 I/O 口使用。
- 若 LXT 振荡器被用于一些时钟源，32.768kHz 晶体应被连接至 XT1/XT2 脚。

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他们之间的连线都应尽可能的接近单片机。



Note: 1. R_p , 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 的值		
晶体振荡器频率	C1	C2
32.768kHz	10pF	10pF
注：1. C1 和 C2 的值仅供参考 2. 推荐 $R_p=5M\sim10M\Omega$		

32.768kHz 晶体振荡器电容推荐值

LXT 振荡器低功耗功能

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

LXTLP 位	LXT 模式
0	快速启动
1	低功耗

系统上电时会清零 LXTLP 位来快速启动 LXT 振荡器。在快速启动模式，LXT 振荡器将起振并快速稳定下来。LXT 振荡器完全起振后，可以通过设置 LXTLP 位为高进入低功耗模式。振荡器可以继续运行，其间耗电将少于快速启动模式。在功耗敏感的应用领域如电池应用方面，功耗必须限制为一个最小值。为了降低功耗，建议系统上电 2 秒后，在应用程序中将 LXTLP 位设为“1”。

应注意的是，无论 LXTLP 位是什么值，LXT 振荡器会一直运作，不同的只是在低功耗模式时启动时间更长。

内部 32kHz 振荡器 – LIRC

内部 32kHz 系统振荡器也是一个低速振荡器，可通过配置选项选择。它是一个完全集成 RC 振荡器，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 电压下的精度保持在 10% 以内。

辅助振荡器

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

工作模式和系统时钟

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

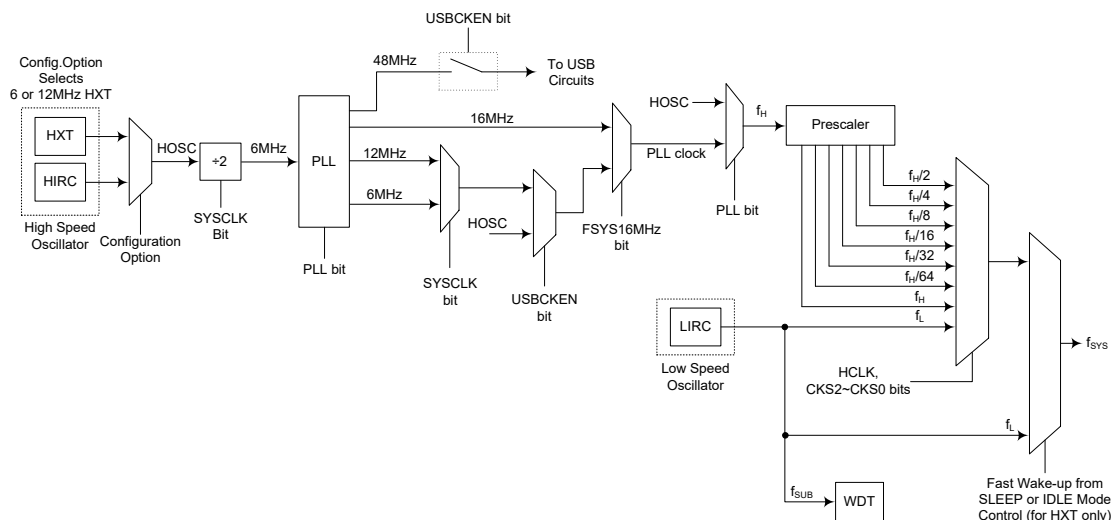
系统时钟

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

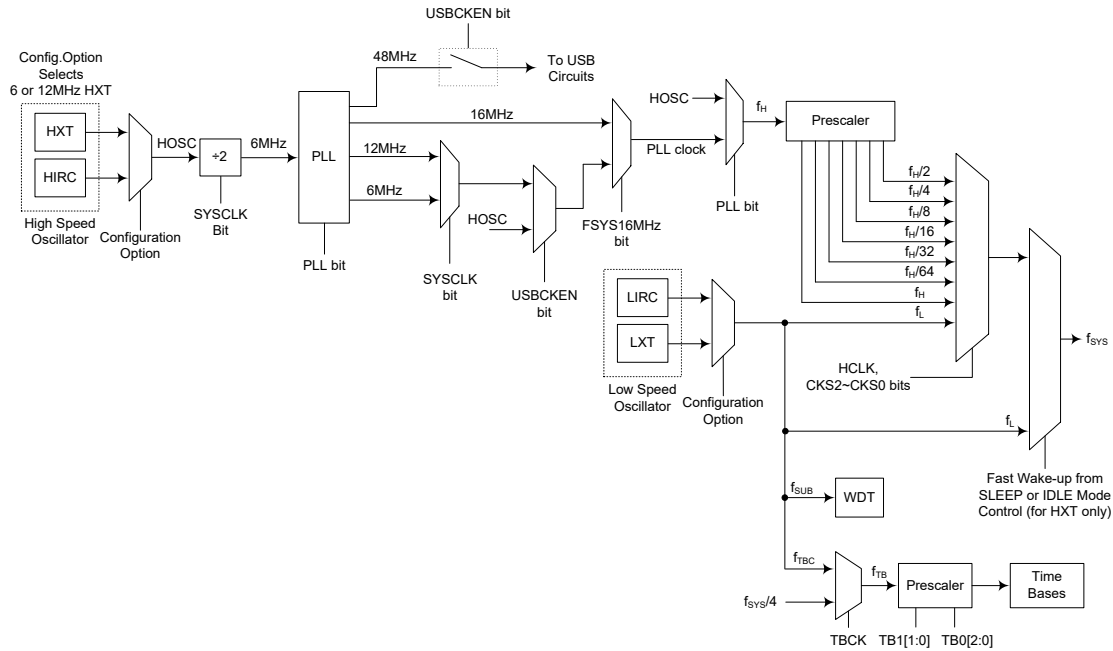
主系统时钟可来自高频时钟源 f_H 或低频时钟源 f_L ，通过 SMOD 寄存器中的 HLCLK 位及 CKS2~CKS0 位进行选择。高频时钟可通过配置选项以及 PLL、USBCKEN 和 Fsys16MHZ 控制位一起选择来自 HXT、PLL 频率发生器或 HIRC 振荡器，低频系统时钟源来自 f_L ，可通过配置选项选择来自 LIRC 或 LXT 振荡器。其它系统时钟还有高速系统振荡器的分频 $f_H/2 \sim f_H/64$ 。

另外两个内部时钟用于外围电路，次时钟源 f_{SUB} 和时基时钟 f_{TBC} 。这两个时钟源可通过配置选项选择来自 LIRC 或 LXT 振荡器。快速唤醒发生后， f_{SUB} 为单片机提供一个次时钟。对于 HT66FB540/HT66FB550/HT66FB560， f_{TBC} 是用于时基中断功能和 TM 的时钟源。而对于 HT66FB542， f_L 是用于时基中断功能和 TM 的时钟源。

注意，HT66FB542 不存在 LXT 振荡器和时基功能，此章中所有 LXT 和时基相关的功能都不适用于 HT66FB542。



单片机时钟配置 – HT66FB542



单片机时钟配置 – HT66FB540/HT66FB550/HT66FB560

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

系统工作模式

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

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

注： f_{TBC} 时钟不适用于 HT66FB542。

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 PLL 频率发生器、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 SMOD 寄存器中的 CKS2~CKS0 位及 HLCLK 位选择的。单片机使用高速振荡器分频作为系统时钟可减少工作电流。

低速模式

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

休眠模式 0

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

休眠模式 1

在 HALT 指令执行后且 SMOD 寄存器中 IDLEN 位为低时，系统进入休眠模式。在休眠模式 1 中，CPU 停止运行。然而若 LVDEN 位为“1”或看门狗定时器功能使能， f_{SUB} 继续运行。

空闲模式 0

执行 HALT 指令后且 SMOD 寄存器中 IDLEN 位为高，CTRL 寄存器中 FSYSON 位为低时，系统进入空闲模式 0。在空闲模式 0 中，CPU 停止，但一些外围功能如看门狗定时器、TMs 和 SIM 将继续工作。在空闲模式 0 中，系统振荡器停止，看门狗定时器时钟 f_{SUB} 开启。

空闲模式 1

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

控制寄存器

寄存器 SMOD 用于控制单片机内部时钟。

• SMOD 寄存器

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

Bit 7~5 **CKS2~CKS0:** 当 HLCLK 为“0”时系统时钟选择位

000: f_L (f_{LXT} 或 f_{LIRC})

001: f_L (f_{LXT} 或 f_{LIRC})

010: $f_H/64$

011: $f_H/32$

100: $f_H/16$

101: $f_H/8$

110: $f_H/4$

111: $f_H/2$

这三位于用于选择系统时钟源。除了 LXT 或 LIRC 振荡器提供的系统时钟源外，也可使用高频振荡器的分频作为系统时钟。注： f_{LXT} 时钟不适用于 HT66FB542。

- Bit 4 **FSTEN**: 快速唤醒控制位 (仅用于 HXT)
0: 除能
1: 使能
此位为快速唤醒控制位, 用于决定单片机被唤醒后 f_{SUB} 是否开始工作。若此位为高, 当 f_{SUB} 有效时, 此时钟源可作为暂用系统时钟以提供一个较快的唤醒时间。
- Bit 3 **LTO**: 低速振荡器就绪标志位
0: 未就绪
1: 就绪
此位为低速系统振荡器就绪标志位, 用于表明低速系统振荡器在系统上电复位或经唤醒后何时稳定下来。当系统处于 SLEEP0 模式时, 该标志为低。系统时钟来自 LIRC 振荡器, 该位转换为高需 1~2 个时钟周期; 若系统时钟来自 LXT 振荡器, 系统唤醒后该位转换为高需 1024 个时钟周期。
- Bit 2 **HTO**: 高速振荡器就绪标志位
0: 未就绪
1: 就绪
此位为高速系统振荡器就绪标志位, 用于表明高速系统振荡器何时稳定下来。此标志在系统上电后经硬件清零, 高速系统振荡器稳定后变为高电平。因此, 此位在单片机上电后由应用程序读取的总为“1”。该标志由休眠模式或空闲模式 0 中唤醒后会处于低电平状态, 若使用 HXT 振荡器, 该位将在 1024 个时钟周期后变为高电平状态, 若使用 HIRC 振荡器则只需 1024 个时钟周期即可。
- Bit 1 **IDLEN**: 空闲模式控制位
0: 除能
1: 使能
此位为空闲模式控制位, 用于决定 HALT 指令执行后发生的动作。若此位为高, 当指令 HALT 执行后, 单片机进入空闲模式。若 FSYSON 位为高, 在空闲模式 1 中 CPU 停止运行, 系统时钟将继续工作以保持外围功能继续工作; 若 FSYSON 为低, 在空闲模式 0 中 CPU 和系统时钟都将停止运行。若此位为低, 单片机将在 HALT 指令执行后进入休眠模式。
- Bit 0 **HLCLK**: 系统时钟选择位
0: $f_H/2 \sim f_H/64$ 或 f_L
1: f_H
此位用于选择 f_H 或 $f_H/2 \sim f_H/64$ 还是 f_L 作为系统时钟。该位为高时选择 f_H 作为系统时钟, 为低时则选择 $f_H/2 \sim f_H/64$ 或 f_L 作为系统时钟。当系统时钟由 f_H 时钟向 f_L 时钟转换时, f_H 将自动关闭以降低功耗。

• CTRL 寄存器

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

“x”: 未知

- Bit 7 **FSYSON**: IDLE 模式下 f_{SYS} 控制位
0: 除能
1: 使能
- Bit 6~3 未使用, 读为“0”
- Bit 2 **LVRF**: LVR 复位标志位
详见其它章节
- Bit 1 **LRF**: LVRC 控制的复位标志位
详见其它章节
- Bit 0 **WRF**: WDTC 控制的复位标志位
详见其它章节

快速唤醒

单片机进入休眠模式或空闲模式 0 后，系统时钟将停止以降低功耗。然而单片机再次唤醒，原来的系统时钟重新起振、稳定且恢复正常工作需要一定的时间。为确保单片机能够尽快的开始工作，系统提供了一个快速唤醒功能。需提供一临时时钟源 f_{SUB} 先驱动系统直至原系统振荡器稳定，这个临时时钟来自 LXT 或 LIRC 振荡器。快速启动功能的时钟源为 f_{SUB} ，该功能仅在休眠模式 1 和空闲模式 0 中有效。当单片机由休眠模式 0 唤醒时，因 f_{SUB} 已停止，故快速唤醒功能无效。快速唤醒功能使能 / 除能由 SMOD 寄存器中 FSTEN 位控制的。

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

若系统振荡器选用 HIRC，将系统从休眠模式或空闲模式 0 中唤醒需 1024 个时钟周期；若选用 LIRC，则需 1~2 个周期。快速唤醒位 FSTEN 在这些情况下不受影响。

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

“×”表示未知

唤醒时间

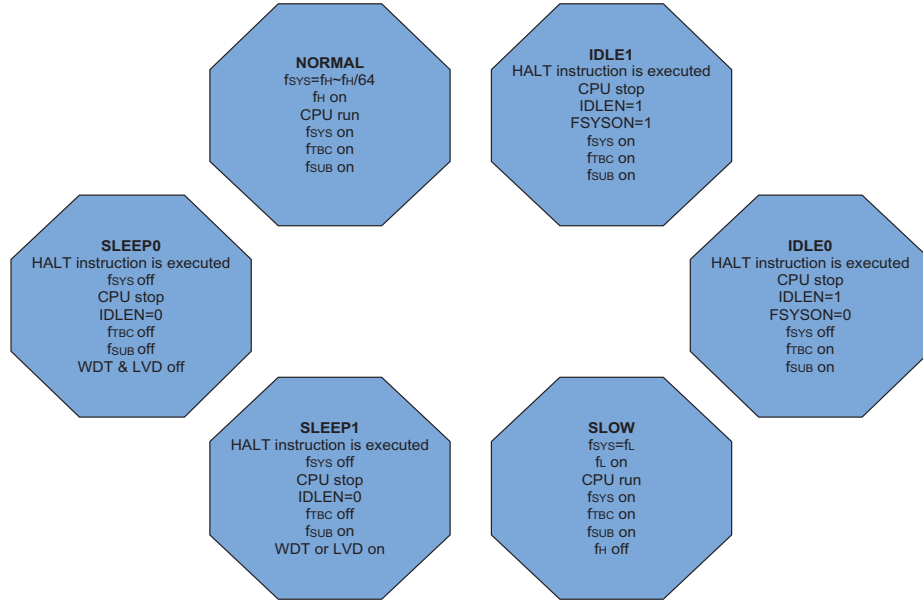
注：若看门狗定时器除能，意味着 LXT 和 LIRC 都关闭，当单片机由休眠模式 0 中唤醒时，快速唤醒功能不可用。HT66FB542 不存在 LXT 振荡器。

工作模式切换和唤醒

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

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

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

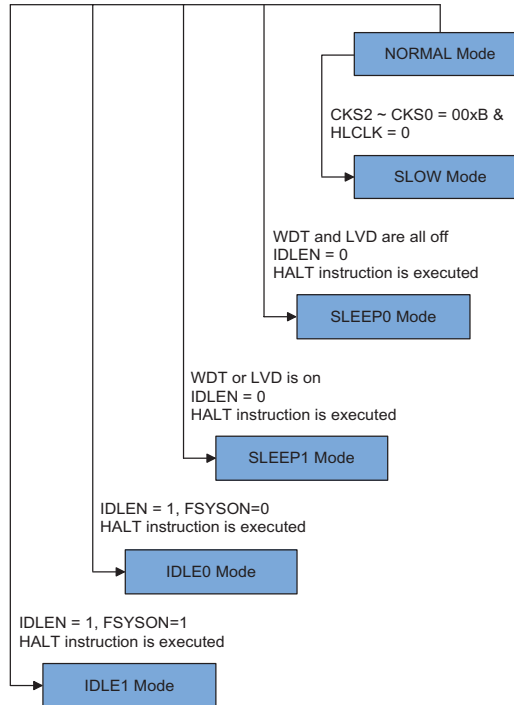


注：frec 时钟不适用于 HT66FB542。

正常模式切换到低速模式

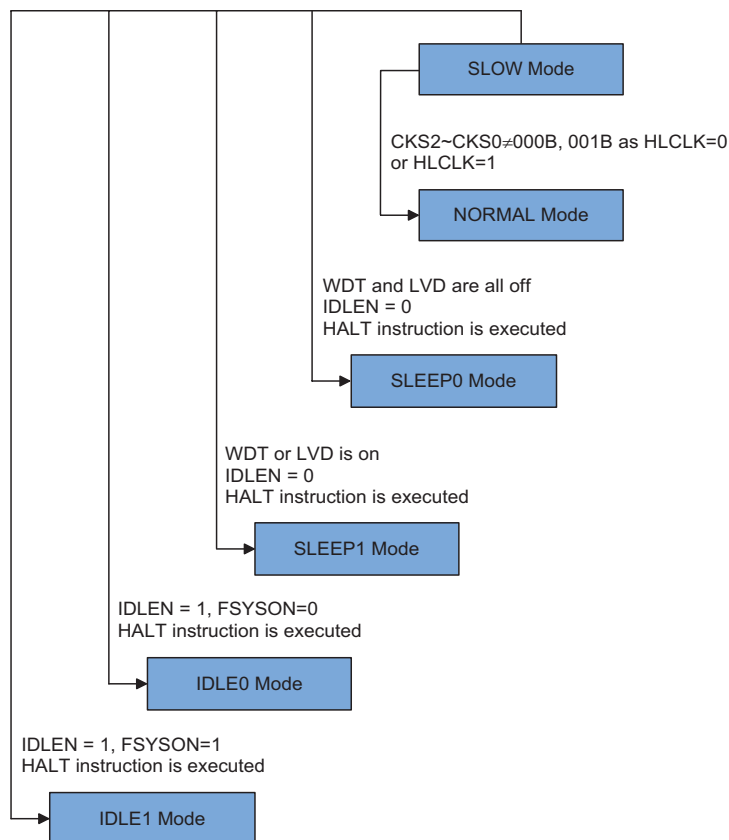
系统运行在正常模式时使用高速系统振荡器，因此较为耗电。可通过设置 SMOD 寄存器中的 HLCLK 位为“0”及 CKS2~CKS0 位为“000”或“001”使系统时钟切换至运行在低速模式下。此时将使用低速系统振荡器以节省耗电。用户可在对性能要求不高的操作中使用此方法以减少耗电。

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



低速模式切换到正常模式

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



进入休眠模式 0

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

- 系统时钟、WDT 时钟和时基时钟停止运行，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- WDT 将被清除并停止运行。
- 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。

进入休眠模式 1

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

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

进入空闲模式 0

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

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

进入空闲模式 1

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

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

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 除外），所以如果要将电路的电流进一步降低，电路设计者还应有其它的考虑。应该特别注意的是单片机的输入 / 输出引脚。所有高阻抗输入脚都必须连接到固定的高或低电平，因为引脚浮空会造成内部振荡并导致耗电增加。这也应用于有不同封装的单片机，因为它们可能含有未引出的引脚，这些引脚也必须设为输出或带有上拉电阻的输入。

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

唤醒

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- 外部复位或 USB 复位

- I/O 口上升沿或下降沿
- 系统中断
- WDT 溢出

若由外部复位或 USB 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。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。

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

编程注意事项

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

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

看门狗定时器

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

看门狗定时器时钟源

WDT 定时器时钟源来自于内部时钟 f_{SUB} ，而 f_{SUB} 的时钟源由 LIRC 或 LXT 振荡器提供，可由配置选项选择。注意，HT66FB542 不存在 LXT 振荡器。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。电压为 5V 时内部振荡器 LIRC 的周期大约为 32kHz。

需要注意的是，这个特殊的内部时钟周期随 V_{DD} 、温度和制成的不同而变化。可以通过设置 WDTC 寄存器来使能 / 除能 WDT 功能。

看门狗定时器控制寄存器

WDTC 寄存器用于控制 WDT 功能的使能 / 除能及选择溢出周期。CTRL 寄存器可以反映出 WRF 软件复位的状态。

• 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 软件控制位

10101: 除能

01010: 使能

其它: MCU 复位

如果单片机复位是由环境干扰引起的，则需要 2~3 个 LIRC 时钟周期响应复位，在复位后 CTRL 寄存器中的 WRF 标志位会被置位。

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: $2^{18}/f_{SUB}$

这三位控制 WDT 时钟源的分频比，从而实现对 WDT 溢出周期的控制。

• CTRL 寄存器

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

“x”: 未知

Bit 7 **FSYSON**: IDLE 模式下 f_{SYS} 控制位

详见其它章节

Bit 6~3 未使用，读为 “0”

- Bit 2

LVRF: LVR 复位标志位
详见其它章节
- Bit 1

LRF: LVRC 控制的复位标志位
详见其它章节
- Bit 0

WRF: WDTC 控制的复位标志位
0: 未发生
1: 发生

WDTC 控制寄存器复位时，该位被置为“1”，且通过应用程序清除。注意，该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。WDTC 寄存器中的 WE4~WE0 位可用来使能 / 除能看门狗定时器及其复位控制。

使用 WDTC 寄存器使能 / 除能看门狗定时器

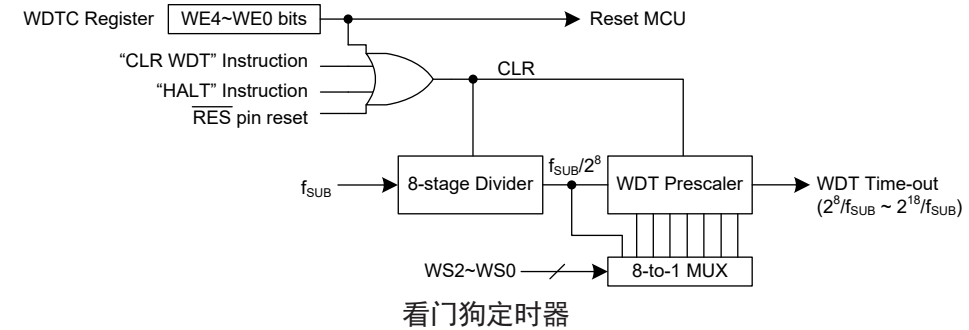
看门狗定时器的使能 / 除能是由 WDTC 寄存器控制的，WE4~WE0 位可以决定 WDT 在哪种模式下工作。如果 WE4~WE0 为 10101B，则 WDT 除能；如果 WE4~WE0 为 01010B，则 WDT 使能；如果 WE4~WE0 为其它任意值，则经过 2~3 个 LIRC 时钟周期后单片机复位。上电后这几位的值为 01010B。

WDT	WE4~WE0 位	WDT 功能
由 WDTC 寄存器控制	10101	除能
	01010	使能
	其它值	单片机复位

看门狗定时器使能 / 除能控制

程序正常运行时，WDT 溢出将导致芯片复位，并置位状态标志位 TO。若系统处于休眠或空闲模式，当 WDT 发生溢出时，状态寄存器中的 TO 应置位，仅 PC 和堆栈指针复位。有四种方法可以用来清除 WDT 的内容。第一种是 WDT 复位，即写入除 01010B 和 10101B 外任何值到 WE4~WE0 位，第二种是通过软件清除指令，第三种是通过“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。只有一种软件指令用于清除看门狗寄存器。因此只要执行“CLR WDT”便清除 WDT。

当设置分频比为 2^{18} 时，溢出周期最大。例如，时钟源为 32kHz LIRC 振荡器，分频比为 2^{18} 时最大溢出周期约 8s，分频比为 2^8 时最小溢出周期约 7.8ms。



复位和初始化

复位功能是任何单片机中基本的部分，使得单片机可以设定一些与外部参数无关的先置条件。单片机在上电后会自动执行硬件复位，但还有一些其它的硬件和软件复位源也可以在单片机运行时动态地实现复位。

复位概述

最重要的复位条件是在单片机首次上电以后，经过短暂的延迟，内部硬件电路使得单片机处于预期的稳定状态并开始执行第一条程序指令。上电复位以后，在程序未执行之前，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。

该系列单片机提供了几个复位源产生内部复位信号，从而增强了单片机的保护。下表列出了不同类型的复位。

序号	复位名称	缩写	标志位	寄存器	注释
1	上电复位	POR	—	—	上电后自动产生
2	复位引脚	$\overline{\text{RES}}$	—	—	硬件复位
3	低电压复位	LVR	LVRF	CTRL	V_{DD} 低电压
4	LVRC 寄存器控制的软件复位	—	LRF	CTRL	往 LVRC 寄存器写值
5	看门狗复位	WDT	TO	STATUS	WDT 溢出
6	WDTC 寄存器控制的软件复位	—	WRF	CTRL	往 WDTC 寄存器写值

复位源列表

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

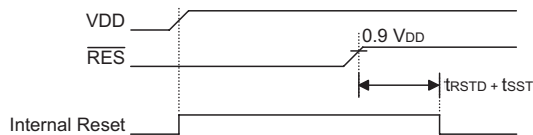
另一种复位为看门狗溢出单片机复位，不同方式的复位操作会对寄存器产生不同的影响。另一种复位为低电压复位即 LVR 复位，在电源供应电压低于 LVR 设定值时，系统会产生 LVR 复位，这种复位与 $\overline{\text{RES}}$ 脚拉低复位方式相同。

复位功能

包括内部和外部事件触发复位，单片机共有多种复位方式：

上电复位

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



注： t_{rSTD} 为上电延迟时间，典型值为 100ms。

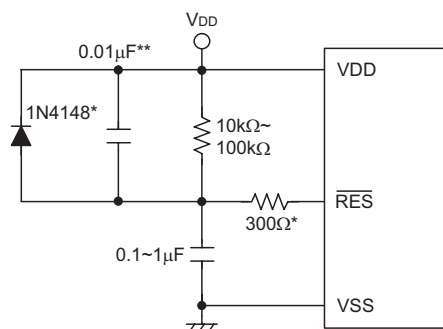
上电复位时序图

RES 引脚复位

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

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

当系统在较强干扰的场合工作时，建议使用增强型的复位电路，如下图所示。



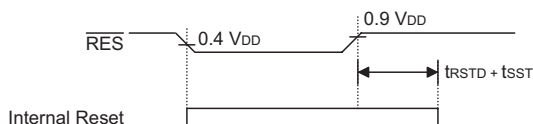
注：“*”表示建议加上此元件以加强静电保护。

“**”表示建议在电源有较强干扰场合加上此元件。

外部 RES 电路

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

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

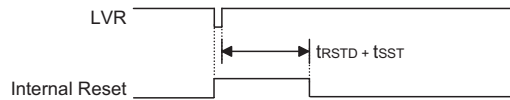


注： t_{RSTD} 为上电延迟时间，典型值为 16.7ms。

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。低电压复位功能始终使能于特定的电压值， V_{LVR} 。例如在更换电池的情况下，单片机供应的电压可能会在 $0.9V \sim V_{LVR}$ 之间，这时 LVR 将会自动复位单片机且 CTRL 寄存器中的 LVRF 标志位置位。LVR 包含以下的规格：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。如果低电压存在不超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。 V_{LVR} 参数值可通过 LVRC 寄存器中的 LVS 位进行设置。若由于受到干扰 LVS7~LVS0 变为其它值时，需经过 2~3 个 LIRC 周期响应复位。此时 CTRL 寄存器的 LRF 位被置位。上电后寄存器的值为 01010101B。正常执行时 LVR 会于休眠或空闲时自动除能关闭。



注：trSTD 为上电延迟时间，典型值为 16.7ms。

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/W	R/W	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~3 **LVS7~LVS0**: LVR 电压选择

01010101: 2.1V

00110011: 2.55V

10011001: 3.15V

10101010: 3.8V

其它值: MCU 复位 (寄存器设置为 POR 值)

若低电压情况发生且满足以上定义的低电压复位值，则单片机复位。需要经过 2~3 个 LIRC 时钟周期响应复位。此时复位后寄存器内容保持不变。

除了以上定义的四个低电压复位值外，其它值也能导致单片机复位。需要经过 2~3 个 LIRC 时钟周期响应复位。但此时复位后寄存器内容恢复到 POR 值。

• CTRL 寄存器

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

“x”：未知

Bit 7 **FSYSON**: IDLE 模式下 f_{sys} 控制位

详见其它章节

Bit 6~3 未使用，读为“0”

Bit 2 **LVRF**: LVR 复位标志位

0: 未发生

1: 发生

当特定的低电压复位条件发生时，该位被置为“1”。该位只能由应用程序清零。

Bit 1 **LRF**: LVRC 控制的复位标志位

0: 未发生

1: 发生

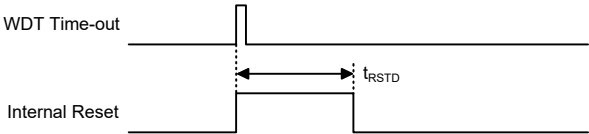
如果 LVRC 寄存器包含任何非定义的 LVR 电压值，该位被置为“1”，这类似于软件复位功能。该位只能由应用程序清零。

Bit 0 **WRF**: WDTC 控制的复位标志位

详见其它章节

正常运行时看门狗溢出复位

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

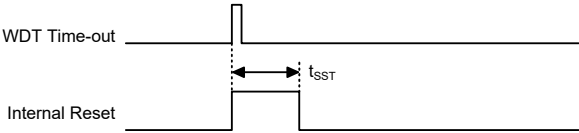


注：t_{RSTD} 为上电延迟时间，典型值为 16.7ms。

正常运行时看门狗溢出时序图

休眠或空闲时看门狗溢出复位

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



注：如果系统时钟为 HIRC，t_{SST} 为 1024 个时钟周期。
如果系统时钟为 HXT 或者 LXT，t_{SST} 为 1024 个时钟周期。
如果系统时钟为 LIRC，则 t_{SST} 为 1~2 个时钟周期。

休眠或空闲时看门狗溢出复位时序图

WDTC 寄存器软件复位

当 WDTC 寄存器的高五位为除“10101”或“01010”外其它值时，将发生 WDTC 软件复位。这时 CTRL 寄存器中的 WRF 位会被置位，表示 WDTC 软件复位已经发生。

• 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 软件控制位
10101: 除能
01010: 使能 (默认)
其它: MCU 复位

Bit 2~0 **WS2~WS0:** WDT 溢出周期选择位
详见其它章节

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，由休眠或空闲功能或看门狗计数器等几种控制器操作控制。复位标志位如下所示：

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 $\overline{\text{RES}}$ 、LVR 或 USB 复位
1	u	正常模式或低速模式时的 WDT 溢出复位
1	1	休眠或空闲模式时的 WDT 溢出复位

注：“u”代表不改变

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

项目	复位后情况
程序计数器	清除为零
中断	所有中断被除能
看门狗定时器	WDT 清除并重新计时
定时器模块	所有定时器模块停止
输入 / 输出	I/O 口设为输入模式，AN0~ANn 作为 A/D 输入脚
堆栈指针	堆栈指针指向堆栈顶端

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

HT66FB540

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu	--uu uuuu	--uu uuuu
BP	---- --00	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00
SMOD	0000 0011	0000 0011	0000 0011	0000 0011	uuuu uuuu	0000 0011	0000 0011
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
LVDC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu	--00 -000	--00 -000
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu	-000 0000	-000 0000
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC3	-000 -000	-000 -000	-000 -000	-000 -000	-uuu -uuu	-000 -000	-000 -000
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI2	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-u-u -u-u	-0-0 -0-0	-0-0 -0-0
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PB	-111 1111	-111 1111	-111 1111	-111 1111	-uuu uuuu	-111 1111	-111 1111
PBC	-111 1111	-111 1111	-111 1111	-111 1111	-uuu uuuu	-111 1111	-111 1111
PD	--11 1111	--11 1111	--11 1111	--11 1111	--uu uuuu	--11 1111	--11 1111
PDC	--11 1111	--11 1111	--11 1111	--11 1111	--uu uuuu	--11 1111	--11 1111

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	$\overline{\text{RES}}/\text{LVRC}$ 软件复位 (正常模式)	$\overline{\text{RES}}$ 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
PE	---1 1101	---1 1101	---1 1101	---1 1101	---u uuuu	---1 1101	---1 1101
PEC	---1 1111	---1 1111	---1 1111	---1 1111	---u uuuu	---1 1111	---1 1111
ADRL (ADRFs=0)	xxxx ----	xxxx ----	xxxx ----	xxxx ----	uuuu ----	xxxx ----	xxxx ----
ADRL (ADRFs=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=1)	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
ADCR0	0110 0000	0110 0000	0110 0000	0110 0000	uuuu uuuu	0110 0000	0110 0000
ADCR1	00-0 -000	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu	00-0 -000	00-0 -000
ACER0	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu	0101 0011	0101 0011
TBC	0011 0111	0011 0111	0011 0111	0011 0111	uuuu uuuu	0011 0111	0011 0111
FRCR	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u	---0 ---0	---0 ---0
FCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
FARL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FARH	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
FD0L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD0H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
I2CTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SIMC0	1110 000-	1110 000-	1110 000-	1110 000-	uuuu uu-	1110 000-	1110 000-
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu	1000 0001	1000 0001
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SIMA/ SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SPIAC0	111- --0-	111- --0-	111- --0-	111- --0-	uuu- --u-	111- --0-	111- --0-
SPIAC1	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu	--00 0000	--00 0000
SPIAD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SBSC	0-00 ---0	0-00 ---0	0-00 ---0	0-00 ---0	u-uu ---u	0-00 ---0	0-00 ---0
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PADIR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
PXPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PSLEW	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
CP0C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
CP1C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
TMPC0	--01 --01	--01 --01	--01 --01	--01 --01	--uu --uu	--01 --01	--01 --01
TMPC1	--0- --0-	--0- --0-	--0- --0-	--0- --0-	--u- --u-	--0- --0-	--0- --0-
TM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---	0000 0---	0000 0---
TM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM2C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM2AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM3C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM3AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
USB_STAT	11xx 000x	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux
UINT	---- 0000	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
USC	1000 xxxx	uuuu xuux	uuuu xuux	uuuu xuux	uuuu xuux	uuuu 0100	uuuu 0100
USR	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
UCC	000x 0-xx	uuuu u-uu	uuuu u-uu	uuuu u-uu	uuuu u-uu	uuu0 u-00	uuu0 u-00
AWR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	$\overline{\text{RES}}$ /LVRC 软件复位 (正常模式)	$\overline{\text{RES}}$ 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
STLI	---- XXXX	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- 0000	---- 0000
STLO	---- xxx-	---- uuu-	---- uuu-	---- uuu-	---- uuu-	---- 000-	---- 000-
SIES	xx-x xxxx	uu-x xuuu	uu-x xuuu	uu-x xuuu	uu-x xuuu	00-0 0000	00-0 0000
MISC	xxx- xxxx	xxu- uuxx	xxu- uuxx	xxu- uuxx	xxu- uuxx	000- 0000	000- 0000
UFIEN	---- 0000	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
FIFO0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO2	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO3	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
UFOEN	---- 0000	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
UFC0	0000 00--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--
PAPS0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPS1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SYSC	000- -0--	000- -0--	000- -0--	000- -0--	uuu- -u--	000- -0--	000- -0--
CTRL	0--- -x00	0--- -x00	0--- -x00	0--- -x00	u--- -xuu	0--- -x00	0--- -x00
LVRC	0101 0101	0101 0101	0101 0101	0101 0101	uuuu uuuu	0101 0101	0101 0101

注：“*”表示热复位
“-”表示未定义
“u”表示不改变
“x”表示未知

HT66FB542

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	$\overline{\text{RES}}$ /LVRC 软件复位 (正常模式)	$\overline{\text{RES}}$ 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu	--uu uuuu	--uu uuuu
BP	---- --0	---- --0	---- --0	---- --0	---- --u	---- --0	---- --0
SMOD	0000 0011	0000 0011	0000 0011	0000 0011	uuuu uuuu	0000 0011	0000 0011
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
LVDC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu	--00 -000	--00 -000
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu	-000 0000	-000 0000
INTC1	00-0 00-0	00-0 00-0	00-0 00-0	00-0 00-0	uu-u uu-u	00-0 00-0	00-0 00-0

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
INTC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC3	-0-- -0--	-0-- -0--	-0-- -0--	-0-- -0--	-u-- -u--	-0-- -0--	-0-- -0--
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI2	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-u-u -u-u	-0-0 -0-0	-0-0 -0-0
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PB	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu	---- 1111	---- 1111
PBC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu	---- 1111	---- 1111
PD	---1 -111	---1 -111	---1 -111	---1 -111	---u -uuu	---1 -111	---1 -111
PDC	---1 -111	---1 -111	---1 -111	---1 -111	---u -uuu	---1 -111	---1 -111
PE	---- -01	---- -01	---- -01	---- -01	---- -uu	---- -01	---- -01
PEC	---- -11	---- -11	---- -11	---- -11	---- -uu	---- -11	---- -11
ADRL (ADRFs=0)	xxxx ----	xxxx ----	xxxx ----	xxxx ----	uuuu ----	xxxx ----	xxxx ----
ADRL (ADRFs=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=1)	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
ADCR0	0110 --00	0110 --00	0110 --00	0110 --00	uuuu --uu	0110 --00	0110 --00
ADCR1	00-0 -000	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu	00-0 -000	00-0 -000
ACER0	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu	---- 1111	---- 1111
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu	0101 0011	0101 0011
FRCR	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u	---0 ---0	---0 ---0
FCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
FARL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FARH	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
FD0L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD0H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
I2CTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SIMC0	1110 000-	1110 000-	1110 000-	1110 000-	uuuu uu-	1110 000-	1110 000-
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu	1000 0001	1000 0001

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SIMA/ SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SPIAC0	111- --0-	111- --0-	111- --0-	111- --0-	uuu- --u-	111- --0-	111- --0-
SPIAC1	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu	--00 0000	--00 0000
SPIAD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SBSC	0-00 ---0	0-00 ---0	0-00 ---0	0-00 ---0	u-uu ---u	0-00 ---0	0-00 ---0
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PADIR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXPU	-000 ---0	-000 ---0	-000 ---0	-000 ---0	-uuu ---u	-000 ---0	-000 ---0
PAOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PSLEW	00-- 0000	00-- 0000	00-- 0000	00-- 0000	uu-- uuuu	00-- 0000	00-- 0000
PXWU	-000 ---0	-000 ---0	-000 ---0	-000 ---0	-uuu ---u	-000 ---0	-000 ---0
PXOI	-000 ---0	-000 ---0	-000 ---0	-000 ---0	-uuu ---u	-000 ---0	-000 ---0
CP0C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
TMPC0	---1 ---1	---1 ---1	---1 ---1	---1 ---1	---u ---u	---1 ---1	---1 ---1
TMPC1	--0- --0-	--0- --0-	--0- --0-	--0- --0-	--u- --u-	--0- --0-	--0- --0-
TM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---	0000 0---	0000 0---
TM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0RP	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM2C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM2AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM3C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
TM3DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
TM3AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3AH	---- --00	---- --00	---- --00	---- --00	---- --uu	---- --00	---- --00
USB_STAT	11xx 000-	11xx 000-	11xx 000-	11xx 000-	11xx 000-	11xx 000-	11xx 000-
UINT	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
USC	1000 0000	uuuu xuux	1000 0000	1000 0000	uuuu xuux	1uuu 0100	1uuu 0100
USR	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
UCC	0000 0-00	uuuu u-uu	0000 0-00	0000 0-00	uuuu u-uu	0uu0 u-00	0uu0 u-00
AWR	0000 0000	uuuu uuuu	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
STLI	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
STLO	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
SIES	00-0 0000	uu-x xuuu	00-0 0000	00-0 0000	uu-x xuuu	00-0 0000	00-0 0000
MISC	000- 0000	xxu- uuuu	000- 0000	000- 0000	xxu- uuuu	000- 0000	000- 0000
UFIE	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
FIFO0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO2	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO3	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
UFOEN	---- 0000	---- uuuu	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
UFC0	0000 00--	uuuu uu--	0000 00--	0000 00--	uuuu uu--	0000 00--	0000 00--
PAPS0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPS1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SYSC	000- -0--	000- -0--	000- -0--	000- -0--	uuu- -u--	000- -0--	000- -0--
CTRL	0--- -x00	0--- -x00	0--- -x00	0--- -x00	u--- -xuu	0--- -x00	0--- -x00
LVRC	0101 0101	0101 0101	0101 0101	0101 0101	uuuu uuuu	0101 0101	0101 0101

注：“*”表示热复位
“-”表示未定义
“u”表示不改变
“x”表示未知

HT66FB550

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---x xxxx	---u uuuu	---u uuuu	---u uuuu	---u uuuu	---u uuuu	---u uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu	--uu uuuu	--uu uuuu
BP	---- -000	---- -000	---- -000	---- -000	---- -000	---- -uuu	---- -000
SMOD	0000 0011	0000 0011	0000 0011	0000 0011	uuuu uuuu	0000 0011	0000 0011
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
LVDC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu	--00 -000	--00 -000
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu	-000 0000	-000 0000
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC3	-000 -000	-000 -000	-000 -000	-000 -000	-uuu -uuu	-000 -000	-000 -000
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI2	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-u-u -u-u	-0-0 -0-0	-0-0 -0-0
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PCC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PE	--11 1101	--11 1101	--11 1101	--11 1101	--uu uuuu	--11 1101	--11 1101
PEC	--11 1111	--11 1111	--11 1111	--11 1111	--uu uuuu	--11 1111	--11 1111
ADRL (ADRFs=0)	xxxx ----	xxxx ----	xxxx ----	xxxx ----	uuuu ----	xxxx ----	xxxx ----
ADRL (ADRFs=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=1)	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
ADCR0	0110 0000	0110 0000	0110 0000	0110 0000	uuuu uuuu	0110 0000	0110 0000
ADCR1	00-0 -000	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu	00-0 -000	00-0 -000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
ACER0	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
ACER1	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu	0101 0011	0101 0011
TBC	0011 0111	0011 0111	0011 0111	0011 0111	uuuu uuuu	0011 0111	0011 0111
FRRCR	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u	---0 ---0	---0 ---0
FCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
FARL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FARH	---x xxxx	---x xxxx	---x xxxx	---x xxxx	---u uuuu	---x xxxx	---x xxxx
FD0L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD0H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
I2CTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SIMC0	1110 000-	1110 000-	1110 000-	1110 000-	uuuu uu--	1110 000-	1110 000-
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu	1000 0001	1000 0001
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SIMA/ SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SPIAC0	111- --0-	111- --0-	111- --0-	111- --0-	uuu- --u-	111- --0-	111- --0-
SPIAC1	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu	--00 0000	--00 0000
SPIAD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SBSC	0-00 ---0	0-00 ---0	0-00 ---0	0-00 ---0	u-uu ---u	0-00 ---0	0-00 ---0
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PADIR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PSLEW	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
CP0C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
CP1C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
TMPC0	--01 --01	--01 --01	--01 --01	--01 --01	--uu --uu	--01 --01	--01 --01
TMPC1	--01 --01	--01 --01	--01 --01	--01 --01	--uu --uu	--01 --01	--01 --01
TM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---	0000 0---	0000 0---

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	$\overline{\text{RES}}/\text{LVRC}$ 软件复位 (正常模式)	$\overline{\text{RES}}$ 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
TM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
TM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1AH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
TM2C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
TM2AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2AH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
TM3C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
TM3AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3AH	---- -00	---- -00	---- -00	---- -00	---- -uu	---- -00	---- -00
USB_STAT	11xx 000x	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux
UINT	--00 0000	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
USC	1000 xxxx	uuuu xuux	uuuu xuux	uuuu xuux	uuuu xuux	uuuu 0100	uuuu 0100
USR	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
UCC	000x 0xxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuu0 u000	uuu0 u000
AWR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000
STLI	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--00 0000	--00 0000
STLO	--xx xxx-	--uu uuu-	--uu uuu-	--uu uuu-	--uu uuu-	--00 000-	--00 000-
SIES	xx-x xxxx	uu-x xuuu	uu-x xuuu	uu-x xuuu	uu-x xuuu	00-0 0000	00-0 0000
MISC	xxxx xxxx	xxuu uuux	xxuu uuux	xxuu uuux	xxuu uuux	0000 0000	0000 0000
UFIEN	--00 0000	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
FIFO0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO2	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO3	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO4	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
FIFO5	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
UFOEN	--00 0000	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
UFC0	0000 00--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--
UFC1	---- 0000	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
PAPS0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPS1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SYSC	000- -0--	000- -0--	000- -0--	000- -0--	uuu- -u--	000- -0--	000- -0--
CTRL	0--- -x00	0--- -x00	0--- -x00	0--- -x00	u--- -xuu	0--- -x00	0--- -x00
LVRC	0101 0101	0101 0101	0101 0101	0101 0101	uuuu uuuu	0101 0101	0101 0101

注：“*”表示热复位
“-”表示未定义
“u”表示不改变
“x”表示未知

HT66FB560

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu	--uu uuuu	--uu uuuu
BP	--0- -000	--0- -000	--0- -000	--0- -000	--0- -000	--u- -uuu	--0- -000
SMOD	0000 0011	0000 0011	0000 0011	0000 0011	uuuu uuuu	0000 0011	0000 0011
INTEG	---- 0000	---- 0000	---- 0000	---- 0000	---- uuuu	---- 0000	---- 0000
LVDC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu	--00 -000	--00 -000
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu	-000 0000	-000 0000
INTC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
INTC3	-000 -000	-000 -000	-000 -000	-000 -000	-uuu -uuu	-000 -000	-000 -000
MFI0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
MFI1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
MFI2	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-u-u -u-u	-0-0 -0-0	-0-0 -0-0
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PCC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PE	--11 1101	--11 1101	--11 1101	--11 1101	--uu uuuu	--11 1101	--11 1101
PEC	--11 1111	--11 1111	--11 1111	--11 1111	--uu uuuu	--11 1111	--11 1111
PF	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PFC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
ADRL (ADRFs=0)	xxxx ----	xxxx ----	xxxx ----	xxxx ----	uuuu ----	xxxx ----	xxxx ----
ADRL (ADRFs=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
ADRH (ADRFs=1)	---- xxxx	---- xxxx	---- xxxx	---- xxxx	---- uuuu	---- xxxx	---- xxxx
ADCR0	0110 0000	0110 0000	0110 0000	0110 0000	uuuu uuuu	0110 0000	0110 0000
ADCR1	00-0 -000	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu	00-0 -000	00-0 -000
ACER0	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
ACER1	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
WDTC	0101 0011	0101 0011	0101 0011	0101 0011	uuuu uuuu	0101 0011	0101 0011
TBC	0011 0111	0011 0111	0011 0111	0011 0111	uuuu uuuu	0011 0111	0011 0111
FRCR	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u	---0 ---0	---0 ---0
FCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
FARL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FARH	--xx xxxx	--xx xxxx	--xx xxxx	--xx xxxx	--uu uuuu	--xx xxxx	--xx xxxx
FD0L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD0H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD2L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
FD2H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
FD3H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
I2CTOC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SIMC0	1110 000-	1110 000-	1110 000-	1110 000-	uuuu uu-	1110 000-	1110 000-
SIMC1	1000 0001	1000 0001	1000 0001	1000 0001	uuuu uuuu	1000 0001	1000 0001
SIMD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SIMA/SIMC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SPIAC0	111- --0-	111- --0-	111- --0-	111- --0-	uuu- --u-	111- --0-	111- --0-
SPIAC1	--00 0000	--00 0000	--00 0000	--00 0000	--uu uuuu	--00 0000	--00 0000
SPIAD	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
SBSC	0-00 ---0	0-00 ---0	0-00 ---0	0-00 ---0	u-uu ---u	0-00 ---0	0-00 ---0
PAWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PADIR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PSLEW	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PXOI	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PFPU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PFWU	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
CP0C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
CP1C	1000 0--1	1000 0--1	1000 0--1	1000 0--1	uuuu u--u	1000 0--1	1000 0--1
TMPC0	--01 --01	--01 --01	--01 --01	--01 --01	--uu --uu	--01 --01	--01 --01
TMPC1	--01 --01	--01 --01	--01 --01	--01 --01	--uu --uu	--01 --01	--01 --01
TM0C0	0000 0---	0000 0---	0000 0---	0000 0---	uuuu u---	0000 0---	0000 0---
TM0C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0DH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM0AH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	$\overline{\text{RES}}/\text{LVRC}$ 软件复位 (正常模式)	$\overline{\text{RES}}$ 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
TM1C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1DH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
TM1AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM1AH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
TM2C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2DH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
TM2AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM2AH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
TM3C0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3C1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3DH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
TM3AL	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
TM3AH	---- -00	---- -00	---- -00	---- -00	---- --uu	---- -00	---- -00
USB_STAT	11xx 000x	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux	uuxx uuux
UINT	0000 0000	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
USC	1000 xxxx	uuuu xuux	uuuu xuux	uuuu xuux	uuuu xuux	uuuu 0100	uuuu 0100
USR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
UCC	000x 0xxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuu0 u000	uuu0 u000
AWR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000
STLI	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000
STLO	xxxx xxx-	uuuu uu-	uuuu uu-	uuuu uu-	uuuu uu-	0000 000-	0000 000-
SIES	xx-x xxxx	uu-x xuuu	uu-x xuuu	uu-x xuuu	uu-x xuuu	00-0 0000	00-0 0000
MISC	xxxx xxxx	xxuu uuxx	xxuu uuxx	xxuu uuxx	xxuu uuxx	0000 0000	0000 0000
UFIE	0000 0000	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
FIFO0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO2	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO3	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx
FIFO4	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx

寄存器	上电复位	WDT 溢出 /WDTC 软件复位 (正常模式)	RES/LVRC 软件复位 (正常模式)	RES 复位 (HALT)	WDT 溢出 (HALT)*	USB 复位 (正常模式)	USB 复位 (HALT)
FIFO5	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX
FIFO6	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX
FIFO7	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX	XXXX XXXX
UFOEN	0000 0000	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
UFC0	0000 00--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--	uuuu uu--
UFC1	0000 0000	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PAPS0	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
PAPS1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
SYSC	000- -0--	000- -0--	000- -0--	000- -0--	uuu- -u--	000- -0--	000- -0--
CTRL	0--- -x00	0--- -x00	0--- -x00	0--- -x00	u--- -xuu	0--- -x00	0--- -x00
LVRC	0101 0101	0101 0101	0101 0101	0101 0101	uuuu uuuu	0101 0101	0101 0101

注：“*”表示热复位
“-”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。每一个引脚都在用户程序控制下都可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~PF 双向输入 / 输出口。这些寄存器在数据存储器有特定的地址。所有 I/O 口用于输入输出操作。作为输入操作，输入引脚无锁存功能，也就是说输出数据必须在执行“MOV A, [m]”，T2 的上升沿准备好，m 为端口地址。对于输出操作，所有数据都是被锁存的，且保持不变直到输出锁存被重写。

寄存器 名称	位							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PADIR	D7	D6	D5	D4	D3	D2	D1	D0
PAOI	D7	D6	D5	D4	D3	D2	D1	D0
PSLEW	PDSLEW1	PDSLEW0	—	—	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
PXWU	PEHWU	PELWU	PDHWU	PDLWU	—	—	PBHWU	PBLWU

寄存器名称	位							
	7	6	5	4	3	2	1	0
PXPU	PEHPU	PELPU	PDHPU	PDLPU	—	—	PBHPU	PBLPU
PXOI	PEHI	PELI	PDHI	PDLI	—	—	PBHI	PBLI
PAPS0	PA3S1	PA3S0	PA2S1	PA2S0	PA1S1	PA1S0	PA0S1	PA0S0
PAPS1	PA7S1	PA7S0	PA6S1	PA6S0	PA5S1	PA5S0	PA4S1	PA4S0
PB	—	D6	D5	D4	D3	D2	D1	D0
PBC	—	D6	D5	D4	D3	D2	D1	D0
PD	—	—	D5	D4	D3	D2	D1	D0
PDC	—	—	D5	D4	D3	D2	D1	D0
PE	—	—	—	D4	D3	D2	D1	D0
PEC	—	—	—	D4	D3	D2	D1	D0

输入 / 输出寄存器列表 – HT66FB540

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PADIR	D7	D6	D5	D4	D3	D2	D1	D0
PAOI	D7	D6	D5	D4	D3	D2	D1	D0
PSLEW	PDSLEW1	PDSLEW0	—	—	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
PXWU	—	PELWU	PDHWU	PDLWU	—	—	—	PBLWU
PXPU	—	PELPU	PDHPU	PDLPU	—	—	—	PBLPU
PXOI	—	PELI	PDHI	PDLI	—	—	—	PBLI
PAPS0	PA3S1	PA3S0	PA2S1	PA2S0	PA1S1	PA1S0	PA0S1	PA0S0
PAPS1	PA7S1	PA7S0	PA6S1	PA6S0	PA5S1	PA5S0	PA4S1	PA4S0
PB	—	—	—	—	D3	D2	D1	D0
PBC	—	—	—	—	D3	D2	D1	D0
PD	—	—	—	D4	—	D2	D1	D0
PDC	—	—	—	D4	—	D2	D1	D0
PE	—	—	—	—	—	—	D1	D0
PEC	—	—	—	—	—	—	D1	D0

输入 / 输出寄存器列表 – HT66FB542

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PADIR	D7	D6	D5	D4	D3	D2	D1	D0
PAOI	D7	D6	D5	D4	D3	D2	D1	D0
PSLEW	PDSLEW1	PDSLEW0	PCSLEW1	PCSLEW0	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
PXWU	PEHWU	PELWU	PDHWU	PDLWU	PCHWU	PCLWU	PBHWU	PBLWU
PXPU	PEHPU	PELPU	PDHPU	PDLPU	PCHPU	PCLPU	PBHPU	PBLPU
PXOI	PEHI	PELI	PDHI	PDLI	PCHI	PCLI	PBHI	PBLI
PAPS0	PA3S1	PA3S0	PA2S1	PA2S0	PA1S1	PA1S0	PA0S1	PA0S0
PAPS1	PA7S1	PA7S0	PA6S1	PA6S0	PA5S1	PA5S0	PA4S1	PA4S0
PB	D7	D6	D5	D4	D3	D2	D1	D0
PBC	D7	D6	D5	D4	D3	D2	D1	D0
PC	D7	D6	D5	D4	D3	D2	D1	D0
PCC	D7	D6	D5	D4	D3	D2	D1	D0
PD	D7	D6	D5	D4	D3	D2	D1	D0
PDC	D7	D6	D5	D4	D3	D2	D1	D0
PE	—	—	D5	D4	D3	D2	D1	D0
PEC	—	—	D5	D4	D3	D2	D1	D0

输入 / 输出寄存器列表 – HT66FB550

寄存器名称	位							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PADIR	D7	D6	D5	D4	D3	D2	D1	D0
PAOI	D7	D6	D5	D4	D3	D2	D1	D0
PSLEW	PDSLEW1	PDSLEW0	PCSLEW1	PCSLEW0	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
PXWU	PEHWU	PELWU	PDHWU	PDLWU	PCHWU	PCLWU	PBHWU	PBLWU
PXPU	PEHPU	PELPU	PDHPU	PDLPU	PCHPU	PCLPU	PBHPU	PBLPU
PXOI	PEHI	PELI	PDHI	PDLI	PCHI	PCLI	PBHI	PBLI
PAPS0	PA3S1	PA3S0	PA2S1	PA2S0	PA1S1	PA1S0	PA0S1	PA0S0
PAPS1	PA7S1	PA7S0	PA6S1	PA6S0	PA5S1	PA5S0	PA4S1	PA4S0
PB	D7	D6	D5	D4	D3	D2	D1	D0
PBC	D7	D6	D5	D4	D3	D2	D1	D0
PC	D7	D6	D5	D4	D3	D2	D1	D0
PCC	D7	D6	D5	D4	D3	D2	D1	D0
PD	D7	D6	D5	D4	D3	D2	D1	D0
PDC	D7	D6	D5	D4	D3	D2	D1	D0

寄存器名称	位							
	7	6	5	4	3	2	1	0
PE	—	—	D5	D4	D3	D2	D1	D0
PEC	—	—	D5	D4	D3	D2	D1	D0
PFWU	D7	D6	D5	D4	D3	D2	D1	D0
PFPU	D7	D6	D5	D4	D3	D2	D1	D0
PF	D7	D6	D5	D4	D3	D2	D1	D0
PFC	D7	D6	D5	D4	D3	D2	D1	D0

输入 / 输出寄存器列表 – HT66FB560

上拉电阻

许多产品应用在端口处于输入状态时需要外加一个上拉电阻来实现上拉的功能。为了免去外部上拉电阻，当引脚规划为输入时，可由内部连接到一个上拉电阻，这些上拉电阻可通过寄存器 PAPU、PXPUP 和 PFPU 来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。注意，PA 和 PF 的上拉电阻由 PAPU 和 PFPU 寄存器按位控制，而 PB、PC、PD 和 PE 的上拉电阻由 PXPUP 寄存器按半字节控制。

• PAPU 寄存器

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

Bit 7~0 **PAPU**: PA 口 bit 7~bit 0 上拉电阻控制位
0: 除能
1: 使能

• PXPUP 寄存器 – HT66FB540

Bit	7	6	5	4	3	2	1	0
Name	PEHPU	PELPU	PDHPU	PDLPU	—	—	PBHPU	PBLPU
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

Bit 7 **PEHPU**: PE4 上拉电阻控制位
0: 除能
1: 使能

Bit 6 **PELPU**: PE3、PE2 和 PE0 上拉电阻控制位
0: 除能
1: 使能
注意，PE1 引脚没有上拉电阻功能。

Bit 5 **PDHPU**: PD7~PD4 上拉电阻控制位
0: 除能
1: 使能

Bit 4 **PDLPU**: PD3~PD0 上拉电阻控制位
0: 除能
1: 使能

Bit 3~2 未使用，读为“0”

- Bit 1 **PBHPU**: PB6~PB4 上拉电阻控制位
0: 除能
1: 使能
- Bit 0 **PBLPU**: PB3~PB0 上拉电阻控制位
0: 除能
1: 使能

● **PXPU 寄存器 – HT66FB542**

Bit	7	6	5	4	3	2	1	0
Name	—	PELPU	PDHPU	PDLPU	—	—	—	PBLPU
R/W	—	R/W	R/W	R/W	—	—	—	R/W
POR	—	0	0	0	—	—	—	0

- Bit 7 未使用，读为“0”
- Bit 6 **PELPU**: PE0 上拉电阻控制位
0: 除能
1: 使能
注意，PE1 引脚没有上拉电阻功能。
- Bit 5 **PDHPU**: PD4 上拉电阻控制位
0: 除能
1: 使能
- Bit 4 **PDLPU**: PD2~PD0 上拉电阻控制位
0: 除能
1: 使能
- Bit 3~1 未使用，读为“0”
- Bit 0 **PBLPU**: PB3~PB0 上拉电阻控制位
0: 除能
1: 使能

● **PXPU 寄存器 – HT66FB550/HT66FB560**

Bit	7	6	5	4	3	2	1	0
Name	PEHPU	PELPU	PDHPU	PDLPU	PCHPU	PCLPU	PBHPU	PBLPU
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 **PEHPU**: PE5~PE4 上拉电阻控制位
0: 除能
1: 使能
- Bit 6 **PELPU**: PE3、PE2 和 PE0 上拉电阻控制位
0: 除能
1: 使能
注意，PE1 引脚没有上拉电阻功能。
- Bit 5 **PDHPU**: PD7~PD4 上拉电阻控制位
0: 除能
1: 使能
- Bit 4 **PDLPU**: PD3~PD0 上拉电阻控制位
0: 除能
1: 使能
- Bit 3 **PCHPU**: PC7~PC4 上拉电阻控制位
0: 除能
1: 使能

- Bit 2 **PCLPU**: PC3~PC0 上拉电阻控制位
0: 除能
1: 使能
- Bit 1 **PBHPU**: PB7~PB4 上拉电阻控制位
0: 除能
1: 使能
- Bit 0 **PBLPU**: PB3~PB0 上拉电阻控制位
0: 除能
1: 使能

● **PFPU 寄存器 – HT66FB560**

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

- Bit 7~0 **PFPU**: PF 口 bit 7~bit 0 上拉电阻控制位
0: 除能
1: 使能

I/O 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状态，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~PF 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。注意，PA~PF 是可以通过设置 PAWU、PXWU 和 PFWU 寄存器来按位或按半字节选择是否具有唤醒功能。

● **PAWU 寄存器**

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

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

● **PXWU 寄存器 – HT66FB540**

Bit	7	6	5	4	3	2	1	0
Name	PEHWU	PELWU	PDHWU	PDLWU	—	—	PBHWU	PBLWU
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

- Bit 7 **PEHWU**: PE4 唤醒功能控制位
0: 除能
1: 使能
- Bit 6 **PELWU**: PE3、PE2 和 PE0 唤醒功能控制位
0: 除能
1: 使能
注意，PE1 引脚没有唤醒功能。

- Bit 5 **PDHWU**: PD7~PD4 唤醒功能控制位
0: 除能
1: 使能
- Bit 4 **PDLWU**: PD3~PD0 唤醒功能控制位
0: 除能
1: 使能
- Bit 3~2 未使用, 读为 “0”
- Bit 1 **PBHWU**: PB6~PB4 唤醒功能控制位
0: 除能
1: 使能
- Bit 0 **PBLWU**: PB3~PB0 唤醒功能控制位
0: 除能
1: 使能

● **PXWU 寄存器 – HT66FB542**

Bit	7	6	5	4	3	2	1	0
Name	—	PELWU	PDHWU	PDLWU	—	—	—	PBLWU
R/W	—	R/W	R/W	R/W	—	—	—	R/W
POR	—	0	0	0	—	—	—	0

- Bit 7 未使用, 读为 “0”
- Bit 6 **PELWU**: PE0 唤醒功能控制位
0: 除能
1: 使能
注意, PE1 引脚没有唤醒功能。
- Bit 5 **PDHWU**: PD4 唤醒功能控制位
0: 除能
1: 使能
- Bit 4 **PDLWU**: PD2~PD0 唤醒功能控制位
0: 除能
1: 使能
- Bit 3~1 未使用, 读为 “0”
- Bit 0 **PBLWU**: PB3~PB0 唤醒功能控制位
0: 除能
1: 使能

● **PXWU 寄存器 – HT66FB550/HT66FB560**

Bit	7	6	5	4	3	2	1	0
Name	PEHWU	PELWU	PDHWU	PDLWU	PCHWU	PCLWU	PBHWU	PBLWU
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 **PEHWU**: PE5~PE4 唤醒功能控制位
0: 除能
1: 使能
- Bit 6 **PELWU**: PE3、PE2 和 PE0 唤醒功能控制位
0: 除能
1: 使能
注意, PE1 引脚没有唤醒功能。

Bit 5	PDHWU: PD7~PD4 唤醒功能控制位 0: 除能 1: 使能
Bit 4	PDLWU: PD3~PD0 唤醒功能控制位 0: 除能 1: 使能
Bit 3	PCHWU: PC7~PC4 唤醒功能控制位 0: 除能 1: 使能
Bit 2	PCLWU: PC3~PC0 唤醒功能控制位 0: 除能 1: 使能
Bit 1	PBHWU: PB7~PB4 唤醒功能控制位 0: 除能 1: 使能
Bit 0	PBLWU: PB3~PB0 唤醒功能控制位 0: 除能 1: 使能

● PFWU 寄存器 – HT66FB560

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	PFWU: PF 口 bit 7~bit 0 唤醒功能控制位 0: 除能 1: 使能
---------	---

PA 口唤醒极性控制寄存器

PA 口的每个引脚都可以通过设置 PADIR 寄存器来单独选择是否具有唤醒极性功能。

● PADIR 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	PADIR: PA7~PA0 唤醒边沿控制位 0: 上升沿 1: 下降沿
---------	---

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 (PAC~PFC) 用来控制输入 / 输出状态。从而每个 I/O 引脚都可以通过软件控制，动态的设置带上拉电阻或者不带上拉电阻。所有的 I/O 端口的引脚都各自对应于 I/O 端口控制的某一位。若 I/O 引脚要实现输入功能，则对应的控制寄存器的位需要设置为“1”，这时程序指令可以直接读取输入脚的逻辑状态。若控制寄存器相应的位被设定为“0”，则此引脚被设置为 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。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。

● PAC 寄存器

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

● PBC 寄存器 – HT66FB540

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	—	1	1	1	1	1	1	1

● PBC 寄存器 – HT66FB542

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

● PBC 寄存器 – HT66FB550/HT66FB560

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

● PCC 寄存器 – HT66FB550/HT66FB560

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

● PDC 寄存器 – HT66FB540

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	—	—	1	1	1	1	1	1

● PDC 寄存器 – HT66FB542

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

● PDC 寄存器 – HT66FB550/HT66FB560

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

● PEC 寄存器 – HT66FB540

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

● PEC 寄存器 – HT66FB542

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

● PEC 寄存器 – HT66FB550/HT66FB560

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	—	—	1	1	1	1	1	1

● PFC 寄存器 – HT66FB560

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

Bit 7~0 I/O 口 bit 7~bit 0 输入 / 输出控制位

0: 输出

1: 输入

注: PE1 引脚仅输入。

I/O 口输出电流控制寄存器

PA~PE 可以通过设置特定的寄存器来选择高或低驱动电流。PA 口的每个引脚都可以通过设置 PAOI 寄存器来单独选择是否具有高输出电流功能。PB~PE 需要通过设置 PXOI 寄存器按半字节选择输出电流。请注意, PF 口默认为具有低输出电流, $V_{DD}=5V$ 时, $I_{OL}=4mA$, $I_{OH}=-4mA$ 。

● PAOI 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 **PAOI**: PA7~PA0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

● PXOI 寄存器 – HT66FB540

Bit	7	6	5	4	3	2	1	0
Name	PEHI	PELI	PDHI	PDLI	—	—	PBHI	PBLI
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

Bit 7 **PEHI**: PE4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 6 **PELI**: PE3~PE0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 5 **PDHI**: PD5~PD4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 4 **PDLI**: PD3~PD0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 3~2 未使用, 读为 “0”

Bit 1 **PBHI**: PB6~PB4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 0 **PBLI**: PB3~PB0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

● PXOI 寄存器 – HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	PELI	PDHI	PDLI	—	—	—	PBLI
R/W	—	R/W	R/W	R/W	—	—	—	R/W
POR	—	0	0	0	—	—	—	0

Bit 7 未使用, 读为 “0”

Bit 6 **PELI**: PE1~PE0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 5 **PDHI**: PD4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

Bit 4 **PDLI**: PD2~PD0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
 0: I_{OL}/I_{OH} 低电流驱动
 1: I_{OL}/I_{OH} 高电流驱动

- Bit 3~1 未使用，读为“0”
- Bit 0 **PBLI**: PB3~PB0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动

● PXOI 寄存器 – HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	PEHI	PELI	PDHI	PDLI	PCHI	PCLI	PBHI	PBLI
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 **PEHI**: PE5~PE4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 6 **PELI**: PE3~PE0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 5 **PDHI**: PD7~PD4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 4 **PDLI**: PD3~PD0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 3 **PCHI**: PC7~PC4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 2 **PCLI**: PC3~PC0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 1 **PBHI**: PB7~PB4 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动
- Bit 0 **PBLI**: PB3~PB0 输出电流控制位 (典型值 @ $V_{DD}=5V$)
0: I_{OL}/I_{OH} 低电流驱动
1: I_{OL}/I_{OH} 高电流驱动

I/O 口输出摆率控制寄存器

PA~PD 可以通过设置 PSLEW 寄存器按半字节选择不同的摆率。请注意，PE 和 PF 的摆率默认固定在 200ns。

● PSLEW 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	PDSLEW1	PDSLEW0	—	—	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W
POR	0	0	—	—	0	0	0	0

- Bit 7~6 **PDSLEW1~PDSLEW0**: PD 输出摆率控制位
00: 200ns
01: 100ns
10: 50ns
11: 未定义
- Bit 5~4 未使用，读为“0”

- Bit 3~2 **PBSLEW1~PBSLEW0**: PB 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义
- Bit 1~0 **PASLEW1~PASLEW0**: PA 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义

● **PSLEW 寄存器 – HT66FB550/HT66FB560**

Bit	7	6	5	4	3	2	1	0
Name	PDSLEW1	PDSLEW0	PCSLEW1	PCSLEW0	PBSLEW1	PBSLEW0	PASLEW1	PASLEW0
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 **PDSLEW1~PDSLEW0**: PD 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义
- Bit 5~4 **PCSLEW1~PCSLEW0**: PC 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义
- Bit 3~2 **PBSLEW1~PBSLEW0**: PB 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义
- Bit 1~0 **PASLEW1~PASLEW0**: PA 输出摆率控制位
 00: 200ns
 01: 100ns
 10: 50ns
 11: 未定义

PA 口电源控制寄存器

PA 口的每个引脚都可以通过设置 PAPS0 和 PAPS1 寄存器来单独选择不同的电源。

● **PAPS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PA3S1	PA3S0	PA2S1	PA2S0	PA1S1	PA1S0	PA0S1	PA0S0
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 **PA3S1~PA3S0**: PA3 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出

- Bit 5~4 **PA2S1~PA2S0:** PA2 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出
- Bit 3~2 **PA1S1~PA1S0:** PA1 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出
- Bit 1~0 **PA0S1~PA0S0:** PA0 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出

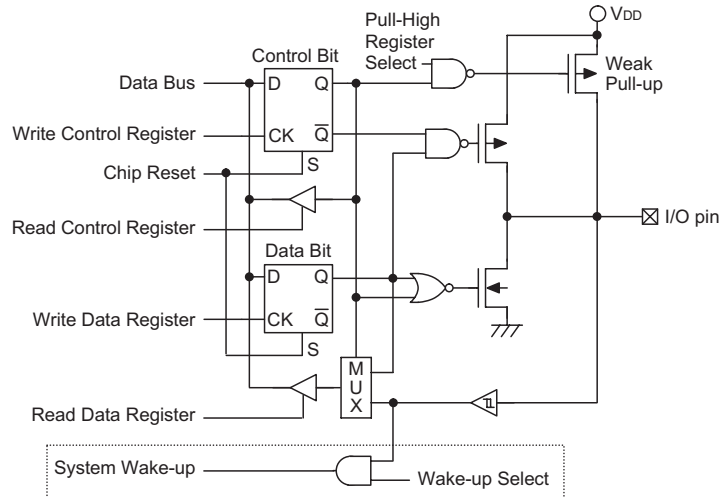
● PAPS1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PA7S1	PA7S0	PA6S1	PA6S0	PA5S1	PA5S0	PA4S1	PA4S0
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 **PA7S1~PA7S0:** PA7 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出
- Bit 5~4 **PA6S1~PA6S0:** PA6 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出
- Bit 3~2 **PA5S1~PA5S0:** PA5 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出
- Bit 1~0 **PA4S1~PA4S0:** PA4 电源控制位
 00: VDD
 01: VDD
 10: VDDIO
 11: V33O, 3.3V 稳压器输出

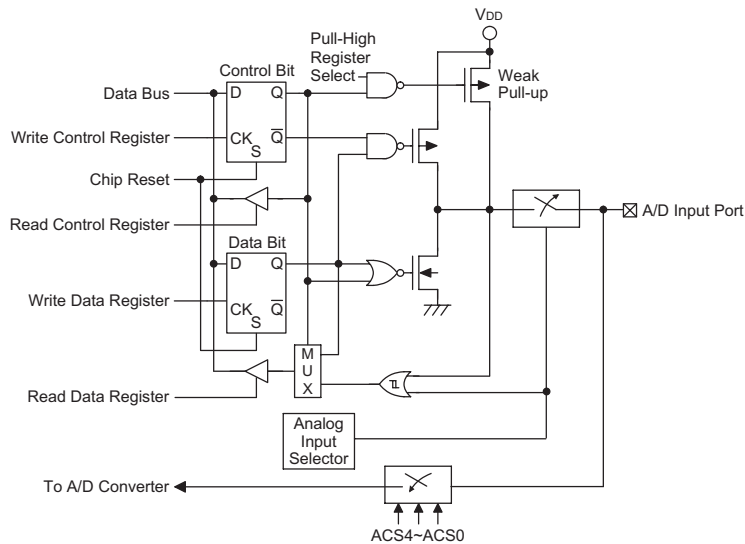
输入 / 输出引脚结构

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。



注：PE1 引脚仅输入，且无上拉电阻和唤醒功能。

通用输入 / 输出端口



A/D 输入 / 输出端口

编程注意事项

在编程中，最先要考虑的是端口的初始化。复位之后，所有的输入 / 输出数据及端口控制寄存器都将被设为逻辑高。所有输入 / 输出引脚默认为输入状态，而其电平则取决于其它相连接电路以及是否选择了上拉电阻。如果端口控制寄存器 PAC~PFC 某些引脚位被设定输出状态，这些输出引脚会有初始高电平输出，除非数据寄存器端口 PA~PF 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，或者使用指令“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 和 PF 的每个引脚以及 PB~PE 的每四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 I/O 口任一引脚电平从高到低的转换的方式，可以设置 I/O 口一个或者多个引脚具有唤醒功能。

定时器模块 – TM

控制和测量时间在任何单片机中都是一个很重要的部分。每个单片机提供几个定时器模块 (简称 TM)，来实现和时间有关的功能。定时器模块是包括多种操作的定时单元，提供的操作有：定时 / 事件计数器，捕捉输入，比较匹配输出，单脉冲输出以及 PWM 输出等功能。每个定时器模块有两个独立中断。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和标准型定时器章节。

简介

该系列单片机包含 4 个 TM，分别命名为 TM0，TM1，TM2 和 TM3。每个 TM 可被划分为一个特定的类型，即简易型 TM 或标准型 TM。虽然性质相似，但不同 TM 特性复杂度不同。本章介绍简易型和标准型 TM 的共性，更多详细资料分别见后面各章。两种类型 TM 的特性和区别见下表。

功能	CTM	STM
定时 / 计数器	√	√
捕捉输入	—	√
比较匹配输出	√	√
PWM 通道数	1	1
单脉冲输出	—	1
PWM 对齐方式	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

TM 功能概要

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

单片机	TM0	TM1	TM2	TM3
HT66FB540/HT66FB542/ HT66FB550/HT66FB560	16-bit STM	10-bit STM	10-bit CTM	10-bit CTM

TM 名称 / 类型参考

TM 操作

两种不同类型的 TM 提供从简单的定时操作到 PWM 信号产生等多种功能。理解 TM 操作的关键是比较 TM 内独立运行的计数器的值与内部比较器的预置值。当计数器的值与比较器的预置值相同时，则比较匹配，TM 中断信号产生，清零计数器并改变 TM 输出引脚的状态。用户选择内部时钟或外部时钟来驱动内部 TM 计数器。

TM 时钟源

驱动 TM 计数器的时钟源很多。通过设置 TM 控制寄存器的 TnCK2~TnCK0 位，选择所需的时钟源。该时钟源来自系统时钟 f_{sys} 或内部高速时钟 f_H 或 f_{TBC} 时钟源 (HT66FB542 为 f_L 时钟源) 或外部 TCKn 引脚时钟的分频比。注意：设置 TnCK2~TnCK0 为 101，将选择 TM 预设时钟输入，有效切断 TM 时钟源。TCKn 引脚时钟源用于允许外部信号作为 TM 时钟源或用于事件计数。

TM 中断

简易型和标准型 TM 都拥有两个内部中断，分别是内部比较器 A 或比较器 P，

当比较匹配发生时产生 TM 中断。当 TM 中断产生时，计数器清零并改变 TM 输出引脚的状态。

TM 外部引脚

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

每个 TM 有一个或两个输出引脚 TP_n。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 TP_n 输出引脚也被 TM 用来产生 PWM 输出波形。当 TM 输出引脚与其它功能共用时，TM 输出功能需要通过寄存器先被设置。寄存器中的一个单独位用于决定其相关引脚用于外部 TM 输出还是用于其它功能。每个单片机和不同类型 TM 中输出引脚的个数是不同的，详见下表。

所有 TM 输出引脚都以“_n”作后缀。引脚后缀名为“_0”或“_1”表示它们来自同一个 TM 的多个输出脚。由此该引脚会产生一对互补输出，可通过 I/O 寄存器数据位来选择。

单片机	CTM	STM	寄存器
HT66FB542	TP2_1, TP3_1	TP0_0, TP1_0	TMPC0, TMPC1
HT66FB540	TP2_1, TP3_1	TP0_0, TP0_1 TP1_0, TP1_1	
HT66FB550	TP2_0, TP3_0		
HT66FB560	TP2_1, TP3_1		

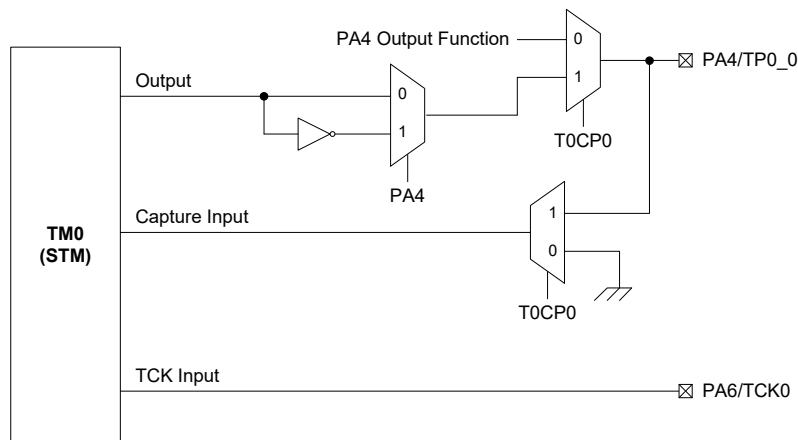
TM 输出引脚

TM 输入 / 输出引脚控制寄存器

通过设置两个与 TM 输入 / 输出引脚相关的寄存器的一位，选择作为 TM 输入 / 输出功能或其它共用功能。设定为高时，相关引脚用作 TM 输入 / 输出，清零时将保持原来的功能。

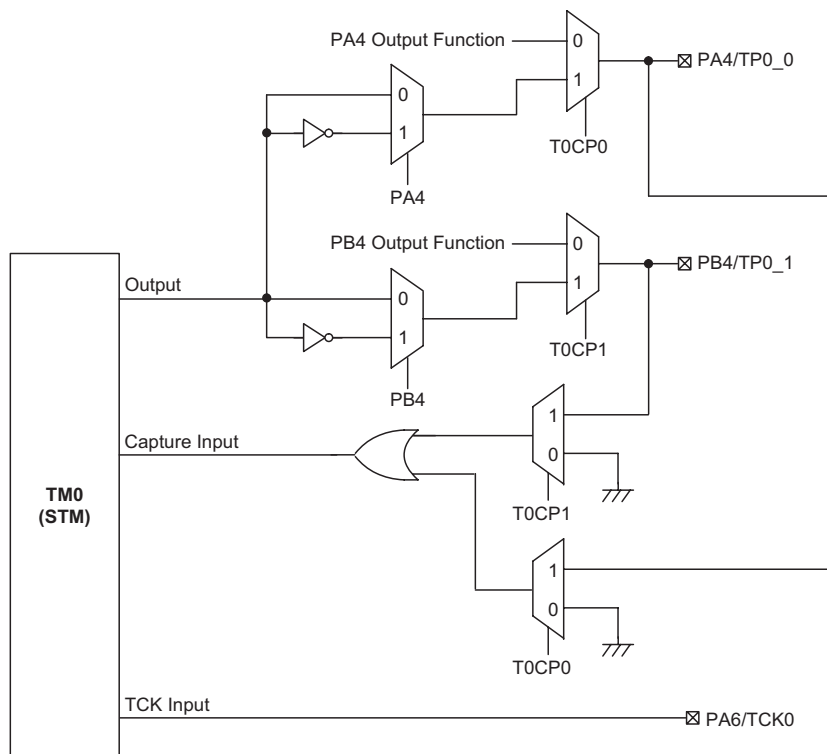
寄存器	单片机	位							
		7	6	5	4	3	2	1	0
TMPC0	HT66FB542	—	—	—	T1CP0	—	—	—	T0CP0
	HT66FB540	—	—	T1CP1	T1CP0	—	—	T0CP1	T0CP0
	HT66FB550	—	—	T1CP1	T1CP0	—	—	T0CP1	T0CP0
	HT66FB560	—	—	T1CP1	T1CP0	—	—	T0CP1	T0CP0
TMPC1	HT66FB540	—	—	T3CP1	—	—	—	T2CP1	—
	HT66FB542	—	—	T3CP1	—	—	—	T2CP1	—
	HT66FB550	—	—	T3CP1	T3CP0	—	—	T2CP1	T2CP0
	HT66FB560	—	—	T3CP1	T3CP0	—	—	T2CP1	T2CP0

TM 输入 / 输出引脚控制寄存器列表



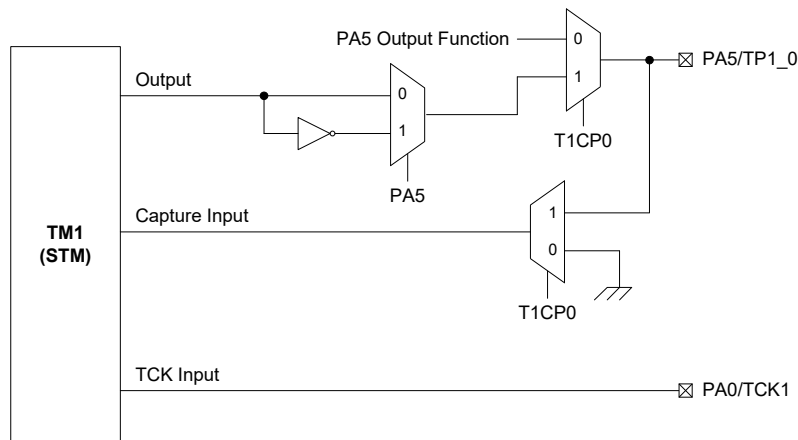
TM0 功能引脚控制方框图 – HT66FB542

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。



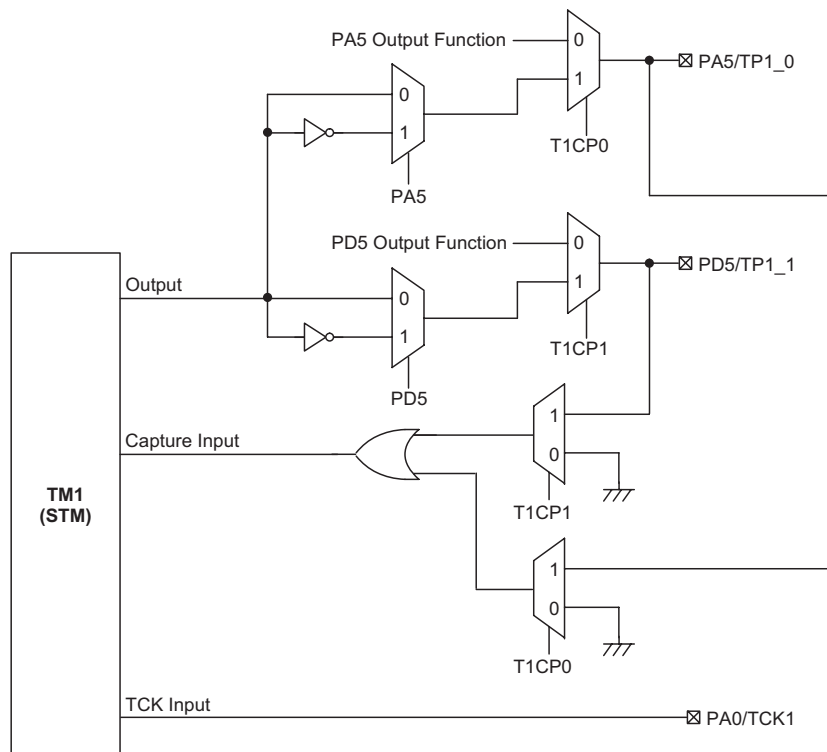
TM0 功能引脚控制方框图 – HT66FB540/HT66FB550/HT66FB560

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



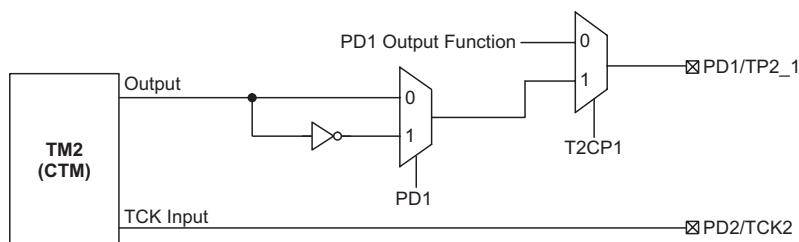
TM1 功能引脚控制方框图 – HT66FB542

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。



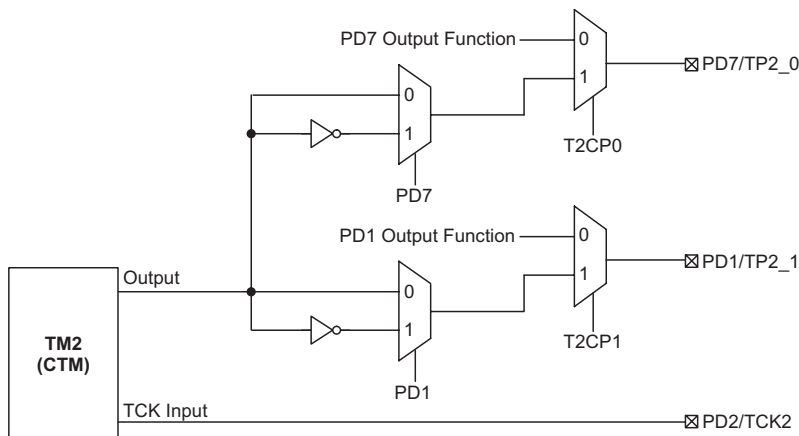
TM1 功能引脚控制方框图 – HT66FB540/HT66FB550/HT66FB560

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



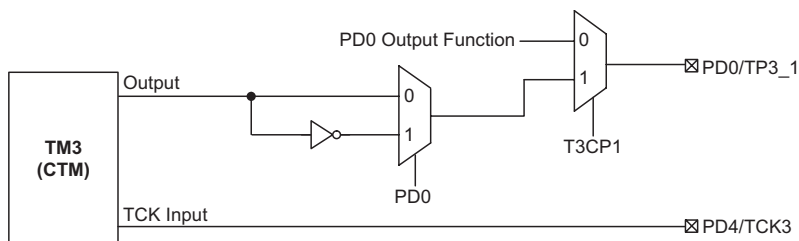
TM2 功能引脚控制方框图 – HT66FB540/HT66FB542

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。



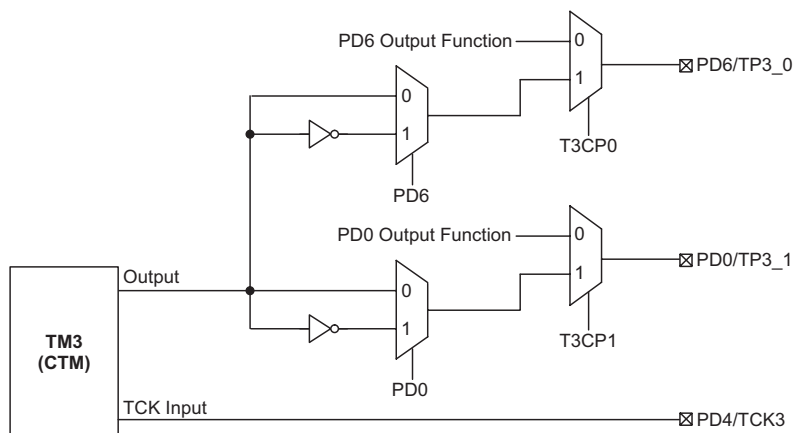
TM2 功能引脚控制方框图 – HT66FB550/HT66FB560

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。



TM3 功能引脚控制方框图 – HT66FB540/HT66FB542

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。



TM3 功能引脚控制方框图 – HT66FB550/HT66FB560

注：上图所示输入 / 输出寄存器数据位用于 TM 输出反向控制。

● TMPC0 寄存器 – HT66FB542

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

Bit 7~5 未定义，读为“0”

Bit 4 **T1CP0**: TP1_0 引脚控制位
0: 除能
1: 使能

Bit 3~1 未定义，读为“0”

Bit 0 **T0CP0**: TP0_0 引脚控制位
0: 除能
1: 使能

● TMPC0 寄存器 – HT66FB540/HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	—	—	T1CP1	T1CP0	—	—	T0CP1	T0CP0
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	1	—	—	0	1

Bit 7~6 未定义，读为“0”

Bit 5 **T1CP1**: TP1_1 引脚控制位
0: 除能
1: 使能

Bit 4 **T1CP0**: TP1_0 引脚控制位
0: 除能
1: 使能

Bit 3~2 未定义，读为“0”

Bit 1 **T0CP1**: TP0_1 引脚控制位
0: 除能
1: 使能

Bit 0 **T0CP0**: TP0_0 引脚控制位
0: 除能
1: 使能

● TMPC1 寄存器 – HT66FB540/HT66FB542

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

Bit 7~6 未定义，读为“0”

Bit 5 **T3CP1**: TP3_1 引脚控制位
0: 除能
1: 使能

Bit 4~2 未定义，读为“0”

Bit 1 **T2CP1**: TP2_1 引脚控制位
0: 除能
1: 使能

Bit 0 未定义，读为“0”

● TMPC1 寄存器 – HT66FB550/HT66FB560

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

Bit 7~6 未定义，读为“0”

Bit 5 **T3CP1**: TP3_1 引脚控制位
0: 除能
1: 使能

Bit 4 **T3CP0**: TP3_0 引脚控制位
0: 除能
1: 使能

Bit 3~2 未定义，读为“0”

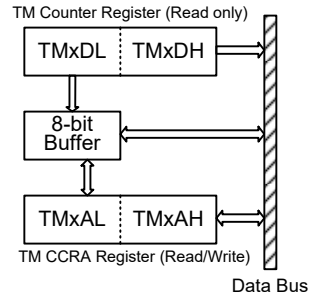
Bit 1 **T2CP1**: TP2_1 引脚控制位
0: 除能
1: 使能

Bit 0 **T2CP0**: TP2_0 引脚控制位
0: 除能
1: 使能

编程注意事项

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

CCRA 寄存器的操作如下图所示，同时因为该寄存器用上述所说的特殊的方式进行读写，因此建议使用“MOV”指令用如下方式对 CCRA 低字节寄存器 TMxAL 操作。若不用下述过程进行操作，将会导致不可预知的结果。



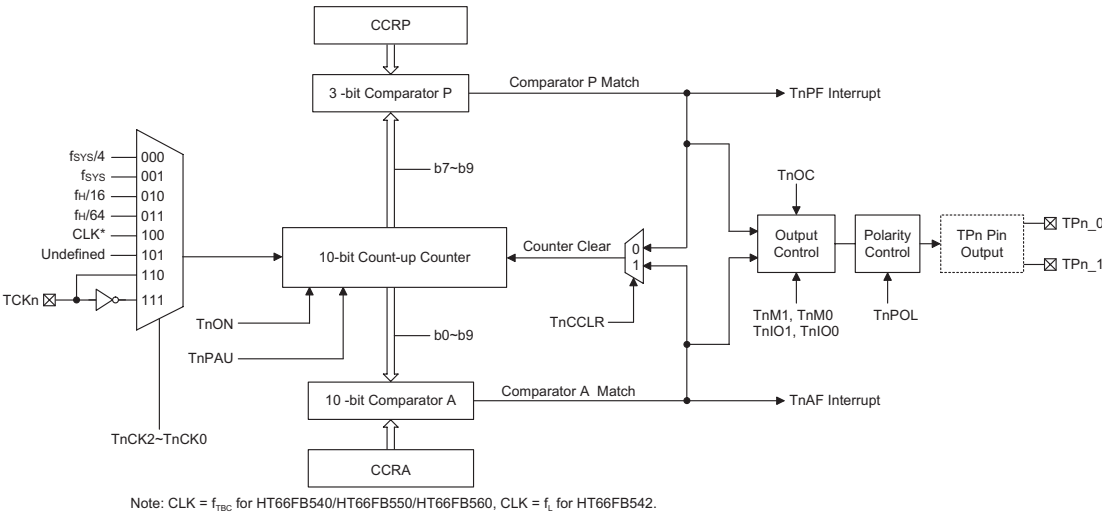
读写流程如下步骤所示：

- 写数据至 CCRA
 - ◆ 步骤 1. 写数据至低字节寄存器 TMxAL
 - 注意，此时数据仅写入 8-bit 缓存器。
 - ◆ 步骤 2. 写数据至高字节寄存器 TMxAH
 - 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 中读取数据
 - ◆ 步骤 1. 由高字节寄存器 TMxDH 或 TMxAH 读取数据
 - 注意，此时高字节寄存器中的数据直接读取，同时由低字节寄存器读取的数据锁存至 8-bit 缓存器中。
 - ◆ 步骤 2. 由低字节寄存器 TMxDL 或 TMxAL 读取数据
 - 注意，此时读取 8-bit 缓存器中的数据。

简易型 TM – CTM

虽然简易型 TM 是两种 TM 类型中最简单的形式，但仍然包括三种工作模式，即比较匹配输出，定时 / 事件计数器和 PWM 输出模式。简易型 TM 也由外部输入脚控制并驱动一个或两个外部输出脚。两个外部输出脚信号可以相同也可以相反。

CTM	名称	TM 编号	TM 输入引脚	TM 输出引脚
HT66FB540 HT66FB542	10-bit CTM	2, 3	TCK2, TCK3	TP2_1, TP3_1
HT66FB550 HT66FB560	10-bit CTM	2, 3	TCK2, TCK3	TP2_0, TP2_1; TP3_0, TP3_1



Note: CLK = f_{TBC} for HT66FB540/HT66FB550/HT66FB560, CLK = f_i for HT66FB542.

简易型 TM 方框图

简易型 TM 操作

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

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

简易型 TM 寄存器介绍

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

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

简易型 TM 寄存器列表 (n=2 或 3)

• TMnDL 寄存器

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

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

• TMnDH 寄存器

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

Bit 7~2 未定义, 读为 “0”
Bit 1~0 **TMnDH**: TMn 计数器高字节寄存器 bit 1~bit 0
TMn 10-bit 计数器 bit 9~bit 8

• TMnAL 寄存器

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

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

• TMnAH 寄存器

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

Bit 7~2 未定义, 读为 “0”
Bit 1~0 **TMnAH**: TMn CCRA 高字节寄存器 bit 1~bit 0
TMn 10-bit CCRA bit 9~bit 8

• TMnC0 寄存器

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

- Bit 7 TnPAU:** TMn 计数器暂停控制位
0: 运行
1: 暂停
通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，TM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其剩余值，直到此位再次改变为低电平，从此值开始继续计数。
- Bit 6~4 TnCK2~TnCK0:** 选择 TMn 计数时钟位
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
101: 保留位
110: TCKn 上升沿时钟
111: TCKn 下降沿时钟
此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。外部引脚时钟源能被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{TBC} (f_L) 是其它的内部时钟源，细节方面请参考振荡器章节。
- Bit 3 TnON:** TMn 计数器 On/Off 控制位
0: Off
1: On
此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转变时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。若 TMn 处于比较匹配输出模式时，当 TnON 位经由低到高转换时，TMn 输出脚将复位至 TnOC 位指定的初始值。
- Bit 2~0 TnRP2~TnRP0:** TMn CCRP 3-bit 寄存器，与 TMn 计数器 bit 9~bit 7 进行比较。比较器 P 匹配周期
000: 1024 个 TMn 时钟周期
001: 128 个 TMn 时钟周期
010: 256 个 TMn 时钟周期
011: 384 个 TMn 时钟周期
100: 512 个 TMn 时钟周期
101: 640 个 TMn 时钟周期
110: 768 个 TMn 时钟周期
111: 896 个 TMn 时钟周期
此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 TnCCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。TnCCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大值溢出。

• TMnC1 寄存器

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

- Bit 7~6 TnM1~TnM0:** 选择 TMn 工作模式位
 00: 比较匹配输出模式
 01: 未定义模式
 10: PWM 模式
 11: 定时 / 计数器模式
 这两位设置 TM 需要的工作模式。为了确保操作可靠, TM 应在 TnM1 和 TnM0 位有任何改变前先关掉。在定时 / 计数器模式, TM 输出脚控制必须除能。
- Bit 5~4 TnIO1~TnIO0:** 选择 TPn_0, TPn_1 输出功能位
 比较匹配输出模式
 00: 无变化
 01: 输出低
 10: 输出高
 11: 输出翻转
 PWM 模式
 00: 强制无效状态
 01: 强制有效状态
 10: PWM 输出
 11: 未定义
 定时 / 计数器模式
 未使用
 此两位用于决定在一定条件达到时 TM 输出脚如何改变状态。这两位值的选择决定 TM 运行在哪种模式下。
 在比较匹配输出模式下, TnIO1 和 TnIO0 位决定当比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时, 这个输出将不会改变。TM 输出脚的初始值通过 TMnC1 寄存器的 TnOC 位设置取得。注意, 由 TnIO1 和 TnIO0 位得到的输出电平必须与通过 TnOC 位设置的初始值不同, 否则当比较匹配发生时, TM 输出脚将不会发生变化。在 TM 输出脚改变状态后, 通过 TnON 位由低到高电平的转换复位至初始值。
 在 PWM 模式, TnIO1 和 TnIO0 用于决定比较匹配条件发生时怎样改变 TM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 TMn 关闭时改变 TnIO1 和 TnIO0 位的值是很有必要的。若在 TM 运行时改变 TnIO1 和 TnIO0 的值, PWM 输出的值是无法预料的。
- Bit 3 TnOC:** TPn_0, TPn_1 输出控制位
 比较匹配输出模式
 0: 初始低
 1: 初始高
 PWM 模式
 0: 低有效
 1: 高有效
 这是 TM 输出脚输出控制位。它取决于 TM 此时正运行于比较匹配输出模式还是 PWM 模式。若 TM 处于定时 / 计数器模式, 则其不受影响。在比较匹配输出模式时, 比较匹配发生前其决定 TM 输出脚的逻辑电平值。在 PWM 模式时, 其决定 PWM 信号是高有效还是低有效。
- Bit 2 TnPOL:** TPn_0, TPn_1 输出极性控制位
 0: 同相
 1: 反相
 此位控制 TPn_0 或 TPn_1 输出脚的极性。此位为高时 TM 输出脚反相, 为低时

	TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。
Bit 1	TnDPX: TMn PWM 周期 / 占空比控制位 0: CCRP - 周期; CCRA - 占空比 1: CCRP - 占空比; CCRA - 周期 此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。
Bit 0	TnCCLR: 选择 TMn 计数器清零条件位 0: TM0 比较器 P 匹配 1: TM0 比较器 A 匹配 此位用于选择清除计数器的方法。简易型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。TnCCLR 位设为高, 计数器在比较器 A 比较匹配发生时被清除; 此位设为低, 计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。TnCCLR 位在 PWM 模式时未使用。

简易型 TM 工作模式

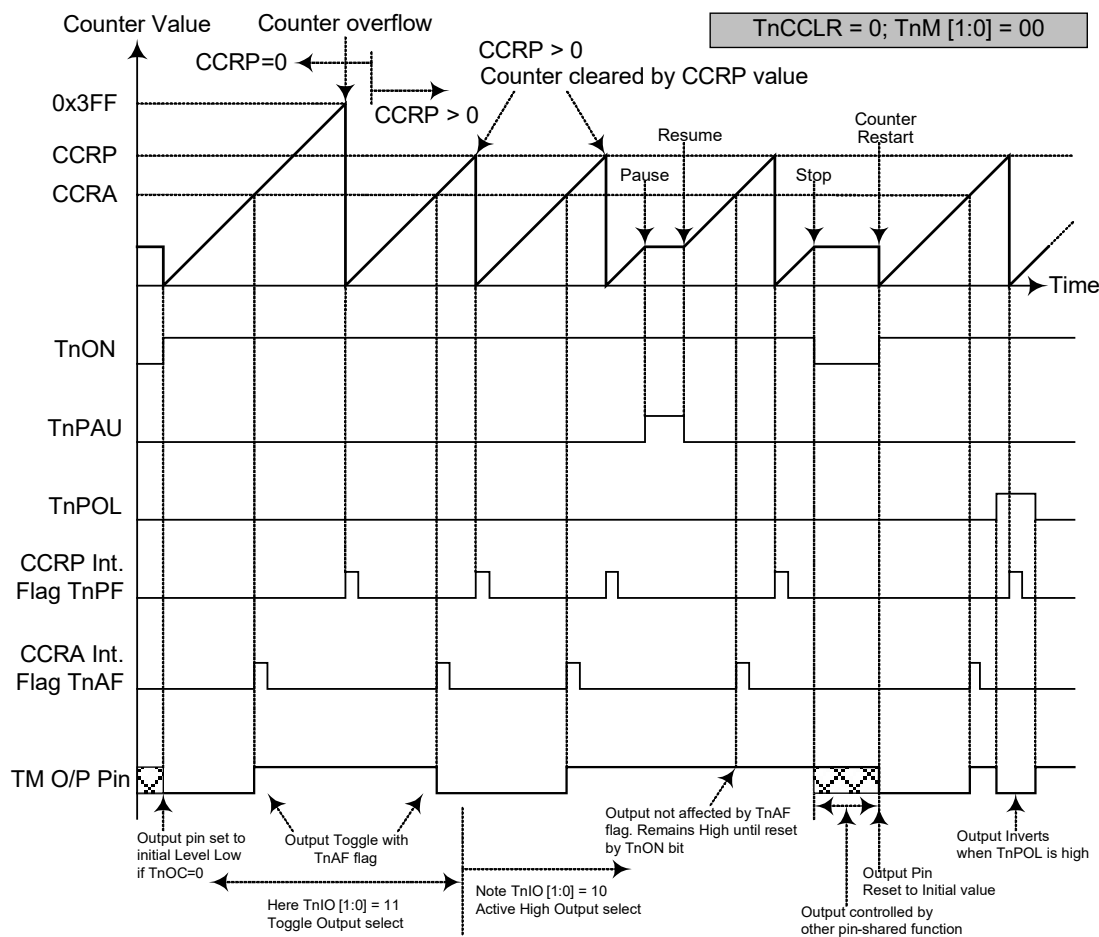
简易型 TM 有三种工作模式, 即比较匹配输出模式, PWM 模式或定时 / 计数器模式。通过设置 TMnC1 寄存器的 TnM1 和 TnM0 位选择任意工作模式。

比较匹配输出模式

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

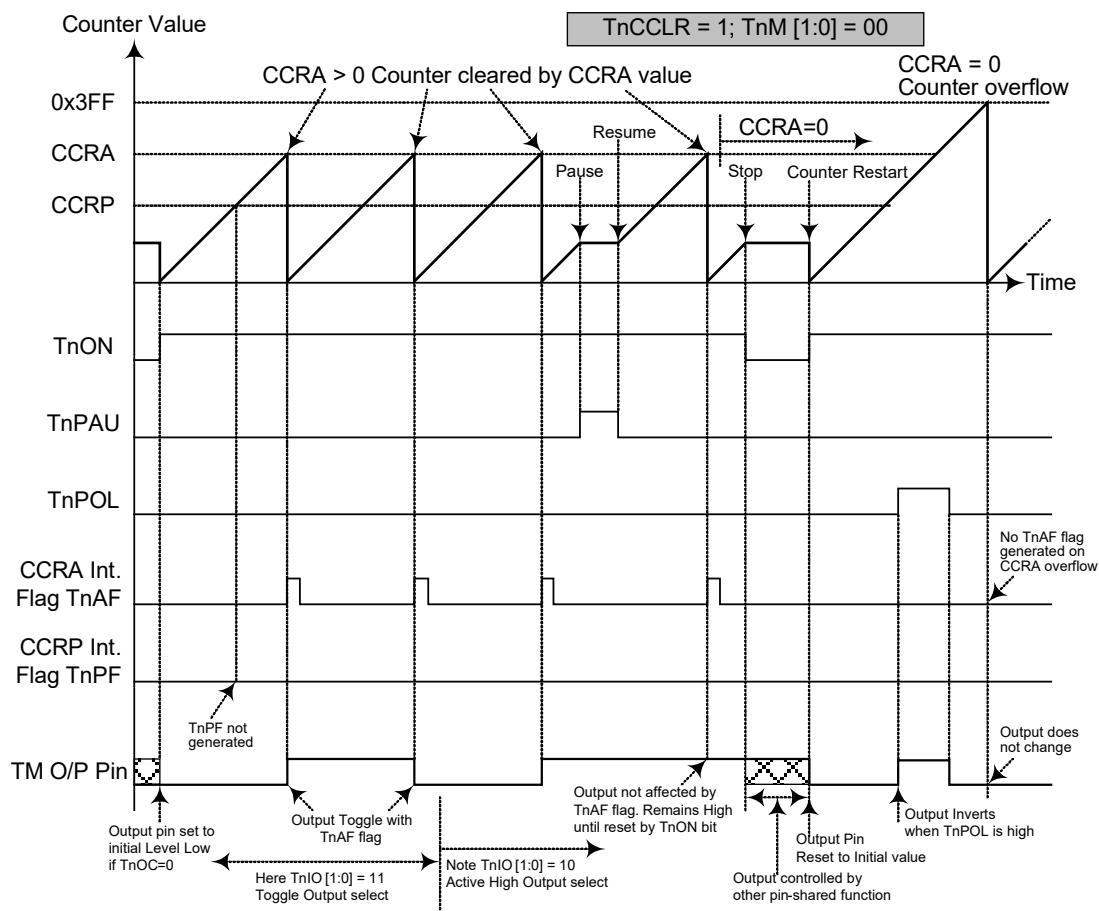
如果 TMnC1 寄存器的 TnCCLR 位设置为高, 当比较器 A 比较匹配发生时计数器被清零。此时, 即使 CCRP 寄存器的值小于 CCRA 寄存器的值, 仅 TnAF 中断请求标志产生。所以当 TnCCLR 为高时, 不产生 TnPF 中断请求标志。如果 CCRA 被清零, 当计数达到最大值 3FFH 时, 计数器溢出, 而此时不产生 TnAF 请求标志。

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



比较匹配输出模式 – TnCCLR=0

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



比较匹配输出模式 – TnCCLR=1

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

PWM 输出模式

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

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

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

• CTM, PWM 模式，边沿对齐模式，TnDPX=0

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

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

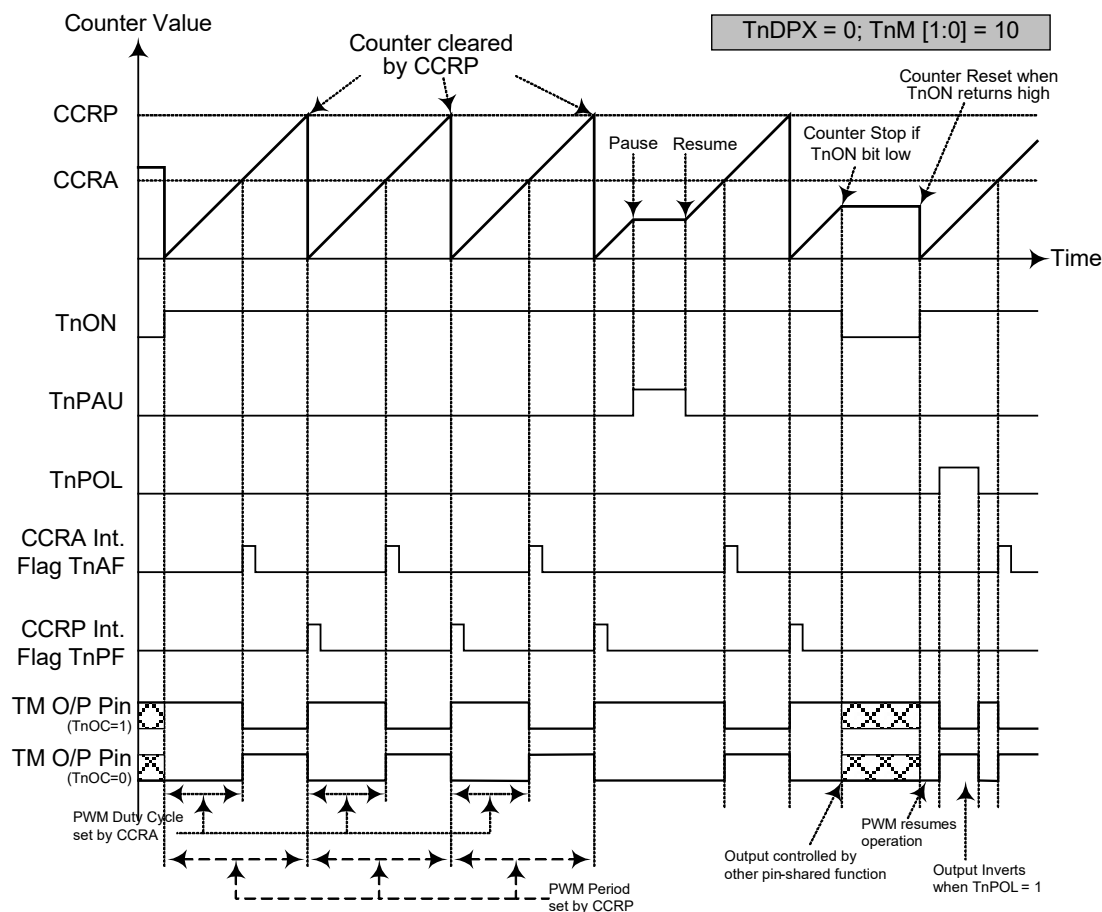
CTM PWM 输出频率 $= (f_{sys}/4)/512 = f_{sys}/2048 = 7.8125\text{kHz}$ ， $duty=128/512=25\%$

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%。

• CTM, PWM 模式，边沿对齐模式，TnDPX=1

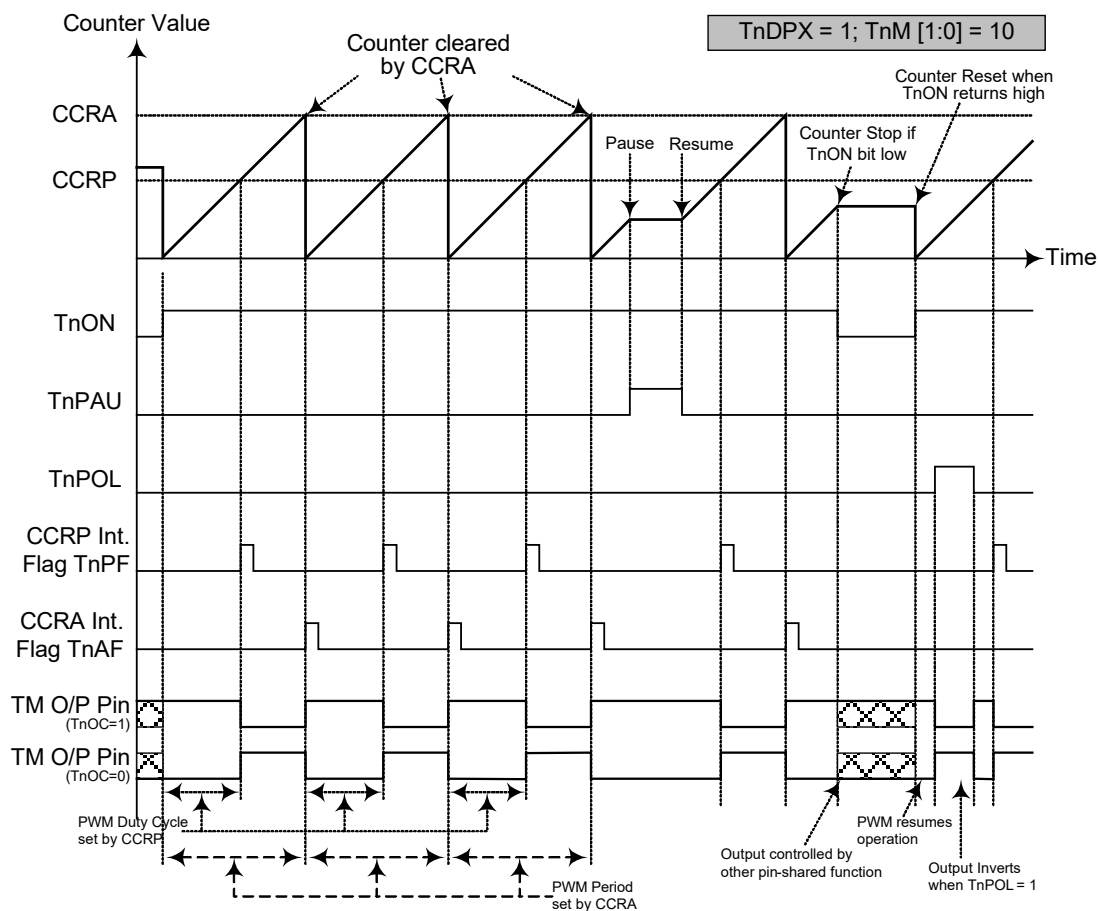
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

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



PWM 模式 – TnDPX=0

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



PWM 模式 – TnDPX=1

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

标准型 TM – STM

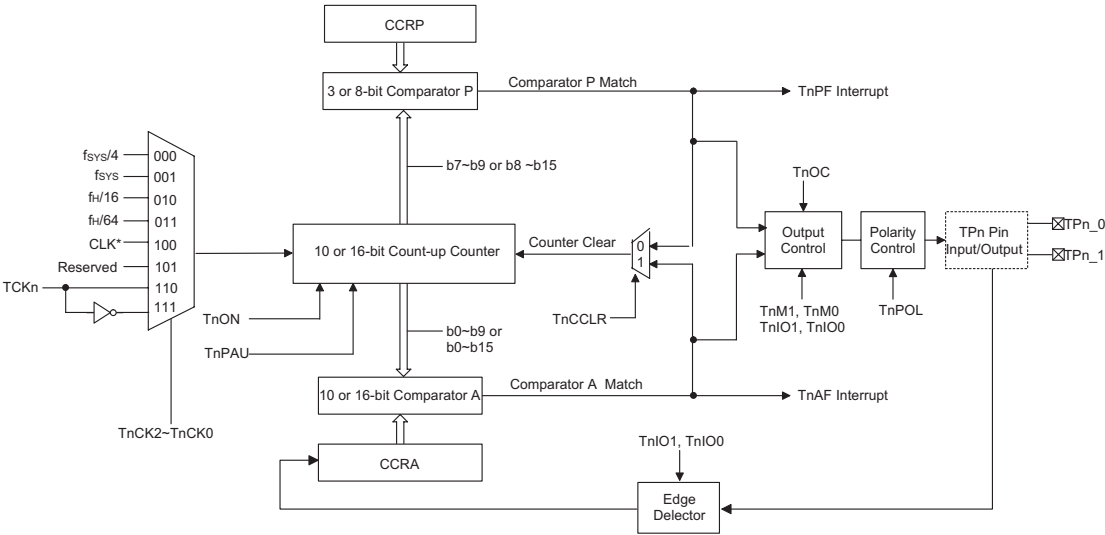
标准型 TM 包括 5 种工作模式，即比较匹配输出，定时 / 事件计数器，捕捉输入，单脉冲输出和 PWM 输出模式。标准型 TM 也由外部输入脚控制并驱动两个外部输出脚。

STM	名称	TM 编号	TM 输入引脚	TM 输出引脚
HT66FB542	16-bit STM 10-bit STM	0, 1	TCK0, TCK1	TP0_0, TP1_0
HT66FB540 HT66FB550 HT66FB560	16-bit STM 10-bit STM	0, 1	TCK0, TCK1	TP0_0, TP0_1 TP1_0, TP1_1

标准型 TM 操作

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

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



Note: CLK = f_{TBC} for HT66FB540/HT66FB550/HT66FB560, CLK = f_i for HT66FB542.

标准型 TM 框图

标准型 TM 寄存器介绍

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

寄存器名称	位							
	7	6	5	4	3	2	1	0
TM0C0	T0PAU	T0CK2	T0CK1	T0CK0	T0ON	—	—	—
TM0C1	T0M1	T0M0	T0IO1	T0IO0	T0OC	T0POL	T0DPX	T0CCLR
TM0DL	D7	D6	D5	D4	D3	D2	D1	D0
TM0DH	D15	D14	D13	D12	D11	D10	D9	D8
TM0AL	D7	D6	D5	D4	D3	D2	D1	D0
TM0AH	D15	D14	D13	D12	D11	D10	D9	D8
TM0RP	D7	D6	D5	D4	D3	D2	D1	D0

16-bit 标准型 TM 寄存器列表

• TM0C0 寄存器 – 16-bit STM

Bit	7	6	5	4	3	2	1	0
Name	T0PAU	T0CK2	T0CK1	T0CK0	T0ON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **T0PAU**: TM0 计数器暂停控制位

0: 运行
1: 暂停

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

Bit 6~4 **T0CK2~T0CK0**: 选择 TM0 计数时钟位

000: $f_{SYS}/4$
001: f_{SYS}
010: $f_{H}/16$
011: $f_{H}/64$
100: f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
101: 保留位
110: TCK0 上升沿时钟
111: TCK0 下降沿时钟

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

Bit 3 **T0ON**: TM0 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变时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。若 TM0 处于比较匹配输出模式时，当 T0ON 位经由低到高转换时，TM0 输出脚将复位至 T0OC 位指定的初始值。

Bit 2~0 未定义，读为“0”

• TM0C1 寄存器 – 16-bit STM

Bit	7	6	5	4	3	2	1	0
Name	T0M1	T0M0	T0IO1	T0IO0	T0OC	T0POL	T0DPX	T0CCLR
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 **T0M1~T0M0**: 选择 TM0 工作模式位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 模式或单脉冲输出模式
- 11: 定时 / 计数器模式

这两位设置 TM 需要的工作模式。为了确保操作可靠, TM 应在 T0M1 和 T0M0 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 **T0IO1~T0IO0**: 选择 TP0_0, TP0_1 输出功能位

比较匹配输出模式

- 00: 无变化
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 模式 / 单脉冲输出模式

- 00: 强制无效状态
- 01: 强制有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 TP0_0, TP0_1 上升沿输入捕捉
- 01: 在 TP0_0, TP0_1 下降沿输入捕捉
- 10: 在 TP0_0, TP0_1 双沿输入捕捉
- 11: 输入捕捉除能

定时 / 计数器模式

未使用

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

在比较匹配输出模式下, T0IO1 和 T0IO0 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 TM0C1 寄存器的 T0OC 位设置取得。注意, 由 T0IO1 和 T0IO0 位得到的输出电平必须与通过 T0OC 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 T0ON 位由低到高电平的转换复位至初始值。

Bit 3 **T0OC**: TP0_0, TP0_1 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 模式 / 单脉冲输出模式

- 0: 低有效
- 1: 高有效

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

Bit 2 **T0POL**: TP0_0, TP0_1 输出极性控制位

- 0: 同相
- 1: 反相

此位控制 TP0_0 或 TP0_1 输出脚的极性。此位为高时 TM 输出脚反相, 为低时

TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1 **T0DPX**: TM0 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 **T0CCLR**: 选择 TM0 计数器清零条件位
0: TM0 比较器 P 匹配
1: TM0 比较器 A 匹配

此位用于选择清除计数器的方法。标准型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。T0CCLR 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。T0CCLR 位在 PWM, 单脉冲或输入捕捉模式时未使用。

• TM0DL 寄存器 – 16-bit STM

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

Bit 7~0 **TM0DL**: TM0 计数器低字节寄存器 bit 7~bit 0
TM0 16-bit 计数器 bit 7~bit 0

• TM0DH 寄存器 – 16-bit STM

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

Bit 7~0 **TM0DH**: TM0 计数器高字节寄存器 bit 7~bit 0
TM0 16-bit 计数器 bit 15~bit 8

• TM0AL 寄存器 – 16-bit STM

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

Bit 7~0 **TM0AL**: TM0 CCRA 低字节寄存器 bit 7~bit 0
TM0 16-bit CCRA bit 7~bit 0

• TM0AH 寄存器 – 16-bit STM

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

Bit 7~0 **TM0AH**: TM0 CCRA 高字节寄存器 bit 7~bit 0
TM0 16-bit CCRA bit 15~bit 8

• TM0RP 寄存器 – 16-bit STM

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

Bit 7~0 TM0RP: TM0 CCRP 高字节寄存器 bit 7~bit 0
TM0 CCRP 8 位寄存器，与 TM0 计数器 bit 15~bit 8 比较。比较器 P 匹配周期
0: 65536 个 TM0 时钟周期
1~255: 256×(1~255) 个 TM0 时钟周期
此八位设定内部 CCRP 8-bit 寄存器的值，然后与内部计数器的高八位进行比较。
如果 T0CCLR 位设为 0 时，比较结果为 0 并清除内部计数器。T0CCLR 位设为低，
CCRP 比较匹配结果将重置内部计数器。由于 CCRP 只与计数器高八位比较，
比较结果是 256 时钟周期的倍数。CCRP 被清零时，实际上会使得计数器在最大
值溢出。

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

10-bit 标准型 TM 寄存器列表

• TM1C0 寄存器 – 10-bit STM

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

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

Bit 6~4 T1CK2~T1CK0: 选择 TM1 计数器时钟位
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
101: 保留位
110: TCK1 上升沿时钟
111: TCK1 下降沿时钟
此三位用于选择 TM 的时钟源。选择保留时钟输入将有效地除能内部计数器。
外部引脚时钟源能被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{TBC} (f_L)
是其它的内部时钟源，细节方面请参考振荡器章节。

- Bit 3 **T1ON**: TM1 计数器 On/Off 控制位
0: Off
1: On
此位控制 TM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 TM。清零此位将停止计数器并关闭 TM 减少耗电。当此位经由低到高转变时，内部计数器将复位清零；当此位经由高到低转换时，内部计数器将保持其剩余值，直到此位再次改变为高电平。若 TM1 处于比较匹配输出模式时，当 T1ON 位经由低到高转换时，TM1 输出脚将复位至 T1OC 位指定的初始值。
- Bit 2~0 **T1RP2~T1RP0**: TM1 CCRP 3-bit 寄存器，与 TM1 计数器 bit 9~bit 7 进行比较。
比较器 P 匹配周期
000: 1024 个 TM1 时钟周期
001: 128 个 TM1 时钟周期
010: 256 个 TM1 时钟周期
011: 384 个 TM1 时钟周期
100: 512 个 TM1 时钟周期
101: 640 个 TM1 时钟周期
110: 768 个 TM1 时钟周期
111: 896 个 TM1 时钟周期
此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 T1CCLR 位设定为 0 时，比较结果为 0 并清除内部计数器。T1CCLR 位设为低，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，实际上会使计数器在最大值溢出。

• TM1C1 寄存器 – 10-bit STM

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

- Bit 7~6 **T1M1~T1M0**: 选择 TM1 工作模式位
00: 比较匹配输出模式
01: 捕捉输入模式
10: PWM 模式或单脉冲输出模式
11: 定时 / 计数器模式
这两位设置 TM 需要的工作模式。为了确保操作可靠，TM 应在 T1M1 和 T1M0 位有任何改变前先关掉。在定时 / 计数器模式，TM 输出脚控制必须除能。
- Bit 5~4 **T1IO1~T1IO0**: 选择 TP1_0, TP1_1 输出功能位
比较匹配输出模式
00: 无变化
01: 输出低
10: 输出高
11: 输出翻转
PWM 模式 / 单脉冲输出模式
00: 强制无效状态
01: 强制有效状态
10: PWM 输出
11: 单脉冲输出
捕捉输入模式
00: 在 TP1_0, TP1_1 上升沿输入捕捉
01: 在 TP1_0, TP1_1 下降沿输入捕捉
10: 在 TP1_0, TP1_1 双沿输入捕捉
11: 输入捕捉除能
定时 / 计数器模式
未使用

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

在比较匹配输出模式下，T1IO1 和 T1IO0 位决定当从比较器 A 比较匹配输出发生时 TM 输出脚如何改变状态。当从比较器 A 比较匹配输出发生时 TM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，这个输出将不会改变。TM 输出脚的初始值通过 TMIC1 寄存器的 T1OC 位设置取得。注意，由 T1IO1 和 T1IO0 位得到的输出电平必须与通过 T1OC 位设置的初始值不同，否则当比较匹配发生时，TM 输出脚将不会发生变化。在 TM 输出脚改变状态后，通过 T1ON 位由低到高电平的转换复位至初始值。

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

Bit 3 **T1OC**: TP1_0, TP1_1 输出控制位
比较匹配输出模式
0: 初始低
1: 初始高
PWM 模式 / 单脉冲输出模式
0: 低有效
1: 高有效

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

Bit 2 **T1POL**: TP1_0, TP1_1 输出极性控制位
0: 同相
1: 反相

此位控制 TP1_0 或 TP1_1 输出脚的极性。此位为高时 TM 输出脚反相，为低时 TM 输出脚同相。若 TM 处于定时 / 计数器模式时其不受影响。

Bit 1 **T1DPX**: TM1 PWM 周期 / 占空比控制位
0: CCRP - 周期; CCRA - 占空比
1: CCRP - 占空比; CCRA - 周期

此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。

Bit 0 **T1CCLR**: 选择 TM1 计数器清零条件位
0: TM1 比较器 P 匹配
1: TM1 比较器 A 匹配

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

● TM1DL 寄存器 – 10-bit STM

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

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

● **TM1DH 寄存器 – 10-bit STM**

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

Bit 7~2 未定义，读为“0”

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

● **TM1AL 寄存器 – 10-bit STM**

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

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

● **TM1AH 寄存器 – 10-bit STM**

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

Bit 7~2 未定义，读为“0”

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

标准型 TM 工作模式

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

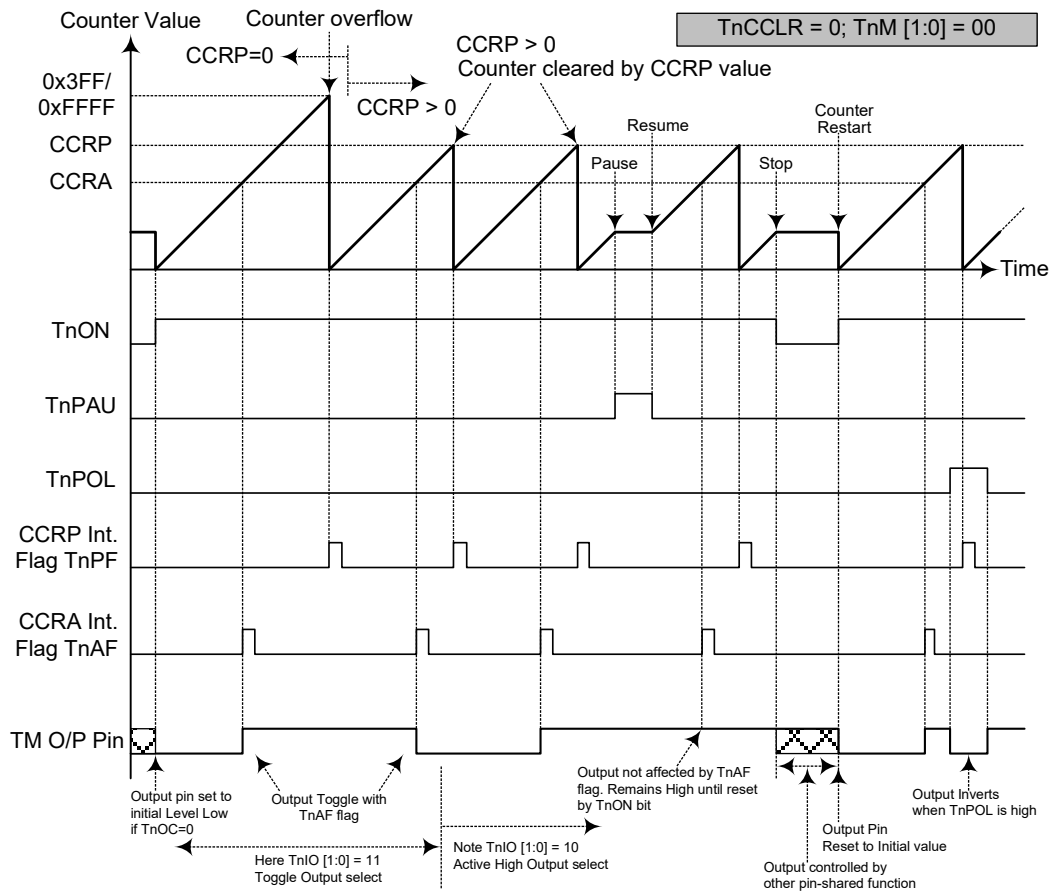
比较匹配输出模式

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

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

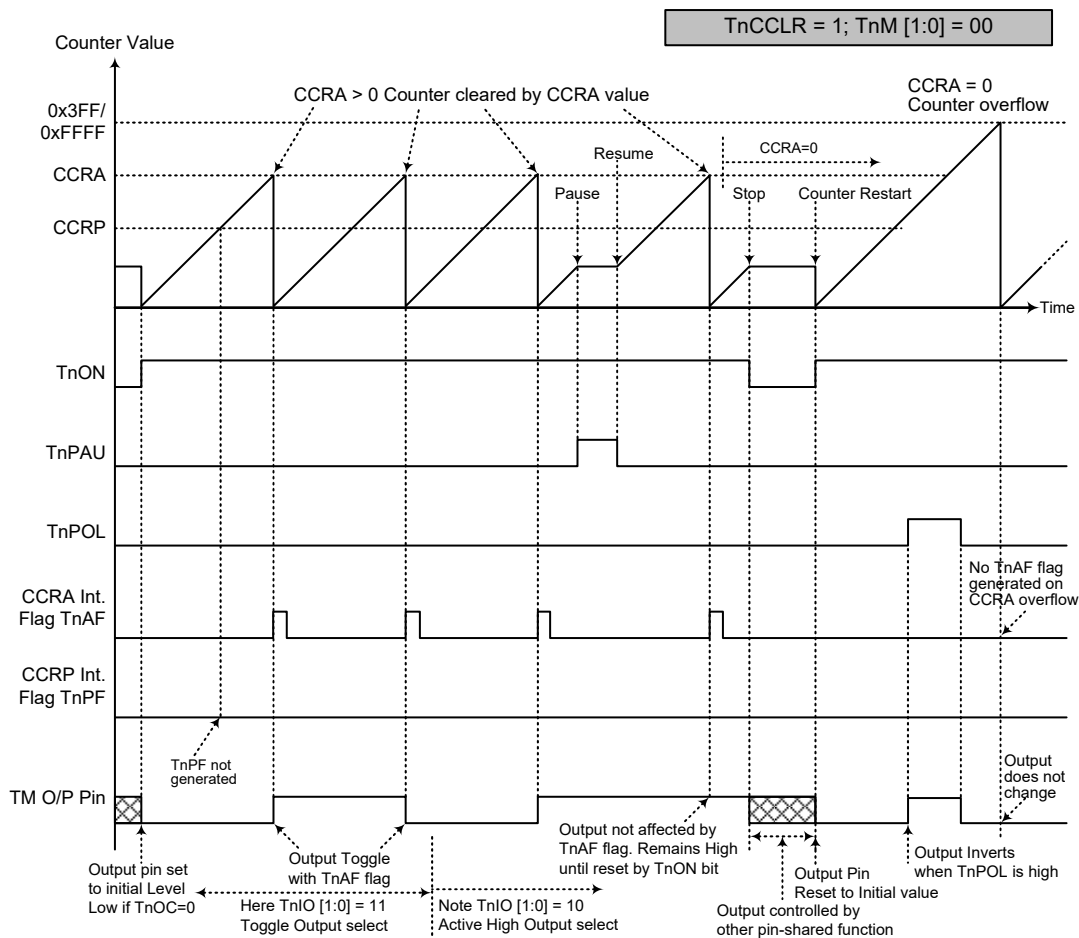
正如该模式名所言，当比较匹配发生后，TM 输出脚状态改变。当比较器 A 比较匹配发生后 TnAF 标志产生时，TM 输出脚状态改变。比较器 P 比较匹配发生时产生的 TnPF 标志不影响 TM 输出脚。TM 输出脚状态改变方式由 TMnC1 寄存器中 TnIO1 和 TnIO0 位决定。当比较器 A 比较匹配发生时，TnIO1 和

TnIO0 位决定 TM 输出脚输出高，低或翻转当前状态。TMn 输出脚初始值，在 TnON 位由低到高电平的变化后通过 TnOC 位设置。注意，若 TnIO1 和 TnIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – TnCCLR=0

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



比较匹配输出模式 – TnCCLR=1

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

定时 / 计数器模式

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

PWM 输出模式

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

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

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

● 16-bit STM, PWM 模式，边沿对齐模式，T0DPX=0

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

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

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

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

● 16-bit STM, PWM 模式，边沿对齐模式，T0DPX=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。

● 10-bit STM, PWM 模式，边沿对齐模式，T1DPX=0

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

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

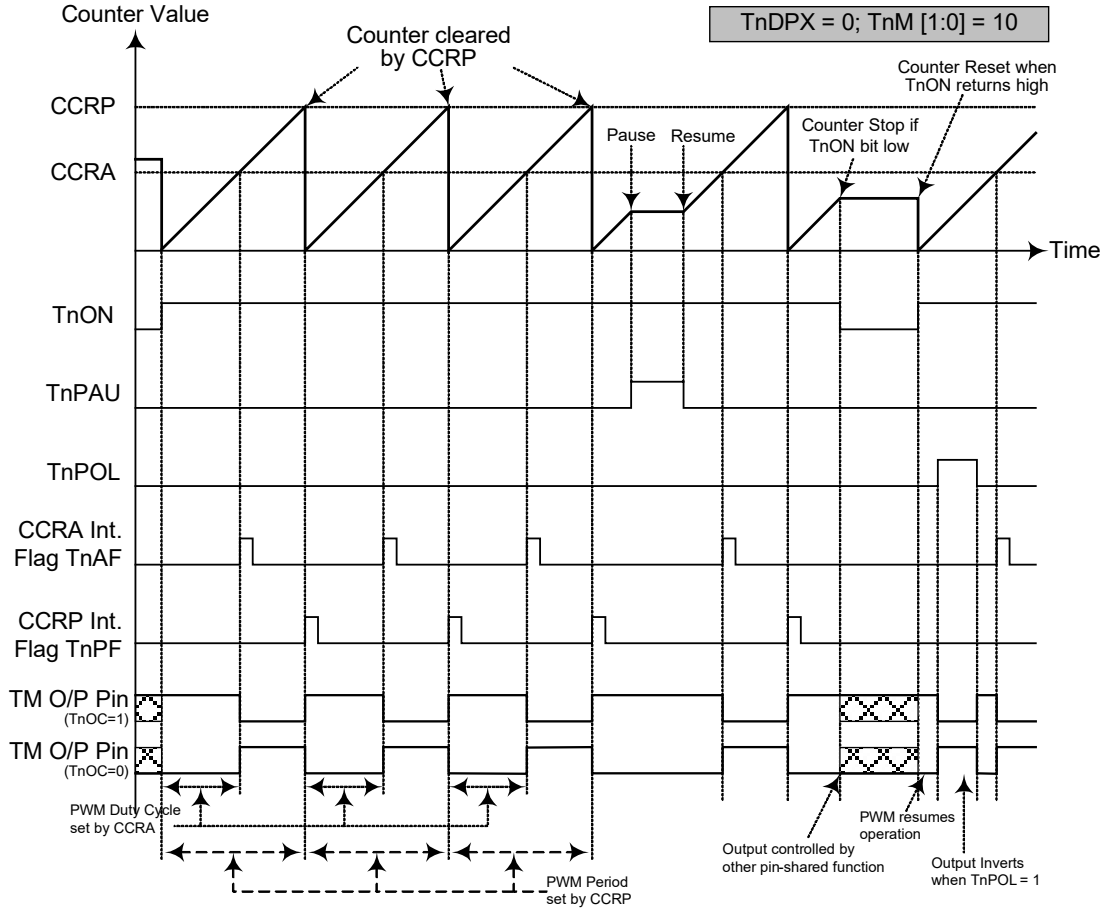
STM PWM 输出频率 = $(f_{SYS}/4)/512 = f_{SYS}/2048 = 7.8125\text{kHz}$ ，duty = $128/512 = 25\%$

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 STM, PWM 模式, 边沿对齐模式, T1DPX=1

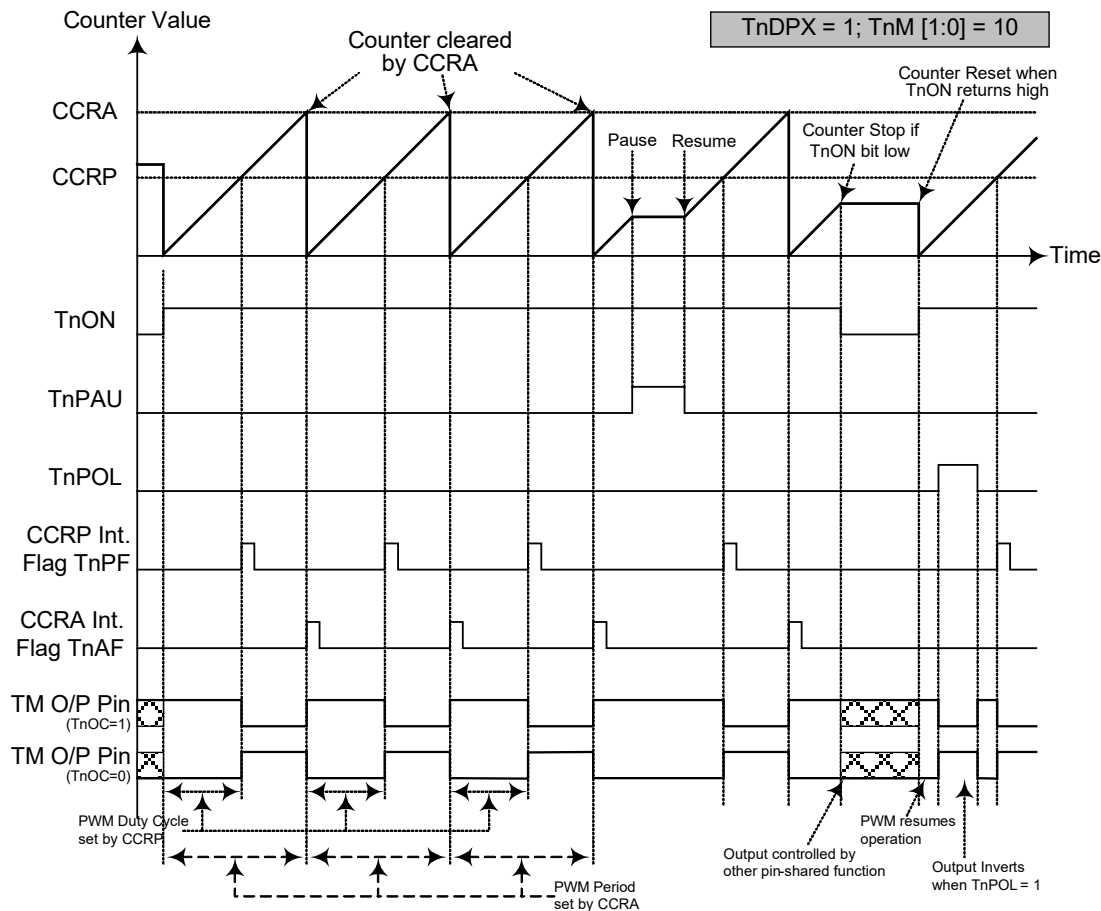
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

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



PWM 模式 – TnDPX=0

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



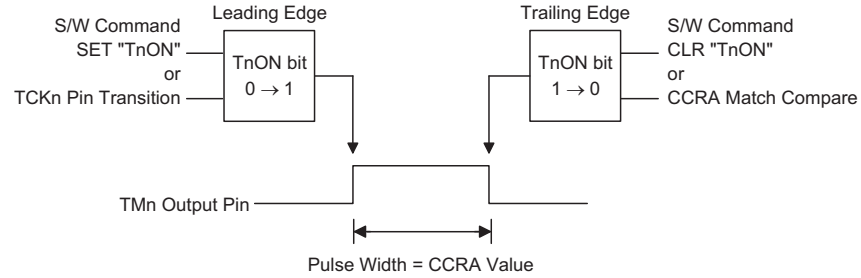
PWM 模式 – TnDPX=1

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

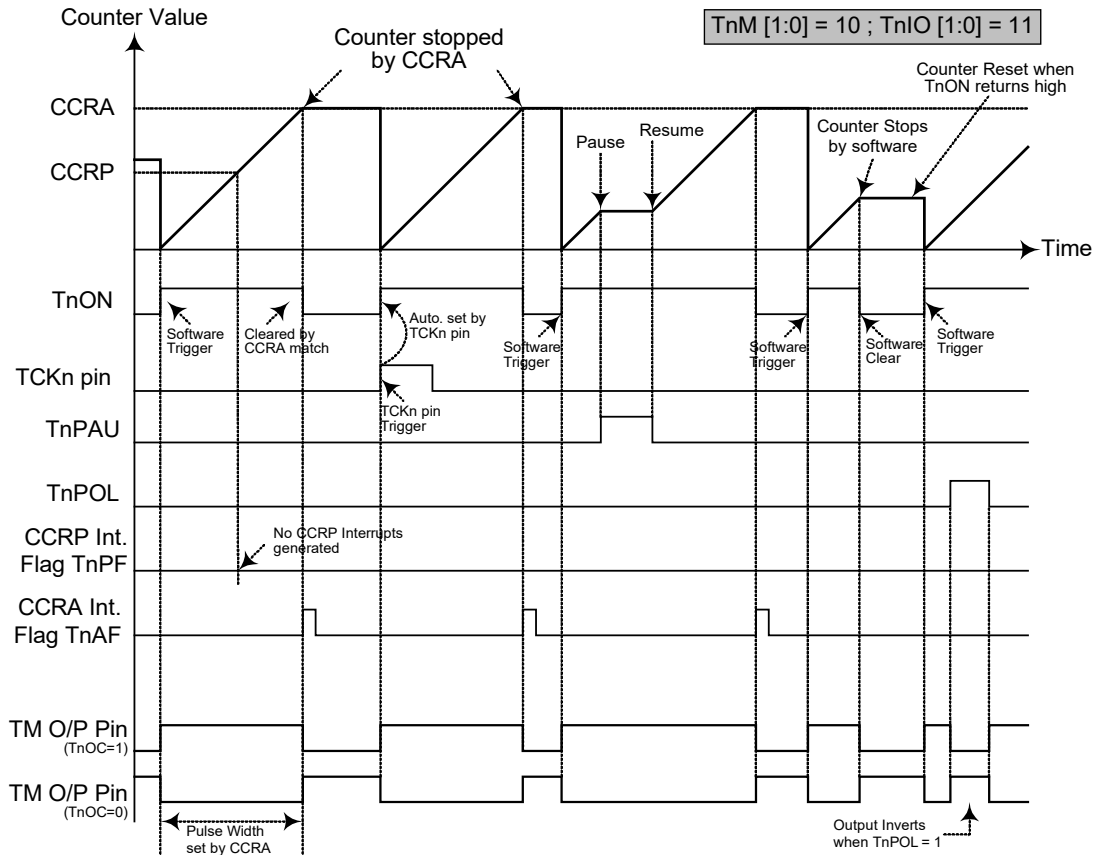
单脉冲模式

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

脉冲输出可以通过应用程序控制 TnON 位由低到高的转变来触发。而处于单脉冲模式时，TnON 位在 TCKn 脚发生有效边沿跳转时自动由低转变为高，进而开始单脉冲输出。当 TnON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 TnON 位保持高电平。通过应用程序使 TnON 位清零或比较器 A 比较匹配发生时，产生脉冲后沿。



单脉冲产生示意图



单脉冲模式

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

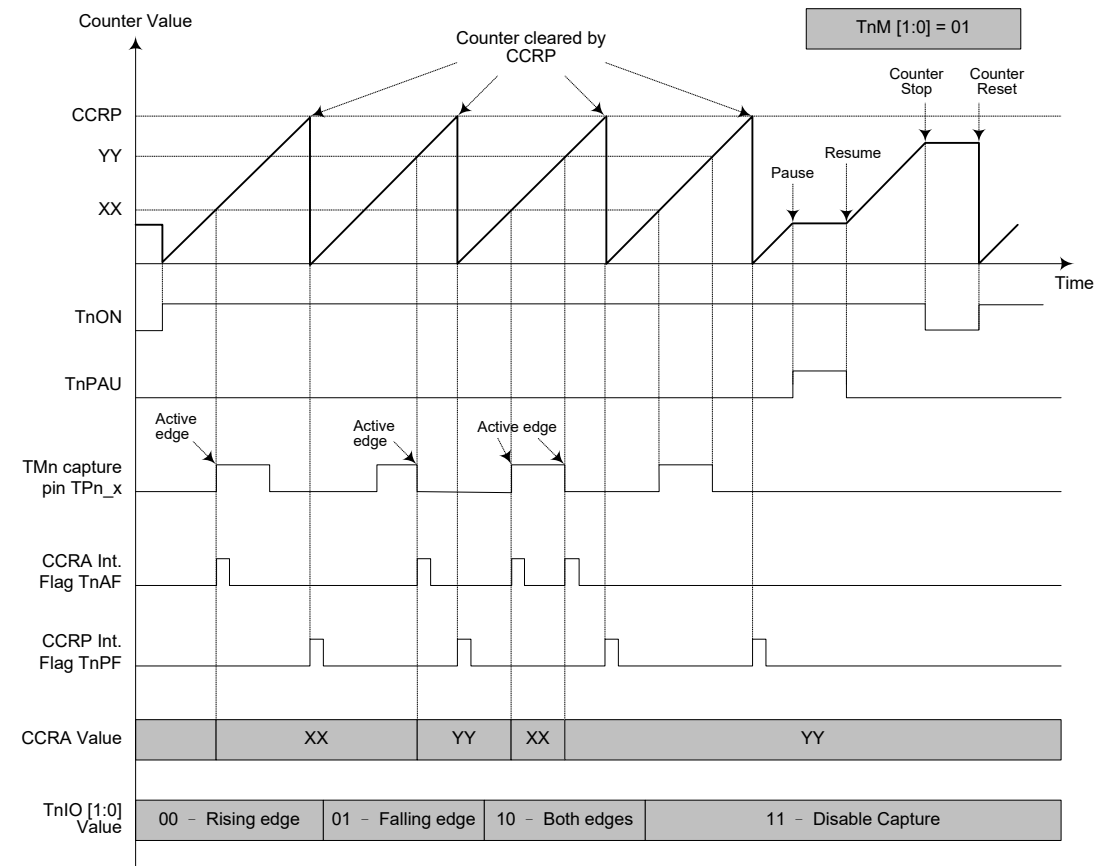
然而，比较器 A 比较匹配发生时，会自动清除 TnON 位并产生单脉冲输出边沿跳变。CCRA 的值通过这种方式控制脉冲宽度。比较器 A 比较匹配发生时，也会产生 TM 中断。TnON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲模式中，CCRP 寄存器，TnCCLR 和 TnDPX 位未使用。

捕捉输入模式

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

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

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



捕捉输入模式

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

A/D 转换器

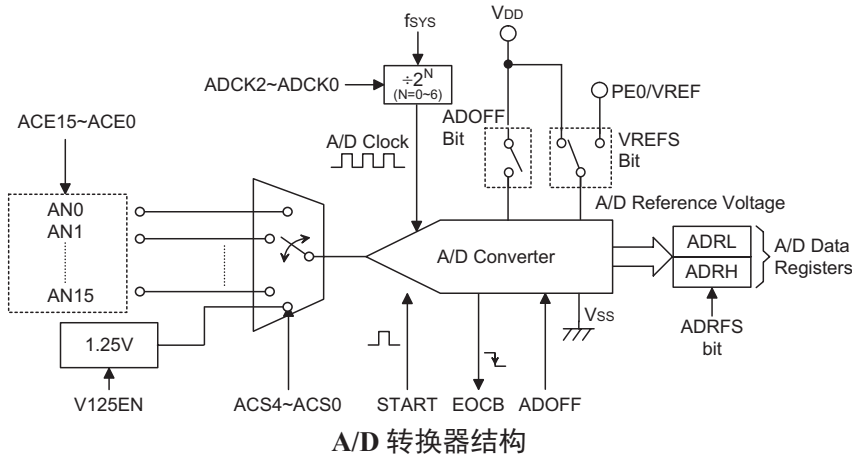
对于大多数的电子系统而言，处理现实世界的模拟信号是共同的需求。为了完全由单片机来处理这些信号，首先需要通过 A/D 转换器将模拟信号转换成数字信号。将 A/D 转换器电路集成入单片机，可有效的减少外部器件，随之而来，具有降低成本和减少器件空间需求的优势。

A/D 简介

此系列单片机包含了一个多通道的 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）并直接将这些信号转换成 12 位的数字量。

单片机型号	输入通道数	A/D 通道选择数	输入引脚
HT66FB540	8	ACS4, ACS2~ACS0	AN0~AN7
HT66FB542	4	ACS4, ACS1~ACS0	AN0~AN3
HT66FB550 HT66FB560	16	ACS4~ACS0	AN0~AN15

下图显示了 A/D 转换器内部结构和相关的寄存器。



A/D 转换器结构

A/D 转换寄存器介绍

A/D 转换器的所有工作由多个寄存器控制。一对只读寄存器来存放 12 位 ADC 数据的值。剩下三个或四个控制寄存器设置 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
ADRL (ADRFS=0)	D3	D2	D1	D0	—	—	—	—
ADRL (ADRFS=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH (ADRFS=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH (ADRFS=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFS	—	ACS2	ACS1	ACS0
ADCR1	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACER0	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0

A/D 转换寄存器列表 – HT66FB540

寄存器名称	位							
	7	6	5	4	3	2	1	0
ADRL (ADRFSS=0)	D3	D2	D1	D0	—	—	—	—
ADRL (ADRFSS=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH (ADRFSS=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH (ADRFSS=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFSS	—	—	ACS1	ACS0
ADCR1	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACER0	—	—	—	—	ACE3	ACE2	ACE1	ACE0

A/D 转换寄存器列表 – HT66FB542

寄存器名称	位							
	7	6	5	4	3	2	1	0
ADRL (ADRFSS=0)	D3	D2	D1	D0	—	—	—	—
ADRL (ADRFSS=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH (ADRFSS=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH (ADRFSS=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFSS	ACS3	ACS2	ACS1	ACS0
ADCR1	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACER0	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
ACER1	ACE15	ACE14	ACE13	ACE12	ACE11	ACE10	ACE9	ACE8

A/D 转换寄存器列表 – HT66FB550/HT66FB560

A/D 转换器数据寄存器 – ADRL, ADRH

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 ADRH 和一个低字节寄存器 ADRL。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 ADCR0 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”。

ADRFSS	ADRH								ADRL							
	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 数据寄存器

A/D 转换控制寄存器 – ADCR0, ADCR1, ACER0, ACER1

寄存器 ADCR0, ADCR1, ACER0 和 ACER1 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，并控制和监视 A/D 转换器的开始和转换结束状态。寄存器 ADCR0 的 ACS3~ACS0 位和 ADCR1 的 ACS4 位定义 ADC 输入通道编号。由于每个单片机只包含一个实际的模数转换电路，因此这 4、8 或 16 个模拟输入中的每一个都需要分别被发送到转换器。ACS4~ACS0 位的功能决定选择哪个模拟输入通道或内部 1.25V 电路是否被连接到内部 A/D 转换器。

ACER1 和 ACER0 控制寄存器中的 ACE15~ACE0 位，用来定义 PA7，PB 口和 PC 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。相应位设为高将选择 A/D 输入功能，清零将选择 I/O 或其它引脚共用功能。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。

● ADCR0 寄存器 – HT66FB540

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRF5	—	ACS2	ACS1	ACS0
R/W	R/W	R	R/W	R/W	—	R/W	R/W	R/W
POR	0	1	1	0	—	0	0	0

- Bit 7 **START**: 启动 A/D 转换
 0→1→0: 启动
 0→1: 重置 A/D 转换，并且设置 EOCB 为“1”
 此位用于初始化 A/D 转换过程。通常此位为低，但如果设为高再被清零，将初始化 A/D 转换过程。当此位为高，将重置 A/D 转换器。
- Bit 6 **EOCB**: A/D 转换标志
 0: A/D 转换结束
 1: A/D 转换中
 此位用于表明 A/D 转换过程的完成。当转换正在进行时，此位为高。
- Bit 5 **ADOFF**: ADC 模块电源开 / 关控制位
 0: ADC 模块电源开
 1: ADC 模块电源关
 注意: 1. 建议在进入休眠或空闲模式之前，设置 ADOFF=1 以减小功耗。
 2. ADOFF=1 将关闭 ADC 模块的电源。
- Bit 4 **ADRF5**: ADC 数据格式控制位
 0: ADC 数据高字节是 ADRH 的 bit 7~bit 0，低字节是 ADRL 的 bit 7~bit 4
 1: ADC 数据高字节是 ADRH 的 bit 3~bit 0，低字节是 ADRL 的 bit 7~bit 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 未定义，读为“0”
- Bit 2~0 **ACS2~ACS0**: 选择 A/D 通道 (ACS4 为“0”)
 000: AN0
 001: AN1
 010: AN2
 011: AN3
 100: AN4
 101: AN5
 110: AN6
 111: AN7
 这几位是 A/D 通道选择控制位。由于只包含一个内部 A/D 转换电路，因此可通过这些位将 8 个 A/D 输入连接到内部转换器。如果 ADCR1 寄存器中的 ACS4 设为高，内部 1.25V 电路将被连接到 A/D 转换器。

● ADCR0 寄存器 – HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRF5	—	—	ACS1	ACS0
R/W	R/W	R	R/W	R/W	—	—	R/W	R/W
POR	0	1	1	0	—	—	0	0

- Bit 7 START:** 启动 A/D 转换
0→1→0: 启动
0→1: 重置 A/D 转换, 并且设置 EOCB 为 “1”
此位用于初始化 A/D 转换过程。通常此位为低, 但如果设为高再被清零, 将初始化 A/D 转换过程。当此位为高, 将重置 A/D 转换器。
- Bit 6 EOCB:** A/D 转换标志
0: A/D 转换结束
1: A/D 转换中
此位用于表明 A/D 转换过程的完成。当转换正在进行时, 此位为高。
- Bit 5 ADOFF:** ADC 模块电源开 / 关控制位
0: ADC 模块电源开
1: ADC 模块电源关
注: 1. 建议在进入休眠或空闲模式之前, 设置 ADOFF=1 以减小功耗。
2. ADOFF=1 将关闭 ADC 模块的电源。
- Bit 4 ADRFS:** ADC 数据格式控制位
0: ADC 数据高字节是 ADRH 的 bit 7~bit 0, 低字节是 ADRL 的 bit 7~bit 4
1: ADC 数据高字节是 ADRH 的 bit 3~bit 0, 低字节是 ADRL 的 bit 7~bit 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 3~2** 未定义, 读为 “0”
- Bit 1~0 ACS1~ACS0:** 选择 A/D 通道 (ACS4 为 “0”)
00: AN0
01: AN1
10: AN2
11: AN3
这几位是 A/D 通道选择控制位。由于只包含一个内部 A/D 转换电路, 因此可通过这些位将 4 个 A/D 输入连接到内部转换器。如果 ADCR1 寄存器中的 ACS4 设为高, 内部 1.25V 电路将被连接到 A/D 转换器。

● ADCR0 寄存器 – HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRF5	ACS3	ACS2	ACS1	ACS0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	0	0	0	0	0

- Bit 7 START:** 启动 A/D 转换
0→1→0: 启动
0→1: 重置 A/D 转换, 并且设置 EOCB 为 “1”
此位用于初始化 A/D 转换过程。通常此位为低, 但如果设为高再被清零, 将初始化 A/D 转换过程。当此位为高, 将重置 A/D 转换器。
- Bit 6 EOCB:** A/D 转换标志
0: A/D 转换结束
1: A/D 转换中
此位用于表明 A/D 转换过程的完成。当转换正在进行时, 此位为高。

- Bit 5 **ADOFF**: ADC 模块电源开 / 关控制位
0: ADC 模块电源开
1: ADC 模块电源关
注意: 1. 建议在进入休眠或空闲模式之前, 设置 ADOFF=1 以减小功耗。
2. ADOFF=1 将关闭 ADC 模块的电源。
- Bit 4 **ADRF5**: ADC 数据格式控制位
0: ADC 数据高字节是 ADRH 的 bit 7~bit 0, 低字节是 ADRL 的 bit 7~bit 4
1: ADC 数据高字节是 ADRH 的 bit 3~bit 0, 低字节是 ADRL 的 bit 7~bit 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 **ACS3~ACS0**: 选择 A/D 通道 (ACS4 为 “0”)
0000: AN0
0001: AN1
0010: AN2
0011: AN3
0100: AN4
0101: AN5
0110: AN6
0111: AN7
1000: AN8
1001: AN9
1010: AN10
1011: AN11
1100: AN12
1101: AN13
1110: AN14
1111: AN15
这几位是 A/D 通道选择控制位。由于只包含一个内部 A/D 转换电路, 因此可通过这些位将 16 个 A/D 输入连接到内部转换器。如果 ADCR1 寄存器中的 ACS4 设为高, 内部 1.25V 电路将被连接到 A/D 转换器。

● ADCR1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
R/W	R/W	R/W	—	R/W	—	R/W	R/W	R/W
POR	0	0	—	0	—	0	0	0

- Bit 7 **ACS4**: 选择内部 1.25V 作为 ADC 输入控制位
0: 除能
1: 使能
此位使能 1.25V 连接到 A/D 转换器。V125EN 位必须先被置位使能 1.25V 电压能隙电路被用于 A/D 转换器。当 ACS4 设为高, 1.25V 能隙电压将连接到 A/D 转换器, 其它 A/D 输入通道断开。
- Bit 6 **V125EN**: 内部 1.25V 控制位
0: 除能
1: 使能
此位控制连接到 A/D 转换器的内部充电泵电路开 / 关功能。当此位设为高, 充电泵电压 1.25V 连接至 A/D 转换器。如果 1.25V 未连接至 A/D 转换器且 LVR/LVD 除能, 充电泵参考电压电路自动关闭以减少功耗。当 1.25V 打开连接至 A/D 转换器, 在 A/D 转换动作执行前, 充电泵电路稳定需一段时间 t_{BG} 。
- Bit 5 未定义, 读为 “0”
- Bit 4 **VREFS**: 选择 ADC 参考电压
0: 内部 ADC 电源
1: VREF 引脚

此位用于选择 A/D 转换器的参考电压。如果该位设为高，A/D 转换器参考电压来源于外部 VREF 引脚。如果该位设为低，内部参考电压来源于电源电压 VDD。

- Bit 3 未定义，读为“0”
- Bit 2~0 **ADCK2~ADCK0**: 选择 ADC 时钟源
- 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: 未定义

这三位用于选择 A/D 转换器的时钟源。

● **ACER0 寄存器 – HT66FB540/HT66FB550/HT66FB560**

Bit	7	6	5	4	3	2	1	0
Name	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
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 **ACE7**: 定义 PA7 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN7
- Bit 6 **ACE6**: 定义 PB6 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN6
- Bit 5 **ACE5**: 定义 PB5 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN5
- Bit 4 **ACE4**: 定义 PB4 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN4
- Bit 3 **ACE3**: 定义 PB3 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN3
- Bit 2 **ACE2**: 定义 PB2 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN2
- Bit 1 **ACE1**: 定义 PB1 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN1
- Bit 0 **ACE0**: 定义 PB0 是否为 A/D 输入
- 0: 不是 A/D 输入
 - 1: A/D 输入, AN0

● ACER0 寄存器 – HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	ACE3	ACE2	ACE1	ACE0
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 **ACE3**: 定义 PA7 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN3

Bit 2 **ACE2**: 定义 PB2 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN2

Bit 1 **ACE1**: 定义 PB1 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN1

Bit 0 **ACE0**: 定义 PB0 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN0

● ACER1 寄存器 – HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	ACE15	ACE14	ACE13	ACE12	ACE11	ACE10	ACE9	ACE8
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 **ACE15**: 定义 PC7 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN15

Bit 6 **ACE14**: 定义 PC6 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN14

Bit 5 **ACE13**: 定义 PC5 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN13

Bit 4 **ACE12**: 定义 PC4 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN12

Bit 3 **ACE11**: 定义 PC3 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN11

Bit 2 **ACE10**: 定义 PC2 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN10

Bit 1 **ACE9**: 定义 PC1 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN9

Bit 0 **ACE8**: 定义 PC0 是否为 A/D 输入
0: 不是 A/D 输入
1: A/D 输入, AN8

A/D 操作

ADCR0 寄存器中的 START 位，用于打开和复位 A/D 转换器。当单片机设定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。当 START 位从逻辑低到逻辑高，但不再回到逻辑低时，ADCR0 寄存器中的 EOCB 位置“1”，复位模数转换器。START 位用于控制内部模数转换器的开启动作。

ADCR0 寄存器中的 EOCB 位用于表明模数转换过程的完成。在转换周期结束后，EOCB 位会被单片机自动地置为“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到相应的 A/D 内部中断入口。如果 A/D 内部中断被禁止，可以让单片机轮询 ADCR0 寄存器中的 EOCB 位，检查此位是否被清除，以作为另一种侦测 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} 分频，而分频系数由 ADCR1 寄存器中的 ADCK2~ADCK0 位决定。

虽然 A/D 时钟源是由系统时钟 f_{sys} ，ADCK2~ADCK0 位决定，但可选择的最大 A/D 时钟源则有一些限制。允许的 A/D 时钟周期 t_{AD} 的最小值为 $0.5\mu s$ ，当系统时钟速度等于或超过 4MHz 时就必须小心。如果系统时钟速度为 4MHz 时，ADCK2~ADCK0 位不能设为“000”。必须保证设定的 A/D 转换时钟周期不小于时钟周期的最小值，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，被标上星号 * 的数值是不允许的，因为它们的 A/D 转换时钟周期小于规定的最小值。

f_{sys}	A/D 时钟周期 (t_{AD})							
	ADCK2, ADCK1, ADCK0 =000 (f_{sys})	ADCK2, ADCK1, ADCK0 =001 ($f_{sys}/2$)	ADCK2, ADCK1, ADCK0 =010 ($f_{sys}/4$)	ADCK2, ADCK1, ADCK0 =011 ($f_{sys}/8$)	ADCK2, ADCK1, ADCK0 =100 ($f_{sys}/16$)	ADCK2, ADCK1, ADCK0 =101 ($f_{sys}/32$)	ADCK2, ADCK1, ADCK0 =110 ($f_{sys}/64$)	ADCK2, ADCK1, ADCK0 =111
1MHz	1 μs	2 μs	4 μs	8 μs	16 μs	32 μs	64 μs	未定义
2MHz	500ns	1 μs	2 μs	4 μs	8 μs	16 μs	32 μs	未定义
4MHz	250ns*	500ns	1 μs	2 μs	4 μs	8 μs	16 μs	未定义
8MHz	125ns*	250ns*	500ns	1 μs	2 μs	4 μs	8 μs	未定义
12MHz	83ns*	167ns*	333ns*	667ns	1.33 μs	2.67 μs	5.33 μs	未定义
16MHz	62ns*	125ns*	250ns*	500ns	1 μs	2 μs	4 μs	未定义

A/D 时钟周期范例

ADCR0 寄存器的 ADOFF 位用于控制 A/D 转换电路电源的开 / 关。该位必须清零以开启 A/D 转换器电源。即使通过清除 ACER1~ACER0 寄存器的 ACE15~ACE0 位，选择无引脚作为 A/D 输入，如果 ADOFF 设为“0”，那么仍然会产生功耗。因此当未使用 A/D 转换器功能时，在功耗敏感的应用中建议设置 ADOFF 为高以减少功耗。

A/D 转换器参考电压来自正电源电压 VDD 或外部参考源引脚 VREF，可通过 VREFS 位来选择。由于 VREF 引脚与其它功能共用，当 VREFS 设为高，选择 VREF 引脚功能且其它引脚功能将自动除能。

A/D 输入引脚

所有的 A/D 模拟输入引脚都与 I/O 引脚及其它功能共用。使用 ACER1~ACER0 寄存器中的 ACE15~ACE0 位，可以将它们设置为 A/D 转换器模拟输入脚或具有其它功能。如果引脚的对应位 ACE15~ACE0 设为高，那么该引脚作为 A/D 转换输入且原引脚功能除能。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，PAC、PBC、PCC 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 ACE15~ACE0 位使能 A/D 输入时，端口控制寄存器的状态将被重置。

A/D 转换器有自己的参考电压引脚 VREF，而通过设置 ADCR1 寄存器的 VREFS 位，参考电压也可以选择来自电源电压引脚。模拟输入值一定不能超过 V_{REF} 值。

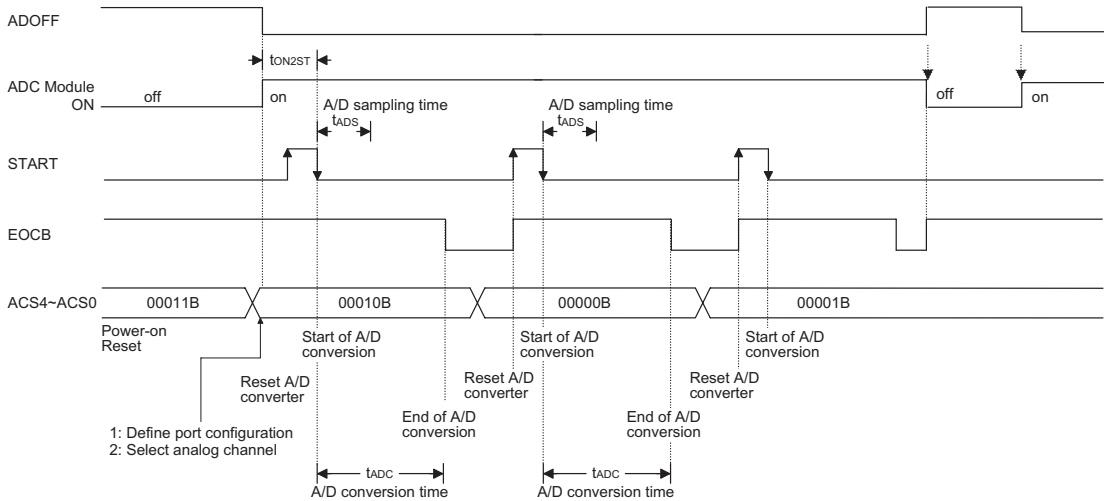
A/D 转换步骤

下面概述实现 A/D 转换过程的各个步骤。

- 步骤 1
通过 ADCR1 寄存器中的 ADCK2~ADCK0 位，选择所需的 A/D 转换时钟。
- 步骤 2
清零 ADCR0 寄存器中的 ADOFF 位使能 A/D。
- 步骤 3
通过 ADCR1 和 ADCR0 寄存器中 ACS4~ACS0 位，选择连接至内部 A/D 转换器的通道。
- 步骤 4
通过 ACER1~ACER0 寄存器中的 ACE15~ACE0 位，选择哪些引脚规划为 A/D 输入引脚。
- 步骤 5
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 转换功能是激活的。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 6
现在可以通过设定 ADCR0 寄存器中的 START 位从“0”到“1”再回到“0”，开始模数转换的过程。注意，该位需初始化为“0”。
- 步骤 7
可以轮询 ADCR0 寄存器中的 EOCB 位，检查模数转换过程是否完成。当此位成为逻辑低时，表示转换过程已经完成。转换完成后，可读取 A/D 数据寄存器 ADRL 和 ADRH 获得转换后的值。另一种方法是，若中断使能且堆栈未满，则程序等待 A/D 中断发生。

注：若使用轮询 ADCR0 寄存器中 EOCB 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。

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 $16t_{AD}$ ， t_{AD} 为 A/D 时钟周期。



A/D 转换时序图

编程注意事项

在编程时，如果 A/D 转换器未使用，通过设置 ADCR0 寄存器中的 ADOFF 为高，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

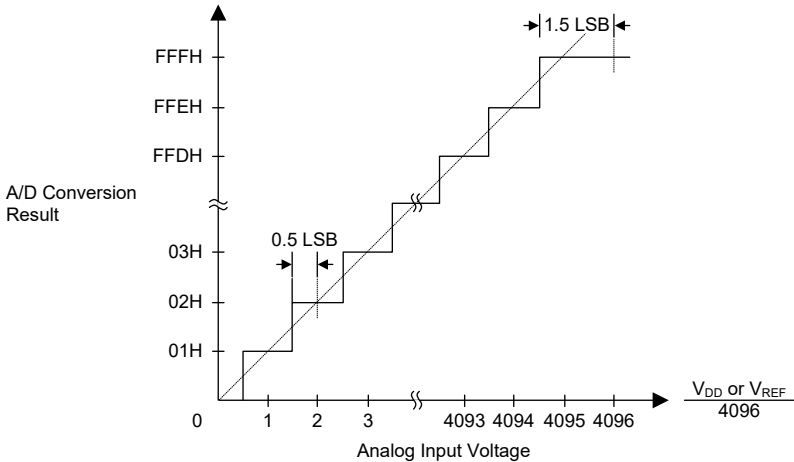
此系列单片机含有一组 12 位的 A/D 转换器，它们转换的最大值可达 FFFH。由于模拟输入最大值等于 V_{DD} 或 V_{REF} 的电压值，因此每一位可表示 $(V_{DD}$ 或 $V_{REF})/4096$ 的模拟输入值。

$1\text{ LSB}=(V_{DD}\text{ 或 }V_{REF})\div4096$

通过下面的等式可估算 A/D 转换器输入电压值：

$A/D\text{ 输入电压}=A/D\text{ 数字输出值}\times(V_{DD}\text{ 或 }V_{REF})\div4096$

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_{DD} 或 V_{REF} 之前的 1.5 LSB 处改变。



理想的 A/D 转换功能

A/D 转换应用范例

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 ADCR0 寄存器中的 EOCB 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：使用查询 EOCB 的方式来检测转换结束

```
clr ADE                      ; disable ADC interrupt
mov a,03H
mov ADCR1,a                  ; select fsys/8 as A/D clock and switch off 1.25V
clr ADOFF
clr ACER1
mov a,0Fh                    ; setup ACER to configure pins AN0~AN3
mov ACER0,a
mov a,00h
mov ADCR0,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 EOCB                      ; poll the ADCR0 register EOCB bit to detect end
                                ; of A/D conversion
jmp polling_EOC              ; continue polling
mov a,ADRL                   ; read low byte conversion result value
mov ADRL_buffer,a            ; save result to user defined register
mov a,ADRH                   ; read high byte conversion result value
mov ADRH_buffer,a            ; save result to user defined register
:
:
jmp start_conversion          ; start next A/D conversion
```

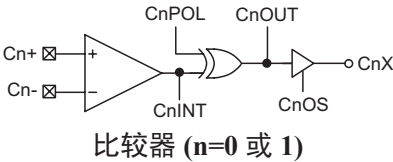
范例：使用中断的方式来检测转换结束

```
clr ADE                      ; disable ADC interrupt
mov a,03H
mov ADCR1,a                  ; select fsys/8 as A/D clock and switch off 1.25V
clr ADOFF
clr ACER1
mov a,0Fh                    ; setup ACER to configure pins AN0~AN3
mov ACER0,a
mov a,00h
mov ADCR0,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,ADRL              ; read low byte conversion result value
mov adrl_buffer,a       ; save result to user defined register
mov a,ADRH              ; read high byte conversion result value
mov adrh_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
```

比较器

HT66FB540/HT66FB550/HT66FB560 含有两个独立的模拟比较器，而 HT66FB542 只含有一个模拟比较器。它们具有暂停、极性选择、迟滞等功能，可通过寄存器进行灵活配置。比较器的引脚与普通 I/O 引脚共用，当比较器功能未使用时，此引脚可做普通引脚使用而不浪费 I/O 资源。



比较器操作

此系列单片机包含一个或两个比较器功能，用于比较两个模拟电压，基于它们的差值上提供一个输出。控制寄存器 CP0C 和 CP1C 可分别控制相应的内部比较器。比较器的输出可由各自寄存器的一位记录，并且在共用的 I/O 口上输出。此外，比较器功能有输出极性，迟滞功能和暂停控制。

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

比较器寄存器

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

寄存器名称	单片机型号	位							
		7	6	5	4	3	2	1	0
CP0C	HT66FB540 HT66FB542 HT66FB550 HT66FB560	C0SEL	C0EN	C0POL	C0OUT	C0OS	—	—	C0HYEN

寄存器名称	单片机型号	位							
		7	6	5	4	3	2	1	0
CP1C	HT66FB540 HT66FB550 HT66FB560	C1SEL	C1EN	C1POL	C1OUT	C1OS	—	—	C1HYEN

比较器寄存器列表

• CP0C 寄存器

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

- Bit 7 C0SEL:** 比较器引脚或 I/O 引脚选择位
0: 输入 / 输出引脚
1: 比较器引脚
此位为比较器引脚或输入 / 输出引脚选择位。为高时，两个比较器输入引脚使能。此时，这两个引脚的输入 / 输出功能失效，任何与比较器共用引脚的上拉电阻配置选项将自动失效。
- Bit 6 C0EN:** 比较器开 / 关控制位
0: 关闭
1: 开启
此位为比较器开 / 关控制位。为“0”时，比较器关闭，即使其引脚上加有模拟电压也不会产生功耗。对功耗要求严格的应用中，当比较器未使用或单片机进入休眠或空闲模式之前，此位应清零。
- Bit 5 C0POL:** 比较器输出极性位
0: 输出同相
1: 输出反相
此位决定比较器极性。为“0”时，C0OUT 位与比较器输出条件同相；为“1”时，C0OUT 位与比较器输出条件反相。
- Bit 4 C0OUT:** 比较器输出位
C0POL=0
0: $C0+ < C0-$
1: $C0+ > C0-$
C0POL=1
0: $C0+ > C0-$
1: $C0+ < C0-$
此位为比较器输出位。此位的极性由比较器输入电压和 C0POL 位的状态决定。
- Bit 3 C0OS:** 输出路径选择位
0: C0X 引脚
1: 内部使用
此位为比较器输出路径选择控制位。当此位为“0”且 C0SEL 位为“1”时，比较器输出连接到外部 C0X 引脚。当此位为“1”或 C0SEL 位为“0”时，比较器输出信号仅供单片机内部使用，且共用输出引脚作为普通输入 / 输出引脚使用。
- Bit 2~1** 未使用，读为“0”
- Bit 0 C0HYEN:** 迟滞控制位
0: 关闭
1: 开启
此位为迟滞控制位。为“1”时，比较器有一定量迟滞，具体见比较器电气特性表。滞后产生的正反馈将减少比较器门槛附近的伪开关效应的影响。

● CP1C 寄存器 – HT66FB540/HT66FB550/HT66FB560

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

- Bit 7 **C1SEL:** 比较器引脚或输入 / 输出引脚选择位
0: 输入 / 输出引脚
1: 比较器引脚
此位为比较器引脚或输入 / 输出引脚选择位。为高时，两个比较器输入引脚使能。此时，这两个引脚的输入 / 输出功能失效，任何与比较器共用引脚的上拉电阻配置选项将自动失效。
- Bit 6 **C1EN:** 比较器开 / 关控制位
0: 关闭
1: 开启
此位为比较器开 / 关控制位。为“0”时，比较器关闭，即使其引脚上加有模拟电压也不会产生功耗。对功耗要求严格的应用中，当比较器未使用或单片机进入休眠或空闲模式之前，此位应清零。
- Bit 5 **C1POL:** 比较器输出极性位
0: 输出同相
1: 输出反相
此位决定比较器极性。为“0”时，C1OUT 位与比较器输出条件同相；为“1”时，C1OUT 位与比较器输出条件反相。
- Bit 4 **C1OUT:** 比较器输出位
C1POL=0
0: $C1+ < C1-$
1: $C1+ > C1-$
C1POL=1
0: $C1+ > C1-$
1: $C1+ < C1-$
此位为比较器输出位。此位的极性由比较器输入电压和 C1POL 位的状态决定。
- Bit 3 **C1OS:** 输出路径选择位
0: C1X 引脚
1: 内部使用
此位为比较器输出路径选择控制位。当此位为“0”且 C1SEL 位为“1”时，比较器输出连接到外部 C1X 引脚。当此位为“1”或 C1SEL 位为“0”时，比较器输出信号仅供单片机内部使用，且共用输出引脚作为普通输入 / 输出引脚使用。
- Bit 2~1 未使用，读为“0”
- Bit 0 **C1HYEN:** 迟滞控制位
0: 关闭
1: 开启
此位为迟滞控制位。为“1”时，比较器有一定量迟滞，具体见比较器电气特性表。滞后产生的正反馈将减少比较器门槛附近的伪开关效应的影响。

比较器中断

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

编程注意事项

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

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

串行接口模块 – SIM

该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 I/O 引脚共用，所以要使用 SIM 功能时应先在配置选项中选中 SIM 功能。因为这两种接口共用引脚和寄存器，所以要通过一个 SIMC0 寄存器中的 SIM2~SIM0 位来选择哪一种通信接口。若 SIM 功能使能，可通过上拉电阻控制寄存器选择与输入 / 输出共用的 SIM 脚的上拉电阻。

串行接口有一个控制寄存器 SBSC，用于使能 SIM WCOL 功能、SPIA WCOL 功能以及 I²C 的去抖选择。

该系列单片机提供两种 SPI 功能，SPI 和 SPIA，它们都有相应的 WCOL 控制位 (SIM_WCOL 和 SA_WCOL) 来使能 SIM WCOL 和 SPIA WCOL。此外，I2CDB1 和 I2CDB0 位用于选择 I²C 的去抖时间。

SPI 接口

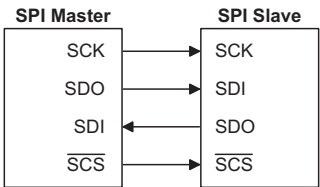
SPI 属于串行接口模块部分，不要与其它独立的 SPI 功能混淆，如 SPIA，此功能详细描述见规格书其它章节。

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

SPI 通信模式为全双工模式，且能以主 / 从模式的工作方式进行通信，单片机既可以作为主机，也可以作为从机。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 接口是一个全双工串行数据传输器。SPI 接口的四线为：SDI、SDO、SCK 和 $\overline{SCS}$ 。SDI 和 SDO 是数据的输入和输出线。SCK 是串行时钟线， $\overline{SCS}$ 是从机的选择线。SPI 的接口引脚与普通 I/O 口和 I²C 的功能脚共用。通过设定 SIMC0/SIMC2 寄存器的对应位，来使能 SPI 接口。SPI 设定好后，还可以通过 SIMC0 寄存器中的 SIMEN 位来除能或使能。

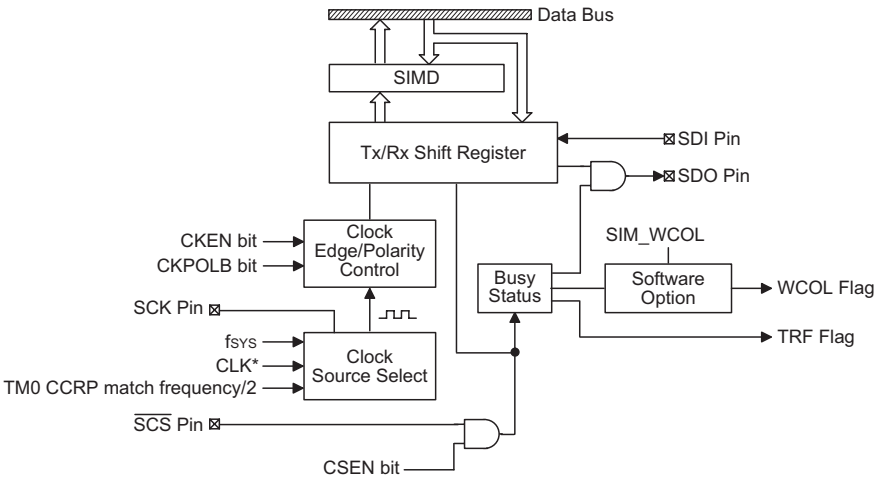


SPI 主 / 从机连接方式

该系列单片机的 SPI 功能具有以下特点：

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

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



Note: CLK = f_{TBC} for HT66FB540/HT66FB550/HT66FB560, CLK = f_L for HT66FB542.

SPI 方框图

SPI 寄存器

有四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、三个控制寄存器 SIMC0、SIMC2 和 SBSC。注意，SIMC1 寄存器仅用于 I²C 接口。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMC2	—	—	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
SBSC	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL

SIM 寄存器列表

SIMD 用于存储发送和接收的数据。这个寄存器由 SPI 和 I²C 功能所共用。在单片机尚未将数据写入到 SPI 总线中时，要传输的数据应先存在 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

单片机中也有三个控制 SPI 接口功能的寄存器，SIMC0、SIMC2 和 SBSC。应注意的是 SIMC2 与 I²C 接口功能中的的寄存器 SIMA 是同一个寄存器。SPI 功能不会用到寄存器 SIMC1，SIMC1 只适用于 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。虽然 SIMC0 与 SPI 功能无关，但是也用于控制外部时钟分频。寄存器 SIMC2 用于其它的控制功能如 LSB/MSB 选择，写冲突标志位等。SBSC 寄存器中的 SIM_WCOL 位用于控制 SPI WCOL 功能。

● SIMC0 寄存器

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

Bit 7~5 SIM2~SIM0: SIM 工作模式控制位

- 000: SPI 主机模式；SPI 时钟为 $f_{sys}/4$
- 001: SPI 主机模式；SPI 时钟为 $f_{sys}/16$
- 010: SPI 主机模式；SPI 时钟为 $f_{sys}/64$
- 011: SPI 主机模式；SPI 时钟为 f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
- 100: SPI 主机模式；SPI 时钟为 TM0 CCRP 匹配频率 /2
- 101: SPI 从机模式
- 110: I²C 从机模式
- 111: 未使用模式

这几位用于设置 SIM 功能的工作模式，用于选择 SPI 的主从模式和 SPI 的主机时钟频率及 I²C 或 SPI 功能。SPI 时钟源可来自于系统时钟也可以选择来自 TM0 或 f_{TBC} (f_L)。若选择的是作为 SPI 从机，则其时钟源从外部主机而得。

Bit 4 PCKEN: PCK 输出脚控制位

- 0: 除能
- 1: 使能

Bit 3~2 PCKP1~PCKP0: 选择 PCK 输出脚的频率

- 00: f_{sys}
- 01: $f_{sys}/4$
- 10: $f_{sys}/8$
- 11: TM0 CCRP 匹配频率 /2

Bit 1 SIMEN: SIM 控制位

- 0: 除能
- 1: 使能

此位为 SIM 接口的开 / 关控制位。此位为“0”时，SIM 接口除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 脚处于浮空状态，SIM 工作电流减小到最小值。此位为“1”时，SIM 接口使能。配置选项中首先将 SIM 接口使能才能使此位有效。若 SIM 经由 SIM2~SIM0 位设置为工作在 SPI 接口，当 SIMEN 位由低到

高转变时，SPI 控制寄存器中的设置不会发生变化，其首先应在应用程序中初始化。若 SIM 经由 SIM2~SIM0 位设置为工作在 I²C 接口，当 SIMEN 位由低到高转变时，I²C 控制寄存器中的设置，如 HTX 和 TXAK，将不会发生变化，其首先应在应用程序中初始化，此时相关 I²C 标志，如 HCF、HAAS、HBB、SRW 和 RXAK，将被设置为其默认状态。

Bit 0 未使用，读为“0”

• SIMC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
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 引脚为高电平

1: 当时钟无效时，SCK 引脚为低电平

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 SCS 引脚控制位

0: 除能

1: 使能

CSEN 位用于 $\overline{\text{SCS}}$ 引脚的使能 / 除能控制。此位为低时， $\overline{\text{SCS}}$ 除能并处于浮空状态。此位为高时，SCS 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。注意，WCOL 位的使能 / 除能可通过 SBSC 寄存器中的 SIM_WCOL 位设置。

Bit 0 **TRF**: SPI 发送 / 接收结束标志位

0: 数据正在发送

1: 数据发送结束

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

• SBSC 寄存器

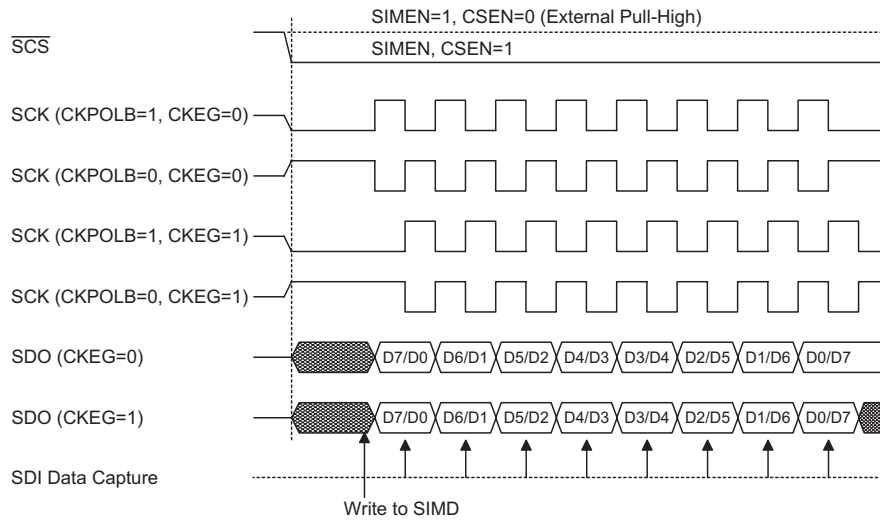
Bit	7	6	5	4	3	2	1	0
Name	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL
R/W	R/W	—	R/W	R/W	—	—	—	R/W
POR	0	—	0	0	—	—	—	0

- Bit 7 **SIM_WCOL**: SIM WCOL 功能控制位
0: 除能
1: 使能
- Bit 6 未使用, 读为 “0”
- Bit 5~4 **I2CDB1~I2CDB0**: I²C 去抖时间选择位
I²C 功能相关位, 具体描述详见其它章节
- Bit 3~1 未使用, 读为 “0”
- Bit 0 **SA_WCOL**: SPIA WCOL 功能控制位
SPIA 功能相关位, 具体描述详见其它章节

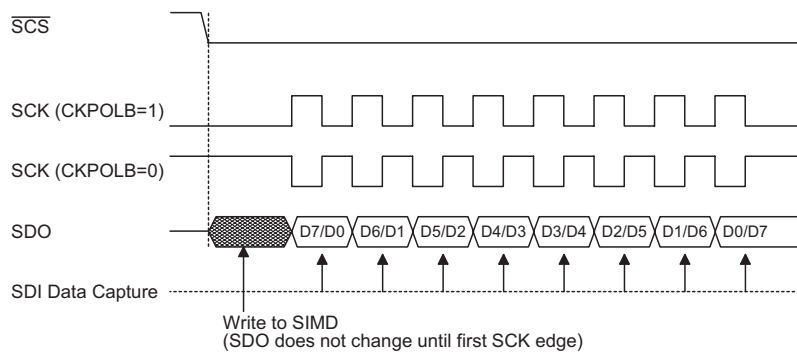
SPI 通信

将 SIMEN 设置为高, 使能 SPI 功能之后, 单片机处于主机模式, 当数据写入到寄存器 SIMD 的同时传输 / 接收开始进行。数据传输完成时, TRF 位将自动被置位但清除只能通过应用程序完成。单片机处于从机模式时, 收到主机发来的信号之后, 会传输 SIMD 中的数据, 而且在 SDI 引脚上的数据也会被移位到 SIMD 寄存器中。主机应在输出时钟信号之前先输出一个 SCS 信号以使能从机, 从机的数据传输功能也应在与 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 SCS 信号的关系。

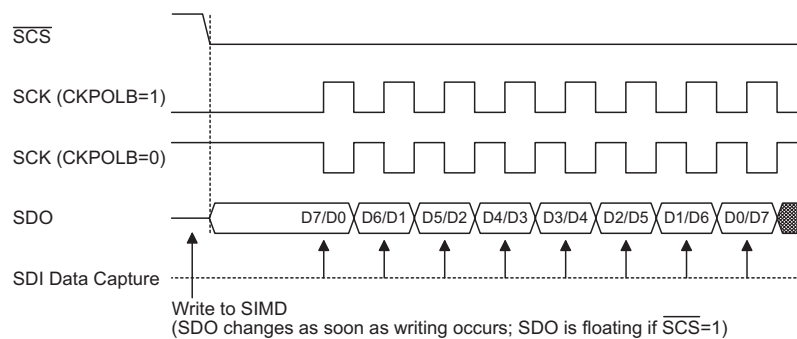
SPI 主机模式在 SPI 时钟运行情况下可以进行正常传输。



SPI 主机模式时序

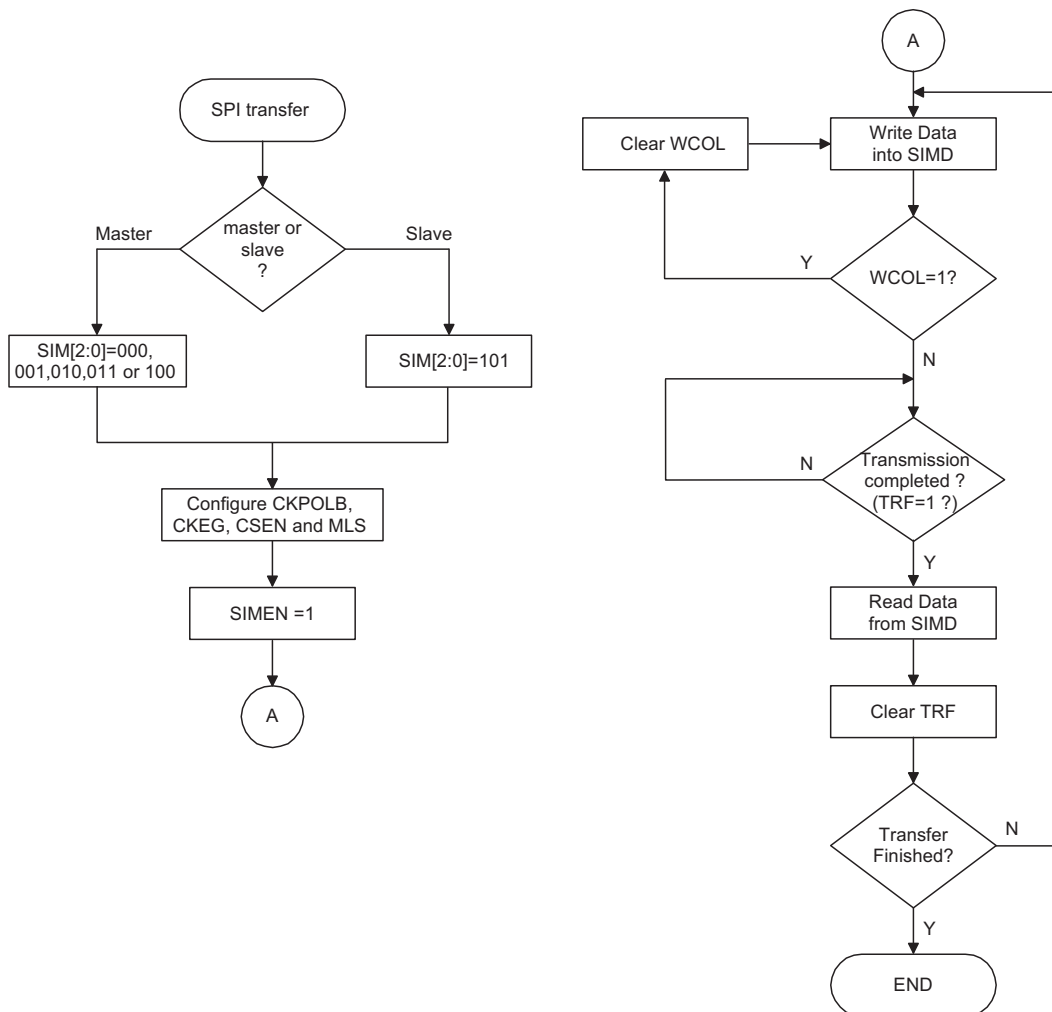


SPI 从机模式时序 – CKEG=0



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

SPI 从机模式时序 – CKEG=1



SPI 传输控制流程图

SPI 使能 / 除能

设置 $\overline{\text{CS}}\text{EN}=1$ 、 $\overline{\text{SCS}}=0$ 将使能 SPI 总线，然后等待写数据到 SIMD 寄存器 (TXRX 缓存器)。单片机处于主机模式，数据写入 SIMD 寄存器后，自动开始数据传输或接收操作。数据传输完成时，TRF 位将自动被置位。单片机处于从机模式，SCK 引脚上收到脉冲信号之后，会传输 TXRX 中的数据，或 SDI 引脚上的数据也会被移入。

除能 SPI 总线，SCK、SDI、SDO、 $\overline{\text{SCS}}$ 将作为 I/O 口或其它功能引脚使用。

SPI 操作

四线制 SPI 接口可完成所有主 / 从模式通信工作。

在 SIMC2 寄存器中，CSEN 位控制 SPI 接口的所有功能。设置此位为高， $\overline{\text{SCS}}$ 信号线有效将使能 SPI 接口。设置此位为低，SPI 接口将除能， $\overline{\text{SCS}}$ 信号线作为 I/O 口或其它功能引脚使用，此时不能使用 SPI 功能。CSEN 位和 SIMC0 寄存器中的 SIMEN 位设置为高，使得 SDI 信号线处于浮空状态且 SDO 信号线为高电平。主机模式中，如果 SCK 信号线为高还是低取决于 SIMC2 寄存器中的时钟极性选择位 CKPOLB。从机模式中，SCK 信号线处于浮空状态。如果

SIMEN 位设置为低，SPI 接口被除能，且 $\overline{SCS}$ 、SDI、SDO 和 SCK 将作为 I/O 口或其它功能引脚使用。主机模式中，当数据被写入 SIMD 寄存器后，主机完成所有的数据传输初始化，并控制时钟信号。从机模式中，由外部主机发出数据传送 / 接收时钟信号。下面介绍主从模式中数据传输步骤。

主机模式：

- 步骤 1
设置 SIMC0 控制寄存器中的 SIM2~SIM0 位，选择 SPI 主机模式和时钟源。
- 步骤 2
设置 CSEN 和 MLS 位，选择高位或低位数据优先传送，这必须与从机设备一致。
- 步骤 3
设置 SIMC0 控制寄存器中的 SIMEN 位，使能 SPI 接口功能。
- 步骤 4
对于写操作：写数据到 SIMD 寄存器，实际上，数据被存储在 TXRX 缓存器中。再使用 SCK 和 SDO 信号线将数据输出。跳至步骤 5。对于读操作：使用 SDI 信号线将 TXRX 缓存器中的数据移出，并全部锁存至 SIMD 寄存器。
- 步骤 5
检测 WCOL 位，若此位为高，则发生数据冲突并跳回至步骤 4；若为低，则继续执行下面的步骤。
- 步骤 6
检测 TRF 位或等待 SPI 串行总线中断发生。
- 步骤 7
从 SIMD 寄存器中读数据。
- 步骤 8
清除 TRF。
- 步骤 9
跳回至步骤 4。

从机模式：

- 步骤 1
设置 SIMC0 控制寄存器中的 SIM2~SIM0 位，选择 SPI 从机模式。
- 步骤 2
设置 CSEN 和 MLS 位，选择高位或低位数据优先传送，这必须与主机设备一致。
- 步骤 3
设置 SIMEN 位，使能 SPI 接口功能。
- 步骤 4
对于写操作：写数据到 SIMD 寄存器，实际上，数据被存储在 TXRX 缓存器中。等待主机时钟 SCK 信号和 $\overline{SCS}$ 信号。跳至步骤 5。对于读操作：使用 SDI 信号线将 TXRX 缓存器中的数据移出，并全部锁存至 SIMD 寄存器。
- 步骤 5
检测 WCOL 位，若此位为高，则发生数据冲突并跳回至步骤 4；若为低，则继续执行下面的步骤。

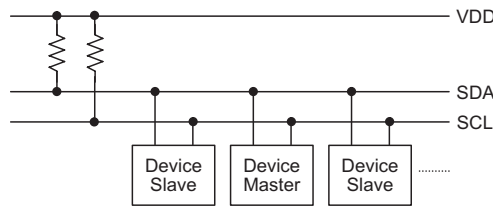
- 步骤 6
检测 TRF 位或等待 SPI 串行总线中断发生。
- 步骤 7
从 SIMD 寄存器中读数据。
- 步骤 8
清除 TRF。
- 步骤 9
跳回至步骤 4。

错误侦测

SIMC2 寄存器中的 WCOL 位用于数据传输期间监测数据冲突的发生。此位由 SPI 串行接口设置为高，而由应用程序来清除为零。在数据传输期间，如果写数据到 SIMD 时又阻止数据继续被写入，此时数据冲突发生。可通过 SBSC 寄存器中的 SIM_WCOL 位除能或使能。

I²C 接口

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



I²C 主从总线连接图

I²C 接口操作

I²C 串行接口是一个双线的接口，有一条串行数据线 SDA 和一条串行时钟线 SCL。由于可能有多个设备在同一条总线上相互连接，所以这些设备的输出都是开漏型输出。因此应在这些输出上都加上拉电阻。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。

SBSC 寄存器中的 I2CDB1 和 I2CDB0 位用于控制 I²C 接口的去抖时间大小。这个功能可以使内部时钟在外部时钟上增加一个去抖间隔，会减小时钟线上毛刺发生的可能性，以避免单片机发生误动作。如果选择了这个功能，去抖时间可以选择 1 个或 2 个系统时钟。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
SIMC1	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMA	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	—
SBSC	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL

I²C 寄存器列表

● SIMC0 寄存器

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

Bit 7~5 **SIM2~SIM0**: SIM 工作模式控制位

- 000: SPI 主机模式; SPI 时钟为 $f_{sys}/4$
- 001: SPI 主机模式; SPI 时钟为 $f_{sys}/16$
- 010: SPI 主机模式; SPI 时钟为 $f_{sys}/64$
- 011: SPI 主机模式; SPI 时钟为 f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
- 100: SPI 主机模式; SPI 时钟为 TM0 CCRP 匹配频率 /2
- 101: SPI 从机模式
- 110: I²C 从机模式
- 111: 未使用模式

这几位用于设置 SIM 功能的工作模式, 用于选择 SPI 的主从模式和 SPI 的主机时钟频率及 I²C 或 SPI 功能。SPI 时钟源可来自于系统时钟也可以选择来自 TM0 或 f_{TBC} (f_L)。若选择的是作为 SPI 从机, 则其时钟源从外部主机而得。

Bit 4 **PCKEN**: PCK 输出脚控制位

- 0: 除能
- 1: 使能

Bit 3~2 **PCKP1~PCKP0**: 选择 PCK 输出脚的频率

- 00: f_{sys}
- 01: $f_{sys}/4$
- 10: $f_{sys}/8$
- 11: TM0 CCRP 匹配频率 /2

Bit 1 **SIMEN**: SIM 控制位

- 0: 除能
- 1: 使能

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

Bit 0 未使用, 读为“0”

● SIMC1 寄存器

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

- Bit 7 HCF:** 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
数据正在传输时该位为低。当 8 位数据传输完成时, 此位为高并产生一个中断。
- Bit 6 HAAS:** I²C 地址匹配标志位
0: 地址不匹配
1: 地址匹配
此标志位用于决定从机地址是否与主机发送地址相同。若地址匹配此位为高, 否则此位为低。
- Bit 5 HBB:** I²C 总线忙标志位
0: I²C 总线闲
1: I²C 总线忙
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:** 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
单片机接收 8 位数据之后会将该位在第九个时钟传到总线上。如果单片机想要接收更多的数据, 则应在接收数据之前将此位设置为“0”。
- Bit 2 SRW:** 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 位是从机读写位。决定主机是否希望传输或接收来自 I²C 总线的的数据。当传输地址和从机的地址相同时, HAAS 位会被设置为高, 主机将检测 SRW 位来决定进入发送模式还是接收模式。如果 SRW 位为高时, 主机会请求从总线上读数据, 此时设备处于传输模式。当 SRW 位为“0”时, 主机往总线上写数据, 设备处于接收模式以读取该数据。
- Bit 1 IAMWU:** I²C 地址匹配唤醒控制位
0: 除能
1: 使能
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。
- 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受到一个正确的确认位。如果单片机处于发送状态, 发送方会检查 RXAK 位来判断接收方是否愿意继续接收下一个字节。因此直到 RXAK 为“1”时, 传输方停止发送数据。这时, 传输方将释放 SDA 线, 主机发出停止信号。

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

• SIMD 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	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	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	—
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”：未知

Bit 7~1 **IICA6~IICA0**: I²C 从机地址位

IICA6~IICA0 是从机地址对应的 6~0 位。

此寄存器也在 SPI 接口功能中使用，但其名称改为 SIMC2。SIMA 寄存器用于存放 7 位从机地址，寄存器 SIMA 中的第 7~1 位是单片机的从机地址，位 0 未定义。

如果接至 I²C 的主机发送处的地址和寄存器 SIMA 中存储的地址相符，那么就选中了这个从机。应注意的是寄存器 SIMA 和 SPI 接口使用的寄存器 SIMC2 是同一个寄存器。

Bit 0 未定义位

此位可通过软件程序进行读写。

• SBSC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL
R/W	R/W	—	R/W	R/W	—	—	—	R/W
POR	0	—	0	0	—	—	—	0

Bit 7 **SIM_WCOL**: SIM WCOL 功能控制位

SPI 功能相关位，具体描述详见其它章节

Bit 6 未使用，读为“0”

Bit 5~4 **I2CDB1~I2CDB0**: I²C 去抖时间选择位

00: 无去抖 (默认)

01: 1 个系统时钟去抖

1X: 2 个系统时钟去抖

Bit 3~1 未使用，读为“0”

Bit 0 **SA_WCOL**: SPIA WCOL 功能控制位

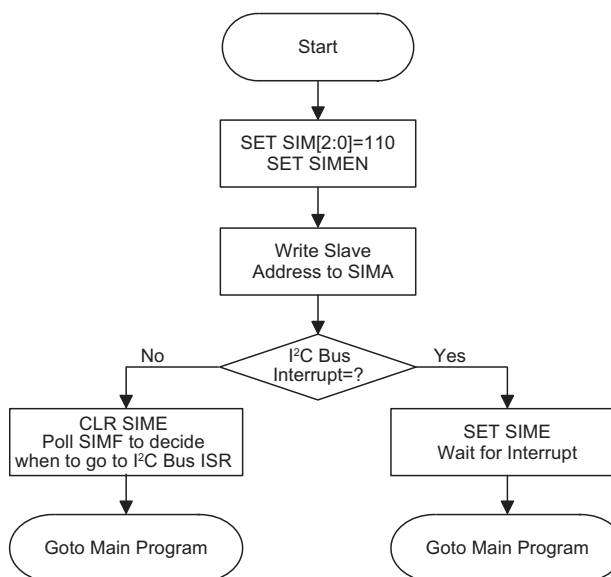
SPIA 功能相关位，具体描述详见其它章节

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 总线超时。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”，以使能 I²C 总线。
- 步骤 2
向 I²C 总线地址寄存器 SIMA 写入从机地址。
- 步骤 3
设置 SIME 位，以使能 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 I²C 总线主机产生，而不是由只做从机的 MCU 产生。总线上的所有从机都可以侦测到起始信号。如果有从机侦测到起始信号，则表明 I²C 总线处于忙碌状态，并会置位 HBB。起始信号是指在 SCL 为高电平时，SDA 线上发生从高到低的电平变化。

从机地址

总线上的所有从机都会侦测由主机发出的起始信号。发送起始信号后，紧接着主机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 总线有三个中断源，当程序运行至中断服务子程序时，通过检测 HAAS 位

和 I2CTOF 位以确定 I²C 总线中断是来自从机地址匹配，还是来自 8 位数据传输完毕，或是来自 I²C 总线超时。当是从机地址匹配发生中断时，则从机或是用于发送模式并将数据写进 SIMD 寄存器，或是用于接收模式并从 SIMD 寄存器中读取空值以释放 SCL 线。

I²C 总线读 / 写信号

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

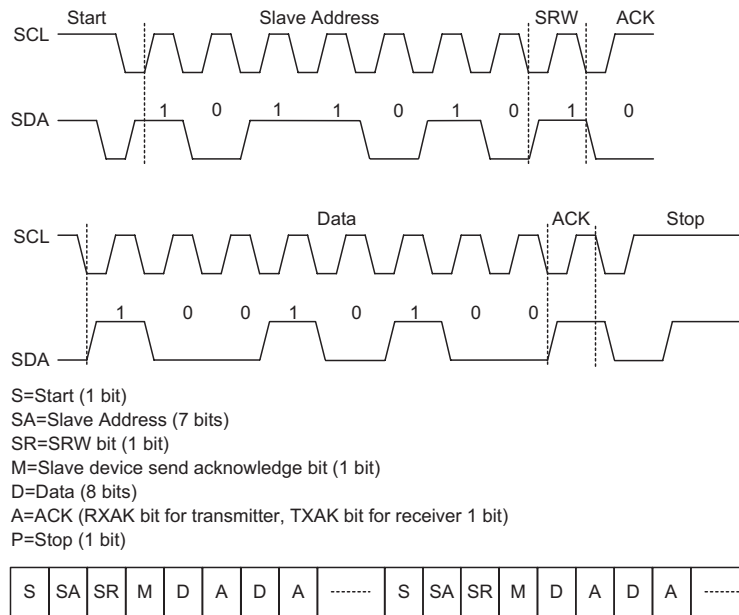
I²C 总线从机地址确认信号

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

I²C 总线数据和确认信号

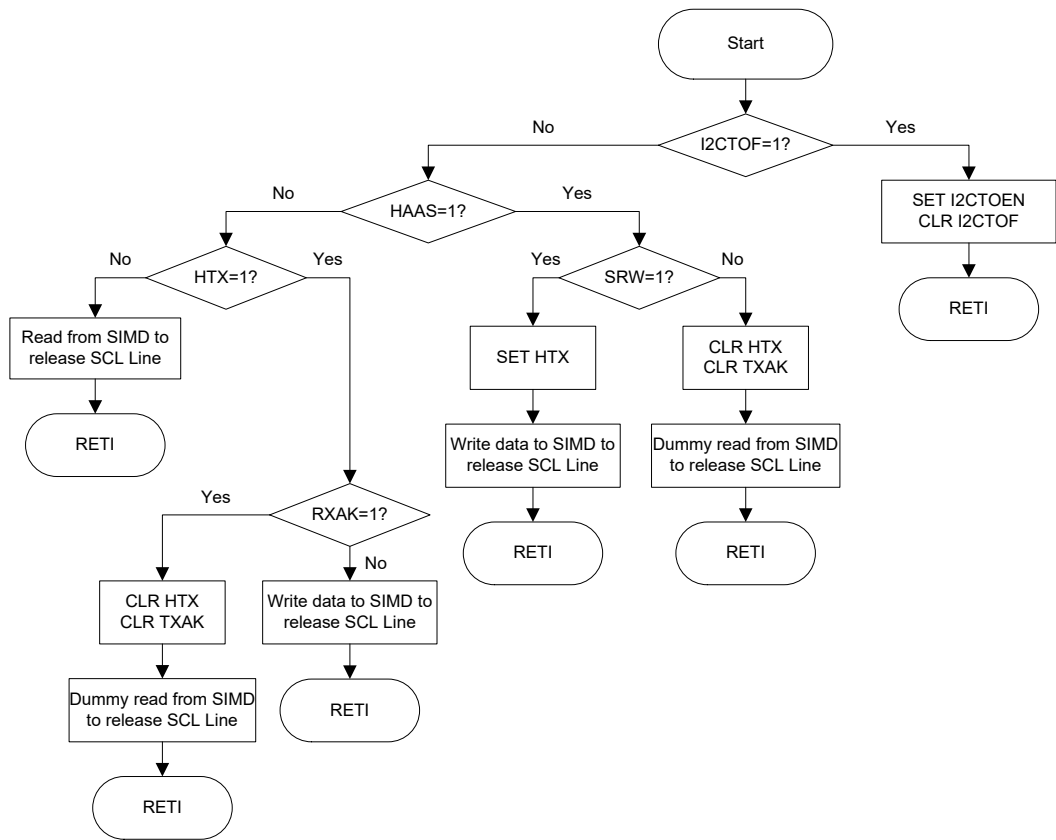
在从机确认接收到从机地址后，会进行 8 位宽度的数据传输。这个数据传输顺序是的高位在前，低位在后。接收方在接收到 8 位数据后必须发出一个应答信号 (“0”) 以继续接收下一个数据。如果发送方没接收到应答信号，发送方将释放 SDA 线，同时，主机将发出 STOP 信号以释放 I²C 总线。所传送的数据存储在 SIMD 寄存器中。如果设置成发送方，从机必须先将欲传输的数据写到 SIMD 寄存器中；如果设置成接收方，从机必须从 SIMD 寄存器读取数据。

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



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

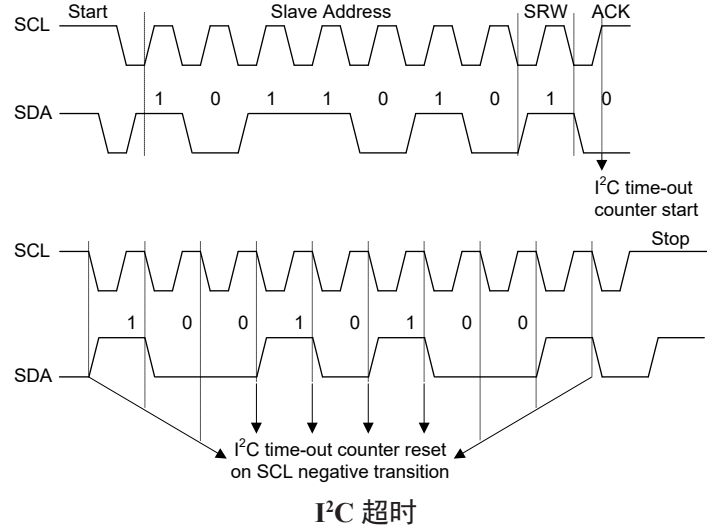
I²C 通信时序图



I²C 总线 ISR 流程图

I²C 超时功能

为了减少由于接收错误时钟源而产生的 I²C 锁定问题，系统提供了超时功能。在固定时间内如果 I²C 总线未接收到时钟源，则 I²C 电路和 SIMC1 寄存器将会复位，I2CTOC 寄存器的 I2CTOF 位将置高。超时功能的使能 / 除能和超时时间都由 I2CTOC 寄存器控制。



I²C 超时操作

超时计数器在 I²C 总线接收到“START”信号和“地址匹配”条件时，超时计数器开始计数，并在 SCL 下降沿处清零。在下一个 SCL 下降沿来临之前，如果等待时间大于 I2CTOC 寄存器设定的超时时间，则会发生超时现象，超时计数器由硬件自动停止，I2CTOF 位被置高，产生 I²C 中断。注意，当 I²C 接收到“STOP”信号，超时计数器将停止计数。多种超时时间可由 I2CTOC 寄存器的 I2CTOS0~I2CTOS5 位选择。

• I2CTOC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	I2CTOEN	I2CTOF	I2CTOS5	I2CTOS4	I2CTOS3	I2CTOS2	I2CTOS1	I2CTOS0
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 **I2CTOEN:** I²C 超时控制位

0: 除能
1: 使能

Bit 6 **I2CTOF:** I²C 超时标志位

0: 未发生
1: 发生

Bit 5~0 **I2CTOS5~I2CTOS0:** I²C 超时时间选择位

I²C 超时时钟由 $f_{SUB}/32$ 提供，超时时间可以根据以下公式计算：

$$([I2CTOS5:I2CTOS0]+1) \times (32/f_{SUB})$$

SPIA 串行接口模块 – SPIA

单片机内含一个独立的 SPI 功能。重要的是，不要将 SPI 功能与 SIM 模块中的 SPI 功能混淆，其具体描述详见规格书的另一章节。将这个独立的 SPI 命名为 SPIA 以区别于 SIM 中的 SPI。

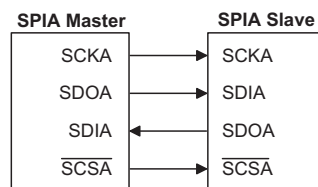
SPIA 接口常用于与外部设备如传感器、闪存或 EEPROM 内存等通信。四线 SPIA 接口最初是由摩托罗拉公司研制，是一个有相当简单的通信协议的串行数据接口，这个协议可以简化与外部硬件的编程要求。

SPIA 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，单片机既可以作为主机，也可以作为从机。虽然 SPIA 接口理论上允许一个主机控制多个从机，但此处的 SPIA 中只有一个片选信号引脚 $\overline{\text{SCSA}}$ 。若主机需要控制多个从机，可使用输入 / 输出引脚选择从机。

SPIA 接口操作

SPIA 接口是一个全双工串行数据传输器。SPIA 接口的四线为：SDIA、SDOA、SCKA 和 $\overline{\text{SCSA}}$ 。SDIA 和 SDOA 是数据的输入和输出线。SCKA 是串行时钟线， $\overline{\text{SCSA}}$ 是从机的选择线。SPIA 的接口引脚与普通 I/O 口共用。通过设定 SPIAC0/SPIAC1 寄存器的对应位，来使能 SPIA 接口。SPIA 可以通过 SPIAC0 寄存器中的 SPIAEN 位来除能或使能。连接到 SPIA 接口的单片机以从主 / 从模式进行通信，且主机完成所有的数据传输初始化，并控制时钟信号。由于单片机只有一个 $\overline{\text{SCSA}}$ 引脚，所以只能拥有一个从机设备。

可通过软件控制 $\overline{\text{SCSA}}$ 引脚使能与除能，设置 SACSSEN 位为“1”使能 $\overline{\text{SCSA}}$ 功能，设置 SACSSEN 位为“0”， $\overline{\text{SCSA}}$ 引脚将处于浮空状态。

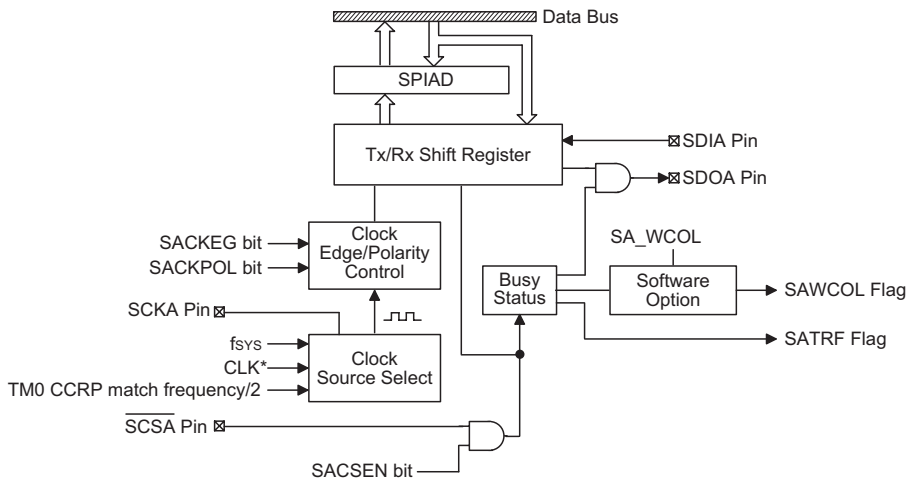


SPIA 主 / 从机连接方式

该系列单片机的 SPIA 功能具有以下特点：

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

SPIA 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 SACSSEN，SPIAEN 位的状态。



Note: CLK = f_{TBC} for HT66FB540/HT66FB550/HT66FB560, CLK = f_i for HT66FB542.

SPIA 方框图

SPIA 寄存器

有四个寄存器用于控制 SPIA 接口的所有操作，其中有一个数据寄存器 SPIAD、三个控制寄存器 SPIAC0、SPIAC1 和 SBSC。SBSC 寄存器中的 SA_WCOL 位用于控制 SPIA WCOL 功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SPIAC0	SASPI2	SASPI1	SASPI0	—	—	—	SPIAEN	—
SPIAC1	—	—	SACKPOL	SACKEG	SAMLS	SACSEN	SAWCOL	SATRF
SPIAD	D7	D6	D5	D4	D3	D2	D1	D0
SBSC	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL

SPIA 寄存器列表

SPIAD 用于存储发送和接收的数据。在单片机尚未将数据写入到 SPIA 总线中时,要传输的数据应先存在 SPIAD 中。SPIA 总线接收到数据之后,单片机就可以从 SPIAD 数据寄存器中读取。所有通过 SPIA 传输或接收的数据都必须通过 SPIAD 实现。

● SPIAD 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

单片机中也有三个控制 SPIA 接口功能的寄存器，SPIAC0、SPIAC1 和 SBSC。寄存器 SPIAC0 用于控制使能 / 除能功能和设置数据传输的时钟频率。虽然 SPIAC0 与 SPIA 功能无关，但是也用于控制外部时钟分频。寄存器 SPIAC1 用于其它的控制功能如 LSB/MSB 选择，写冲突标志位等。

• SPIAC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SASPI2	SASPI1	SASPI0	—	—	—	SPIAEN	—
R/W	R/W	R/W	R/W	—	—	—	R/W	—
POR	1	1	1	—	—	—	0	—

Bit 7~5 SASPI2~SASPI0: SPIA 工作模式控制位
 000: SPIA 主机模式; SPIA 时钟为 $f_{SYS}/4$
 001: SPIA 主机模式; SPIA 时钟为 $f_{SYS}/16$
 010: SPIA 主机模式; SPIA 时钟为 $f_{SYS}/64$
 011: SPIA 主机模式; SPIA 时钟为 f_{TBC} (HT66FB540/550/560), f_L (HT66FB542)
 100: SPIA 主机模式; SPIA 时钟为 TM0 CCRP 匹配频率 /2
 101: SPIA 从机模式
 110: 未使用模式
 111: 未使用模式

这几位用于设置 SPIA 功能的工作模式, 用于选择 SPIA 的主从模式和 SPIA 的主机时钟频率。SPIA 时钟源可来自于系统时钟也可以选择来自 TM0 或 f_{TBC} (f_L)。若选择的是作为 SPIA 从机, 则其时钟源从外部主机而得。

Bit 4~2 未使用, 读为“0”

Bit 1 **SPIAEN**: SPIA 控制位

0: 除能
 1: 使能

此位为 SPIA 接口的开 / 关控制位。此位为“0”时, SPIA 接口除能, SDIA、SDOA、SCKA 和 SCSA 脚处于浮空状态, SPIA 工作电流减小到最小值。此位为“1”时, SPIA 接口使能。

Bit 0 未使用, 读为“0”

• SPIAC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	SACKPOL	SACKEG	SAMLS	SACSEN	SAWCOL	SATRF
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 未定义位
 用户可通过软件程序对这两位进行读写。

Bit 5 **SACKPOL**: 时钟线的基础状态位

0: 当时钟无效时, SCKA 引脚为高电平
 1: 当时钟无效时, SCKA 引脚为低电平

此位决定了时钟线的基础状态, 当时钟无效时, 若此位为高, SCKA 为低电平, 若此位为低, SCKA 为高电平。

Bit 4 **SACKEG**: SPIA 的 SCKA 有效时钟边沿类型位

SACKPOL=0

0: SCKA 为高电平且在 SCKA 上升沿抓取数据
 1: SCKA 为高电平且在 SCKA 下降沿抓取数据

SACKPOL=1

0: SCKA 为低电平且在 SCKA 下降沿抓取数据
 1: SCKA 为低电平且在 SCKA 上升沿抓取数据

SACKEG 和 SACKPOL 位用于设置 SPIA 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。SACKPOL 位决定时钟线的基本状态, 若时钟无效且此位为高, 则 SCKA 为低电平, 若时钟无效且此位为低, 则 SCKA 为高电平。SACKEG 位决定有效时钟边沿类型, 取决于 SACKPOL 的状态。

- Bit 3 **SAMLS**: SPIA 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 **SACSEN**: SPIA $\overline{SCSA}$ 引脚控制位
0: 除能
1: 使能
 $\overline{SACSEN}$ 位用于 $\overline{SCSA}$ 引脚的使能 / 除能控制。此位为低时, $\overline{SCSA}$ 除能并处于浮空状态。此位为高时, $\overline{SCSA}$ 作为选择脚。
- Bit 1 **SAWCOL**: SPIA 写冲突标志位
0: 无冲突
1: 冲突
SAWCOL 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 SPIAD 寄存器。若数据正在被传输时, 此操作无效。此位可被应用程序清零。注意, SAWCOL 位的使能 / 除能可通过 SBSC 寄存器中的 SA_WCOL 位设置。
- Bit 0 **SATRF**: SPIA 发送 / 接收结束标志位
0: 数据正在发送
1: 数据发送结束
SATRF 位为发送 / 接收结束标志位, 当 SPIA 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。

● SBSC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM_WCOL	—	I2CDB1	I2CDB0	—	—	—	SA_WCOL
R/W	R/W	—	R/W	R/W	—	—	—	R/W
POR	0	—	0	0	—	—	—	0

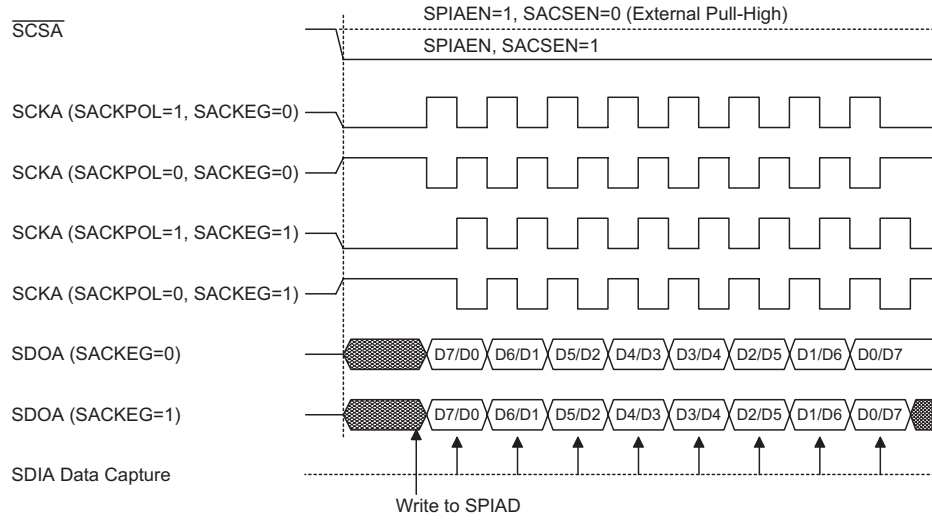
- Bit 7 **SIM_WCOL**: SIM WCOL 功能控制位
SPI 功能相关位, 具体描述详见其它章节
- Bit 6 未使用, 读为“0”
- Bit 5~4 **I2CDB1~I2CDB0**: I²C 去抖时间选择位
I²C 功能相关位, 具体描述详见其它章节
- Bit 3~1 未使用, 读为“0”
- Bit 0 **SA_WCOL**: SPIA WCOL 功能控制位
0: 除能
1: 使能

SPIA 通信

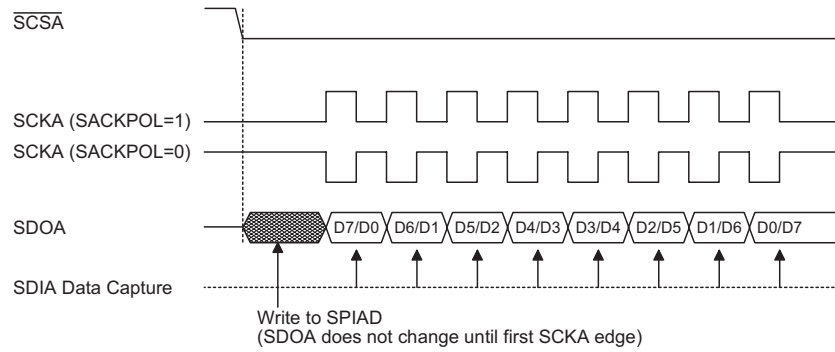
将 SPIAEN 设置为高, 使能 SPIA 功能之后, 单片机处于主机模式, 当数据写入到寄存器 SPIAD 的同时传输 / 接收开始进行。数据传输完成时, SATRF 位将自动被置位但清除只能通过应用程序完成。单片机处于从机模式时, 收到主机发来的信号之后, 会传输 SPIAD 中的数据, 而且在 SDIA 引脚上的数据也会被移位到 SPIAD 寄存器中。

主机应在输出时钟信号之前先输出一个 $\overline{SCSA}$ 信号以使能从机, 从机的数据传输功能也应在与 $\overline{SCSA}$ 信号相关的适当时候准备就绪, 这由 SACKPOL 和 SACKEG 位决定。所附时序图表明了 SACKPOL 和 SACKEG 位各种设置情况下从机数据与 $\overline{SCSA}$ 信号的关系。

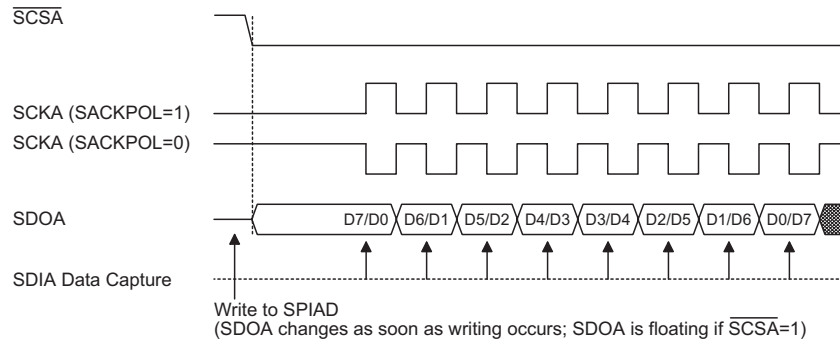
SPIA 主机模式在 SPI 时钟运行情况下可以进行正常传输。



SPIA 主机模式时序

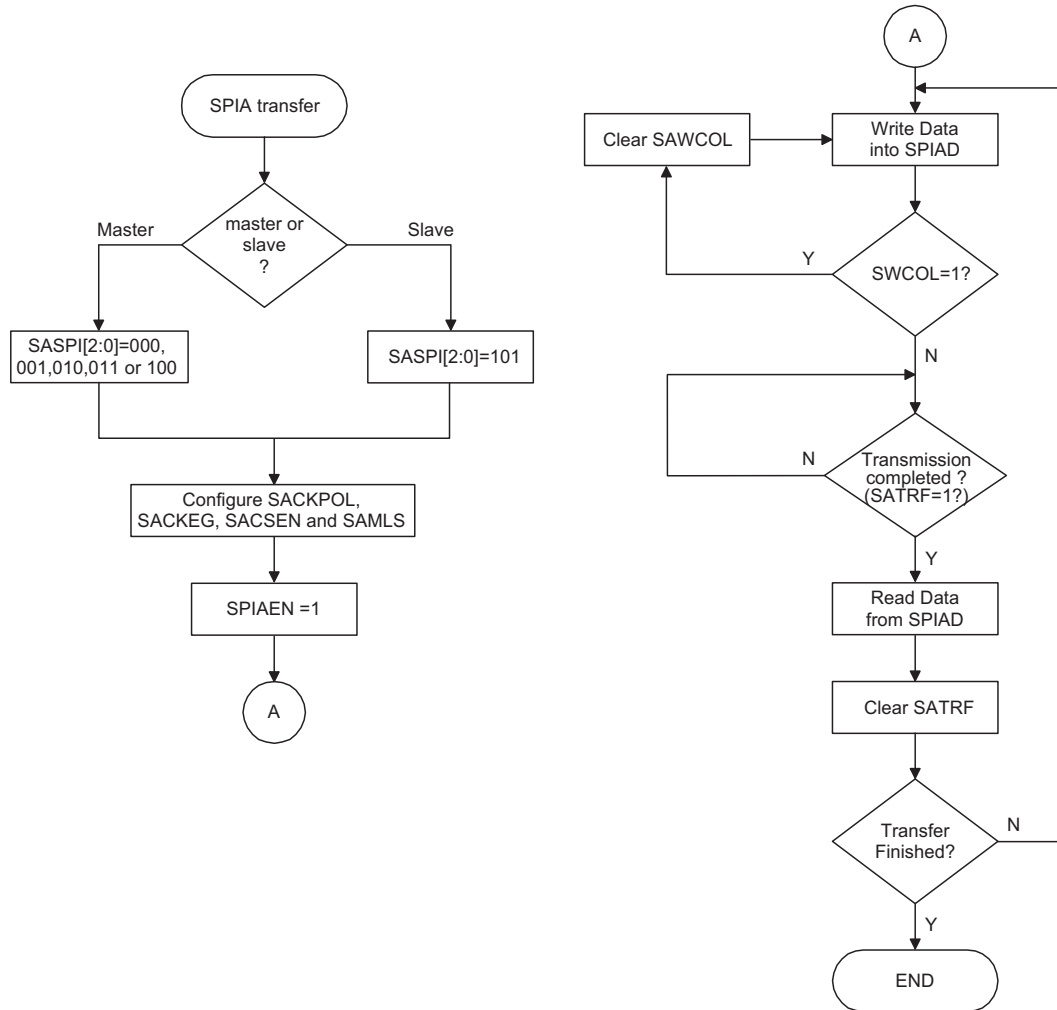


SPIA 从机模式时序 – SACKEG=0



Note: For SPIA slave mode, if SPIAEN=1 and SACSSEN=0, SPIA is always enabled and ignores the $\overline{SCSA}$ level.

SPIA 从机模式时序 – SACKEG=1



SPIA 传输控制流程图

SPIA 使能 / 除能

设置 $SACSEN=1$ 、 $\overline{SCSA}=0$ 将使能 SPIA 总线，然后等待写数据到 SPIAD 寄存器 (TXRX 缓存器)。单片机处于主机模式，数据写入 SPIAD 寄存器后，自动开始数据传输或接收操作。数据传输完成时，SATRF 位将自动被置位。单片机处于从机模式，SCKA 引脚上收到脉冲信号之后，会传输 TXRX 中的数据，或 SDIA 引脚上的数据也会被移入。

设置 SCKA、SDIA、SDOA、 $\overline{SCSA}$ 为浮空状态可除能 SPIA。

SPIA 操作

四线制 SPIA 接口可完成所有主 / 从模式通信工作。由时序图可看出基本的总线工作流程。

在 SPIAC1 寄存器中，SACSEN 位控制 SPIA 接口的所有功能。设置此位为高，SCSA 信号线有效将使能 SPIA 接口。设置此位为低，SCSA 信号线处于浮空状态将除能 SPIA 接口。SASCSEN 位和 SPIAC0 寄存器中的 SPIAEN 位设置为高，使得 SDIA 信号线处于浮空状态且 SDOA 信号线为高电平。主机模式中，如果 SCKA 信号线为高还是低取决于 SPIAC1 寄存器中的时钟极性选择位 SACKPOL。从机模式中，SCKA 信号线处于浮空状态。如果 SPIAEN 位设置为低，SPIA 接口被除能，且 SCSA、SDIA、SDOA 和 SCKA 将处于浮空状态。主机模式中，当数据被写入 SPIAD 寄存器后，主机完成所有的数据传输初始化，并控制时钟信号。从机模式中，由外部主机发出数据传送 / 接收时钟信号。下面介绍主从模式中数据传输步骤。

主机模式：

- 步骤 1
设置 SPIAC0 控制寄存器中的 SASPI2~SASPI0 位，选择 SPIA 主机模式和时钟源。
- 步骤 2
设置 SACSEN 和 SAMLS 位，选择高位或低位数据优先传送，这必须与从机设备一致。
- 步骤 3
设置 SPIAC0 控制寄存器中的 SPIAEN 位，使能 SPIA 接口功能。
- 步骤 4
对于写操作：写数据到 SPIAD 寄存器，实际上，数据被存储在 TXRX 缓存器中。再使用 SCKA 和 SDOA 信号线将数据输出。跳至步骤 5。对于读操作：使用 SDIA 信号线将 TXRX 缓存器中的数据移出，并全部锁存至 SPIAD 寄存器。
- 步骤 5
检测 SAWCOL 位，若此位为高，则发生数据冲突并跳回至步骤 4；若为低，则继续执行下面的步骤。
- 步骤 6
检测 SATRF 位或等待 SPIA 串行总线中断发生。
- 步骤 7
从 SPIAD 寄存器中读数据。
- 步骤 8
清除 SATRF。
- 步骤 9
跳回至步骤 4。

从机模式：

- 步骤 1
设置 SPIAC0 控制寄存器中的 SASPI2~SASPI0 位，选择 SPIA 从机模式。
- 步骤 2
设置 SACSEN 和 SAMLS 位，选择高位或低位数据优先传送，这必须与主机设备一致。

- 步骤 3
设置 SPIAEN 位，使能 SPIA 接口功能。
- 步骤 4
对于写操作：写数据到 SPIAD 寄存器，实际上，数据被存储在 TXRX 缓存器中。等待主机时钟 SCKA 信号和 SCSA 信号。跳至步骤 5。对于读操作：使用 SDIA 信号线将 TXRX 缓存器中的数据移出，并全部锁存至 SPIAD 寄存器。
- 步骤 5
检测 SAWCOL 位，若此位为高，则发生数据冲突并跳回至步骤 4；若为低，则继续执行下面的步骤。
- 步骤 6
检测 SATRF 位或等待 SPIA 串行总线中断发生。
- 步骤 7
从 SPIAD 寄存器中读数据。
- 步骤 8
清除 SATRF。
- 步骤 9
跳回至步骤 4。

错误侦测

SPIAC1 寄存器中的 SAWCOL 位用于数据传输期间监测数据冲突的发生。此位由串行接口设置为高，而由应用程序来清除为零。在数据传输期间，如果写数据到 SPIAD 时又阻止数据继续被写入，此时数据冲突发生。可通过 SBSC 寄存器中的 SA_WCOL 位除能或使能。

外围时钟输出

外围时钟输出功能使单片机能够为外部硬件提供和单片机时钟同步的时钟信号。

外围时钟操作

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

● SIMC0 寄存器

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

Bit 7~5 **SIM2~SIM0**: SIM 工作模式控制位
具体描述详见其它章节

Bit 4 **PCKEN**: PCK 输出脚控制位
0: 除能
1: 使能

- Bit 3~2

PCKP1~PCKP0: 选择 PCK 输出脚的频率位
00: f_{sys}
01: $f_{sys}/4$
10: $f_{sys}/8$
11: TM0 CCRP 匹配频率 /2
- Bit 1

SIMEN: SIM 控制位
具体描述详见其它章节
- Bit 0

未使用, 读为 “0”

中断

中断是单片机一个重要功能。当外部事件或内部功能如定时器模块或 A/D 转换器有效, 并且产生中断时, 系统会暂时中止当前的程序而转到执行相对应的中断服务程序。此单片机提供多个外部中断和内部中断功能, 外部中断由 INT0 和 INT1 引脚动作产生, 而内部中断由各种内部功能, 如定时器模块、比较器、时基、LVD、SIM、SPIA、USB 和 A/D 转换器等产生。

中断寄存器

中断控制基本上是在一定单片机条件发生时设置请求标志位, 应用程序中中断使能位的设置是通过位于专用数据存储器中的一系列寄存器控制的。寄存器的数量由所选单片机的型号决定, 但总的分为三类。第一类是 INTC0~INTC3 寄存器, 用于设置基本的中断; 第二类是 MFI0~MFI2 寄存器, 用于设置多功能中断; 最后一种有 INTEG 寄存器, 用于设置外部中断边沿触发类型。

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

功能	使能位	请求标志	注释
总中断	EMI	—	—
比较器	CPnE	CPnF	对于 HT66FB542, n=0; 对于 HT66FB540/HT66FB550/HT66FB560, n=0 或 1
INTn 脚	INTnE	INTnF	n=0 或 1
A/D 转换器	ADE	ADF	—
多功能	MFnE	MFnF	n=0~4
时基	TBnE	TBnF	对于 HT66FB540/HT66FB550/HT66FB560, n=0 或 1
SIM	SIME	SIMF	—
SPIA	SPIAE	SPIAF	—
LVD	LVE	LVF	—
TM	TnPE	TnPF	n=0~3
	TnAE	TnAF	
USB	USBE	USBF	—

中断寄存器位命名模式

中断寄存器内容

● HT66FB542

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	USBF	INT1F	INT0F	USBE	INT1E	INT0E	EMI
INTC1	MF1F	MF0F	—	CP0F	MF1E	MF0E	—	CP0E
INTC2	SPIAF	SIMF	MF3F	MF2F	SPIAE	SIME	MF3E	MF2E
INTC3	—	MF4F	—	—	—	MF4E	—	—
MF10	T1AF	T1PF	T0AF	T0PF	T1AE	T1PE	T0AE	T0PE
MF11	T3AF	T3PF	T2AF	T2PF	T3AE	T3PE	T2AE	T2PE
MF12	—	ADF	—	LVF	—	ADE	—	LVE

● HT66FB540/HT66FB550/HT66FB560

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	USBF	INT1F	INT0F	USBE	INT1E	INT0E	EMI
INTC1	MF1F	MF0F	CP1F	CP0F	MF1E	MF0E	CP1E	CP0E
INTC2	SPIAF	SIMF	MF3F	MF2F	SPIAE	SIME	MF3E	MF2E
INTC3	—	MF4F	TB1F	TB0F	—	MF4E	TB1E	TB0E
MF10	T1AF	T1PF	T0AF	T0PF	T1AE	T1PE	T0AE	T0PE
MF11	T3AF	T3PF	T2AF	T2PF	T3AE	T3PE	T2AE	T2PE
MF12	—	ADF	—	LVF	—	ADE	—	LVE

● 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 未使用，读为“0”

Bit 3~2 **INT1S1~INT1S0**: INT1 脚中断边沿控制位

00: 除能
01: 上升沿
10: 下降沿
11: 双沿

Bit 1~0 **INT0S1~INT0S0**: INT0 脚中断边沿控制位

00: 除能
01: 上升沿
10: 下降沿
11: 双沿

●INTC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	USBF	INT1F	INT0F	USBE	INT1E	INT0E	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

- Bit 7 未使用，读为“0”
- Bit 6 **USBF**: USB 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 **USBE**: USB 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 – HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	MF1F	MF0F	—	CP0F	MF1E	MF0E	—	CP0E
R/W	R/W	R/W	—	R/W	R/W	R/W	—	R/W
POR	0	0	—	0	0	0	—	0

- Bit 7 **MF1F**: 多功能中断 1 请求标志位
0: 无请求
1: 中断请求
- Bit 6 **MF0F**: 多功能中断 0 请求标志位
0: 无请求
1: 中断请求
- Bit 5 未使用，读为“0”
- Bit 4 **CP0F**: 比较器 0 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **MF1E**: 多功能中断 1 控制位
0: 除能
1: 使能

Bit 2 **MF0E**: 多功能中断 0 控制位
 0: 除能
 1: 使能

Bit 1 未使用, 读为 “0”

Bit 0 **CP0E**: 比较器 0 中断控制位
 0: 除能
 1: 使能

●INTC1 寄存器 – HT66FB540/HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	MF1F	MF0F	CP1F	CP0F	MF1E	MF0E	CP1E	CP0E
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 **MF1F**: 多功能中断 1 请求标志位
 0: 无请求
 1: 中断请求

Bit 6 **MF0F**: 多功能中断 0 请求标志位
 0: 无请求
 1: 中断请求

Bit 5 **CP1F**: 比较器 1 中断请求标志位
 0: 无请求
 1: 中断请求

Bit 4 **CP0F**: 比较器 0 中断请求标志位
 0: 无请求
 1: 中断请求

Bit 3 **MF1E**: 多功能中断 1 控制位
 0: 除能
 1: 使能

Bit 2 **MF0E**: 多功能中断 0 控制位
 0: 除能
 1: 使能

Bit 1 **CP1E**: 比较器 1 中断控制位
 0: 除能
 1: 使能

Bit 0 **CP0E**: 比较器 0 中断控制位
 0: 除能
 1: 使能

●INTC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SPIAF	SIMF	MF3F	MF2F	SPIAE	SIME	MF3E	MF2E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **SPIAF**: SPIA 请求标志位
0: 无请求
1: 中断请求
- Bit 6 **SIMF**: SIM 请求标志位
0: 无请求
1: 中断请求
- Bit 5 **MF3F**: 多功能中断 3 请求标志位
0: 无请求
1: 中断请求
- Bit 4 **MF2F**: 多功能中断 2 请求标志位
0: 无请求
1: 中断请求
- Bit 3 **SPIAE**: SPIA 控制位
0: 除能
1: 使能
- Bit 2 **SIME**: SIM 控制位
0: 除能
1: 使能
- Bit 1 **MF3E**: 多功能中断 3 控制位
0: 除能
1: 使能
- Bit 0 **MF2E**: 多功能中断 2 控制位
0: 除能
1: 使能

●INTC3 寄存器 – HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	MF4F	—	—	—	MF4E	—	—
R/W	—	R/W	—	—	—	R/W	—	—
POR	—	0	—	—	—	0	—	—

- Bit 7 未使用，读为“0”
- Bit 6 **MF4F**: 多功能中断 4 请求标志位
0: 无请求
1: 中断请求
- Bit 5~3 未使用，读为“0”
- Bit 2 **MF4E**: 多功能中断 4 控制位
0: 除能
1: 使能
- Bit 1~0 未使用，读为“0”

●INTC3 寄存器 – HT66FB540/HT66FB550/HT66FB560

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

- Bit 7 未使用，读为“0”
- Bit 6 **MF4F**: 多功能中断 4 请求标志位
0: 无请求
1: 中断请求
- Bit 5 **TB1F**: 时基 1 中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **TB0F**: 时基 0 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 未使用，读为“0”
- Bit 2 **MF4E**: 多功能中断 4 控制位
0: 除能
1: 使能
- Bit 1 **TB1E**: 时基 1 中断控制位
0: 除能
1: 使能
- Bit 0 **TB0E**: 时基 0 中断控制位
0: 除能
1: 使能

●MF10 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1AF	T1PF	T0AF	T0PF	T1AE	T1PE	T0AE	T0PE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **T1AF**: TM1 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 6 **T1PF**: TM1 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 **T0AF**: TM0 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **T0PF**: TM0 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **T1AE**: TM1 比较器 A 匹配中断控制位
0: 除能
1: 使能
- Bit 2 **T1PE**: TM1 比较器 P 匹配中断控制位
0: 除能
1: 使能

- Bit 1 **T0AE:** TM0 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
- Bit 0 **T0PE:** TM0 比较器 P 匹配中断控制位
 0: 除能
 1: 使能

● MF11 寄存器

Bit	7	6	5	4	3	2	1	0
Name	T3AF	T3PF	T2AF	T2PF	T3AE	T3PE	T2AE	T2PE
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 **T3AF:** TM3 比较器 A 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 6 **T3PF:** TM3 比较器 P 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 5 **T2AF:** TM2 比较器 A 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 4 **T2PF:** TM2 比较器 P 匹配中断请求标志位
 0: 无请求
 1: 中断请求
- Bit 3 **T3AE:** TM3 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
- Bit 2 **T3PE:** TM3 比较器 P 匹配中断控制位
 0: 除能
 1: 使能
- Bit 1 **T2AE:** TM2 比较器 A 匹配中断控制位
 0: 除能
 1: 使能
- Bit 0 **T2PE:** TM2 比较器 P 匹配中断控制位
 0: 除能
 1: 使能

● MF12 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	ADF	—	LVF	—	ADE	—	LVE
R/W	—	R/W	—	R/W	—	R/W	—	R/W
POR	—	0	—	0	—	0	—	0

- Bit 7 未使用，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 未使用，读为“0”
- Bit 4 **LVF:** LVD 中断请求标志位
 0: 无请求
 1: 中断请求

Bit 3	未使用，读为“0”
Bit 2	ADE : A/D 转换器中断控制位 0: 除能 1: 使能
Bit 1	未使用，读为“0”
Bit 0	LVE : LVD 中断控制位 0: 除能 1: 使能

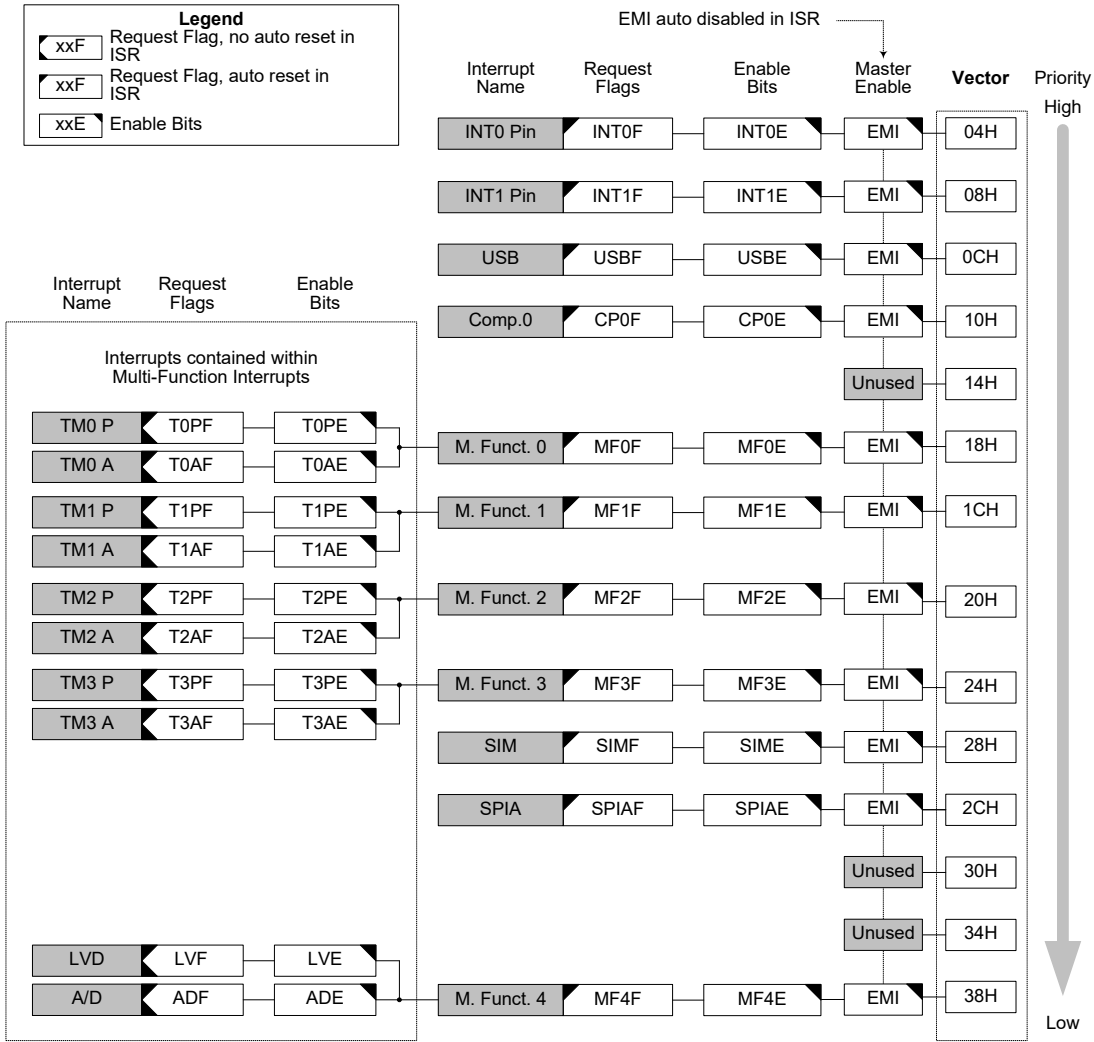
中断操作

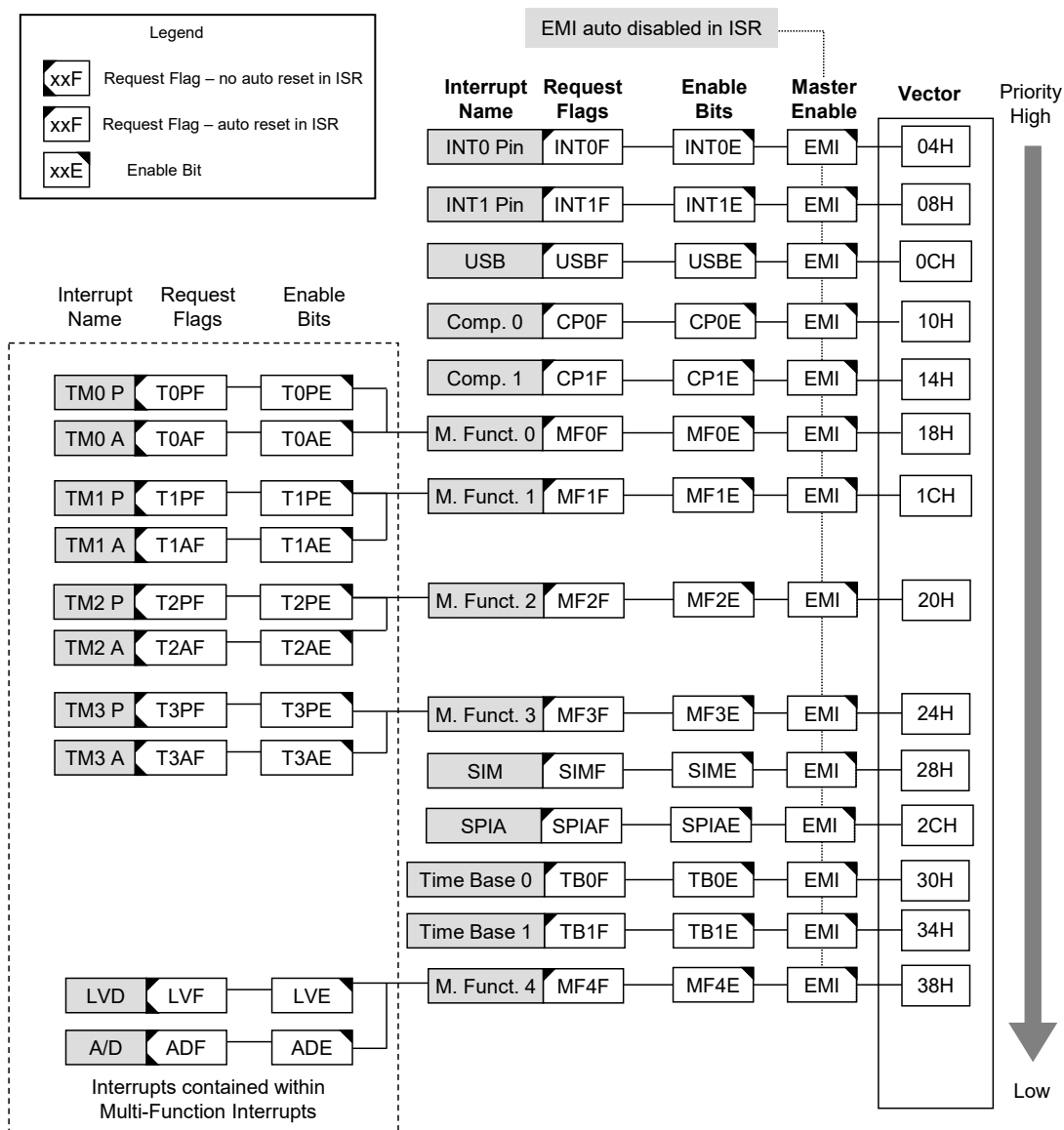
若中断事件条件产生，如一个 TM 比较器 P、比较器 A 匹配或 A/D 转换结束等等，相关中断请求标志将置起。中断标志产生后程序是否会跳转至相关中断向量执行是由中断使能位的条件决定的。若使能位为“1”，程序将跳至相关中断向量中执行；若使能位为“0”，即使中断请求标志置起中断也不会发生，程序也不会跳转至相关中断向量执行。若总中断使能位为“0”，所有中断都将除能。

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

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

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





中断结构 – HT66FB540/HT66FB550/HT66FB560

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，INT0~INT1 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 和相应中断使能位 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。外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，此引脚将被作为外部中断脚使用。此时该引脚必须通过设置控制寄存器，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态改变，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INT0F~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，其上拉电阻仍保持有效。

寄存器 INTEG 被用来选择有效的边沿类型，来触发外部中断。可以选择上升沿还是下降沿或双沿触发都产生外部中断。注意 INTEG 也可以用来除能外部中断功能。

比较器中断

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

注意，HT66FB542 单片机只有比较器 0 中断。

USB 中断

当有端点被访问，USB 中断请求标志 USBF 被置位，USB 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 USB 中断使能位 USBE 需先被置位。当中断使能，堆栈未满并且有端点被访问时，将调用 USB 中断向量子程序。当响应中断服务子程序时，USB 中断请求标志位会自动复位且 EMI 位会被清零以除能其它中断。

时基中断 – HT66FB540/HT66FB550/HT66FB560

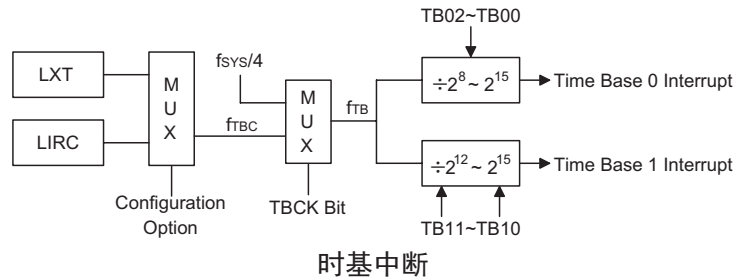
时基中断提供一个固定周期的中断信号，由各自的定时器功能产生溢出信号控制。当各自的中断请求标志 TB0F 或 TB1F 被置位时，中断请求发生。当总中断使能位 EMI 和时基使能位 TB0E 或 TB1E 被置位，允许程序跳转到各自的中断向量地址。当中断使能，堆栈未满且时基溢出时，将调用它们各自的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 TB0F 或 TB1F 会自动复位且 EMI 位会被清零以除能其它中断。

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

• TBC 寄存器

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

- Bit 7 **TBON**: 时基控制位
0: 除能
1: 使能
- Bit 6 **TBCK**: 选择 f_{TB} 时钟位
0: f_{TBC}
1: $f_{SYS}/4$
- Bit 5~4 **TB11~TB10**: 选择时基 1 溢出周期位
00: $4096/f_{TB}$
01: $8192/f_{TB}$
10: $16384/f_{TB}$
11: $32768/f_{TB}$
- Bit 3 **LXTLP**: LXT 低功耗控制位
0: 除能
1: 使能
- Bit 2~0 **TB02~TB00**: 选择时基 0 溢出周期位
000: $256/f_{TB}$
001: $512/f_{TB}$
010: $1024/f_{TB}$
011: $2048/f_{TB}$
100: $4096/f_{TB}$
101: $8192/f_{TB}$
110: $16384/f_{TB}$
111: $32768/f_{TB}$



串行接口模块中断

串行接口模块中断，即 SIM 中断。当一个字节数据已由 SIM 接口接收或发送完，或 I²C 从机地址匹配，或 I²C 超时，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 和串行接口中断使能位 SIME 需先被置位。当中断使能，堆栈未满足且以上任何一种情况发生时，将调用 SIM 中断向量子程序。当响应中断服务子程序时，SIM 中断请求标志位会自动复位且 EMI 位会被清零以除能其它中断。

串行外设接口中断

串行外设接口中断，即 SPIA 中断。当一个字节数据已由 SPIA 接口接收或发送完，中断请求标志 SPIAF 被置位，SPIA 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 和串行接口中断使能位 SPIAE 需先被置位。

当中断使能，堆栈未满且一个字节数据已被传送或接收完毕时，将调用 SPIA 中断向量子程序。当响应中断服务子程序时，SPIA 中断请求标志位会自动复位且 EMI 位会被清零以除能其它中断。

多功能中断

此系列单片机中有多达 3 种多功能中断，与其它中断不同，它没有独立源，但由其它现有的中断源构成，即 TM 中断、ADC 中断和 LVD 中断。

当多功能中断中任何一种中断请求标志 MF0F~MF4F 被置位，多功能中断请求产生。当中断使能，堆栈未满，包括在多功能中断中的任意一个中断发生时，将调用多功能中断向量中的一个子程序。当响应中断服务子程序时，相关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、ADC 中断和 LVD 中断的请求标志位不会自动复位，必须由应用程序清零。

A/D 转换器中断

A/D 中断属于多功能中断。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被置位，即 A/D 转换过程完成时，中断请求发生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、A/D 中断使能位 ADE 及其相应的多功能中断使能位需先被置位。当中断使能，堆栈未满且 A/D 转换动作结束时，可跳转到相应的多功能中断向量中执行。当 A/D 中断响应时，EMI 将被自动清零以除能其它中断，相关多功能中断请求标志也可自动清除，而相应的中断请求标志位 ADF 需在应用程序中手动清除。

LVD 中断

LVD 中断属于多功能中断。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 及其相应的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，可跳转到相应的多功能中断向量中执行。当 LVD 中断响应时，EMI 将被自动清零以除能其它中断，相关多功能中断请求标志也可自动清除，而相应的中断请求标志位 LVF 需在应用程序中手动清除。

TM 中断

简易型和标准型 TM 各有两个中断。所有的 TM 中断都属于多功能中断。简易型和标准型 TM 各有两个中断请求标志位 TnPF、TnAF 及两个使能位 TnPE、TnAE。当 TM 比较器 P、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 MFnE 需先被置位。当中断使能，堆栈未满且 TM 比较器匹配情况发生时，可跳转至相关多功能中断向量子程序中执行。当 TM 中断响应，EMI 将被自动清零以除能其它中断，相关 MFnF 标志也可自动清除，但 TM 中断请求标志需在应用程序中手动清除。

中断唤醒功能

每个中断都具有将处于休眠或空闲模式的单片机唤醒的能力。当中断请求标志由低到高转换时唤醒动作产生，其与中断是否使能无关。因此，尽管单片机处于休眠或空闲模式且系统振荡器停止工作，如有外部中断脚上产生外部边沿跳

变、低电压或比较器输入改变都可能导致其相应的中断标志被置位，由此产生中断，因此必须注意避免伪唤醒情况的发生。若中断唤醒功能被除能，单片机进入休眠或空闲模式前相应中断请求标志应被置起。中断唤醒功能不受中断使能位的影响。

编程注意事项

通过禁止相关中断使能位，可以屏蔽中断请求，然而，一旦中断请求标志位被设定，它们会被保留在中断控制寄存器内，直到相应的中断服务子程序执行或请求标志位被软件指令清除。

多功能中断中所含中断相应程序执行时，多功能中断请求标志 MF0F~MF4F 可以自动清零，但各自的请求标志需在应用程序中手动清除。

建议在中断服务子程序中不要使用“CALL 子程序”指令。中断通常发生在不可预料的情况或是需要立刻执行的某些应用。假如只剩下一层堆栈且没有控制好中断，当“CALL 子程序”在中断服务子程序中执行时，将破坏原来的控制序列。

所有中断在休眠或空闲模式下都具有唤醒功能，当中断请求标志发生由低到高的转变时都可产生唤醒功能。若要避免相应中断产生唤醒动作，在单片机进入休眠或空闲模式前需先将相应请求标志置为高。

当进入中断服务程序，系统仅将程序计数器的内容压入堆栈，如果中断服务程序会改变状态寄存器或其它的寄存器的内容而破坏控制流程，应事先将这些数据保存起来。

若从中断子程序中返回可执行 RET 或 RETI 指令。除了能返回至主程序外，RETI 指令还能自动设置 EMI 位为高，允许进一步中断。RET 指令只能返回至主程序，清除 EMI 位，除能进一步中断。

低电压检测 – LVD

此系列单片机都具有低电压检测功能，即 LVD。该功能使能用于监测电源电压 V_{DD} ，若电源电压低于一定值可提供一个警告信号。此功能在电池类产品中非常有用，在电池电压较低时产生警告信号。低电压检测也可产生中断信号。

LVD 寄存器

低电压检测功能由 LVDC 寄存器控制。VLVD2~VLVD0 位用于选择 8 个固定的电压参考点。LVDO 位被置位时低电压情况发生，若 LVDO 位为低表明 V_{DD} 电压工作在当前所设置低电压水平值之上。LVDEN 位用于控制低电压检测功能的开启 / 关闭，设置此位为高使能此功能，反之，关闭内部低电压检测电路。低电压检测会有一定的功耗，在不使用时可考虑关闭此功能，此举在功耗要求严格的电池供电应用中值得考虑。

• LVDC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	—	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

Bit 7~6 未定义，读为“0”

Bit 5 **LVDO**: LVD 输出标志位

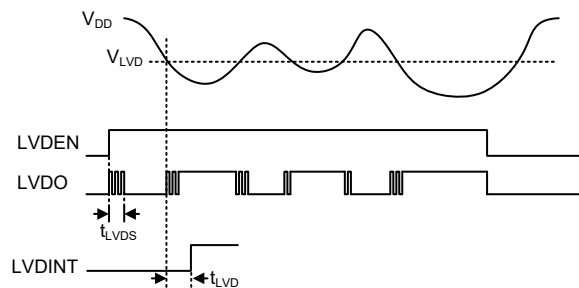
0: 未检测到低电压

1: 检测到低电压

Bit 4	LV DEN: 低电压检测控制位 0: 除能 1: 使能
Bit 3	未定义, 读为 “0”
Bit 2~0	VLVD2~VLVD0: 选择 LVD 电压位 000: 2.0V 001: 2.2V 010: 2.4V 011: 2.7V 100: 3.0V 101: 3.3V 110: 3.6V 111: 4.0V

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 2.0V~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。若 LV DEN 位为高, 当单片机掉电时低电压检测器保持有效状态。低电压检测器使能后, 读取 LVDO 位前, 电路稳定需要一定的延时 t_{LVDS} 。注意, V_{DD} 电压可能上升或下降比较缓慢, 在 V_{LVD} 电压值附近时, LVDO 位可能有多种变化。



LVD 操作

低电压检测器也有自己的中断功能, 它是除了轮询 LVDO 位之外的另一种检测低电压的方法。中断条件产生置位 LVDO 并延时 t_{LVD} 后, 中断产生。若 LV DEN 位为高, 当单片机掉电时低电压检测器保持有效状态。此种情况下, 若 V_{DD} 降至小于 LVD 预置电压值时, 中断请求标志位 LVF 将被置位, 中断产生, 单片机将被从休眠或空闲模式中唤醒。若不要求低电压检测的唤醒功能使能, 在单片机进入休眠或空闲模式前应将 LVF 标志置为高。

USB 接口

USB 接口是一个 4 线串行总线，允许主机与多达 127 个外围设备在同一总线上进行通信。主机设备使用基于令牌协议的方法进行通信控制。USB 总线的其它优点包括热插拔、动态设备配置。由于 USB 数据传输协议的复杂性，不容许在 datasheet 中提供全面的 USB 操作信息，因此，读者应该查阅其它外部信息以详细了解 USB。

该系列单片机具有 USB 接口功能，使 USB 外围产品的设计更加方便。

SYSC 寄存器中的 USBdis 位可以控制 USB 使能 / 除能。如果 USB 被除能，那么 V33O 将处于浮空状态，UDP/UDN 作为输入 / 输出脚使用，USB SIE 也会被除能。

电源

HT66FB5xx 单片机有三个电源：USB SIE VDD、VDDIO 和 MCU VDD。

USB SIE VDD 为所有 USB SIE 相关电路提供电源，来自于 UBUS 引脚。一旦 USB 从 USB 接口移除，USB 总线上将没有电源，USB SIE 电路也不再运作。

PA 口可以通过 PAPS1 和 PAPS0 寄存器来被配置 PA0~PA7 的电源是来自 MCU VDD、V33O 还是 VDDIO 引脚。

PE0 与 VDDIO 和 VREF 引脚共用。VDDIO 功能可以通过配置选项选择，VREF 功能可以通过 ADCR1 寄存器中的 VREFS 位选择。然而，VREF 具有最高优先级。如果 VREF 功能被选中，即使选择了 VDDIO，PE0 功能和 VDDIO 功能都将被除能。

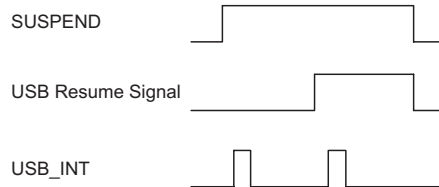
MCU VDD 为 HT66FB5xx 所有电路提供电源，除了 USB SIE 电路，它由 UBUS 提供电源。

PE1 与 UBUS 引脚共用，且仅输入功能。

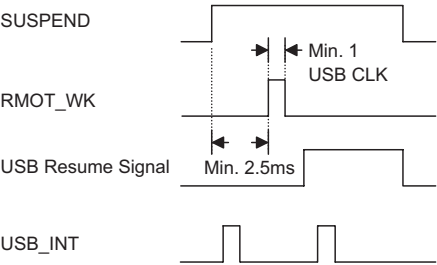
USB 暂停模式唤醒和远程唤醒

如果 USB 总线上超过 3ms 没有信号，单片机将进入暂停模式。USC 寄存器的暂停标志位 SUSP 将被置位并产生 USB 中断，表示单片机应跳转到暂停状态，以满足 USB 暂停电流规范。为了满足暂停电流，固件应通过清除 USBCKEN 位为 0 以除能 USB 时钟。

通过置位 SUSP2 位，可进一步降低暂停电流。当主机发出恢复信号时，单片机将会因 USB 中断而唤醒，USC 寄存器中的恢复标志位被置位。为了使单片机正常运行，程序必须置位 UCC 寄存器中的 USBCKEN 位和清除 UCC 寄存器中的 SUSP2 位。RESUME 位会在主机发出暂停信号前被清除，USC 寄存器中的 SUSP 位也会清为 0。当单片机检测到暂停位时，也必须注意和考虑恢复位。



单片机具有远程唤醒功能，通过 USC 寄存器中的 RMWK 位发送一个唤醒脉冲来唤醒 USB 主机。一旦 USB 主机接收到来自单片机的唤醒信号，会发送一个恢复信号给单片机。



USB 操作

HT66FB540、HT66FB542、HT66FB550 和 HT66FB560 分别有 4 个端点 (EP0~EP3)、4 个端点 (EP0~EP3)、6 个端点 (EP0~EP5) 和 8 个端点 (EP0~EP7)。EP0 支持控制传输，所有的 EP1~EP7 支持中断或批量传输。

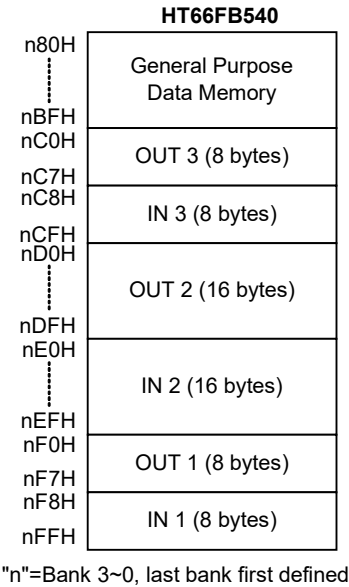
除 EP0 外的所有端点都可以通过 UFC0 和 UFC1 寄存器配置 8、16、32、64 个字节 FIFO。EP0 有 8 字节的 FIFO。

HT66FB540 的 FIFO 大小共 384+8 个字节，HT66FB542 的 FIFO 大小共 256+8 个字节，HT66FB550 的 FIFO 大小共 640+8 个字节，HT66FB560 的 FIFO 大小共 896+8 个字节。

USB FIFO 从数据存储器最后一个 Bank 的起始地址 0FFH 开始向上分配地址，具体多少取决于 FIFO 的大小。如果相应的数据存储器同时被通用数据存储器 和 USB FIFO 使用，需要注意的是 RAM 的 EQU 定义不应与 USB FIFO RAM 地址重叠。

USC 寄存器中的 URD 位是 USB 复位信号控制位。

IN/OUT 控制的 USB FIFO 大小取决于 UFC0、UFC1、UFIEN 和 UFOEN 寄存器。如果 OUT 1 未使用，OUT 1 FIFO 将不会被定义，而 IN 1 其后的 IN 2 将会被定义。



USB FIFO 大小定义

USB 接口寄存器

有几个与 USB 功能相关的寄存器。

• SYSC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CLK_ADJ	USBdis	RUBUS	—	—	HFV	—	—
R/W	R/W	R/W	R/W	—	—	R/W	—	—
POR	0	0	0	—	—	0	—	—

Bit 7 **CLK_ADJ**: PLL 时钟自动调整功能控制位
PLL 相关控制位，具体描述详见其它章节

Bit 6 **USBdis**: USB SIE 控制位
0: 使能
1: 除能

Bit 5 **RUBUS**: USB 引脚下拉电阻
0: 使能
1: 除能

Bit 4~3 未定义，读为“0”

Bit 2 **HFV**: USB 模式高频电压控制位
此位必须设置为 1。

Bit 1~0 未定义，读为“0”

• USB_STAT 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PS2_CKO	PS2_DAO	PS2_CKI	PS2_DAI	SE1	SE0	PU	ESD
R/W	W	W	R	R	R/W	R/W	R/W	R/W
POR	1	1	x	x	0	0	0	x

“x”：未知

Bit 7 **PS2_CKO**: 当工作在 3D PS2 鼠标功能，输出驱动 UDP/GPIO1 引脚，默认值为“1”。

Bit 6 **PS2_DAO**: 当工作在 3D PS2 鼠标功能，输出驱动 UDN/GPIO0 引脚，默认值为“1”。

Bit 5 **PS2_CKI**: UDP/GPIO1 输入

Bit 4 **PS2_DAI**: UDN/GPIO0 输入

Bit 3 **SE1**: 表示 SIE 已检测到 USB 总线的 SE1 噪音
该位由 SIE 置位，由固件清零。

Bit 2 **SE0**: 表示 SIE 已检测到 USB 总线的 SE0 噪音
该位由 SIE 置位，由固件清零。

Bit 1 **PU**: UDP 和 UDN 之间有无 600kΩ 上拉电阻控制位
0: 无上拉电阻 (默认)
1: UDP 和 UDN 之间有 600kΩ 上拉电阻

Bit 0 **ESD**: 有 ESD 问题时此位被置位
该位由 SIE 置位，由固件清零。

• UINT 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	EP3EN	EP2EN	EP1EN	EP0EN
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 **EP3EN**: USB 端点 3 中断控制位
0: 除能
1: 使能

Bit 2 **EP2EN**: USB 端点 2 中断控制位
0: 除能
1: 使能

Bit 1 **EP1EN**: USB 端点 1 中断控制位
0: 除能
1: 使能

Bit 0 **EP0EN**: USB 端点 0 中断控制位
0: 除能
1: 使能

• UINT 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	EP5EN	EP4EN	EP3EN	EP2EN	EP1EN	EP0EN
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 **EP5EN**: USB 端点 5 中断控制位
0: 除能
1: 使能

Bit 4 **EP4EN**: USB 端点 4 中断控制位
0: 除能
1: 使能

Bit 3 **EP3EN**: USB 端点 3 中断控制位
0: 除能
1: 使能

Bit 2 **EP2EN**: USB 端点 2 中断控制位
0: 除能
1: 使能

Bit 1 **EP1EN**: USB 端点 1 中断控制位
0: 除能
1: 使能

Bit 0 **EP0EN**: USB 端点 0 中断控制位
0: 除能
1: 使能

• UINT 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	EP7EN	EP6EN	EP5EN	EP4EN	EP3EN	EP2EN	EP1EN	EP0EN
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 **EP7EN**: USB 端点 7 中断控制位
0: 除能
1: 使能

Bit 6 **EP6EN**: USB 端点 6 中断控制位
0: 除能
1: 使能

Bit 5 **EP5EN**: USB 端点 5 中断控制位
0: 除能
1: 使能

Bit 4 **EP4EN**: USB 端点 4 中断控制位
0: 除能
1: 使能

Bit 3 **EP3EN**: USB 端点 3 中断控制位
0: 除能
1: 使能

Bit 2 **EP2EN**: USB 端点 2 中断控制位
0: 除能
1: 使能

Bit 1 **EP1EN**: USB 端点 1 中断控制位
0: 除能
1: 使能

Bit 0 **EP0EN**: USB 端点 0 中断控制位
0: 除能
1: 使能

• USC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	URD	SELPS2	PLL	SELUSB	RESUME	URST	RMWK	SUSP
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R
POR	1	0	0	0	x	x	x	x

“x”：未知

Bit 7 **URD**: USB 复位信号控制位
0: USB 复位信号不能复位单片机
1: USB 复位信号将复位单片机

Bit 6 **SELPS2**: PS2 模式选择位
0: 非 PS2 模式
1: PS2 模式
当 SELPS2 位被置高, PS2 功能被选中, 共用引脚 UDN/GPIO0 和 UDP/GPIO1 将作为普通输入 / 输出引脚使用, 可以用作 PS2 的 DATA 和 CLK 引脚。

Bit 5 **PLL**: PLL 控制位
0: 打开 PLL
1: 关闭 PLL

Bit 4 **SELUSB**: USB 模式和 V330 开 / 关选择位
0: 非 USB 模式, 关闭 V330
1: USB 模式, 打开 V330

当 SELUSB 位被置高，USB 和 V33O 功能被选中，共用引脚 UDN/GPIO0 和 UDP/GPIO1 将作为 USB 的 UDN 和 UDP 引脚使用。

- Bit 3 **RESUME:** USB 恢复标志位
 0: SUSP 位为 0
 1: 离开暂停模式
- 当 USB 离开暂停模式，此位会被 SIE 置位。当 RESUME 位被 SIE 置位，会产生一个中断并唤醒单片机。为了检测到暂停状态，单片机必须置高 USBCKEN 位和清除 SUSP2 位，以使能 SIE 检测功能。SUSP 位清零同时 RESUME 位也会被清除。当单片机检测 SUSP 时，必须考虑能唤醒单片机的恢复信号 RESUME。
- Bit 2 **URST:** USB 复位标志位
 0: 没有 USB 复位发生
 1: USB 复位发生
- 此位由 USB SIE 置位 / 清零。若 URST 位被置位，表明发生了 USB 复位，并且会触发 USB 中断。
- Bit 1 **RMWK:** USB 远程唤醒命令
 0: 无远程唤醒
 1: 远程唤醒
- 此位可由 MCU 设置为 1，以强制 USB 主机离开暂停模式。设置 RMWK 位为“1”以启动远程唤醒命令。该位被置位后会产生一个高电平脉冲信号，表明 USB 主机已经离开暂停模式。
- Bit 0 **SUSP:** USB 暂停标志位
 0: 离开暂停模式
 1: 进入暂停模式
- 当 USB 总线进入暂停模式，此位会被 SIE 置位。此位由低到高变化会触发 USB 中断。

SELUSB	SELPS2	USB 和 PS2 模式描述
0	0	1. 未使用 2. V33O 引脚浮空，无输出 3. UDN/GPIO0 和 UDP/GPIO1 引脚不输出
0	1	1. PS2 模式 2. V33O 引脚输出 VDD 3. UDN/GPIO0 和 UDP/GPIO1 引脚将作为普通 GPIO0 和 GPIO1 脚使用，可由固件输出
1	x	1. USB 模式 2. V33O 输出 3.3V 3. UDN/GPIO0 和 UDP/GPIO1 引脚将作为 UDN 和 UDP 脚使用

“x”：未知

● **USR 寄存器 – HT66FB540/HT66FB542**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	EP3F	EP2F	EP1F	EP0F
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	x	x	x	x

“x”：未知

- Bit 7~4 未定义，读为“0”
- Bit 3 **EP3F:** 端点 3 访问检测标志位
 0: 没被访问
 1: 被访问
- Bit 2 **EP2F:** 端点 2 访问检测标志位
 0: 没被访问

- 1: 被访问
- Bit 1 **EP1F**: 端点 1 访问检测标志位
0: 没被访问
1: 被访问
- Bit 0 **EP0F**: 端点 0 访问检测标志位
0: 没被访问
1: 被访问

● USB 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	EP5F	EP4F	EP3F	EP2F	EP1F	EP0F
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	x	x	x	x	x	x

“x”：未知

- Bit 7~6 未定义，读为“0”
- Bit 5 **EP5F**: 端点 5 访问检测标志位
0: 没被访问
1: 被访问
- Bit 4 **EP4F**: 端点 4 访问检测标志位
0: 没被访问
1: 被访问
- Bit 3 **EP3F**: 端点 3 访问检测标志位
0: 没被访问
1: 被访问
- Bit 2 **EP2F**: 端点 2 访问检测标志位
0: 没被访问
1: 被访问
- Bit 1 **EP1F**: 端点 1 访问检测标志位
0: 没被访问
1: 被访问
- Bit 0 **EP0F**: 端点 0 访问检测标志位
0: 没被访问
1: 被访问

● USB 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	EP7F	EP6F	EP5F	EP4F	EP3F	EP2F	EP1F	EP0F
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 **EP7F**: 端点 7 访问检测标志位
0: 没被访问
1: 被访问
- Bit 6 **EP6F**: 端点 6 访问检测标志位
0: 没被访问
1: 被访问
- Bit 5 **EP5F**: 端点 5 访问检测标志位
0: 没被访问
1: 被访问
- Bit 4 **EP4F**: 端点 4 访问检测标志位
0: 没被访问

- 1: 被访问
- Bit 3 **EP3F:** 端点 3 访问检测标志位
0: 没被访问
1: 被访问
- Bit 2 **EP2F:** 端点 2 访问检测标志位
0: 没被访问
1: 被访问
- Bit 1 **EP1F:** 端点 1 访问检测标志位
0: 没被访问
1: 被访问
- Bit 0 **EP0F:** 端点 0 访问检测标志位
0: 没被访问
1: 被访问

● UCC 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	Rctrl	SYSCLK	Fsys16MHz	SUSP2	USBCKEN	—	EPS1	EPS0
R/W	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W
POR	0	0	0	x	0	—	x	x

“x”：未知

- Bit 7 **Rctrl:** UDP 和 UBUS 之间有无 7.5kΩ 电阻控制位
0: UDP 和 UBUS 之间无 7.5kΩ 电阻
1: UDP 和 UBUS 之间有 7.5kΩ 电阻
- Bit 6 **SYSCLK:** 系统时钟频率选择位
0: 12MHz
1: 6MHz
- Bit 5 **Fsys16MHz:** 单片机系统时钟源控制位
0: HXT
1: PLL 16MHz
- Bit 4 **SUSP2:** 暂停模式下降低功耗控制位
0: 正常模式
1: 暂停模式，置起此位可以降低功耗
- Bit 3 **USBCKEN:** USB 时钟使能控制位
0: 除能
1: 使能
- Bit 2 未定义，读为 “0”
- Bit 1~0 **EPS1~EPS0:** 选择要访问端点的 FIFO
00: 选择端点 0 的 FIFO
01: 选择端点 1 的 FIFO
10: 选择端点 2 的 FIFO
11: 选择端点 3 的 FIFO

• UCC 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	Rctrl	SYSCLK	Fsys16MHz	SUSP2	USBCKEN	EPS2	EPS1	EPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	x	0	x	x	x

“x”：未知

- Bit 7 **Rctrl**: UDP 和 UBUS 之间 7.5kΩ 电阻控制位
0: UDP 和 UBUS 之间无 7.5kΩ 电阻
1: UDP 和 UBUS 之间有 7.5kΩ 电阻
- Bit 6 **SYSCLK**: 系统时钟频率选择位
0: 12MHz
1: 6MHz
- Bit 5 **Fsys16MHz**: PLL 16MHz 输出控制位
0: HXT
1: PLL 16MHz
- Bit 4 **SUSP2**: 暂停模式下降低功耗控制位
0: 正常模式
1: 暂停模式, 置起此位可以降低功耗
- Bit 3 **USBCKEN**: USB 时钟使能控制位
0: 除能
1: 使能
- Bit 2~0 **EPS2~EPS0**: 选择要访问端点的 FIFO
000: 选择端点 0 的 FIFO
001: 选择端点 1 的 FIFO
010: 选择端点 2 的 FIFO
011: 选择端点 3 的 FIFO
100: 选择端点 4 的 FIFO
101~111: 选择端点 5 的 FIFO

• UCC 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	Rctrl	SYSCLK	Fsys16MHz	SUSP2	USBCKEN	EPS2	EPS1	EPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	x	0	x	x	x

“x”：未知

- Bit 7 **Rctrl**: UDP 和 UBUS 之间 7.5kΩ 电阻控制位
0: UDP 和 UBUS 之间无 7.5kΩ 电阻
1: UDP 和 UBUS 之间有 7.5kΩ 电阻
- Bit 6 **SYSCLK**: 系统时钟频率选择位
0: 12MHz
1: 6MHz
- Bit 5 **Fsys16MHz**: PLL 16MHz 输出控制位
0: HXT
1: PLL 16MHz
- Bit 4 **SUSP2**: 暂停模式下降低功耗控制位
0: 正常模式
1: 暂停模式, 置起此位可以降低功耗
- Bit 3 **USBCKEN**: USB 时钟使能控制位
0: 除能
1: 使能

- Bit 2~0 **EPS2~EPS0**: 选择要访问端点的 FIFO
 000: 选择端点 0 的 FIFO
 001: 选择端点 1 的 FIFO
 010: 选择端点 2 的 FIFO
 011: 选择端点 3 的 FIFO
 100: 选择端点 4 的 FIFO
 101: 选择端点 5 的 FIFO
 110: 选择端点 6 的 FIFO
 111: 选择端点 7 的 FIFO

● AWR 寄存器

Bit	7	6	5	4	3	2	1	0
Name	AD6	AD5	AD4	AD3	AD2	AD1	AD0	WKEN
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 7~1 **AD6~AD0**: USB 设备地址
 Bit 0 **WKEN**: USB 远程唤醒控制位
 0: 除能
 1: 使能

AWR 寄存器由当前地址和远程唤醒功能控制位组成。AWR 寄存器的初始值为“00H”。写 AWR 寄存器直到 SETUP 阶段结束才会生效。

● STLO 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	STLO3	STLO2	STLO1	—
R/W	—	—	—	—	R/W	R/W	R/W	—
POR	—	—	—	—	x	x	x	—

“x”：未知

- Bit 7~4 未定义，读为“0”
 Bit 3~1 **STLO3~STLO1**: 端点 FIFO OUT 停止标志位
 0: 正常
 1: 停止

STLO 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLO 寄存器相应的位必须置“1”。USB 复位将会清除 STLO 寄存器。

- Bit 0 未定义，读为“0”

● STLO 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	STLO5	STLO4	STLO3	STLO2	STLO1	—
R/W	—	—	R/W	R/W	R/W	R/W	R/W	—
POR	—	—	x	x	x	x	x	—

“x”：未知

- Bit 7~6 未定义，读为“0”
 Bit 5~1 **STLO5~STLO1**: 端点 FIFO OUT 停止标志位
 0: 正常
 1: 停止

STLO 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLO 寄存器相应的位必须置“1”。USB 复位将会清除 STLO 寄存器。

- Bit 0 未定义，读为“0”

• STLO 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	STLO7	STLO6	STLO5	STLO4	STLO3	STLO2	STLO1	—
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”：未知

Bit 7~1 **STLO7~STLO1**：端点 FIFO OUT 停止标志位

0：正常

1：停止

STLO 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLO 寄存器相应的位必须置“1”。USB 复位将会清除 STLO 寄存器。

Bit 0 未定义，读为“0”

• STLI 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	STLI3	STLI2	STLI1	STLI0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	x	x	x	x

“x”：未知

Bit 7~4 未定义，读为“0”

Bit 3~0 **STLI3~STLI0**：端点 FIFO IN 停止标志位

0：正常

1：停止

STLI 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLI 寄存器相应的位必须置“1”。USB 复位将会清除 STLI 寄存器。

• STLI 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	STLI5	STLI4	STLI3	STLI2	STLI1	STLI0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	x	x	x	x	x	x

“x”：未知

Bit 7~6 未定义，读为“0”

Bit 5~0 **STLI5~STLI0**：端点 FIFO IN 停止标志位

0：正常

1：停止

STLI 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLI 寄存器相应的位必须置“1”。USB 复位将会清除 STLI 寄存器。

●STLI 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	STLI7	STLI6	STLI5	STLI4	STLI3	STLI2	STLI1	STLI0
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 **STLI7~STLI0**: 端点 FIFO IN 停止标志位

0: 正常

1: 停止

STLI 寄存器用于表明相应的端点工作是否正常。若端点工作存在异常，STLI 寄存器相应的位必须置“1”。USB 复位将会清除 STLI 寄存器。

●SIES 寄存器

Bit	7	6	5	4	3	2	1	0
Name	NMI	CRCF	—	NAK	IN	OUT	ERR	ASET
R/W	R/W	R/W	—	R	R	R/W	R/W	R/W
POR	x	x	—	x	x	x	x	x

“x”：未知

Bit 7 **NMI**: NAK 令牌中断控制位

0: 使能

1: 除能

如果该位被置位，当单片机发送 NAK 令牌到主机，不会产生中断，否则，当该位被清零，单片机发送 NAK 令牌到 USB 主机时，将进入中断子程序。该位可用于所有端点。

Bit 6 **CRCF**: CRC 错误标志位

0: 无错误

1: 错误

当以下三种情况发生，此位将被置位：CRC 错误、PID 错误和位填充错误。此位由 SIE 置位，由固件清零。

Bit 5 未定义，读为“0”

Bit 4 **NAK**: ACK 错误标志位

0: 无错误

1: 错误

SIE 发现错误时此位会被置位，因此 SIE 不会响应 USB 令牌 (NAK/ACK/DATA)。此位由 SIE 置位，由固件清零。

Bit 3 **IN**: 当前 USB 接收信号标志位

0: 不是 IN 令牌

1: 是 IN 令牌

该位用来表示当前 USB 从 PC 主机收到的信号为 IN 令牌。

Bit 2 **OUT**: USB OUT 令牌标志位

0: 不是 OUT 令牌

1: 是 OUT 令牌

该位用来表示接收到的信号为 OUT 令牌 (零长度的 OUT 令牌除外)。OUT 数据被读取后，该位由固件清除。此外，在下一个有效的 SETUP 令牌接收后，该位将由 SIE 清除。

Bit 1 **ERR**: 访问 FIFO 错误标志位

0: 无错误

1: 错误

该位用来表示访问 FIFO 时出现错误。该位由 SIE 置位，由固件清零。

Bit 0 **ASET:** 设备地址更新方式控制位
 0: 写入地址到 AWR 寄存器后立即更新设备地址
 1: USB 设置地址命令完成后更新设备地址
 该位用来设置 SIE 自动从 AWR 寄存器中加载设备地址。若此位由固件置位，当 PC 主机通过 IN 操作成功地从设备中读取到数据时，SIE 将会从 AWR 寄存器中加载地址。否则，若此位清零，写入 AWR 寄存器中地址将会立即生效。所以，为了能够正常工作，接收到下一个有效的 SETUP 令牌后，固件必须清除此位。

● MISC 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	LEN0	READY	SETCMD	—	E3IDF	CLEAR	TX	REQUEST
R/W	R	R	R/W	—	R/W	R/W	R/W	R/W
POR	x	x	x	—	x	x	x	x

“x”：未知

Bit 7 **LEN0:** 零大小数据包标志位
 0: 无零大小数据包
 1: 零长度的数据包
 此位表示主机发送了零大小数据包给 MCU。读取相应的 FIFO 后必须清除此位。

Bit 6 **READY:** 所需 FIFO 就绪标志位
 0: 未就绪
 1: 就绪

Bit 5 **SETCMD:** SETUP 命令标志位
 0: 不是 SETUP 命令
 1: SETUP 令牌
 此位用来表示 FIFO 内数据是否为 SETUP 命令。它由硬件置位，由固件清除。

Bit 4 未定义，读为“0”

Bit 3 **E3IDF:** 端点 3 输入 FIFO 选择位
 0: 单缓存
 1: 双缓存

Bit 2 **CLEAR:** 清除 FIFO 控制位
 0: 除能
 1: 使能
 MCU 请求清除 FIFO，即使 FIFO 尚未准备好。FIFO 清除后，USB 接口将发送 force_tx_err 告诉主机正在运行的数据是否为主机要读取的数据。

Bit 1 **TX:** 数据写入 FIFO 状态标志位
 0: 从 FIFO 读取数据
 1: 写入数据到 FIFO
 表示单片机访问方向和传输是否结束。当此位设置为高，意味着单片机想要写数据到 FIFO。写操作完成后，在终止 FIFO 请求之前需将其清零，以表示数据传输结束。对于单片机的读操作，需将该位清零，读操作完成后需将其置高。

Bit 0 **REQUEST:** 所需 FIFO 请求状态标志位
 0: 无请求
 1: 请求
 选择所需端点并将此位置高，便可访问相应的 FIFO。完成后，此位必须清零。

● MISC 寄存器 – HT66FB550/HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	LEN0	READY	SETCMD	E4ODF	E3IDF	CLEAR	TX	REQUEST
R/W	R	R	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 **LEN0**: 零大小数据包标志位
 0: 无零大小数据包
 1: 零大小的数据包
 此位表示主机发送了零大小数据包给 MCU。读取相应的 FIFO 后必须清除此位。
- Bit 6 **READY**: 所需 FIFO 就绪标志位
 0: 未就绪
 1: 就绪
- Bit 5 **SETCMD**: SETUP 命令标志位
 0: 不是 SETUP 命令
 1: SETUP 令牌
 此位用来表示 FIFO 内数据是否为 SETUP 命令。它由硬件置位，由固件清除。
- Bit 4 **E4ODF**: 端点 4 输出 FIFO 选择位
 0: 单缓存
 1: 双缓存
- Bit 3 **E3IDF**: 端点 3 输入 FIFO 选择位
 0: 单缓存
 1: 双缓存
- Bit 2 **CLEAR**: 清除 FIFO 控制位
 0: 除能
 1: 使能
 MCU 请求清除 FIFO，即使 FIFO 尚未准备好。FIFO 清除后，USB 接口将发送 force_tx_err 告诉主机正在运行的数据是否为主机要读取的数据。
- Bit 1 **TX**: 数据写入 FIFO 状态标志位
 0: 从 FIFO 读取数据
 1: 写入数据到 FIFO
 表示单片机访问方向和传输是否结束。当此位设置为高，意味着单片机想要写数据到 FIFO。写操作完成后，在终止 FIFO 请求之前需将其清零，以表示数据传输结束。对于单片机的读操作，需将该位清零，读操作完成后需将其置高。
- Bit 0 **REQUEST**: 所需 FIFO 请求状态标志位
 0: 无请求
 1: 请求
 选择所需端点并将此位置高，便可访问相应的 FIFO。完成后，此位必须清零。

● UFOEN 寄存器 – HT66FB540/HT66FB542

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	SETO3	SETO2	SETO1	DATATG
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 **SETO3**: EP3 输出 FIFO 控制位
 0: 除能
 1: 使能

- Bit 2 **SETO2:** EP2 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETO1:** EP1 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **DATATG:** DATA 令牌触发位
0: 低
1: 高

● **UFOEN 寄存器 – HT66FB550**

Bit	7	6	5	4	3	2	1	0
Name	—	—	SETO5	SETO4	SETO3	SETO2	SETO1	DATATG
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 **SETO5:** EP5 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 4 **SETO4:** EP4 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 3 **SETO3:** EP3 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 2 **SETO2:** EP2 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETO1:** EP1 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **DATATG:** DATA 令牌触发位
0: 低
1: 高

● **UFOEN 寄存器 – HT66FB560**

Bit	7	6	5	4	3	2	1	0
Name	SETO7	SETO6	SETO5	SETO4	SETO3	SETO2	SETO1	DATATG
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 **SETO7:** EP7 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 6 **SETO6:** EP6 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 5 **SETO5:** EP5 输出 FIFO 控制位
0: 除能
1: 使能

- Bit 4 **SETO4:** EP4 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 3 **SETO3:** EP3 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 2 **SETO2:** EP2 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETO1:** EP1 输出 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **DATATG:** DATA 令牌触发位
0: 低
1: 高

● **UFIEN 寄存器 – HT66FB540/HT66FB542**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	SETI3	SETI2	SETI1	FIFO_def
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 **SETI3:** EP3 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 2 **SETI2:** EP2 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETI1:** EP1 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **FIFO_def:** FIFO 配置重新定义控制位
0: 除能
1: 使能

若此位置高，SIE 需重新定义 FIFO 的配置。该位由 SIE 自动清零。

● **UFIEN 寄存器 – HT66FB550**

Bit	7	6	5	4	3	2	1	0
Name	—	—	SETI5	SETI4	SETI3	SETI2	SETI1	FIFO_def
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 **SETI5:** EP5 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 4 **SETI4:** EP4 输入 FIFO 控制位
0: 除能
1: 使能

- Bit 3 **SETI3:** EP3 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 2 **SETI2:** EP2 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETI1:** EP1 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **FIFO_def:** FIFO 配置重新定义控制位
0: 除能
1: 使能

若此位置高，SIE 需重新定义 FIFO 的配置。该位由 SIE 自动清零。

• UFIEN 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	SETI7	SETI6	SETI5	SETI4	SETI3	SETI2	SETI1	FIFO_def
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 **SETI7:** EP7 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 6 **SETI6:** EP6 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 5 **SETI5:** EP5 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 4 **SETI4:** EP4 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 3 **SETI3:** EP3 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 2 **SETI2:** EP2 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 1 **SETI1:** EP1 输入 FIFO 控制位
0: 除能
1: 使能
- Bit 0 **FIFO_def:** FIFO 配置重新定义控制位
0: 除能
1: 使能

若此位置高，SIE 需重新定义 FIFO 的配置。该位由 SIE 自动清零。

●UFC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	E3FS1	E3FS0	E2FS1	E2FS0	E1FS1	E1FS0	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	—
POR	0	0	0	0	0	0	—	—

Bit 7~6 **E3FS1~E3FS0**: 端点 3 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 5~4 **E2FS1~E2FS0**: 端点 2 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 3~2 **E1FS1~E1FS0**: 端点 1 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 1~0 未定义, 读为“0”

●UFC1 寄存器 – HT66FB550

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	E5FS1	E5FS0	E4FS1	E4FS0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~6 **E7FS1~E7FS0**: 端点 7 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 5~4 **E6FS1~E6FS0**: 端点 6 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 3~2 **E5FS1~E5FS0**: 端点 5 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

Bit 1~0 **E4FS1~E4FS0**: 端点 4 的 FIFO 大小选择位
 00: 8-byte
 01: 16-byte
 10: 32-byte
 11: 64-byte

●UFC1 寄存器 – HT66FB560

Bit	7	6	5	4	3	2	1	0
Name	E7FS1	E7FS0	E6FS1	E6FS0	E5FS1	E5FS0	E4FS1	E4FS0
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 **E7FS1~E7FS0**: 端点 7 的 FIFO 大小选择位
00: 8-byte
01: 16-byte
10: 32-byte
11: 64-byte

Bit 5~4 **E6FS1~E6FS0**: 端点 6 的 FIFO 大小选择位
00: 8-byte
01: 16-byte
10: 32-byte
11: 64-byte

Bit 3~2 **E5FS1~E5FS0**: 端点 5 的 FIFO 大小选择位
00: 8-byte
01: 16-byte
10: 32-byte
11: 64-byte

Bit 1~0 **E4FS1~E4FS0**: 端点 4 的 FIFO 大小选择位
00: 8-byte
01: 16-byte
10: 32-byte
11: 64-byte

寄存器名称	位							
	7	6	5	4	3	2	1	0
FIFO0	D7	D6	D5	D4	D3	D2	D1	D0
FIFO1	D7	D6	D5	D4	D3	D2	D1	D0
FIFO2	D7	D6	D5	D4	D3	D2	D1	D0
FIFO3	D7	D6	D5	D4	D3	D2	D1	D0

USB 端点访问寄存器 – HT66FB540/HT66FB542

寄存器名称	位							
	7	6	5	4	3	2	1	0
FIFO0	D7	D6	D5	D4	D3	D2	D1	D0
FIFO1	D7	D6	D5	D4	D3	D2	D1	D0
FIFO2	D7	D6	D5	D4	D3	D2	D1	D0
FIFO3	D7	D6	D5	D4	D3	D2	D1	D0
FIFO4	D7	D6	D5	D4	D3	D2	D1	D0
FIFO5	D7	D6	D5	D4	D3	D2	D1	D0

USB 端点访问寄存器 – HT66FB550

寄存器名称	位							
	7	6	5	4	3	2	1	0
FIFO0	D7	D6	D5	D4	D3	D2	D1	D0
FIFO1	D7	D6	D5	D4	D3	D2	D1	D0
FIFO2	D7	D6	D5	D4	D3	D2	D1	D0
FIFO3	D7	D6	D5	D4	D3	D2	D1	D0
FIFO4	D7	D6	D5	D4	D3	D2	D1	D0
FIFO5	D7	D6	D5	D4	D3	D2	D1	D0
FIFO6	D7	D6	D5	D4	D3	D2	D1	D0
FIFO7	D7	D6	D5	D4	D3	D2	D1	D0

USB 端点访问寄存器 – HT66FB560

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。所有位需要按系统的需要定义，具体内容可参考下表：

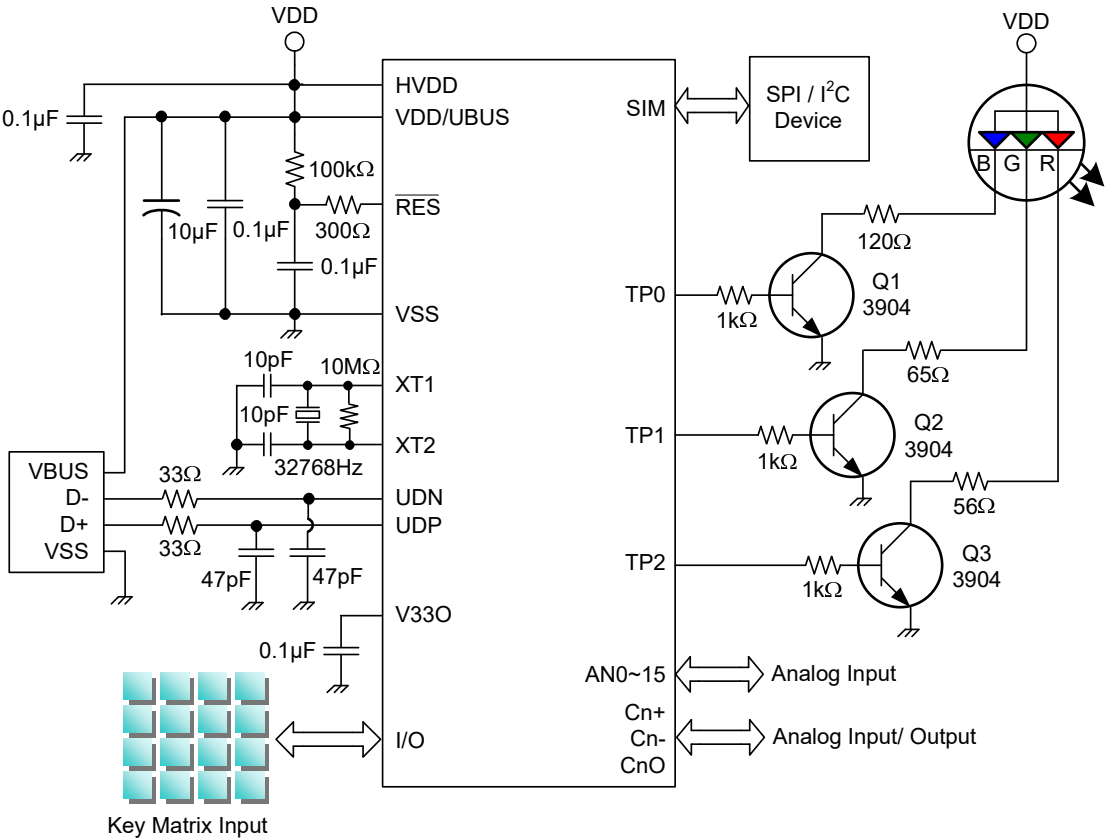
● HT66FB540/HT66FB550/HT66FB560

编号	选项
振荡器选项	
1	高速振荡器类型选择 – f_H : 1. HIRC (默认) 2. HXT
2	低速振荡器类型选择 – f_L : 1. LIRC (默认) 2. LXT
3	f_{SUB} 时钟选择: 1. LIRC (默认) 2. LXT
4	时基时钟选择 – f_{TBC} : 1. LIRC (默认) 2. LXT
晶振频率选项	
5	时钟频率选择: 1. 12MHz 2. 6MHz
I/O 或 VDDIO 选项	
6	I/O 或 VDDIO 引脚控制位: 1. VDDIO (默认) 2. I/O (PE0)

● HT66FB542

编号	选项
振荡器选项	
1	高速振荡器类型选择 – f_H : 1. HIRC (默认) 2. HXT
晶振频率选项	
2	时钟频率选择: 1. 12MHz 2. 6MHz
I/O 或 VDDIO 选项	
3	I/O 或 VDDIO 引脚控制位: 1. VDDIO (默认) 2. I/O (PE0)

应用电路



指令集

简介

任何单片机成功运作的核心在于它的指令集，此指令集为一组程序指令码，用来指导单片机如何去执行指定的工作。在 Holtek 单片机中，提供了丰富且灵活的指令，共超过六十条，程序设计者可以事半功倍地实现他们的应用。

为了更加容易理解各种各样的指令码，接下来按功能分组介绍它们。

指令周期

大部分的操作均只需要一个指令周期来执行。分支、调用或查表则需要两个指令周期。一个指令周期相当于四个系统时钟周期，因此如果在 8MHz 的系统时钟振荡器下，大部分的操作将在 0.5 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者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，或者是内部数据位的值。

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

下表中说明了按功能分类的指令集，用户可以将该表作为基本的指令参考。

惯例

- x: 立即数
- m: 数据存储器地址
- A: 累加器
- i: 第 0~7 位
- addr: 程序存储器地址

助记符	说明	指令周期	影响标志位
算术运算			
ADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	1	Z, C, AC, OV
ADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	1 注	Z, C, AC, OV
ADD A, x	ACC 与立即数相加，结果放入 ACC	1	Z, C, AC, OV
ADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	1	Z, C, AC, OV
ADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	1 注	Z, C, AC, OV
SUB A, x	ACC 与立即数相减，结果放入 ACC	1	Z, C, AC, OV
SUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z, C, AC, OV
SUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	1 注	Z, C, AC, OV
SBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z, C, AC, OV
SBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	1 注	Z, C, AC, OV
DAA [m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	1 注	C
逻辑运算			
AND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	1	Z
OR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	1	Z
XOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	1	Z
ANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	1 注	Z
ORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	1 注	Z
XORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	1 注	Z
AND A, x	ACC 与立即数做“与”运算，结果放入 ACC	1	Z
OR A, x	ACC 与立即数做“或”运算，结果放入 ACC	1	Z
XOR A, x	ACC 与立即数做“异或”运算，结果放入 ACC	1	Z
CPL [m]	对数据存储器取反，结果放入数据存储器	1 注	Z
CPLA [m]	对数据存储器取反，结果放入 ACC	1	Z
递增和递减			
INCA [m]	递增数据存储器，结果放入 ACC	1	Z
INC [m]	递增数据存储器，结果放入数据存储器	1 注	Z
DECA [m]	递减数据存储器，结果放入 ACC	1	Z
DEC [m]	递减数据存储器，结果放入数据存储器	1 注	Z
移位			
RRA [m]	数据存储器右移一位，结果放入 ACC	1	无
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

助记符	说明	指令周期	影响标志位
RL [m]	数据存储器左移一位，结果放入数据存储器	1 ^注	无
RLCA [m]	带进位将数据存储器左移一位，结果放入 ACC	1	C
RLC [m]	带进位将数据存储器左移一位，结果放入数据存储器	1 ^注	C
数据传送			
MOV A,[m]	将数据存储器送至 ACC	1	无
MOV [m],A	将 ACC 送至数据存储器	1 ^注	无
MOV A, x	将立即数送至 ACC	1	无
位运算			
CLR [m].i	清除数据存储器的位	1 ^注	无
SET [m].i	置位数据存储器的位	1 ^注	无
转移			
JMP addr	无条件跳转	2	无
SZ [m]	如果数据存储器为零，则跳过下一条指令	1 ^注	无
SZA [m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	1 ^注	无
SZ [m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	1 ^注	无
SNZ [m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	1 ^注	无
SIZ [m]	递增数据存储器，如果结果为零，则跳过下一条指令	1 ^注	无
SDZ [m]	递减数据存储器，如果结果为零，则跳过下一条指令	1 ^注	无
SIZA [m]	递增数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1 ^注	无
SDZA [m]	递减数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1 ^注	无
CALL addr	子程序调用	2	无
RET	从子程序返回	2	无
RET A, x	从子程序返回，并将立即数放入 ACC	2	无
RETI	从中断返回	2	无
查表			
TABRD [m]	读取特定页或当前页的 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
其它指令			
NOP	空指令	1	无
CLR [m]	清除数据存储器	1 ^注	无
SET [m]	置位数据存储器	1 ^注	无
CLR WDT	清除看门狗定时器	1	TO, PDF
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。

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
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
ADD 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
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
ADDM 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
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
ANDMA 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	$[m] \leftarrow ACC$
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	$ACC \leftarrow ACC \text{ "OR" } [m]$
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	$ACC \leftarrow ACC \text{ "OR" } x$
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	$[m] \leftarrow ACC \text{ "OR" } [m]$
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	$Program\ Counter \leftarrow Stack$
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	$Program\ Counter \leftarrow Stack$ $ACC \leftarrow x$
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 EMI $\leftarrow$ 1
影响标志位	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

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	$[m] \leftarrow ACC - [m] - \overline{C}$
影响标志位	OV、Z、AC、C
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
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
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

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
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	$ACC \leftarrow [m]$ ，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 or current page) to TBLH and Data Memory
指令说明	将表格指针 (TBHP 和 TBLP，若无 TBHP 则仅 TBLP) 所指的程序代码低字节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
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 ← ACC “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] ← ACC “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 ← ACC “XOR” x
影响标志位	Z

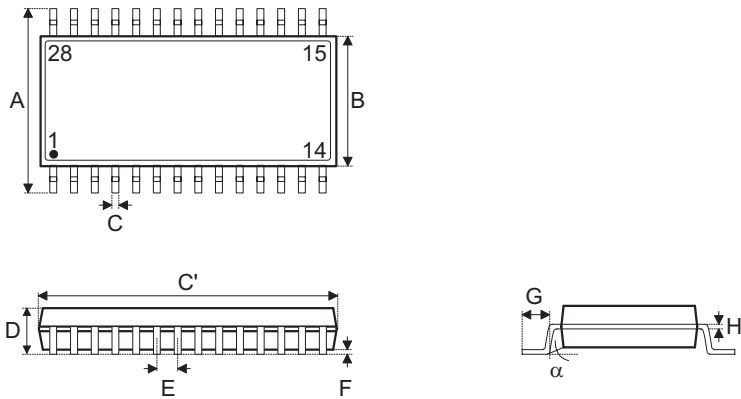
封装信息

请注意，这里提供的封装信息仅作为参考。由于这个信息经常更新，提醒用户咨询 [Holtek 网站](#) 以获取最新版本的 [封装信息](#)。

封装信息的相关内容如下所示，点击可链接至 Holtek 网站相关信息页面。

- 封装信息 (包括外形尺寸、包装带和卷轴规格)
- 封装材料信息
- 纸箱信息

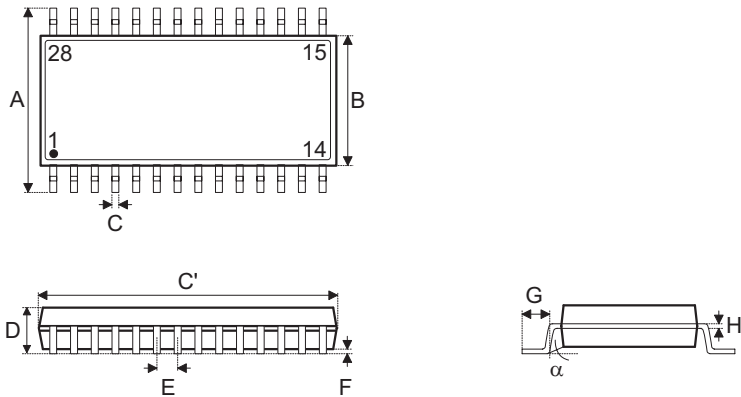
24-pin SSOP (150mil) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	0.236 BSC		
B	0.154 BSC		
C	0.008	—	0.012
C'	0.341 BSC		
D	—	—	0.069
E	0.025 BSC		
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符号	尺寸 (单位: mm)		
	最小值	典型值	最大值
A	6.00 BSC		
B	3.90 BSC		
C	0.20	—	0.30
C'	8.66 BSC		
D	—	—	1.75
E	0.635 BSC		
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

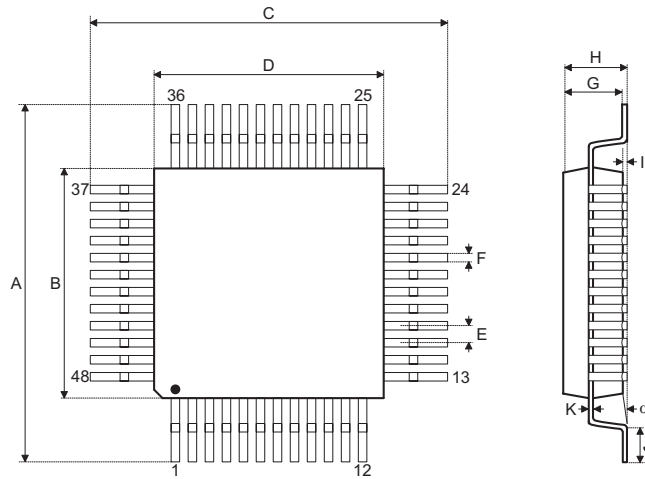
28-pin SSOP (150mil) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	0.236 BSC		
B	0.154 BSC		
C	0.008	—	0.012
C'	0.390 BSC		
D	—	—	0.069
E	0.025 BSC		
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符号	尺寸 (单位: mm)		
	最小值	典型值	最大值
A	6.00 BSC		
B	3.90 BSC		
C	0.20	—	0.30
C'	9.90 BSC		
D	—	—	1.75
E	0.635 BSC		
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

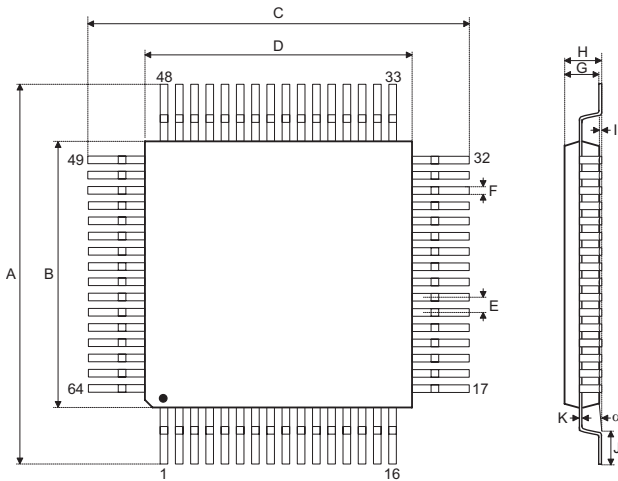
48-pin LQFP (7mm×7mm) 外形尺寸



符号	尺寸 (单位: inch)		
	最小	一般	最大
A	0.354 BSC		
B	0.276 BSC		
C	0.354 BSC		
D	0.276 BSC		
E	0.020 BSC		
F	0.007	0.009	0.011
G	0.053	0.055	0.057
H	—	—	0.063
I	0.002	—	0.006
J	0.018	0.024	0.030
K	0.004	—	0.008
α	0°	—	7°

符号	尺寸 (单位: mm)		
	最小	一般	最大
A	9.00 BSC		
B	7.00 BSC		
C	9.00 BSC		
D	7.00 BSC		
E	0.50 BSC		
F	0.17	0.22	0.27
G	1.35	1.40	1.45
H	—	—	1.60
I	0.05	—	0.15
J	0.45	0.60	0.75
K	0.09	—	0.20
α	0°	—	7°

64-pin LQFP (7mm×7mm) 外形尺寸



符号	尺寸 (单位: inch)		
	最小	一般	最大
A	0.354 BSC		
B	0.276 BSC		
C	0.354 BSC		
D	0.276 BSC		
E	0.016 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	一般	最大
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°

Copyright© 2024 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

本文件出版时 HOLTEK 已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。HOLTEK 不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。HOLTEK 就文中提到的信息及该信息之应用，不承担任何法律责任。此外，HOLTEK 并不推荐将 HOLTEK 的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。HOLTEK 特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用 HOLTEK 产品的风险完全由买方承担，如因该等使用导致 HOLTEK 遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使 HOLTEK 免受损害。HOLTEK (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。HOLTEK 在此并未明示或暗示授予任何知识产权。HOLTEK 拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。