



深圳富芯电子

RC6T8X634 数据手册

增强型 8 位触控类单片机

Rev. V1.6

重要声明：本公司保留对以下所有产品在可靠性、功能和设计方面作进一步说明的权利，同时保留在未通知的情况下，对本产品所有文档做更改的权利。客户在使用本产品时，请向我司销售人员索要最新文档，特此声明！

1 产品概述

RC6T8X634 (X 代表不同封装型号) 是一款高性能 8 位触摸型单片机, 最大支持 25 路触摸通道。内部集成高性能 8051 内核, 包含 16K Byte MTP, 64 Byte INFO, 128 Byte EEPROM, 1K Byte SRAM, 6 个 16 位 Timer (3 个可输出 PWM), 每个 Timer 均可作为定时器或计数器, 1 个独立时钟看门狗, 12-bit ADC (最大支持 25 路外部输入和 1 路内部 1/4VDD), 2 路 UART (两个独立定时器做波特率控制)、1 路 I2C 接口, 片内 POR, BOR 和 LVD, 内部 24MHz RC 振荡器, 内部 32KHz RC 振荡器。具有两种低功耗模式。

2 功能特性

◆ 内核:

- 高速 8051 内核 (1T)

- 指令周期可配:

- 2MHz, VDD ≥ 2.6V

- 12MHz, VDD ≥ 4.5V

- ◆ 工作电压: 2.6V~5.5V

- ◆ 工作温度: -40°C~105°C

- ◆ MTP ROM: 16K 字节 MTP (擦写次数 1000 次)

- ◆ EEPROM: 128 字节 (擦写次数 50000 次)

- ◆ SRAM: 1K 字节 SRAM

◆ 时钟:

- 内部 24MHz RC 振荡器 (可微调)

- 误差 ±1% (25°C~65°C, 2.6V~5.5V)

- 误差 ±2% (-40°C~85°C, 2.6V~5.5V)

- 误差 ±3% (-40°C~105°C, 2.6V~5.5V)

- 内部 32KHz 低速 RC 振荡器 (误差 ±10%)

◆ 复位:

- 上电复位

- 欠压复位 (2.40V、2.73V、3.10V、3.81V、4.46V)

- 看门狗溢出复位

- ◆ 低电压检测: LVD 共 5 级 (2.55V、2.98V、3.25V、4.06V、4.82V)

◆ 中断 (INT):

- Timer、VC3、WDT、PWM、CSD、ADC、UART、I2C、LVD、GPIO 共 19 个中断源, 所有 GPIO 可设上升沿、下降沿、双边沿、低电平中断

◆ 数字外设:

- 6 个 16 位高级定时器

- Timer0/Timer3/Timer4 支持 PWM 输出功能

- Timer5 支持捕获功能

- 支持周期中断

- 1 个 16 位睡眠定时器, 2 位看门狗定时器

- 2 路 UART (支持全双工和半双工)

- 1 路 I2C: 支持主机模式和从机模式

- 速率 100KHz/400KHz

◆ CSD:

- 最大支持 25 路触摸通道

- 支持伪随序列生成

- 支持硬件防水

◆ 12 位 ADC:

- 外部输入: 25 路

- 内部输入: 1 路 (1/4 VDD)

- 参考源 — 外部参考: VDD

内部参考: 1.2V/2.4V

- 可以配置管脚的边沿触发采样

◆ 26 个 GPIO:

- PT12、PT13 默认开漏上拉输出, 其余 I/O 默认为输入高阻态

- 所有 I/O 可单独配置 10K 上拉电阻 (匹配精度 5%)

◆ 省电模式:

- 深度休眠可由看门狗复位、睡眠定时器中断、引脚外部中断唤醒

- 深度休眠电流: 6.5uA (典型值)

◆ 仿真和烧录:

- 支持在线调试仿真和 IAP 功能

- 4 线烧写 (VDD, VSS, SDA, SCL)

◆ 封装:

- S0P28、TSS0P28、S0P20、S0P16

目 录

1	产品概述.....	2
2	功能特性.....	2
3	模块框图.....	5
4	引脚说明.....	6
4.1	RC6T80634GA 封装脚位图	6
4.2	RC6T80634GC 封装脚位图	6
4.3	RC6T80634EA 封装脚位图	7
4.4	RC6T81634EA 封装脚位图	7
4.5	RC6T81634DA 封装脚位图	8
4.6	引脚功能描述	8
4.7	引脚复用	9
5	端口结构.....	10
5.1	I/O 口结构框图	10
5.2	配置 GPIO	11
5.3	GPIO 中断	12
5.4	寄存器定义	12
6	CPU 功能简介.....	22
6.1	内核 SFR 寄存器	22
7	中断控制系统.....	26
7.1	中断向量表	26
7.2	中断优先级	26
7.3	寄存器定义	27
7.4	外部中断	29
7.5	外部中断配置流程	29
8	存储器系统.....	30
8.1	存储器系统概述	30
8.2	程序存储器映射	30
8.3	数据存储器映射	30
8.4	MTP 与 SRAM 结构	32
8.5	寄存器映射	32
8.6	MTP 控制器	34
9	时钟控制.....	39
9.1	时钟概述	39
9.2	时钟信号说明	39
9.3	时钟结构框图	39
9.4	内部主振 (IMO)	40
9.5	内部低速振荡器 (ILO)	40
9.6	寄存器定义	40
10	外设资源.....	42
10.1	TIMER.....	42
10.2	CHECKSUM.....	60
10.3	ADC.....	62
10.4	UART0/1	68

10.5	I2C	74
10.6	CSD	80
10.7	PI_MUX	80
11	系统控制.....	100
11.1	复位	100
11.2	电源管理	100
11.3	看门狗	102
11.4	系统模式控制寄存器	105
11.5	模拟电路控制	106
12	电气特性.....	110
12.1	绝对最大额定值	110
12.2	推荐工作条件	110
12.3	电源功耗特性	111
12.4	内部时钟源特性	111
12.5	存储器特性	112
12.6	I/O 引脚特性	112
12.7	ADC 特性	113
12.8	ESD 特性	114
13	封装信息.....	115
13.1	SOP28.....	115
13.2	TSSOP28	116
13.3	SOP20.....	117
13.4	SOP16.....	118
14	命名规则.....	119
15	订单信息.....	119
16	开发工具.....	119
17	版本说明.....	120

3 模块框图

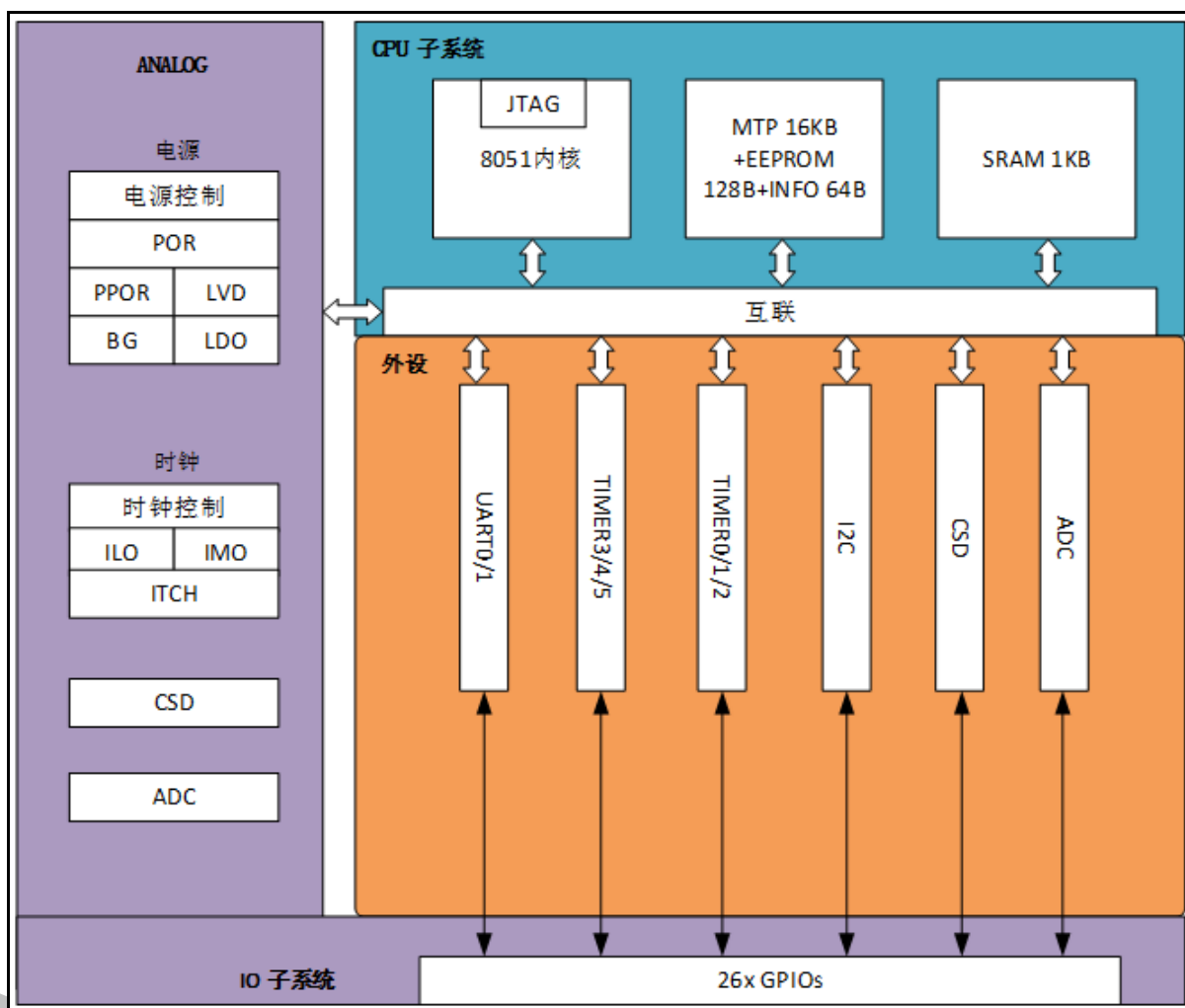


图 3-1 RC6T8X634 模块框图

4 引脚说明

4.1 RC6T80634GA 封装脚位图

VDD	1	28	P0.0
CMOD/P1.0	2	27	P0.1
VSS	3	26	P0.2
ADCREP/P3.1	4	25	P0.3
P3.0	5	24	P0.4
P1.1	6	23	P0.5
TCK/SCL/P1.2	7	22	P0.6
TDIO/SDA/P1.3	8	21	P0.7
P1.4	9	20	P2.0
P1.5	10	19	P2.1
P1.6	11	18	P2.2
P1.7	12	17	P2.3
P2.7	13	16	P2.4
P2.6	14	15	P2.5

SOP28

图 4-1 RC6T80634GA (SOP28) 封装脚位图 (支持引脚功能全映射)

4.2 RC6T80634GC 封装脚位图

VDD	1	28	P0.0
CMOD/P1.0	2	27	P0.1
VSS	3	26	P0.2
ADCREP/P3.1	4	25	P0.3
P3.0	5	24	P0.4
P1.1	6	23	P0.5
TCK/SCL/P1.2	7	22	P0.6
TDIO/SDA/P1.3	8	21	P0.7
P1.4	9	20	P2.0
P1.5	10	19	P2.1
P1.6	11	18	P2.2
P1.7	12	17	P2.3
P2.7	13	16	P2.4
P2.6	14	15	P2.5

TSSOP28

图 4-2 RC6T80634GC (TSSOP28) 封装脚位图 (支持引脚功能全映射)

4.3 RC6T80634EA 封装脚位图

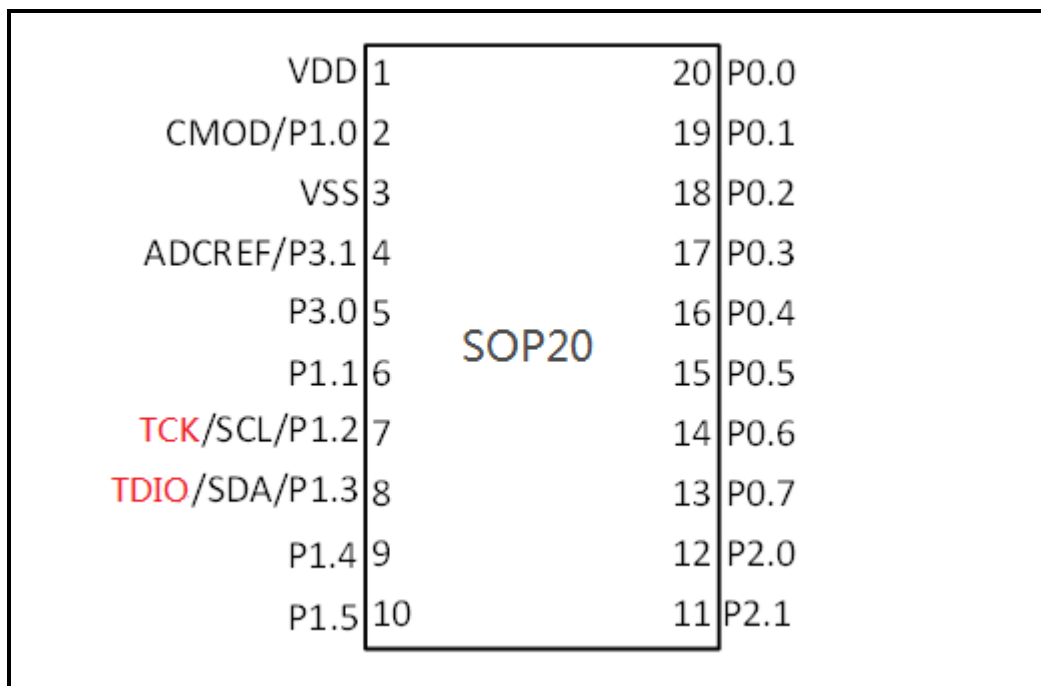


图 4-3 RC6T80634EA (SOP20) 封装脚位图 (支持引脚功能全映射)

4.4 RC6T81634EA 封装脚位图

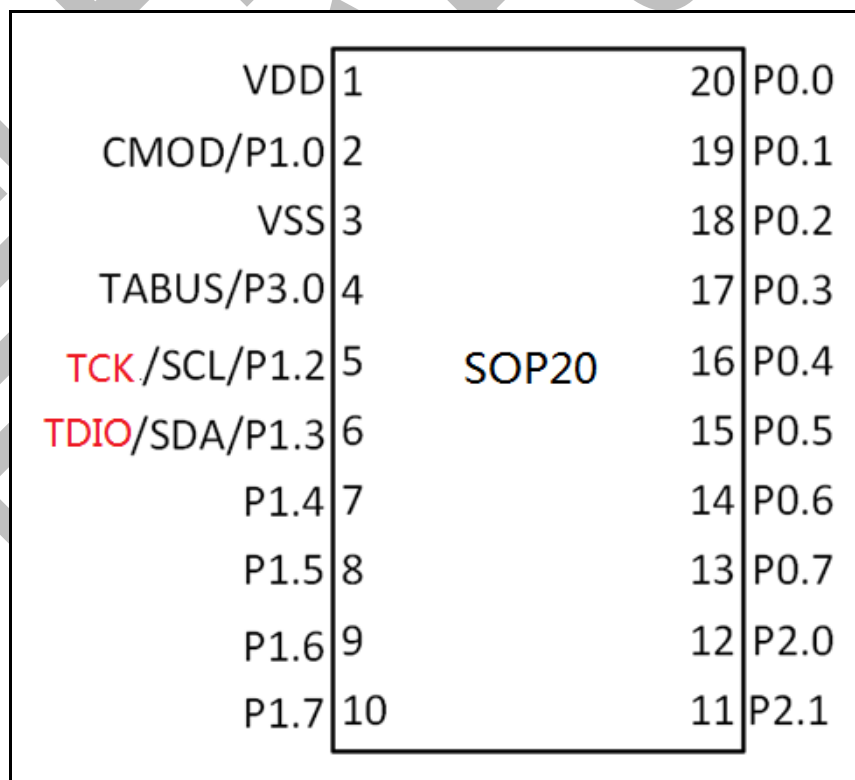


图 4-4 RC6T81634EA (SOP20) 封装脚位图 (支持引脚功能全映射)

4.5 RC6T81634DA 封装脚位图

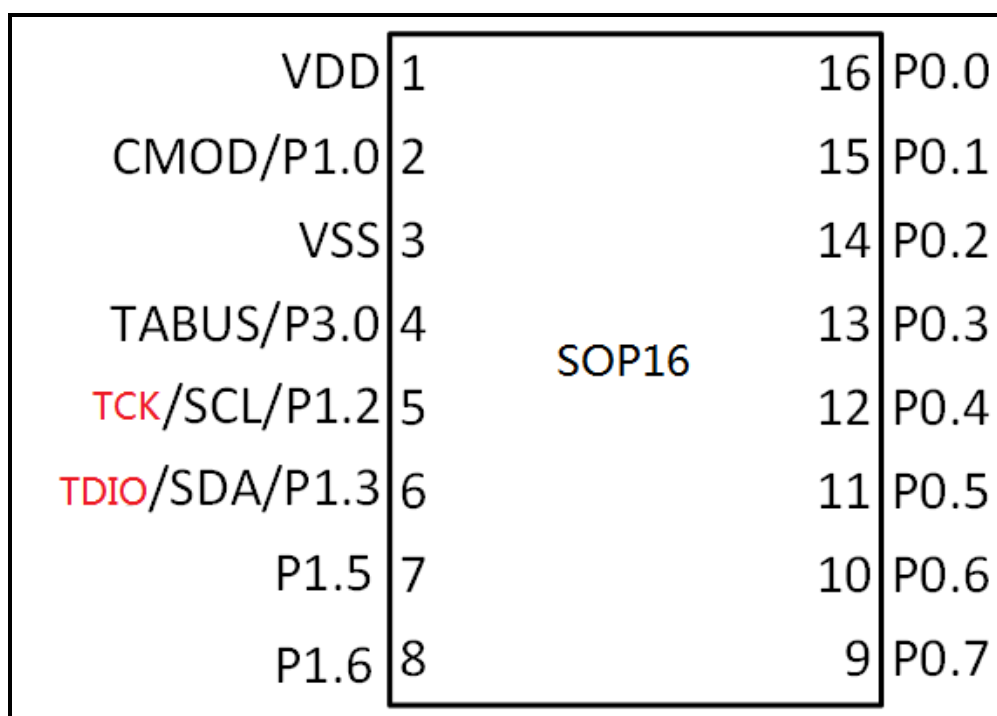


图 4-5 RC6T81634DA (SOP16) 封装脚位图(支持引脚功能全映射)

4.6 引脚功能描述

表 4-1 引脚功能描述

引脚名称	I/O 类型	说明
GPIO 端口 （所有的 GPIO 都可以产生中断）		
P0.0~P0.7	I/O	8 位双向 I/O 端口 P0，可配置为大电流 I/O 驱动 LED
P1.0~P1.7	I/O	8 位双向 I/O 端口 P1（若启用 CSD，则 P1.0 不可作为 GPIO）
P2.0~P2.7	I/O	8 位双向 I/O 端口 P2
P3.0~P3.1	I/O	2 位双向 I/O 端口 P3
CSD （最大支持 25 路 TOUCH）		
TK0~TK24	I	25 路触摸通道，支持硬件防水 TK0~TK7 - P0.0~P0.7 TK8~TK15 - P2.0~P2.7 TK16~TK22 - P1.7~P1.1 TK23~TK24 - P3.0~P3.1
CMOD	I	Touch Key 触摸外接电容
TIMER		
PWM0	0	Timer0 PWM 输出，功能全映射到所有 GPIO
PWM1	0	Timer3 PWM 输出，功能全映射到所有 GPIO
PWM2	0	Timer4 PWM 输出，功能全映射到所有 GPIO
PWM_ST_PIN0	I	Timer4 PWM 启停 I/O，功能全映射到所有 GPIO
PWM_ST_PIN1	I	Timer4 PWM 启停 I/O，功能全映射到所有 GPIO

T5	I	Timer5 计数外部输入, 功能全映射到所有 GPIO
T5EX	I	Timer5 捕获外部输入, 功能全映射到所有 GPIO
UART		
TX0	0	UART0 数据输出引脚, 功能全映射到所有 GPIO
RX0	I	UART0 数据输入引脚, 功能全映射到所有 GPIO
TX1	0	UART1 数据输出引脚, 功能全映射到所有 GPIO
RX1	I	UART1 数据输入引脚, 功能全映射到所有 GPIO
I2C		
SCL	I/O	I2C 时钟引脚, 功能全映射到所有 GPIO
SDA	I/O	I2C 数据引脚, 功能全映射到所有 GPIO (注意: 上电 2.1mS 左右, 该脚会输出 50uS 左右的低电平)
电源		
VDD	POWER	电源 (2.6V~5.5V)
VSS	POWER	地
烧录和仿真		
TCK	I	烧录和仿真口时钟引脚
TDIO	0	烧录和仿真 DATA 引脚
ADC		
AIN0~AIN24	I	25 路 ADC 外部采样输入通道
ADC_ETR	I	ADC 外部触发采样输入引脚, 功能全映射到所有 GPIO
ADC_REF	I	ADC 外部参考引脚
烧录和仿真		
TCK	I	烧录和仿真口时钟引脚
TDIO	0	烧录和仿真 DATA 引脚

4.7 引脚复用

RC6T8X634x 系列芯片支持数字外设使用不同的管脚位置, 通过 DO_SEL0~12, DO_SEL0~8 寄存器来进行设置, 实现数字功能的全映射。

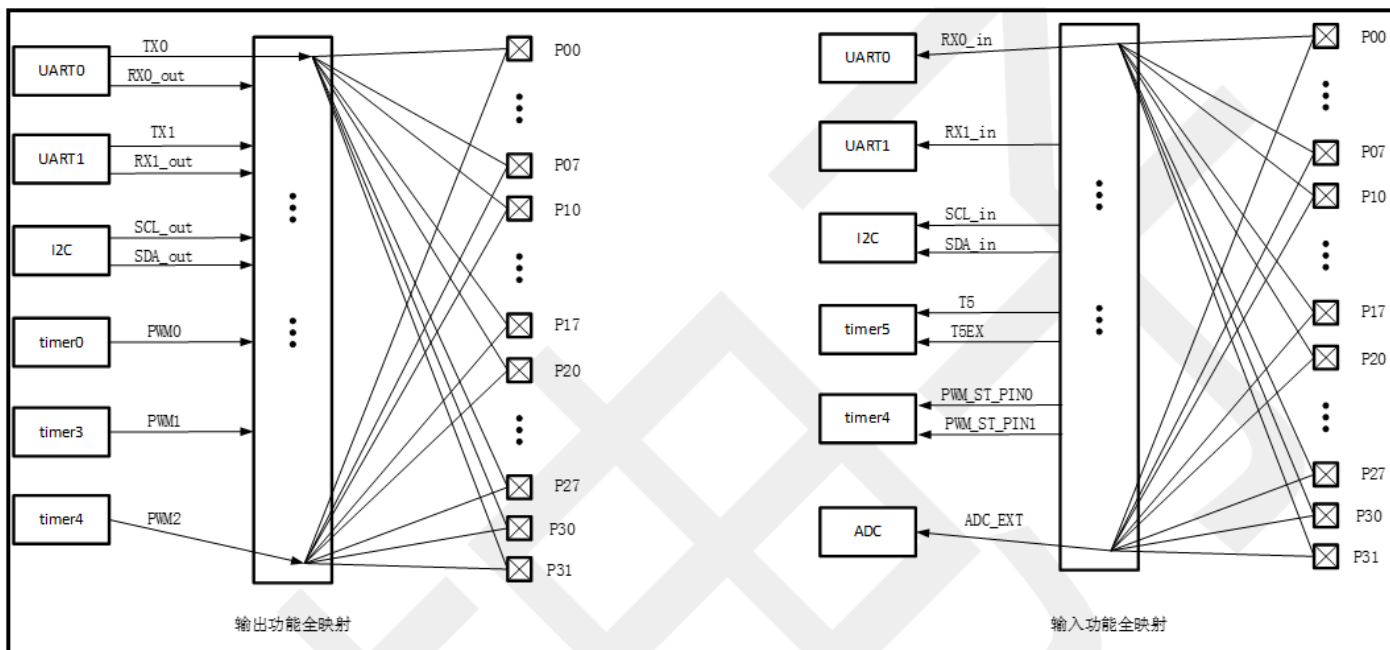


图 4-8 数字功能全映射框图

5 端口结构

5.1 I/O 口结构框图

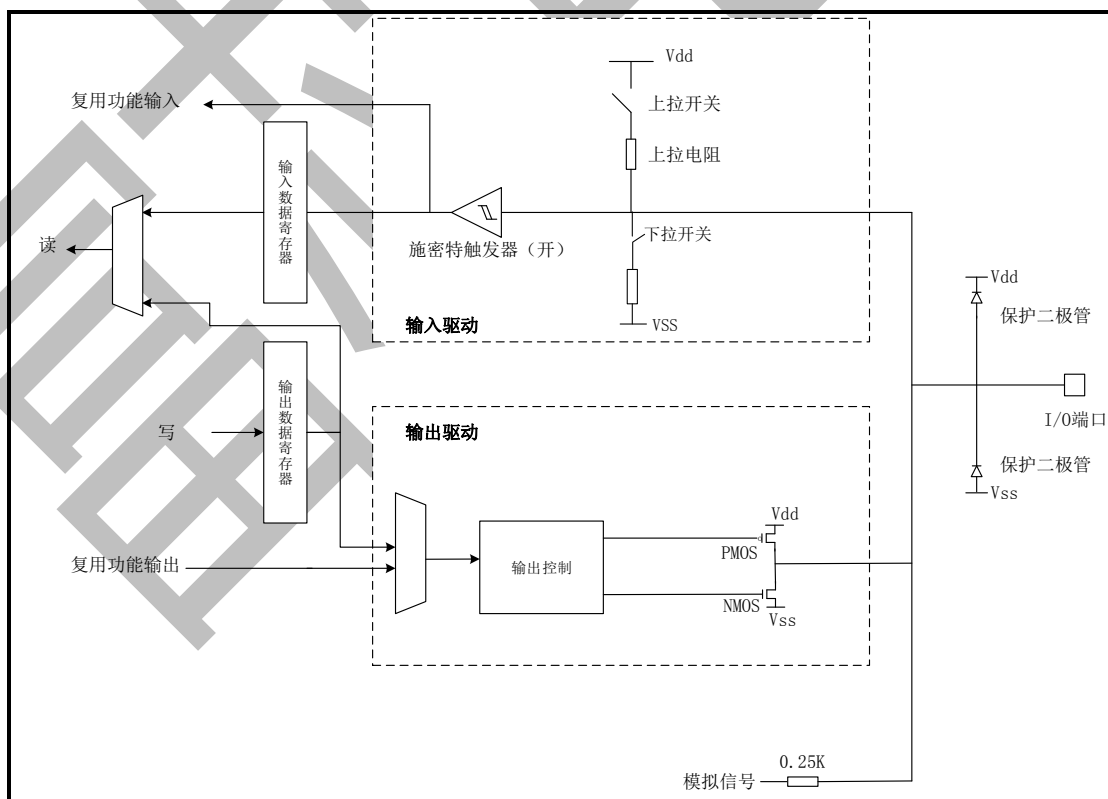


图 5-1 GPIO 结构图

5.2 配置 GPIO

RC6T8X634 的 GPIO 支持多种驱动模式，可通过 Px_DMx 寄存器进行配置。以 P0 口为例，配置 P0 口需操作 P0_DM0~P0_DM3 寄存器，P0_DM0 寄存器可配置 P0_0 与 P0_1 引脚模式（bit0~bit2 位配置 P0_0 引脚模式，bit 4~bit6 位配置 P0_1 引脚模式），P0_DM1 可配置 P0_2 与 P0_3 引脚模式，依次类推，P1~P3 口配置与 P0 口相同。GPIO 模式的配置说明如下表所示：

表 5-1 GPIO 驱动模式

Px_DMx 的 bit2 位	Px_DMx 的 bit1 位	Px_DMx 的 bit0 位	驱动模式	施密特触发器	应用说明
0	0	0	模拟高阻	关	可以用于 ADC 采样。 此时读 DR 寄存器读到的是输入数据寄存器的值。
0	0	1	强推挽输出	开	普通的 GPIO 输出或 PWM 输出、UART 通讯。此时读 DR 寄存器读到的是输出数据寄存器的值。
0	1	0	开漏	开	双向模式或者 I2C 通讯。 配置 Px 的对应 I/O 为开漏低输出，此时数字输入使能打开，此时读 DR 寄存器读到的是输入数据寄存器的值。
1	0	0	数字高阻	开	配置 Px 的对应 I/O 为高阻输出，此时数字输入使能打开，此时读 DR 寄存器读到的是输入数据寄存器的值。
1	0	1	防水扫描	关	配置该模式下的按键，与正常扫描按键进行配合，从而完成防水功能。
1	1	0	电容检测	关	触摸按键
1	1	1	CMOD 模式	关	外挂 Cmod 电容，P1.0 配置成该模式。

注 1：Open Drain 模式下需要配置 Px_PU 寄存器使能上拉电阻。在开漏模式下，当 IO 口为输入时，DR 寄存器需要设置为 1。

注 2：当 GPIO 配置成开漏低输出时，需要配置使能内部上拉或者接外部上拉，此时如果管脚输出低电平，则在该管脚上会形成上拉电阻到地的通路，预计会有 VDD/10K 的电流（比如 5V 电源供电，则会有 500uA 电流）通过。在系统进入 deepsleep 时，需要注意开漏的管脚避免输出低电平。

注 3：配置防水功能时，配置成防水模式的按键数量不能超过 4 个，也就是组成一个防水组的按键不能超过 5 个。

注 4：表格中 Px_DMx 的 bit0~bit2 位可替换为 bit4~bit6 位。

读-修改-写 指令

从 SFR 或内部 RAM 读一个字节，对其进行修改，并重新写回去的指令，叫做读-修改-写指令。当目标是一个 I/O 端口或一个端口位，这些指令读内部输出锁存器而不是外部引脚的状态，这种指令读端口 SFR 的值，对其进行修改，并写回到 SFR 端口。所有读-修改-写的指令如下所列：

指令	描述
ANL	逻辑 与. (ANL direct, A and ANL direct, #data)
ORL	逻辑 或. (ORL direct, A and ORL direct, #data)
XRL	逻辑 异或 OR. (XRL direct, A and XRL direct, #data)
JBC	if bit = 1 转跳指令并清除. (JBC bit, rel)
CPL	位取反. (CPL bit)
INC	加一指令. (INC direct)

DEC 减一指令. (DEC direct)
 DJNZ 减一不为零转跳指令. (DJNZ direct, rel)
 MOV bit, C 移进位标志到位. (MOV bit, C)
 CLR bit 清位. (CLR bit)
 SETB bit 置位. (SETB bit)

5.3 GPIO 中断

GPIO 支持 4 种中断触发模式：上升沿、下降沿、上下沿以及低电平触发；

从图 5-3 中可以看出中断的 4 种模式下的 RTL 实现的门电路，数据进入芯片后，通过三级寄存器同步后，经过一个异或门电路，得到了 3 种边沿触发的中断信号，再通过一个与门即可实现边沿触发中断；低电平实现中断触发则是在第二级寄存器同步后经过非门，再和中断选择寄存器 PT_IC 相与，实现低电平时中断触发。四种触发模式任意一种都能产生中断标志信号。

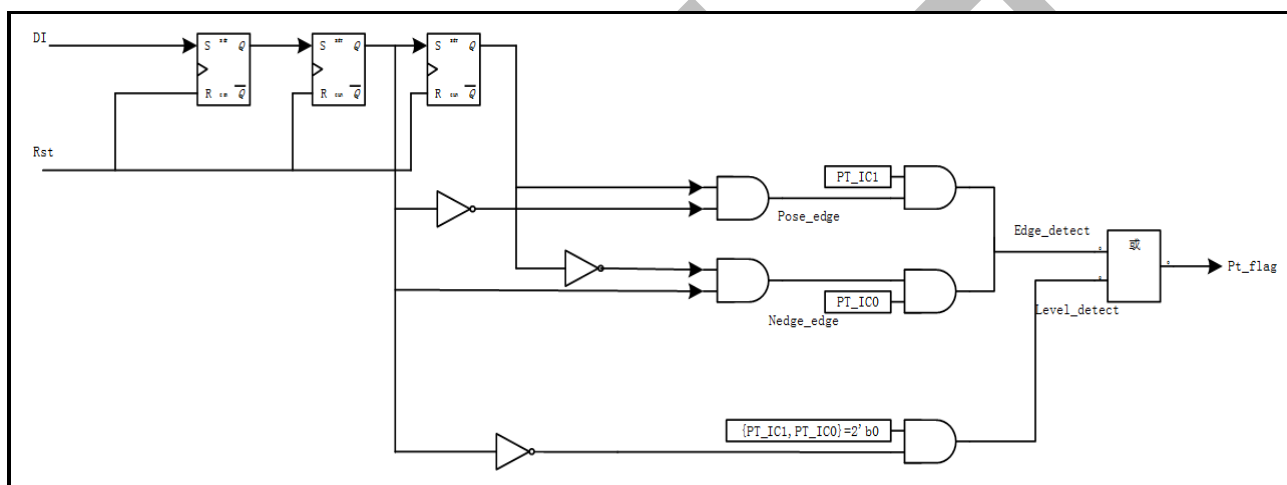


图 5-3 中断触发实现框图

GPIO 中断来自引脚，可以根据寄存器配置来选择中断发生的条件。GPIO 中断可以通过 Px_IC0/1 来选择中断触发条件。寄存器 Px_FLAG 保存每个中断的中断标志。

注：在配置 GPIO 中断时，在 GPIO 中断使能之前，需要清除对应的 FLAG 标志。

5.4 寄存器定义

名字	地址	读写	复位值	描述
P0_DR	0x80	读写	00000000	端口 0 数据寄存器
P0_DM0	0xFF40	读写	00000000	端口 0 模式控制寄存器 0
P0_DM1	0xFF41	读写	00000000	端口 0 模式控制寄存器 1
P0_DM2	0xFF42	读写	00000000	端口 0 模式控制寄存器 2
P0_DM3	0xFF43	读写	00000000	端口 0 模式控制寄存器 3
P0_SOURCE_LEVEL	0xFF44	读写	00000000	端口 0 上拉大电流开关控制寄存器
P0_SINK_LEVEL	0xFF45	读写	00000000	端口 0 下拉大电流开关控制寄存器
P0_FLAG	0xFF10	读写	00000000	端口 0 中断标志位

P0_GE	0xFF11	读写	00000000	端口 0 数字复用使能寄存器
P0_PU	0xFF12	读写	00000000	端口 0 上拉使能寄存器
P0_PD	0xFF13	读写	00000000	端口 0 下拉使能寄存器
P0_IE	0xFF15	读写	00000000	端口 0 中断使能寄存器
P0_IC0	0xFF16	读写	00000000	端口 0 中断控制 0 位
P0_IC1	0xFF17	读写	11111111	端口 0 中断控制 1 位
P1_DR	0x90	读写	00000000	端口 1 数据寄存器
P1_DM0	0xFF48	读写	00000000	端口 1 模式控制寄存器 0
P1_DM1	0xFF49	读写	00100010	端口 1 模式控制寄存器 1
P1_DM2	0xFF4A	读写	00000000	端口 1 模式控制寄存器 2
P1_DM3	0xFF4B	读写	00000000	端口 1 模式控制寄存器 3
P1_FLAG	0xFF18	读写	00000000	端口 1 中断标志位
P1_GE	0xFF19	读写	00001100	端口 1 数字复用使能寄存器
P1_PU	0xFF1A	读写	00001100	端口 1 上拉电阻使能寄存器
P1_PD	0xFF1B	读写	00000000	端口 1 下拉电阻使能寄存器
P1_IE	0xFF1D	读写	00000000	端口 1 中断使能寄存器
P1_IC0	0xFF1E	读写	00000000	端口 1 中断控制 0 位
P1_IC1	0xFF1F	读写	11111111	端口 1 中断控制 1 位
P2_DR	0xA0	读写	00000000	端口 2 数据寄存器
P2_DM0	0xFF50	读写	00000000	端口 2 模式控制寄存器 0
P2_DM1	0xFF51	读写	00000000	端口 2 模式控制寄存器 1
P2_DM2	0xFF52	读写	00000000	端口 2 模式控制寄存器 2
P2_DM3	0xFF53	读写	00000000	端口 2 模式控制寄存器 3
P2_FLAG	0xFF20	读写	00000000	端口 2 中断标志位
P2_GE	0xFF21	读写	00000000	端口 2 数字复用使能寄存器
P2_PU	0xFF22	读写	00000000	端口 2 上拉电阻使能寄存器
P2_PD	0xFF23	读写	00000000	端口 2 下拉电阻使能寄存器
P2_IE	0xFF25	读写	00000000	端口 2 中断使能寄存器
P2_IC0	0xFF26	读写	00000000	端口 2 中断控制 0 位
P2_IC1	0xFF27	读写	11111111	端口 2 中断控制 1 位
P3_DR	0xB0	读写	00000000	端口 3 数据寄存器
P3_DM0	0xFF58	读写	00000000	端口 3 模式控制寄存器 0
P3_FLAG	0xFF28	读写	00000000	端口 3 中断标志位
P3_GE	0xFF29	读写	00000000	端口 3 数字复用使能寄存器

P3_PU	0xFF2A	读写	00000000	端口 3 上拉电阻使能寄存器
P3_PD	0xFF2B	读写	00000000	端口 3 下拉电阻使能寄存器

5.4.1 P0_DR (addr: 0x80) 端口 0 数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	P0_DR							
Reset	0							
Type	R/W							

Bit	Name	Function
7:0	P0_DR	端口 0 的数据寄存器, 写该寄存器会更新端口输出, 读该寄存器得到端口输出值, 读该寄存器详细说明见 5.2。

5.4.2 P0_GE (addr: 0xFF11) 端口 0 数字复用使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	GE0.7	GE0.6	GE0.5	GE0.4	GE0.3	GE0.2	GE0.1	GE0.0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	P0_GE	端口 0 的外设复用功能使能。 0 关闭复用使能, 输出由 Px.DR 决定, 如果输入则输入电平寄存在 Px.DR 中 1 打开复用使能, 输出由 GDO 决定, 如果输入则输入电平到 PINMUX 供需要的模块选择使用

5.4.3 P0_IE (addr: 0xFF15) 端口 0 中断使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	P0_IE							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_IE	中断使能: 0 关闭端口中断 1 打开端口中断

5.4.4 P0_IC0/P0_IC1 (addr: 0xFF16, 0xFF17) 端口 0 中断控制位

Bit	7	6	5	4	3	2	1	0
Name	P0_IC0							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Name	P0_IC1							
Reset	1	1	1	1	1	1	1	1

Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
------	-----	-----	-----	-----	-----	-----	-----	-----

Bit	Name	Function
7:0	P0_IC1	[P0_IC1: P0_IC0] 中断触发模式控制: 00 低电平中断 01 上升沿中断 10 下降沿中断 11 双边沿中断
7:0	P0_IC0	

5.4.5 P0_FLAG (addr: 0xFF10) 端口 0 中断标志位

Bit	7	6	5	4	3	2	1	0
Name	P0_FLAG							
Reset	0x00							
Type	R/W							
Bit	Name	Function						
7:0	P0_FLAG	中断标志： 0 没有中断发生 1 有中断发生 写 1 清除该中断标志						

5.4.6 P1_DR (addr: 0x90) 端口 1 数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	P1_DR							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_DR	端口 1 的数据寄存器，写该寄存器会更新端口输出，读该寄存器得到端口输出值，读该寄存器详细说明见 5.2。

5.4.7 P1_GE (addr: 0xFF19) 端口 1 数字复用使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	GE1.7	GE1.6	GE1.5	GE1.4	GE1.3	GE1.2	GE1.1	GE1.0
Reset	0	0	0	0	1	1	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	P1_GE	端口 1 的外设复用功能使能。 0 关闭复用使能，输出由 Px_DR 决定，如果输入则输入电平寄存在 Px_DR 中 1 打开复用使能，输出由 GDO 决定，如果输入则输入电平到 PINMUX 供需要的模块选择使用

5.4.8 P1_IE (addr: 0xFF1D) 端口 1 中断使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	P1_IE							
Reset	0x00							

Type	R/W
------	-----

Bit	Name	Function
7:0	P1_IE	中断使能: 0 关闭端口中断 1 打开端口中断

5.4.9 P1_IC0/P1_IC1 (addr: 0xFF1E, 0xFF1F) 端口 1 中断控制 0 位

Bit	7	6	5	4	3	2	1	0
Name	P1_IC0							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Name	P1_IC1							
Reset	1	1	1	1	1	1	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	P1_IC1	[P0_IC1: P0_IC0] 中断触发模式控制: 00 低电平中断 01 上升沿中断 10 下降沿中断 11 双边沿中断
7:0	P1_IC0	

5.4.10 P1_FLAG (addr: 0xFF18) 端口 1 中断标志位

Bit	7	6	5	4	3	2	1	0
Name	P1_FLAG							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_FLAG	中断标志: 0 没有中断发生 1 有中断发生 写 1 清除该中断标志

5.4.11 P2_GE (addr: 0xFF21) 端口 2 数字复用使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	GE2. 7	GE2. 6	GE2. 5	GE2. 4	—	—	GE2. 1	GE2. 0
Reset	0	0	0	0	—	—	0	0
Type	R/W	R/W	R/W	R/W	—	—	R/W	R/W

Bit	Name	Function
7:0	P2_GE	端口 0 的外设复用功能使能。

		0 关闭复用使能，输出由 Px.DR 决定，如果输入则输入电平寄存在 Px.DR 中
		1 打开复用使能，输出由 GD0 决定，如果输入则输入电平到 PINMUX 供需要的模块选择使用

5.4.12 P2_DR (addr: 0xA0) 端口 2 数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	P2_DR							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_DR	端口 2 的数据寄存器，写该寄存器会更新端口输出，读该寄存器得到端口输出值，读该寄存器详细说明见 5.2。

5.4.13 P2_IE (addr: 0xFF25) 端口 2 中断使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	P2_IE							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P2_IE	中断使能： 0 关闭端口中断 1 打开端口中断

5.4.14 P2_IC0/P2_IC1 (addr: 0xFF26, 0xFF27) 端口 2 中断控制位

Bit	7	6	5	4	3	2	1	0
Name	P2_IC0							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Name	P2_IC1							
Reset	1	1	1	1	1	1	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	P2_IC1	[P2_IC1: P2_IC0] 中断触发模式控制： 00 低电平中断 01 上升沿中断 10 下降沿中断 11 双边沿中断
7:0	P2_IC0	

5.4.15 P2_FLAG (addr: 0xFF20) 端口 2 中断标志位

Bit	7	6	5	4	3	2	1	0
Name	P2_FLAG							

Reset	0x00
Type	R/W

Bit	Name	Function
7:0	P2_FLAG	中断标志: 0 没有中断发生 1 有中断发生 写 1 清除该中断标志

5.4.16 P3_DR (addr: 0xB0) 端口 3 数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	P3_DR							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P3_DR	端口 3 的数据寄存器, 写该寄存器会更新端口输出, 读该寄存器得到端口输出值, 读该寄存器详细说明见 5.2。

5.4.17 P3_IE (addr: 0xFF2D) 端口 3 中断使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	-						P3_IE	
Reset	-						0x00	
Type	-						R/W	

Bit	Name	Function
7:2	-	-
1:0	P3_IE	中断使能: 0 关闭端口中断 1 打开端口中断

5.4.18 P3_IC0/P3_IC1 (addr: 0xFF2E, 0xFF2F) 端口 3 中断控制位

Bit	7	6	5	4	3	2	1	0
Name	-						P3_IC0	
Reset	-						0x00	
Type	-						R/W	

Bit	7	6	5	4	3	2	1	0
Name	-						P3_IC1	
Reset	-						0x03	
Type	-						R/W	

Bit	Name	Function
7:2	-	-

1:0	P3_IC1	[P3_IC1: P3_IC0] 中断触发模式控制: 00 低电平中断 01 上升沿中断 10 下降沿中断 11 双边沿中断
1:0	P3_IC0	

5.4.19 P3_FLAG (addr: 0xFF28) 端口 3 中断标志位

Bit	7	6	5	4	3	2	1	0
Name	-						P3_FLAG	
Reset	-						0x00	
Type	-						R/W	

Bit	Name	Function
7:2	-	-
1:0	P3_FLAG	中断标志: 0 没有中断发生 1 有中断发生 写 1 清除该中断标志

5.4.20 P3_GE (addr: 0xFF21) 端口 2 数字复用使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	GE2.7	GE2.6	GE2.5	GE2.4	GE2.3	GE2.2	GE2.1	GE2.0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	P2_GE	端口 0 的外设复用功能使能。 0 关闭复用使能, 输出由 Px.DR 决定, 如果输入则输入电平寄存在 Px.DR 中 1 打开复用使能, 输出由 GDO 决定, 如果输入则输入电平到 PINMUX 供需要的模块选择使用

5.4.21 Px_PU 端口上拉电阻使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	Px_PU							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	Px_PU	上拉使能: P1 高 7 位有效、P0 和 P2 为 8 位有效, P3 低 2 位有效 0 关闭上拉 1 打开上拉

5.4.22 Px_PD 端口下拉电阻使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	Px_PD							
Reset	0x00							

Type	R/W
------	-----

Bit	Name	Function
7:0	Px_PD	下拉使能: P1 高 7 位有效、P0 和 P2 为 8 位有效, P3 低 2 位有效 0 关闭下拉 1 打开下拉

5.4.23 Px_DM0 端口模式控制寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	—	Px1_DM			—	Px0_DM		
Reset	—				—			
Type	—	R/W			—	R/W		

Bit	Name	Function
7	—	—
6:4	Px1_DM[2:0]	Px1 模式控制寄存器
3	—	—
2:0	Px0_DM[2:0]	Px0 模式控制寄存器

5.4.24 Px_DM1 端口模式控制寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	—	Px3_DM			—	Px2_DM		
Reset	—				—			
Type	—	R/W			—	R/W		

Bit	Name	Function
7	—	—
6:4	Px3_DM[2:0]	Px1 模式控制寄存器
3	—	—
2:0	Px2_DM[2:0]	Px0 模式控制寄存器

5.4.25 Px_DM2 端口模式控制寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	—	Px5_DM			—	Px4_DM		
Reset	—				—			
Type	—	R/W			—	R/W		

Bit	Name	Function
7	—	—
6:4	Px5_DM[2:0]	Px1 模式控制寄存器
3	—	—
2:0	Px4_DM[2:0]	Px0 模式控制寄存器

5.4.26 Px_DM3 端口模式控制寄存器 0

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	–	Px7_DM	–	Px6_DM
Reset	–		–	
Type	–	R/W	–	R/W

Bit	Name	Function
7	–	–
6:4	Px7_DM[2:0]	Px1 模式控制寄存器
3	–	–
2:0	Px6_DM[2:0]	Px0 模式控制寄存器

5.4.27 P0_SOURCE_LEVEL (addr: 0xFF44) 端口 0 上拉大电流开关控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	P0_SOURCE_LEVEL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_SOURCE_LEVEL	P0 上拉大电流开关 0 关闭上拉大电流, 电流 10mA 1 开启上拉大电流, 电流 20mA

5.4.28 P0_SINK_LEVEL (addr: 0xFF45) 端口 0 下拉大电流开关控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	P0_SINK_LEVEL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_SINK_LEVEL	P0 下拉大电流开关 0 关闭下拉大电流, 电流 15mA 1 开启下拉大电流, 电流 125mA

6 CPU 功能简介

RC6T8X634 系列芯片在传统 8051 体系结构上做了一些优化，采用了单周期 8051 CPU，相比传统的 8051 在性能上面有了很大提升。芯片内部的 ALU 配合内部的 ACC (0xE0)，B (0xF0)，PSW (0xD0) 寄存器可以实现各种 8 位运算操作。

CPU 具备以下特性：

- 1T 8051 CPU
- 兼容 8051 指令集
- 双 DPTR，可快速移动数据

ALU 可以进行典型操作如下：

- 基本算术运算：加法、减法、乘法、除法
- 其他算术运算：自加、自减、BCD 调整、比较
- 逻辑运算：与、或、异或、取反、移位
- 布尔比特运算：置位、清零、取反、按位判断跳转、进位操作

还有一些 8051 核内部使用的寄存器可以通过 SFR 地址访问，包括 SP、DPL0/1、DPH0/1、DPS 等。

6.1 内核 SFR 寄存器

名字	地址	读写	复位值	描述
ACC	0xE0	读写	00000000	累加寄存器
B	0xF0	读写	00000000	B 寄存器
PSW	0xD0	读写	00000000	程序状态字寄存器
IE	0xA8	读写	00000000	系统中断使能寄存器
SP	0x81	读写	00000111	堆栈指针，指向 IDATA 空间
DPL0	0x82	读写	00000000	DPTR0 寄存器的低 8bit
DPH0	0x83	读写	00000000	DPTR0 寄存器的高 8bit
DPL1	0x84	读写	00000000	DPTR1 寄存器的低 8bit
DPH1	0x85	读写	00000000	DPTR1 寄存器的高 8bit
DPS	0x86	读写	00000000	DPTR0/DPTR1 选择寄存器

6.1.1 ACC 寄存器 (addr: 0xE0) 累加寄存器

Bit	7	6	5	4	3	2	1	0
Name	ACC. 7	ACC. 6	ACC. 5	ACC. 4	ACC. 3	ACC. 2	ACC. 1	ACC. 0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	ACC	累加寄存器。

注：累加器 ACC 是一个常用的专用寄存器，指令系统中采用 A 作为累加器的助记符，常用于存放算术或逻辑运算的操作数及运算结果。

6.1.2 B 寄存器 (addr: 0xF0) B 寄存器

Bit	7	6	5	4	3	2	1	0
Name	B. 7	B. 6	B. 5	B. 4	B. 3	B. 2	B. 1	B. 0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	B	乘法运算和除法运算的时候使用，其他情况用作普通寄存器。

注：B 寄存器在乘除法运算中需要和 ACC 配合使用，MUL AB 指令把 ACC 和 B 中 8 位无符号数相乘，所得 16 位乘积的低字节存放在 A 中，高字节存放在 B 中。DIV AB 指令用 B 除以 A，整数商存放在 A 中，余数存放在 B 中。寄存器 B 还可以做通用暂存寄存器。

6.1.3 PSW 寄存器 (addr: 0xD0) 程序状态字寄存器

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	P
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	CY	进位标志
6	AC	辅助进位标志
5	F0	通用标志 0
4:3	RS[1:0]	寄存器组选择： 00 寄存器组 0，数据地址 0x00-0x07 01 寄存器组 1，数据地址 0x08-0x0F 10 寄存器组 2，数据地址 0x10-0x17 11 寄存器组 3，数据地址 0x18-0x1F
2	OV	溢出标志
1	F1	通用标志 1
0	P	奇偶校验标志

注：PSW 是 CPU 的状态寄存器，在 CPU 做算术运算或逻辑运算时，对应的 PSW 状态位会发生变化。

6.1.4 IE 寄存器 (addr: 0xA8) 系统中断使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	IE_EA	-	-	-	-	-	-	-
Reset	0	-	-	-	-	-	-	-
Type	R/W	-	-	-	-	-	-	-

Bit	Name	Function
7	IE_EA	CPU 中断允许位总开关，1 表示使能中断，0 表示不使能中断。
6:0	N/A	保留位，读 0

6.1.5 SP 寄存器 (addr: 0x81) 堆栈指针, 指向 IDATA 空间

Bit	7	6	5	4	3	2	1	0
Name	SP							
Reset	0	0	0	0	0	1	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	SP	堆栈指针, 指向 IDATA 区域。

注: 堆栈指针 SP 是一个 8 位专用寄存器。它指示堆栈顶部在内部 RAM 块中的位置。系统复位后, SP 初始化为 07H, 使得堆栈事实上从 08H 开始。

6.1.6 DPL0 寄存器 (addr: 0x82) DPTR0 寄存器的低 8bit

Bit	7	6	5	4	3	2	1	0
Name	DPTR0[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	DPTR0[7:0]	用于 DPTR0[7:0]。

6.1.7 DPH0 寄存器 (addr: 0x83) DPTR0 寄存器的高 8bit

Bit	7	6	5	4	3	2	1	0
Name	DPTR0[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	DPTR0[15:8]	用于 DPTR0[15:8]。

6.1.8 DPL1 寄存器 (addr: 0x84) DPTR1 寄存器的低 8bit

Bit	7	6	5	4	3	2	1	0
Name	DPTR1[7:0]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	DPTR1[7:0]	用于 DPTR1[7:0]。

6.1.9 DPH1 寄存器 (addr: 0x85) DPTR1 寄存器的高 8bit

Bit	7	6	5	4	3	2	1	0
Name	DPTR1[15:8]							
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:0	DPTR1[15:8]	用于 DPTR1[15:8]。

注：DPTR0/1 是两个 16 位专用寄存器，它们的高字节寄存器用 DP0H/DP1H 表示，低位字节寄存器用 DP0L/DP1L 表示，通过 DPS 可选择使用 DPTR0/DPTR1。每个 DPTR 可以作为一个 16 位寄存器来处理，也可以作为两个独立的 8 位寄存器 DP0H/DP1H 和 DP0L/DP1L 来处理。

6.1.10 DPS 寄存器（addr：0x86）DPTR0/DPTR1 选择寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	SEL
Reset	—	—	—	—	—	—	—	0
Type	—	—	—	—	—	—	—	R/W

Bit	Name	Function
7:1	—	—
0	SEL	SEL=0 时系统使用 DPTR0 寄存器； SEL=1 时系统使用 DPTR1 寄存器。

7 中断控制系统

RC6T8X634 系列芯片有一个增强型的中断控制系统，共有 19 个中断源，中断延时为 5~8 个机器周期，每个中断源有两级中断优先级。每个中断源都有独立的中断向量、优先级设置位、中断使能位、中断标志位。CPU 在响应中断后，进入该中断对应的中断服务程序，接到 RETI 指令后将返回中断前状态。如果同时有多个有效中断产生中断请求，CPU 将根据设置的中断优先级依次响应；如果优先级相同，则根据中断的自然优先级（中断号越小，优先级越高）依次响应。

7.1 中断向量表

中断控制器支持 19 个中断源。当中断发生且中断使能之后，跳转到对应向量地址去执行 LCALL 指令来进入中断服务程序。

表 7-1 中断向量表

中断号	中断源	说明	睡眠唤醒	深睡唤醒	中断地址
0	LVD	低压检测	是	否	0003H
1	P0	GPIO0 脚中断	是	是	000BH
2	P1	GPIO1 脚中断	是	是	0013H
3	P2	GPIO2 脚中断	是	是	001BH
4	P3	GPIO3 脚中断	是	是	0023H
5	VC3	VC3 时钟有效中断	是	否	002BH
6	TIMER0	定时器 0 中断	是	否	0033H
7	TIMER1	定时器 1 中断	是	否	003BH
8	TIMER2	定时器 2 中断	是	否	0043H
9	TIMER3	定时器 3 中断	是	否	004BH
10	TIMER4	定时器 4 中断	是	否	0053H
11	TIMER5	定时器 5 中断	是	否	005BH
12	ADC	ADC 转换完成中断	是	否	0063H
13	PWM	Timer4 的 PWM 中断	是	否	006BH
14	CSD	CSD 中断	是	否	0073H
15	I2C	I2C 状态中断	是	否	007BH
16	UART0	UART0 状态中断	是	否	0083H
17	UART1	UART1 状态中断	是	否	008BH
18	WDT	看门狗中断	是	是	0093H

7.2 中断优先级

每个中断有唯一的中断号。中断号越小，中断的优先级越高。同时，每个中断源都有一个优先级配置位，用户可以根据需要配置该位以提高对应中断的优先级。

7.3 寄存器定义

名字	地址	读写	复位值	描述
IP	0xB8	读写	00000000	中断优先级配置寄存器
IP1	0xB9	读写	00000000	中断优先级配置寄存器 1
IP2	0xBA	读写	00000000	中断优先级配置寄存器 2

7.3.1 IP (addr: 0xB8) 中断优先级配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	T1PRI	TOPRI	VC3PRI	P3PRI	P2PRI	P1PRI	POPRI	LVDPRI
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	T1PRI	0 Timer1 中断为低优先级 1 Timer1 中断为高优先级
6	TOPRI	0 Timer0 中断为低优先级 1 Timer0 中断为高优先级
5	VC3PRI	0 VC3 中断为低优先级 1 VC3 中断为高优先级
4	P3PRI	0 GPIO 3 中断为低优先级 1 GPIO 3 中断为高优先级
3	P2PRI	0 GPIO 2 中断为低优先级 1 GPIO 2 中断为高优先级
2	P1PRI	0 GPIO 1 中断为低优先级 1 GPIO 1 中断为高优先级
1	POPRI	0 GPIO 0 中断为低优先级 1 GPIO 0 中断为高优先级
0	LVDPRI	0 LVD 中断为低优先级 1 LVD 中断为高优先级

7.3.2 IP1 (addr: 0xB9) 中断优先级配置寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	I2CPRI	CSDPRI	PWMPRI	ADCPRI	T5PRI	T4PRI	T3PRI	T2PRI
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	I2CPRI	0 I2C 中断为低优先级 1 I2C 中断为高优先级
6	CSDPRI	0 CSD 中断为低优先级 1 CSD 中断为高优先级
5	PWMPRI	0 PWM 中断为低优先级 1 PWM 中断为高优先级
4	ADCPRI	0 ADC 中断为低优先级 1 ADC 中断为高优先级
3	T5PRI	0 TIMER5 中断为低优先级 1 TIMER5 中断为高优先级
2	T4PRI	0 TIMER4 中断为低优先级 1 TIMER4 中断为高优先级
1	T3PRI	0 TIMER3 中断为低优先级 1 TIMER3 中断为高优先级
0	T2PRI	0 TIMER2 中断为低优先级 1 TIMER2 中断为高优先级

7.3.3 IP2 (addr: 0xBA) 中断优先级配置寄存器 2

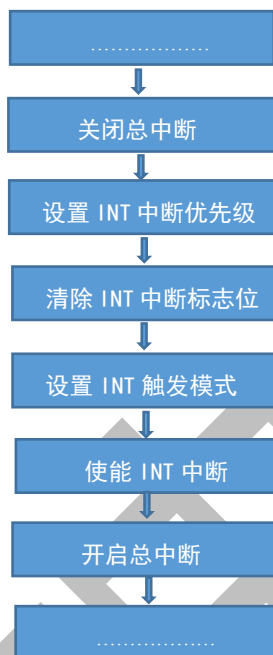
Bit	7	6	5	4	3	2	1	0
Name	-					WDTPRI	UART1PRI	UART0PRI
Reset	-					0	0	0
Type	-					R/W	R/W	R/W

Bit	Name	Function
7:3	-	-
2	WDTPRI	0 WDT 中断为低优先级 1 WDT 中断为高优先级
1	UART1	0 UART1 中断为低优先级 1 UART1 中断为高优先级
0	UART0	0 UART0 中断为低优先级 1 UART0 中断为高优先级

7.4 外部中断

RC6T8X634 在标准 8501 基础上增加了可选择任意输入口作为中断触发源的功能，外部中断可单独设置上升沿下降沿、双边沿、低电平中断。每个外部中断都可以用于休眠唤醒，相关寄存器说明见 5.6 章节。

7.5 外部中断配置流程



外部中断配置流程

8 存储器系统

8.1 存储器系统概述

RC6T8X634 系列芯片提供了 3 种存储器：SFR，内部数据存储器，程序存储器。

程序存储器只能读不能写，大小为 16K 字节。内部数据存储器大小为 1K 字节。SFR 为内部特殊功能寄存器。

8.2 程序存储器映射

本芯片的程序指针为 16 位，最大寻址空间可达 64K 字节，实际只实现了 16K 字节的程序存储空间。

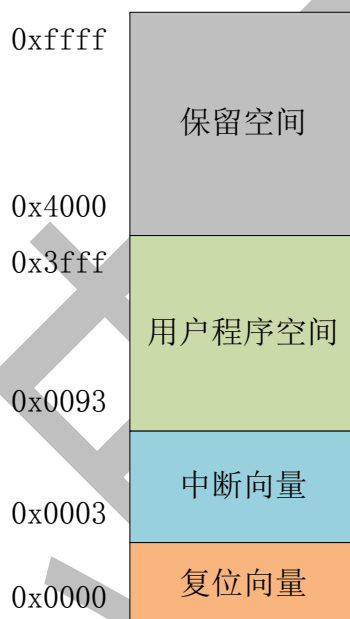


图 8-1 程序存储空间

复位后，MCU 从 0000H 开始执行。中断向量表从 0003H 开始，当发生中断且中断使能后，PC 会跳转到对应的中断向量位置去执行。

8.3 数据存储器映射

数据存储器分内部数据存储器和外部数据存储器，内部数据存储器空间为 256 字节，其中低 128 字节可以直接访问（通过地址 0x00~0x7f），高 128 字节和 SFR 共用一个地址空间（通过地址 0x80~0xff），直接寻址方式可以访问到 SFR 空间，通过间接寻址方式可以访问内部数据存储器的高 128 字节。低 128 字节数据存储空间可以划分为如下图所示的不同空间。

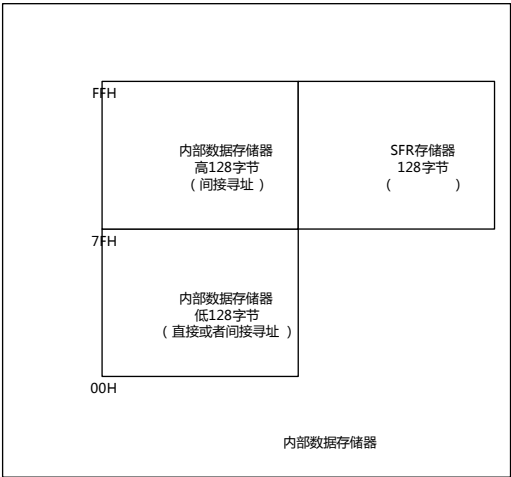


图 8-2 数据存储器

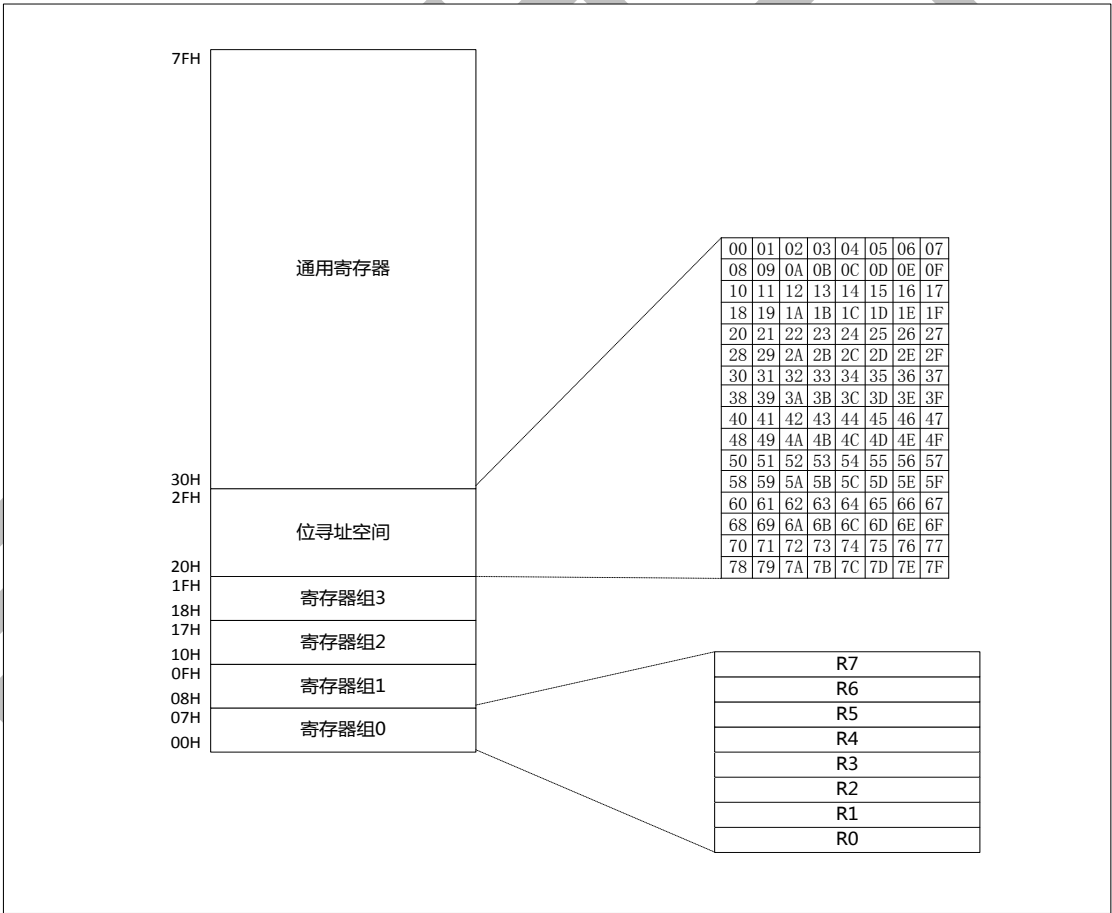


图 8-3 内部低 128 字节数据空间分配

此外，片上还集成了 768 字节的片上 RAM 映射在外部数据存储空间，地址范围为 0x0000~0x02ff。

8.4 MTP 与 SRAM 结构

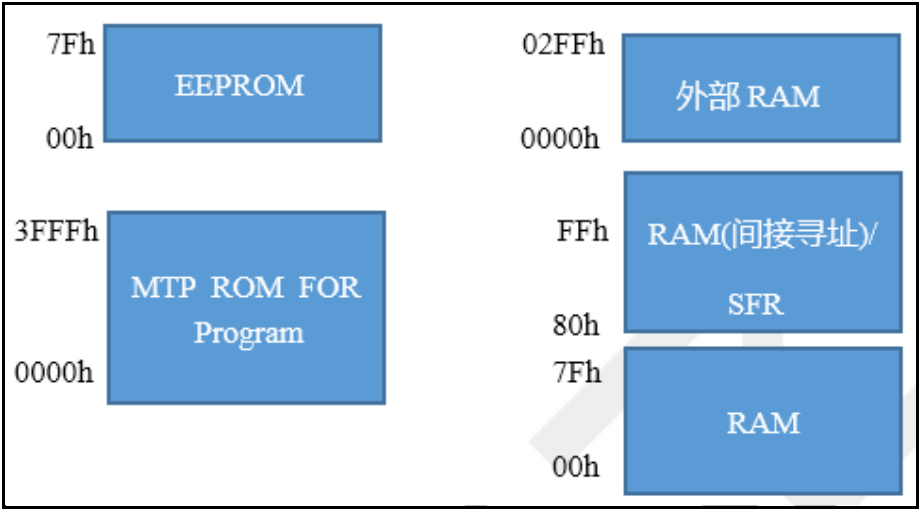


图 8-4 MTP 与 SRAM 结构图

8.5 寄存器映射

表 8-1 寄存器映射表

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	SLPTIM_CR	SLPTIM_SR	SLPTIM_CLR	SLPTIM_WDT	SLPTIM_CNL			SLPTIM_PRDH
F0h	B							
E8h	ADC_CR0	ADC_CR1	ADC_CR2	ADC_CHSEL	ADC_CON	ADC_DLY	ADC_RESL	ADC_RES
E0h	ACC		-	-	-	-	ADC_COMPL	ADC_COMPH
D8h	T5CON							
D0h	PSW	DUTY0	DUTY3L	DUTY3H	DUTY4L	DUTY4H	PWM_CFG	TPCON
C8h	T3CON	T4_PWM_NUM	RCAP2L	RCAP2H	TL2	TH2	TL5	TH5
C0h	T4CON	RCAP4L	RCAP4H	TL4	TH4	T2CON	RCAP5L	RCAP5H
B8h	IP	IP1	IP2		RCAP3L	RCAP3H	TL3	TH3
B0h	P3_DR	-	-	-	-	-	-	-
A8h	IE	-	-	-	-	-	-	-
A0h	P2_DR	I2C_ADDR	I2C_CR	I2C_STAT	I2C_DR	I2C_MCR	-	-
98h	SCON	SBUF	SCON1	SBUF1	CHIP_ID0	CHIP_ID1	ITCH_TRIM	ITCH_CR
90h	P1_DR	SCR_CFG	SCR_SLEEP	MBIST_CFG	CLK_CR	PCLK_CR	PCLK_DIV12	PCLK_DIV3
88h	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	TIRQ
80h	P0_DR	SP	DPL0	DPL1	DPL1	DPH1	DPS	PCON
	位寻址	不可位寻址						

芯片部分寄存器放在外部数据存储空间，该部分地址空间大小为 256 字节，地址范围为 0xFF00~0xFFFF。

	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
FFF8H	CSD_RX0_DRH	CSD_RX0_DRL	CSD_RX1_DRH	CSD_RX1_D RL	CSD_RX2_D RH	CSD_RX2_ DRL	CSD_RX3_DRH	CSD_RX3_DRL
FFF0H	CSD_RX0_CR	CSD_RX0_TH						
FFE8H	CSD_WIN	CSD_VREF_CR	CSD_WPF_CR	CSD_CR2	ITCH_TRIM_ H	ITCH_TRI M1	ITCH_TRIM2	ITCH_TRIM3
FFE0H	CSD_CR	CSD_SIN_CKS EL	CSD_CKSEL	CSD_STAT	CSD_PRSH	CSD_PRSL	CSD_POLYH	CSD_POLYL
FFD8H	CSD_RX0_ICO M_H	CSD_RX0_ICO M_L	CSD_RX1_ICO M_H	CSD_RX1_IC OM_L	CSD_RX2_IC OM_H	CSD_RX2_ ICOM_L	CSD_RX3_ICO M_H	CSD_RX3_ICO M_L
FFD0H	CSD_RX0_IMS _H	CSD_RX0_IMS _L	CSD_RX1_IMS _H	CSD_RX1_IM S_L	CSD_RX2_IM S_H	CSD_RX2_ IMS_L	CSD_RX3_IMS H	CSD_RX3_IMS _L
FFC8H	-	-	-	-	-	-	-	-
FFC0H								
FFB8H	-	-	-	-	-	-	-	-
FFB0H	CHKSUM_CR	CHKSUMH	CHKSUML	-	-	-	-	-
FFA8H								
FFA0H								
FF98H								
FF90H								
FF88H	IMO_CR	IMO_TRIM	ILO_TRIM	ILO_TEST	IMO_TRIMH	-	MBIST_KEY	PT_DIG_TEST
FF80H	BG_CR	BG_VTRIM	BG_ITRIM	BG_TCTRIM	BG_TEST	BORLVD_ CR	BORLVD_STAT	ANA_TEST
FF78H	DO_SEL8	DO_SEL9	DO_SEL10	DO_SEL11	DO_SEL12	-	-	-
FF70H	DO_SEL0	DO_SEL1	DO_SEL2	DO_SEL3	DO_SEL4	DO_SEL5	DO_SEL6	DO_SEL7
FF68H	DI_SEL8							PMUX_AFR
FF60H	DI_SEL0	DI_SEL1	DI_SEL2	DI_SEL3	DI_SEL4	DI_SEL 5	DI_SEL6	DI_SEL7
FF58H	P3_DM0				-	-	-	-
FF50H	P2_DM0	P2_DM1	P2_DM2	P2_DM3	-	-	-	-
FF48H	P1_DM0	P1_DM1	P1_DM2	P1_DM3	-	-	-	-
FF40H	P0_DM0	P0_DM1	P0_DM2	P0_DM3	P0_SOURCE_ LEVEL	P0_SINK_ LEVEL		

FF38H								
FF30H								
FF28H	P3_FLAG	P3_GE	P3_PU	P3_PD	-	P3_IE	P3_IC0	P3_IC1
FF20H	P2_FLAG	P2_GE	P2_PU	P2_PD	-	P2_IE	P2_IC0	P2_IC1
FF18H	P1_FLAG	P1_GE	P1_PU	P1_PD	-	P1_IE	P1_IC0	P1_IC1
FF10H	P0_FLAG	P0_GE	P0_PU	P0_PD	-	P0_IE	P0_IC0	P0_IC1
FF08H			-	-	-	-	-	-
FF00H	MTP_CR	MTP_CFG	MTP_KEY	MTP_ADL	MTP_ADH	MTP_PBU FL	MTP_PBUFH	MTP_DR

8.6 MTP 控制器

RC6T8X634 系列芯片内部实现了一个大小为 16KB 的 MTP 存储器，编程次数可达 1000 次。其中包含了 16K 字节 (8K*16bit) 的主存储区域，用来存放程序；64 字节 (32*16bit) 的 Information 区域，用来存放 FT 测试时的修调数据；128 字节 (128*8bit) 的 EEPROM 区域，用于用户存储数据。MTP 控制器用来控制 8051 访问的 MTP 存储器的读时序，同时编程器通过编程接口对 MTP 存储器进行编程。

8.6.1 MTP 控制器寄存器定义

名字	地址	读写	复位值	描述
MTP_CR	0xFF00	读写	11000000	MTP 控制寄存器
MTP_CFG	0xFF01	读写	00000011	MTP 配置寄存器
MTP_KEY	0xFF02	读写	00000000	MTP key 寄存器
MTP_ADL	0xFF03	读写	00000000	MTP 编程地址低 8 位
MTP_ADH	0xFF04	读写	00000000	MTP 编程地址高 6 位
MTP_PBUFL	0xFF05	读写	00000000	MTP 编程缓冲地址低 8 位
MTP_PBUFH	0xFF06	读写	00000000	MTP 编程缓冲地址高 2 位
MTP_DR	0xFF07	只读	xxxxxxxx	MTP 读数据寄存器

8.6.1.1 MTP_CR (addr: 0xFF00) MTP 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	ISAVB	STATICEN	WRSZ[1:0]		CKEN	FWEEP	IFREN	BUSY
Reset	1	1	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	ISAVB	Option setting for Read operation : 0 ISAVB = L to select the low power or power saving mode. 1 ISAVB = H to select the high speed mode. (默认值)

6	STATICEN	0 STATICEN = L, NON-STATIC mode (INDS <10uA) 1 STATICEN = H, STATIC mode (IDS <500uA). (默认值)
5:4	WRSZ[1:0]	MTP 存储器编程数据大小, 当选择 EEPROM 区域时, 单位为字节, MTP 用户区域或信息区域单位为半字 (2 个字节) 00 = 2 01 = 1 10 = 64 11 = 128
3	CKEN	MTP 时钟使能: 0 = 关闭时钟 1 = 使能时钟
2	FWEEP	0 = 选择 MTP 用户区域 1 = 选择 EEPROM 区域, 注意 FWEEP 和 IFREN 不能同时为 1
1	IFREN	0 = 选择 MTP 用户区域 1 = 选择 MTP 信息区域, 注意 FWEEP 和 IFREN 不能同时为 1
0	BUSY	读模式下 BUSY 的值表示: 0 = MTP 编程完成 1 = MTP 编程没有完成 写 1 开始编程操作。

注: ISAVB, STATICEN 的配置说明参照如下, 可配合选择三种模式, 正常使用默认为推荐的 High speed 模式。当读操作频率大于 4 MHz 时应配置 ISAVB 和 STATICEN 为 H, 选择 High speed 模式。当读操作频率小于等于 500KHz 时应配置 ISAVB 和 STATICEN 为 L, 选择 Power saving mode。当读操作频率介于 4MHz 和 500KHz 之间时可配置 ISAVB 为 L 和 STATICEN 为 H, 选择 Low power mode。修改配置前需将 READ 信号拉低, 并保持低电平至配置完成后 100ns。

8.6.1.2 MTP_CFG (addr: 0xFF01) MTP 配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	FWSEL	CLEAN	SRL	MRGN	SAVPWR1	SAVPWR0	RDCYC[1:0]	
Reset	0	0	0	0	0	0	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	FWSEL	MTP 控制信号选择 0 使用默认的 MTP CLEN, SRL, MRGN, ISAVB, STATICEN 信号 (默认值) 1 使用寄存器定义的 MTP CLEN, SRL, MRGN, ISAVB, STATICEN 信号
6	CLEAN	MTP 存储单元电流测试模式 0 关闭 (默认值) 1 打开
5	SRL	MTP SRL 存储单元 cell 选择 0 默认值 cell (默认值) 1 shadow cell
4	MRGN	MTP MRGN 控制 0 无效 (默认值) 1 有效 (MRGN READ/MRGN INTHV WRITE operation for data retention test when select the EEPROM block)
3	SAVPWR1	SLEEP 模式门控 CS 信号

		0 SLEEP 模式时 CS 信号门控关闭 (CS 有效) 1 SLEEP 模式时 CS 信号门控打开 (CS 无效) 睡眠模式需要配置为 1, 睡眠模式下 MTP 的片选关闭, 可以节省功耗。
2	SAVPWRO	SLEEP 模式门控 READ 信号 0 SLEEP 模式时 READ 信号门控关闭 (READ 有效) 1 SLEEP 模式时 READ 信号门控打开 (READ 无效) 睡眠模式需要配置为 1, 睡眠模式下 MTP 的读关闭, 可以节省功耗。
1: 0	RDCYC[1:0]	MTP 访问周期 00 1 个周期 01 2 个周期 10 5 个周期 11 6 个周期 注意: 当芯片电压低于 4.5V 时候, 要配置 RDCYC 为 11 (6 个周期)。芯片 MTP 内部实现了 2 个字节的缓冲, VDD 电压大于 4.5V 时使用 01 (2 个周期) 配置即可, 这样可以保证性能和功耗的平衡。 读 EEPROM 时, 所有电压下都需要配置 RDCYC 为 11 (6 个周期)。

Read Operation: High speed mode:

Parameter	Symbol	Condition	Min	Max	Unit
READ Cycle Time@ MTP/Information	Trc	$4.5\text{V} \leq \text{VDD} \leq 5.5\text{V}$	60	-	ns
		$1.8\text{V} \leq \text{VDD} \leq 4.5\text{V}$	300	-	ns
READ Cycle Time@ EEPROM		$1.8\text{V} \leq \text{VDD} \leq 5.5\text{V}$	300	-	ns
READ Access Time@ MTP/Information	Trac	$4.5\text{V} \leq \text{VDD} \leq 5.5\text{V}$	-	60	ns
		$1.8\text{V} \leq \text{VDD} \leq 4.5\text{V}$	-	300	ns
READ Access Time@ EEPROM		$1.8\text{V} \leq \text{VDD} \leq 5.5\text{V}$	-	300	ns

Read Operation: Low power mode:

Parameter	Symbol	Condition	Min	Max	Unit
READ Cycle Time@ MTP/Information	Trc	$4.5\text{V} \leq \text{VDD} \leq 5.5\text{V}$	500	-	ns
		$1.8\text{V} \leq \text{VDD} \leq 4.5\text{V}$	1	-	us
READ Access Time@ MTP/Information	Trac	$4.5\text{V} \leq \text{VDD} \leq 5.5\text{V}$	-	500	ns
		$1.8\text{V} \leq \text{VDD} \leq 4.5\text{V}$	-	1	us

Read Operation: Power saving mode:

Parameter	Symbol	Condition	Min	Max	Unit
READ Cycle Time	Trc	$1.8\text{V} \leq \text{VDD} \leq 5.5\text{V}$	2	-	us
READ Access Time	Trac	$1.8\text{V} \leq \text{VDD} \leq 5.5\text{V}$	-	2	us

8.6.1.3 MTP_KEY (addr: 0xFF02) MTP key 寄存器

Bit	7	6	5	4	3	2	1	0
Name	MTP_KEY							
Reset	0x00							
Type	R/W							
Bit	Name	Function						
7:0	MTP_KEY	MTP 烧写密钥, 写 0xCA 打开, 打开后才能向 MTP_CR 的 bit 0 写 1 来启动烧写 MTP。						

8.6.1.4 MTP_ADL (addr: 0xFF03) MTP 编程地址低 8 位

Bit	7	6	5	4	3	2	1	0
Name	ADL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	ADL	MTP 访问地址低 8 位。

8.6.1.5 MTP_ADH (addr: 0xFF04) MTP 编程地址高 6 位

Bit	7	6	5	4	3	2	1	0
Name	-		ADH					
Reset	-		0x00					
Type	-		R/W					

Bit	Name	Function
7:6	-	-
5:0	ADH	MTP 访问地址高 6 位。

8.6.1.6 MTP_PBUFL (addr: 0xFF05) MTP 编程缓冲地址低 8 位

Bit	7	6	5	4	3	2	1	0
Name	PBUFL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	PBUFL	MTP 编程缓冲地址低 8 位 在 MTP MBIST 过程中作为读和写过程中的 Pattern, 供数据写入以及读出校验对比, 高 8bit 和低 8bit 都为 PBUFL

8.6.1.7 MTP_PBUFH (addr: 0xFF06) MTP 编程缓冲地址高 2 位

Bit	7	6	5	4	3	2	1	0
Name	-						PBUFH	
Reset	-						0x00	
Type	-						R/W	

Bit	Name	Function
7:2	—	—
1:0	PBUFH	MTP 编程缓冲地址高 2 位。

8.6.1.8 MTP_DR (addr: 0xFF07) MTP 读数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	DR							
Reset	0xFF							
Type	R							

Bit	Name	Function
7:0	DR	MTP 读数据。

9 时钟控制

9.1 时钟概述

时钟系统负责整个系统时钟的生成、分频以及分配工作。RC6T8X634x 内部包含两个 24MHz 高速 RC 振荡器和 1 个 32KHz 低速 RC 振荡器，系统复位后，CLK_SYS 默认使用 24MHz 高速 RC 振荡器 4 分频的时钟(6MHz)，在程序开始运行后用户可以自行配置修改频率。

9.2 时钟信号说明

RC6T8X634 系列芯片内部实现各种时钟源，增加了整个系统的灵活性，各时钟源详情如下表所示：

CLK_SYS	来源 IMO_0 时钟
VC0	可变时钟 0，和 CLK_SYS 同频
VC1	可变时钟 1，可以是 CLK_SYS 的 1-16 分频
VC2	可变时钟 2，可以是 CLK_SYS 或者 VC1 的 1-16 分频
VC3	可变时钟 3，可以是 CLK_SYS、VC0、VC1、VC2 的 1-256 分频。VC3 时钟自带一个中断，可以单独使能，每次当 VC3 的上升沿到来的时候产生一次中断，用户可用该中断来做定时器。
FCLK	自由时钟，该时钟永远都会工作。
HCLK_CORE	内核时钟，给内核及外设寄存器使用。
HCLK_MEM	存储时钟，给外设寄存器使用。该时钟在测试模式下也工作，需要在测试模式下能否访问的模块，需要连接该时钟。
HCLK_SRAM	给 SRAM 的时钟。
CLK_32K	低速时钟，时钟来自内部 32K ILO 振荡器和晶体振荡器。

注意：VC3 还作为 MTP 烧录时 MTP IP 所需要的时钟，MTP IP 要求的时钟周期范围 10~15us，典型值 12.5us，我们电路实现上使用的是 VC3 时钟的二分频。VC3 的复位默认值约为 160KHz，二分频后约 80KHz，周期约 12.5us，对应 MTP IP 要求的典型值。所有的烧录方式包括 ICP/IAP 等，在开始烧录之前都需要确保 VC3 打开并且频率约为 160KHz，否则 MTP 的烧录不可靠。

9.3 时钟结构框图

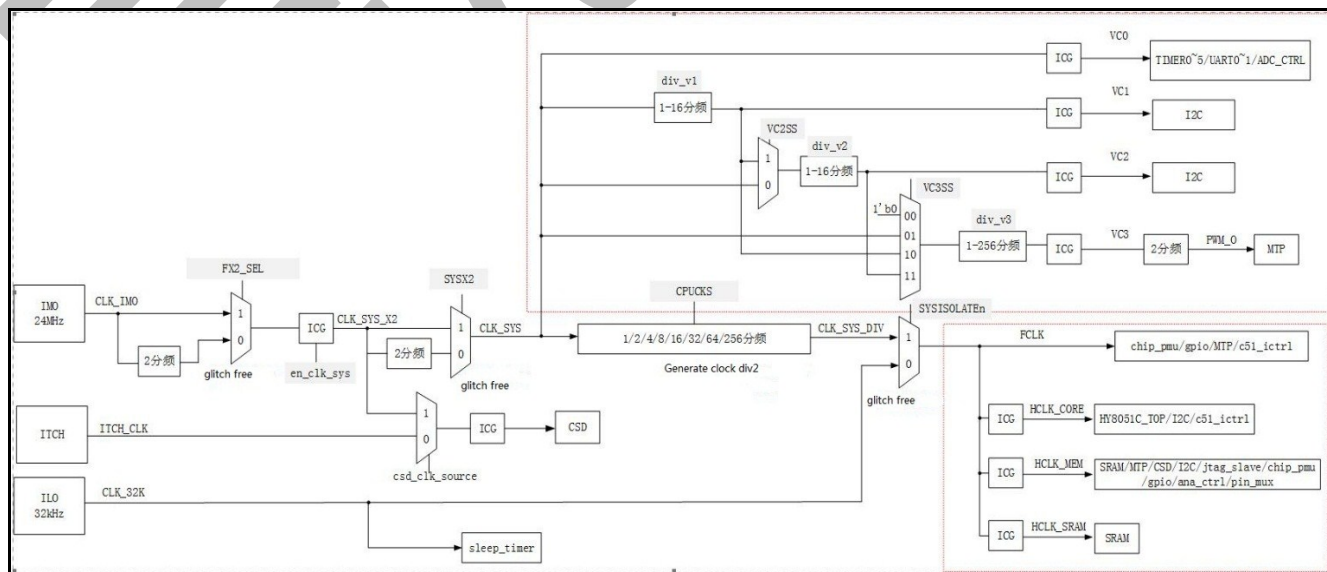


图 9-1 时钟系统结构图

9.4 内部主振（IM0）

内部主振荡器是所有时钟的基本时钟。默认模式 CLK_SYS 使用该时钟的 4 分频做时钟。

9.5 内部低速振荡器（ILO）

内部低速振荡器是低功耗振荡器，不会关闭。也是睡眠定时器和看门狗定时器的时钟。

9.6 寄存器定义

名字	地址	读写	复位值	描述
CLK_CR	0x94	读写	10000011	系统时钟控制寄存器
PCLK_CR	0x95	读写	11110001	外设时钟控制寄存器
PCLK_DIV12	0x96	读写	00001111	VC1、VC2 时钟控制寄存器
PCLK_DIV3	0x97	读写	00100100	VC3 时钟控制寄存器

9.6.1 CLK_CR（addr: 0x94）系统时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	VC3IF	—	—	—	—	CPUCKS[2:0]		
Reset	1	—	—	—	—	0	1	1
Type	R/W	—	—	—	—	R/W	R/W	R/W

Bit	Name	Function
7	VC3IF	0 = 没有 VC3 中断发生 1 = 有 VC3 中断发生 对该位写 1 会将其清零，写 0 无效。 注意：VC3IF 复位值为 0，而 VC3 默认情况下是有效的，而且会在软件启动之前就起振，因此软件看到的复位值为 0x83。
6:3	—	—
2:0	CPUCKS[2:0]	内核工作频率选择： 000 SYSCLK/8 001 SYSCLK/4 010 SYSCLK/2 011 SYSCLK 100 SYSCLK/16 101 SYSCLK/32 110 SYSCLK/64 111 SYSCLK/128

9.6.2 PCLK_CR（addr: 0x95）外设时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	VCOEN	VC1EN	VC2EN	VC3EN	VC3_IE	VC2SS	VC3SS	
Reset	1	1	1	1	0	0	0	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	VCOEN	0 禁止 VCO 时钟

		1 使能 VC0 时钟
6	VC1EN	0 禁止 VC1 时钟 1 使能 VC1 时钟
5	VC2EN	0 禁止 VC2 时钟 1 使能 VC2 时钟
4	VC3EN	0 禁止 VC3 时钟 1 使能 VC3 时钟
3	VC3IE	0 禁止 VC3 时钟中断 1 使能 VC3 时钟中断
2	VC2SS	VC2 时钟源选择，具体使用见 VC2CKS 说明 0 CLK_SYS 作为 VC2 的时钟源 1 VC1 作为 VC2 的时钟源
1:0	VC3SS	VC3 时钟源选择 00 关闭 VC3 时钟 01 来自 CLK_SYS 10 来自 VC1 时钟 11 来自 VC2 时钟

9.6.3 PCLK_DIV12 (addr:0x96) VC1、VC2 时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	V1CKS[3:0]				V2CKS[3:0]			
Reset	0	0	0	0	1	1	1	1
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:4	VC1CKS	控制 VC1 时钟分频 $f_{VC1} = f_{CLK_SYS} / (VC1CKS + 1)$
3:0	VC2CKS	控制 VC2 时钟分频 VC2SS=0 时 $f_{VC2} = f_{CLK_SYS} / (VC2CKS + 1)$ VC2SS=1 时 $f_{VC2} = f_{CLK_SYS} / (VC2CKS + 1) / (VC1CKS + 1)$

9.6.4 PCLK_DIV3 (addr:0x97) VC3 时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	VC3CKS							
Reset	0x24							
Type	R/W							

Bit	Name	Function
7:0	VC3KS[7:0]	控制 VC3 时钟的分频，频率和 VC3SS 的值相关，具体计算方法如下： 当 VC3SS 等于： 00 关闭 VC3 时钟 01 $f_{VC3} = f_{CLK_SYS} / (VC3CKS + 1)$ 10 $f_{VC3} = f_{CLK_SYS} / (VC3CKS + 1) / (VC1CKS + 1)$ 11 $f_{VC3} = f_{VC2} / (VC3CKS + 1)$

10 外设资源

10.1 TIMER

RC6T8X634 系列芯片包含 6 个 16 位 Timer (Timer0/1/2/3/4/5), 每个 Timer 都可以配置为为定时器 (以 VC0 (12Mhz) 作为工作时钟 clk) 或计数器 (以外部管脚的输入脉冲)。当使用 TIMER 作为计数器时, 因需检测外部信号脉冲的上升沿, 所以外部管脚的输入脉冲的频率最高为模块工作频率 VC0 的 2 分频。

每个 Timer 使能置 1 打开定时器, 但不会复位定时器, 定时器寄存器将会从上次 Timer 清零的值开始计数。所以在启动定时器工作前, 应设定定时器寄存器的初始值。

10.1.1 Timer0 与 Timer1

Timer0 可配置为 4 种工作模式, Timer1 可配置为 3 种工作模式, 通过配置 TMOD 寄存器实现:

- (1) 13 位定时器/计数器 (模式 0)
- (2) 16 位定时器/计数器 (模式 1)
- (3) 8 位自动重载定时器/计数器 (模式 2); 生成 PWM 波 (仅 Timer0)
- (4) 2 个 8 位定时器/计数器 (模式 3, 仅 Timer0)

10.1.1.1 模式 0: 13 位定时器/计数器

模式 0 的工作示意图如模式 1 图所示。Timer0 与 Timer1 都能实现该模式。在该模式下, Timer 被配置为 13 位的计数器/定时器, 使用 TL 的低 5 位与 TH 的 8 位组成 13 位。TIMER_MOD 寄存器的 C/T 位用来选择 Timer 的计数来源, 当 Timer 计数至 1FFFh 时会产生溢出事件, 将 TIMER_CON 寄存器的 TF 位置 1, 若中断使能有效, 则会上报溢出中断至 CPU。

TR0/1 置 1 将打开定时器, 但不会复位定时器, 定时器寄存器将会从上次 TR0/1 清 0 的值开始计数。所以, 在启动定时器工作前, 应设定定时器寄存器的初始值。

10.1.1.2 模式 1: 16 位定时器/计数器

模式 1 的工作示意如下图所示。Timer0 与 Timer1 都能实现该模式, 除了 TL 位使用了全部 8 位外, 该模式的工作方式与模式 0 相同。

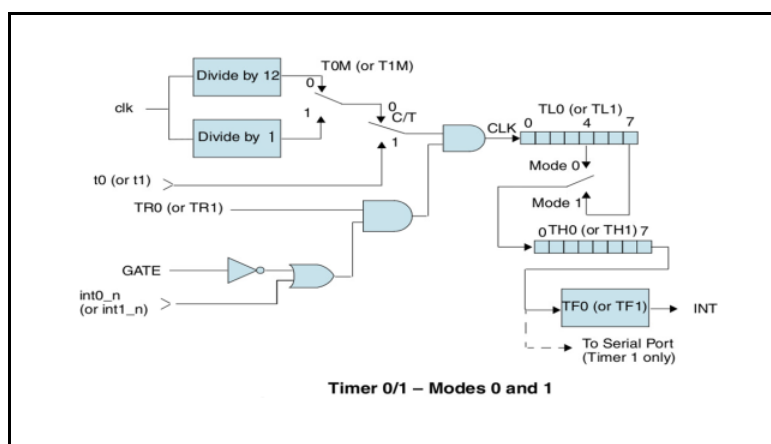


图 10-1

10.1.1.3 模式 2：8 位自动重载定时器/计数器

模式 2 的工作示意如下图所示，Timer0 与 Timer1 都能实现该模式。在模式 2 中，TL 用于存放 Timer 计数值，TH 用于存放重载值，当 TL 发生溢出时，硬件会将 TH 中的值自动装载到 TL 中。除了自动重载功能外，其他工作方式与模式 0/1 相同。

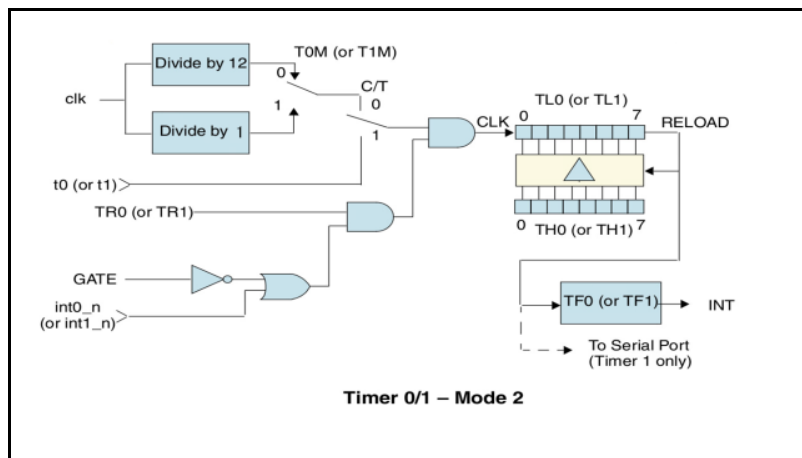


图 10-2

T0_PWM 功能特点说明（仅 Timer0）：

- 1、能覆盖 4KHz PWM 波，可以驱动蜂鸣器。
- 2、8 位计数器，向上计数到 FF。
- 3、计数周期可配置，占空比可调。

T0_PWM 脉宽调制模块为周期和脉宽都可以通过寄存器进行配置。

T0_PWM 信号的周期由周期配置寄存器（TH0）的值确定。

占空比由占空比寄存器（DUTY0）中的设置值确定，取值范围在 TH-FF 之间，0%和 100%的占空比都是可能的。当占空比寄存器（DUTY0）被设为 TH0 时，占空比为 0%；当占空比寄存器（DUTY0）设置为等于 FF 可实现 100%的占空比。T0_PWM 信号的极性由 TCON 控制位中的设置确定，初始设定之后，不支持中途改变。

当极性选择为 0 时，周期= (FF-TH)，占空比= (DUTY0-TH0) / (FF-TH0)。

计数器是自动重载的，不会自行停止，直到寄存器 T0_PWM 使能关闭才会停止。在重新启动 T0_PWM，需设定 timer3 寄存器的初始值。

产生 T0_PWM 波如下图所示。

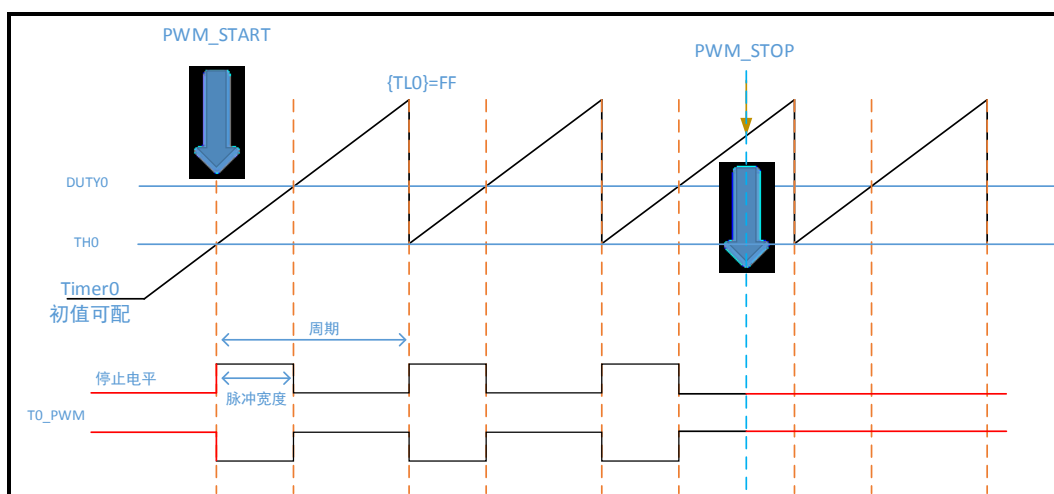


图 10-3

10.1.1.4 模式 3：2 个 8 位定时器/计数器（仅 Timer0）

模式 3 的工作示意如下图所示，仅 Timer0 能实现该模式。当 Timer0 处于模式 3 时，Timer1 将会停止计数并保持当前的计数值。Timer0 的 TL 和 TH 相互独立，TL 可工作为定时器或计时器，TH 只能工作为定时器（无法通过外部引脚触发工作）。在此模式下，Timer1 的使能位（T1_EN）、溢出标志位（T1_TF）以及中断事件都服务于 Timer0 的 TH。

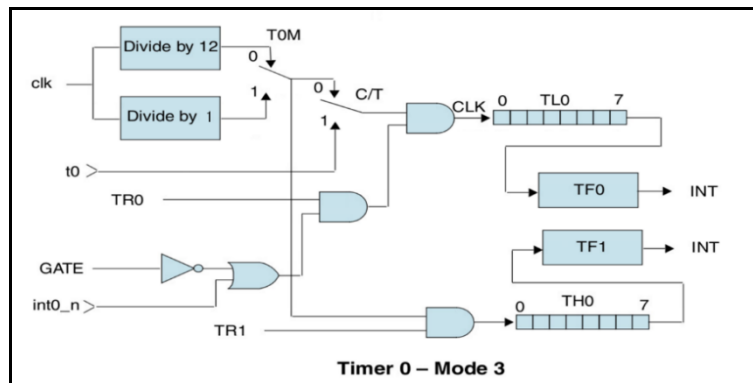


图 10-4

10.1.2 Timer2

Timer2 为 16 位的定时器/计数器，具有 2 种触发中断的方式（T2_IE 为 1）：Timer2 的溢出事件。Timer2 通过系统时钟可以实现以下功能：

- 16 位自动重载的定时器/计数器
- 波特率生成器

通过配置 TIMER2_T2CON 的 CP/RL 位为 0 使得 Timer2 进入自动重载模式。在该模式下，Timer2 的溢出事件将会触发 Timer2 将 TIMER_RCAP2L 和 TIMER_RCAP2H 的值装载至 Timer2 中。

10.1.3 Timer3

Timer3 为 16 位的定时器/计数器，触发中断的方式（T3_IE 为 1）：Timer3 的溢出事件。Timer3 通过系统时钟触发可实现以下功能：

- 16 位自动重载的定时器/计数器；生成 PWM 波

通过配置 TIMER3_T3CON 的 CP/RL 位为 0 使得 Timer3 进入自动重载模式。在该模式下，Timer3 的溢出事件时，将会触发 Timer3 将 TIMER3_RCAP3L 和 TIMER3_RCAP3H 的值装载至 Timer3 中。

T3_PWM 功能说明（仅 Timer3）：

T3_PWM 功能特点：

- 1、16 位计数器，向上计数到 FFFF。
- 2、计数周期可配置，占空比可调。

T3_PWM 脉宽调制模块为周期和脉宽都可以通过寄存器进行配置。

T3_PWM 信号的周期由周期配置寄存器（{RCAP3H, RCAP3L}）的值确定。

占空比由占空比寄存器（{DUTY3H, DUTY3L}）中的设置确定，0%和 100%的占空比都是可能的。当占空比寄存器（{DUTY3H, DUTY3L}）被设为（{CAP3H, CAP3L}）时，占空比为 0%；当占空比寄存器（{DUTY3H, DUTY3L}）设置为等于 FFFF 可实现 100%的占空比。

T3_PWM 信号的极性由 PWM_CFG 控制位中的设置确定，初始设定之后，不支持中途改变。

当极性选择为 0 时, 周期= $(FFFF - \{RCAP3H, RCAP3L\})$,
占空比= $(\{DUTY3H, DUTY3L\} - \{RCAP3H, RCAP3L\}) / (FFFF - \{RCAP3H, RCAP3L\})$ 。

计数器是自动重载的, 不会自行停止, 直到寄存器 T3_PWM 使能关闭才会停止。在重新启动 T3_PWM, 需设定 timer3 寄存器的初始值。

产生 T3_PWM 波如下图所示。

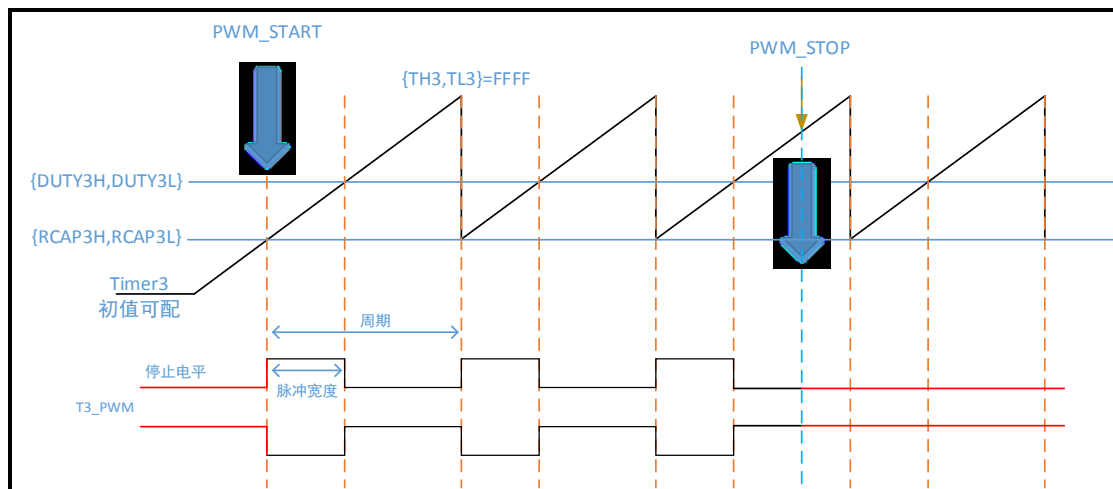


图 10-5

10.1.4 Timer4

Timer4 为 16 位的定时器/计数器, 触发中断的方式 (T4_IE 为 1): Timer4 的溢出事件。Timer4 通过系统时钟触发可实现以下功能:

- 16 位自动重载的定时器/计数器; 生成 PWM 波

通过配置 TIMER4_T4CON 的 CP/RL 位为 0 使得 Timer4 进入自动重载模式。在该模式下, Timer4 的溢出事件将会触发 Timer4 将 TIMER4_RCAP4L 和 TIMER4_RCAP4H 的值装载至 Timer4 中。

T4_PWM 功能说明 (仅 Timer4):

T4_PWM 功能特点:

- 1、较为丰富的 PWM, 分频器范围大。
- 2、16 位计数器, 向上计数到 FFFF。
- 3、计数周期可配置, 占空比可调。
- 4、周期数 N 可选 (N=0; N≠0)。
- 5、可通过软件配置或外部引脚驱动来启动 PWM 波。
- 6、通过软件配置和外部引脚以及周期个数来停止 PWM 波。
- 7、写 0 清除中断状态位 (不管中断状态位为 0/1), 可立即生效周期数 N。
- 8、改变周期和占空比, 则在下一周期生效。

T4_PWM 脉宽调制模块为周期和脉宽都可以通过寄存器进行配置。

T4_PWM 信号的周期由周期配置寄存器 ({RCAP4H, RCAP4L}) 的值确定。

占空比由占空比寄存器 ({DUTY4H, DUTY4L}) 中的设置确定, 0%和 100%的占空比都是可能的。当占空比寄存器 ({DUTY4H, DUTY4L}) 被设为 ({RCAP4H, RCAP4L}) 时, 占空比为 0%; 当占空比寄存器 ({DUTY4H, DUTY4L}) 设置为等于 FFFF 可实现 100%的占空比。

T4_PWM 信号的极性由 PWM_CFG 控制位中的设置确定, 初始设定之后, 不支持中途改变。计数器是自动重载的, 不会自行停止, 直到 T4_PWM 使能关闭才会停止。在重新启动 timer4 工作前, 应设定 timer4 寄存器的初

始值。

当极性选择为 0 时，周期= (FFFF- {RCAP4H, RCAP4L})，
 占空比= ({DUTY4H, DUTY4L} - {RCAP4H, RCAP4L}) / (FFFF- {RCAP4H, RCAP4L})。

PIN_STOP1、PIN_STOP0 分别选择 PIN1、PIN0 的停止电平 (可见寄存器具体说明)，PIN_EN1、PIN_EN0 分别选择 PIN1、PIN0 的通断。

PWM 启停控制如下所示：

- 1、软件控制启动/停止。
- 2、在软件控制启动的前提下，且外部引脚使能有效，根据外部任一引脚 PIN1/PIN0 控制启动/停止。(先配外部引脚使能，再配软件控制启动)。
- 3、软件和外部引脚均未控制停止 PWM 时，此时由 PWM 是否产生 N 个周期以及选停配置 PWM4_SELSTOP 来控制 PWM 停止。

SWEN	PIN_EN0/ PIN_EN1	PIN_STOP0/ PIN_STOP1	PIN0/PIN1	PWM4_ENABLE
0	X	X	X	0
1	0	X	X	1
1	1	1	0	1
1	1	1	1	0
1	1	0	0	0
1	1	0	1	1

判断周期数 N：

当 N=0 时，则循环输出 PWM 波；

当 N≠0 时，当该中断使能位有效时，输出 PWM 中断信号。

- 1、选停配置寄存器 (PWM4_SELSTOP)=1，输出 N 个 PWM 波后停止，置位该中断标志位；
- 2、选停配置寄存器 (PWM4_SELSTOP)=0 时，控制 PWM 不停止，则置位该中断标志位。

产生 T4_PWM 波如下图所示。

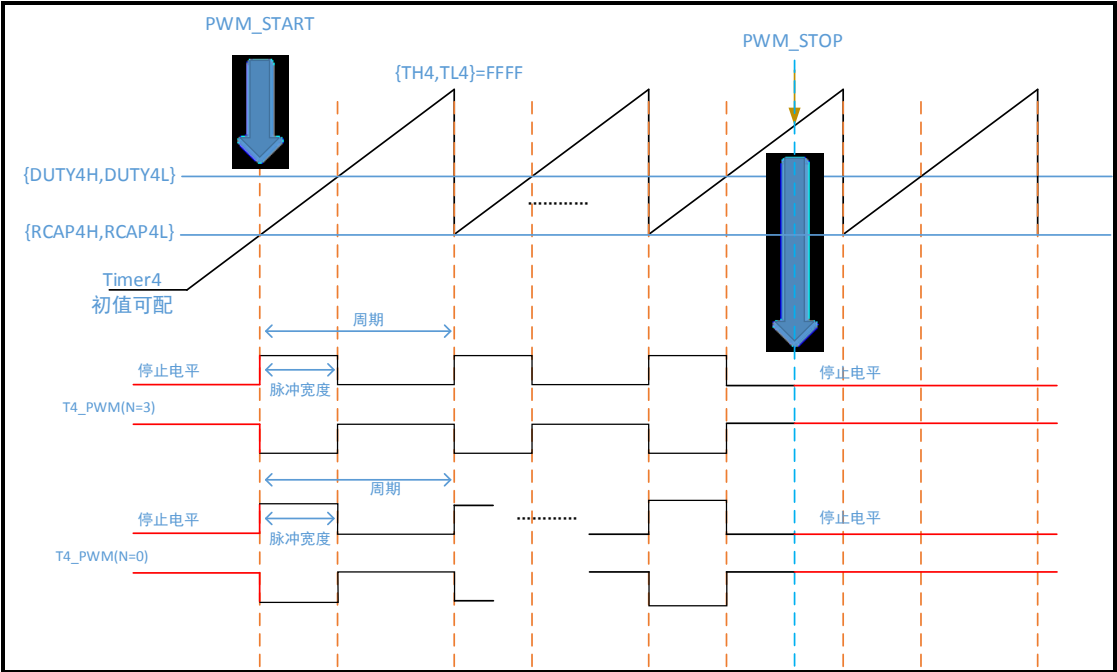


图 10-6

10.1.5 Timer5

Timer5 为 16 位的定时器/计数器，具有 2 种触发中断的方式：Timer5 的溢出事件以及外部引脚 T5EX 的下降沿事件。外部引脚输入脉冲频率不得大于 Timer 工作时钟的 1/2。Timer5 通过系统时钟或外部引脚 T5EX 的下降沿触发可实现以下功能：

- 16 位带捕获功能的定时器/计数器
- 16 位自动重载的定时器/计数器

10.1.5.1 16 位带捕获功能的定时器/计数器

通过配置 TIMER5_CON 的 CP/RL 位以及 EXEN5 位为 1 使得 Timer5 进入该模式。当外部引脚 T5EX 出现下降沿时，将会触发 Timer5 的捕获功能，硬件将此时的 Timer 值存储在 TIMER5_RCAP5L 和 TIMER5_RCAP5H 中。

TR5 置 1 将打开定时器，但是不会复位定时器，定时器寄存器将会从上次 TR5 清 0 的值开始计数。所以，在启动定时器工作前，应设定定时器寄存器的初始值。

10.1.5.2 16 位带自动重载的定时器/计数器

通过配置 TIMER5_CON 的 CP/RL 位为 0 使得 Timer5 进入该模式。在该模式下，Timer5 的溢出事件以及外部引脚 T5EX 出现下降沿时（EXEN5 位为 1）将会触发 Timer5 将 TIMER5_CAPL 和 TIMER5_CAPH 的值装载至 Timer5 中。

10.1.6 波特率发生器

Timer1 可作为串口 0 和串口 1 的波特率生成器，Timer2 仅能作为串口 0 的波特率生成器。Timer3 和 Timer4 和 Timer5 无波特率生成。选中 T2CON 寄存器中的 TCLK 或 RCLK 即可使得 Timer2 进入波特率生成模式，为串口 0 提供对应波特率。同时若使用 Timer1 作为波特率生成器时，需将 Timer1 配置为模式 2:8 位自动重载定时器/计数器（尽管其他模式可用），因为模式 2 的溢出后自动重载重新计数而无需连续软件介入的特征，很适合作为波特率发生器。Timer1 将溢出标识位 TF1 的内部置位信号输出供串口 0/1 作为波特率使用。

Timer2 在波特率生成器模式下计数溢出时不会置位溢出标志位 TF2，而会产生溢出信号作为移位时钟供串口使用。此模式下，若 EXEN2 位被置 1，在 t2ex 引脚上的下降沿会置位 EXF2 位，产生中断，但不会引起重载。此模式下的计数时钟为系统时钟二分频 clk/2。若要使用外部时钟源，则可置位 C/T2 为 1 并将想要使用的外部时钟源接到 t2 引脚。

10.1.7 寄存器定义

名字	地址	读写	复位值	描述
TMOD	89H	R/W	00H	Timer0/1 工作模式配置寄存器
TCON	88H	R/W	00H	Timer0/1 控制寄存器
TL0	8AH	R/W	00H	Timer0 计数值低 8 位
TH0	8CH	R/W	00H	Timer0 计数值高 8 位
TL1	8BH	R/W	00H	Timer1 计数值低 8 位
TH1	8DH	R/W	00H	Timer1 计数值高 8 位
CKCON	8EH	R/W	00H	时钟控制寄存器
T2CON	C5H	R/W	00H	Timer2 控制寄存器

RCAP2L	CAH	R/W	00H	Timer2 重装载值低 8 位
RCAP2H	CBH	R/W	00H	Timer2 重装载值高 8 位
TL2	CCH	R/W	00H	Timer2 计数值低 8 位
TH2	CDH	R/W	00H	Timer2 计数值高 8 位
DUTY0	D1H	R/W	00H	Timer0 输出 PWM 占空比设置
TIRQ	8FH	R/W	00H	Timer3/4/5 中断控制
TPCON	D7H	R/W	00H	TimerPwm 控制寄存器
PWM_CFG	D6H	R/W	00H	Timer3/4 输出 PWM 配置值设置
T4_PWM_NUM	C9H	R/W	00H	T4_PWM 周期数配置
T3CON	C8H	R/W	00H	Timer3 控制寄存器
RCAP3L	BCH	R/W	00H	Timer3 重装载值低 8 位
RCAP3H	BDH	R/W	00H	Timer3 重装载值高 8 位
TL3	BEH	R/W	00H	Timer3 计数值低 8 位
TH3	BFH	R/W	00H	Timer3 计数值高 8 位
DUTY3L	D2H	R/W	00H	Timer3 输出 PWM 占空比设置低 8 位
DUTY3H	D3H	R/W	00H	Timer3 输出 PWM 占空比设置高 8 位
T4CON	C0H	R/W	00H	Timer4 控制寄存器
RCAP4L	C1H	R/W	00H	Timer4 重装载值低 8 位
RCAP4H	C2H	R/W	00H	Timer4 重装载值高 8 位
TL4	C3H	R/W	00H	Timer4 计数值低 8 位
TH4	C4H	R/W	00H	Timer4 计数值高 8 位
DUTY4L	D4H	R/W	00H	Timer4 输出 PWM 占空比设置低 8 位
DUTY4H	D5H	R/W	00H	Timer4 输出 PWM 占空比设置高 8 位
T5CON	D8H	R/W	00H	Timer5 控制寄存器
RCAP5L	C6H	R/W	00H	Timer5 捕获/重装载值低 8 位
RCAP5H	C7H	R/W	00H	Timer5 捕获/重装载值高 8 位
TL5	CEH	R/W	00H	Timer5 计数值低 8 位
TH5	CFH	R/W	00H	Timer5 计数值高 8 位

10.1.7.1 TMOD (addr:0x89) Timer0/1 工作模式配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	–	T1_C/T	T1_M1	T1_M0	–	T0_C/T	T0_M1	T0_M0
Reset	–	0	0	0	–	0	0	0
Type	–	R/W	R/W	R/W	–	R/W	R/W	R/W

Bit	Name	Function
7	–	–
6	T1_C/T	Timer1 计数来源选择

		0 作为定时器, Timer1 计数来源于工作时钟 clk 1 无效
5: 4	[T1_M1]:[T1_M0]	Timer1 工作模式选择 00 13 位定时器/计数器, TL1 的高 3 位无效 01 16 位定时器/计数器 10 8 位自动重载计数器, 溢出时将 TH1 存放的值自动重装入 TL1 11 无效配置, 停止计数
3	—	—
2	T0_C/T	Timer0 计数来源选择 0 作为定时器, Timer0 计数来源于工作时钟 clk 1 无效
1:0	[T0_M1]:[T0_M0]	Timer0 工作模式选择 00 13 位定时器/计数器, TL0 的高 3 位无效 01 16 位定时器/计数器 10 8 位自动重载计数器, 溢出时将 TH0 存放的值自动重装入 TL0 输出 PWM 波形 11 2 个独立的 8 位计数器, 此模式下 TL0 作为一个 8 位定时器/计数器, 通过 TIMER_CON 的 TR0 控制启停, TH0 作为一个 8 位定时器, 通过 TIMER_CON 的 TR1 控制启停

10.1.7.2 TCON (addr:0x88) Timer0/1 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	—	T2_IE	T1_IE	T0_IE
Reset	0	0	0	0	—	0	0	0
Type	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

Bit	Name	Function
7	TF1	Timer1 溢出标志位: 0 未发生溢出 1 Timer1 产生溢出, 该位只能通过写 1 清零
6	TR1	Timer1 工作使能位 0 停止工作 1 开始工作
5	TF0	Timer0 溢出标志位: 0 未发生溢出 1 Timer0 产生溢出, 该位只能通过写 1 清零
4	TR0	Timer0 工作使能位 0 停止工作 1 开始工作 若配置 T0_PWM 使能有效, Timer0 将开始工作, 在产生 T0_PWM 波时, 禁止配置 TR0 位
3	—	—
2	T2_IE	Timer2 中断使能: 0 不上报中断 1 上报中断
1	T1_IE	Timer1 中断使能: 0 不上报中断

		1 上报中断
0	TO_IE	Timer0 中断使能： 0 不上报中断 1 上报中断

10.1.7.3 TL0 (addr:0x8A) Timer0 计数值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL0							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TL0	写：写该寄存器会将写入值装载至 Timer0 的低 8 位，从更新值重新计数 读：读该寄存器将会读出 Timer0 当前的低 8 位值

10.1.7.4 TH0 (addr:0x8C) Timer0 计数值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH0							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH0	写：写该寄存器会将写入值装载至 Timer0 的高 8 位，从更新值重新计数 读：读该寄存器将会读出 Timer0 当前的高 8 位值

10.1.7.5 TL1 (addr:0x8B) Timer1 计数值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL1							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TL1	写：写该寄存器会将写入值装载至 Timer1 的低 8 位，从更新值重新计数 读：读该寄存器将会读出 Timer1 当前的低 8 位值

10.1.7.6 TH1 (addr:0x8D) Timer1 计数值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH1							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH1	写：写该寄存器会将写入值装载至 Timer1 的高 8 位，从更新值重新计数 读：读该寄存器将会读出 Timer1 当前的高 8 位值

10.1.7.7 CKCON (addr:0x8E) 时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	-		T2M	T1M	T0M	T5M	T4M	T3M
Reset	-		0	0	0	0	0	0
Type	-		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	-	-
5	T2M	Timer2 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk
4	T1M	Timer1 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk 时钟
3	T0M	Timer0 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk
2	T5M	Timer5 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk
1	T4M	Timer4 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk 时钟
0	T3M	Timer3 作为定时器时钟来源选择 0 工作时钟 clk 的 12 分频 1 工作时钟 clk

10.1.7.8 T2CON (addr:0xC5) Timer2 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	TF2	-	RCLK	TCLK	-	TR2	T2_C/T	CP/RL
Reset	0	-	0	0	-	0	0	0
Type	R/W	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7	TF2	Timer2 溢出标志位: 0 未发生溢出 1 Timer2 产生溢出, 该位只能通过写 1 清零
6	-	-
5	RCLK	UART 接收时钟控制位 0 Timer1 产生串口 0 的接收波特率 1 Timer2 产生串口 0 的接收波特率
4	TCLK	UART 发送时钟控制位 0 Timer1 产生串口 0 的发送波特率 1 Timer2 产生串口 0 的发送波特率
3	-	-
2	TR2	Timer2 工作使能位

		0 停止工作 1 开始工作
1	T2_C/T	Timer2 计数来源选择 0 作为定时器, Timer2 计数来源于系统时钟 1 作为计数器, Timer2 计数来源于外部管脚脉冲的下降沿
0	GP/RL	捕获/重装载模式选择 0 16 位带重装载功能的定时器/计数器, 重载事件发生在 Timer2 溢出或 T2EX 输入的下降沿 (当 EXEN2 配为 1) 1 16 位带捕获功能的定时器/计数器, 捕获事件发生在 T2EX 输入的下降沿

10.1.7.9 RCAP2L (addr:0xCA) Timer2 捕获/重装载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP2L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP2L	捕获模式下, 该寄存器用于存放捕获时间发生时 Timer2 的低 8 位, 重装载模式下, 该寄存器用于存放用于重装载值的低 8 位

10.1.7.10 RCAP2H (addr:0xCB) Timer2 捕获/重装载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP2H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP2H	捕获模式下, 该寄存器用于存放捕获时间发生时 Timer2 的高 8 位, 重装载模式下, 该寄存器用于存放用于重装载值的高 8 位

10.1.7.11 TL2 (addr:0xCC) Timer2 计数值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL2							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TL2	写: 写该寄存器会将写入值装载至 Timer2 的低 8 位, 每写一次装载一次 读: 读该寄存器将会读出 Timer2 当前的低 8 位值

10.1.7.12 TH2 (addr:0xCD) Timer2 计数值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH2							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH2	写：写该寄存器会将写入值装载至 Timer2 的高 8 位，每写一次装载一次 读：读该寄存器将会读出 Timer2 当前的高 8 位值

10.1.7.13 DUTY0 (addr:0xD1) Timer0 PWM 输出占空比配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY0							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH2	写：写该寄存器，可设置 PWM0 占空比的 8 位值 读：读该寄存器，可读出 PWM0 占空比的 8 位值

10.1.7.14 TIRQ (addr:0x8F) Timer3/4/5 中断控制

Bit	7	6	5	4	3	2	1	0
Name	-					T5_IE	T4_IE	T3_IE
Reset	-					0x00	0x00	0x00
Type	-					R/W	R/W	R/W

Bit	Name	Function
7:3	-	-
2	T5_IE	Timer5 中断使能： 不上报中断 上报中断
1	T4_IE	Timer4 中断使能： 不上报中断 上报中断
0	T3_IE	Timer3 中断使能： 0 不上报中断 1 上报中断

10.1.7.15 TPCON (addr:0xD7) TimerPwm 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	PIN_EN1	PIN_EN0	PIN_STOP1	PIN_STOP0	-	TOM24	PWM0_POLSEL	PWM0_EN
Reset	0x00	0x00	0x00	0x00	-	0x00	0x00	0x00
Type	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7	PIN_EN1	PIN1 控制使能信号 0: PIN1 无效 1: PIN1 有效
6	PIN_EN0	PIN0 控制使能信号 0: PIN0 无效 1: PIN0 有效
5	PIN_STOP1	PIN1 停止电平选择

		0: PIN1: 0 电平为停止电平, 1 电平为启动电平 1: PIN1: 1 电平为停止电平, 0 电平为启动电平
4	PIN_STOP0	PIN0 停止电平选择 0: PIN0: 0 电平为停止电平, 1 电平为启动电平 1: PIN0: 1 电平为停止电平, 0 电平为启动电平
3	—	—
2	TOM24	TOM24 作为 T0_PWM 输出时时钟来源选择 0: T0_PWM 工作时钟来自 timer0 的时钟来源选择 (工作时钟 clk/工作时钟 clk 的 12 分频) 1: T0_PWM 工作时钟与 timer0 的时钟, 都为工作时钟 clk 的 24 分频
1	PWM0_POLSEL	PWM0_EN 不使能, 设置 T0_PWM 的极性: 1: T0_PWM 的电平极性为 1 0: T0_PWM 的电平极性为 0
0	PWM0_EN	T0_PWM 启动控制位, 配置该位, 可硬件置位 TR0 位 0: 设置 T0_PWM 停止 1: 设置 T0_PWM 启动

10.1.7.16 PWM_CFG (addr:0xD6) Timer3/4 输出 PWM 配置值寄存器

Bit	7	6	5	4	3	2	1	0
Name	PWM4_POLSEL	PWM4_ISR	PWM4_EN	PWM4_IEN	PWM4_SELSTOP	—	PWM_POLSEL	PWM3_EN
Reset	0x00	0x00	0x00	0x00	0x00	—	0x00	0x00
Type	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

Bit	Name	Function
7	PWM4_POLSEL	设置 T4_PWM 的极性 1: T4_PWM 的电平为 1 0: T4_PWM 的电平为 0
6	PWM4_ISR	PWM4_ISR 中断状态 当 PWM4_ISR 设置时, 表示已产生 N(N≠0) 个 PWM 波 此位由硬件置位, 软件写 0 操作清除 软件写 0 操作可立即生效周期数 N
5	PWM4_EN	T4_PWM 中断使能信号 0: 中断不使能 1: 中断使能
4	PWM4_IEN	T4_PWM 使能信号: 0: T4_PWM 停止 1: T4_PWM 启动
3	PWM4_SELSTOP	PWM 达到 N 个周期后, 选停信号 0: 选择不停止 1: 选择停止
2	—	—
1	PWM_POLSEL	设置 T3_PWM 的电平极性 1: T3_PWM 的电平为 1 0: T3_PWM 的电平为 0
0	PWM3_EN	T3_PWM 使能 0: 设置 T3_PWM 停止 1: 设置 T3_PWM 启动

10.1.7.17 T4_PWM_NUM (addr:0xC9) T4_PWM 周期数配置值寄存器

Bit	7	6	5	4	3	2	1	0
Name	T4_PWM_NUM							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	T4_PWM_NUM	T4_PWM 周期数配置, 可配置范围 0~255

10.1.7.18 T3CON (addr:0xC8) Timer3 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	TF3	-				TR3	T3_C/T	CP/RL
Reset	0x00	-				0x00	0x00	0x00
Type	R/W	-				R/W	R/W	R/W

Bit	Name	Function
7	TF3	Timer3 溢出标志位: 0 未发生溢出 1 Timer3 产生溢出, 该位只能通过写 1 清零
6:3	-	-
2	TR3	Timer3 工作使能位 若配置 T3_PWM 使能有效, Timer3 将开始工作, 在产生 T3_PWM 波期间, 禁止配置 TR3 位。 0 停止工作 1 开始工作
1	T3_C/T	Timer3 计数来源选择 0 作为定时器, Timer3 计数来源于系统时钟 1 无效
0	CP/RL	重载模式选择 0 16 位带重载功能的定时器/计数器, 重载事件发生在 Timer3 溢出 1 无效

10.1.7.19 RCAP3L (addr:0xBC) Timer3 重载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP3L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP3L	重载模式下, 该寄存器用于存放用于重载值的低 8 位

10.1.7.20 RCAP3H (addr:0xBD) Timer3 重载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP3H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP3H	重装载模式下, 该寄存器用于存放用于重装载值的高 8 位

10.1.7.21 TL3 (addr:0xBE) Timer3 计数值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL3							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TL3	写: 写该寄存器会将写入值装载至 Timer3 的高 8 位, 每写一次装载一次 读: 读该寄存器将会读出 Timer3 当前的高 8 位值

10.1.7.22 TH3 (addr:0xBF) Timer3 计数值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH3							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH3	写: 写该寄存器会将写入值装载至 Timer3 的高 8 位, 每写一次装载一次 读: 读该寄存器将会读出 Timer3 当前的高 8 位值

10.1.7.23 DUTY3L (addr:0xD2) Timer3 输出 PWM 占空比设置低 8 位

Bit	7	6	5	4	3	2	1	0
Name	DUTY3L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY3L	写: 写该寄存器, 可设置 PWM 占空比的低 8 位值 读: 读该寄存器, 可读出 PWM 占空比的低 8 位值

10.1.7.24 DUTY3H (addr:0xD3) Timer3 输出 PWM 占空比设置高 8 位

Bit	7	6	5	4	3	2	1	0
Name	DUTY3H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY3H	写: 写该寄存器, 可设置 PWM 占空比的高 8 位值 读: 读该寄存器, 可读出 PWM 占空比的高 8 位值

10.1.7.25 T4CON (addr:0xC0) Timer4 控制寄存器

Bit	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---

Name	TF4	–	TR4	T4_C/T	CP/RL
Reset	0x00	–	0x00	0x00	0x00
Type	R/W	–	R/W	R/W	R/W

Bit	Name	Function
7	TF4	Timer4 溢出标志位: 0 未发生溢出 1 Timer4 产生溢出, 该位只能通过写 1 清零
6:3	–	–
2	TR4	Timer4 工作使能位 若配置 T4_PWM 使能有效, Timer4 将开始工作, 在产生 T4_PWM 波期间, 禁止配置 TR4 位。 0 停止工作 1 开始工作
1	T4_C/T	Timer4 计数来源选择 0 作为定时器, Timer4 计数来源于系统时钟 1 无效
0	CP/RL	重装载模式选择 0 16 位带重装载功能的定时器/计数器, 重载事件发生在 Timer4 溢出 1 无效

10.1.7.26 RCAP4L (addr:0xC1) Time4 重装载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP4L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP4L	重装载模式下, 该寄存器用于存放用于重装载值的低 8 位

10.1.7.27 RCAP4H (addr:0xC2) Time4 重装载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP4H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP4H	重装载模式下, 该寄存器用于存放用于重装载值的高 8 位

10.1.7.28 TL4 (addr:0xC3) Time4 重装载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL4							
Reset	0x00							
Type	R/W							

Bit	Name	Function
-----	------	----------

7:0	TL4	写：写该寄存器会将写入值装载至 Timer4 的低 8 位，每写一次装载一次 读：读该寄存器将会读出 Timer4 当前的低 8 位值
-----	-----	--

10.1.7.29 TH4 (addr:0xC4) Time4 重载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH4							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TH4	写：写该寄存器会将写入值装载至 Timer4 的高 8 位，每写一次装载一次 读：读该寄存器将会读出 Timer4 当前的高 8 位值

10.1.7.30 DUTY4L (addr:0xD4) Timer4 输出 PWM 占空比设置低 8 位

Bit	7	6	5	4	3	2	1	0
Name	DUTY4L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY4L	写：写该寄存器，可设置 PWM 占空比的低 8 位值 读：读该寄存器，可读出 PWM 占空比的低 8 位值

10.1.7.31 DUTY4H (addr:0xD5) Timer4 输出 PWM 占空比设置高 8 位

Bit	7	6	5	4	3	2	1	0
Name	DUTY4H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY4H	写：写该寄存器，可设置 PWM 占空比的高 8 位值 读：读该寄存器，可读出 PWM 占空比的高 8 位值

10.1.7.32 T5CON (addr:0xD8) Timer5 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	TF5	EXF5	—	—	EXEN5	TR5	T5_C/T	CP/RL
Reset	0x00	0x00	—	—	0x00	0x00	0x00	0x00
Type	R/W	R/W	—	—	R/W	R/W	R/W	R/W

Bit	Name	Function
7	TF5	Timer5 溢出标志位： 0 未发生溢出 1 Timer5 产生溢出，该位只能通过写 1 清零
6	EXF5	Timer5 外部引脚 T5EX 下降沿被检测到的标志位 0 未检测到下降沿 1 检测到外部输入的下降沿，该为只能通过写 1 清零

5:4	-	-
3	EXEN5	Timer5 外部输入事件用作捕获/重装载触发器允许/禁止控制位 0 忽略 T2EX 引脚上的事件 1 检测到外部输入的下降沿时触发一次捕获/重装载
2	TR5	Timer5 工作使能位 0 停止工作 1 开始工作
1	T5_C/T	Timer5 计数来源选择 0 作为定时器, Timer5 计数来源于系统时钟 1 作为计数器, Timer5 计数来源于外部管脚脉冲的下降沿
0	CP/RL	捕获/重装载模式选择 0 16 位带重装载功能的定时器/计数器, 重载事件发生在 Timer5 溢出或 T5EX 输入的下降沿 (当 EXEN5 配为 1) 1 16 位带捕获功能的定时器/计数器, 捕获事件发生在 T5EX 输入的下降沿

10.1.7.33 RCAP5L (addr:0xC6) Time4 重装载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP5L							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP5L	重装载模式下, 该寄存器用于存放用于重装载值的低 8 位

10.1.7.34 RCAP5H (addr:0xC7) Time5 重装载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	RCAP5H							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	RCAP5H	重装载模式下, 该寄存器用于存放用于重装载值的高 8 位

10.1.7.35 TL5 (addr:0xCE) Time5 重装载值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	TL5							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	TL5	写: 写该寄存器会将写入值装载至 Timer5 的低 8 位, 每写一次装载一次 读: 读该寄存器将会读出 Time5 当前的低 8 位值

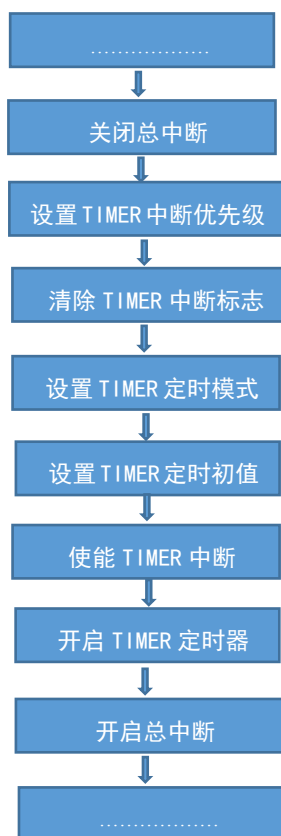
10.1.7.26 TH5 (addr:0xCF) Timer5 重装载值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	TH5							

Reset	0x00
Type	R/W

Bit	Name	Function
7:0	TH5	写：写该寄存器会将写入值装载至 Timer5 的高 8 位，每写一次装载一次 读：读该寄存器将会读出 Timer5 当前的高 8 位值

10.1.8 TIMER 配置流程



TIMER 配置流程图

10.2 CHECKSUM

10.2.1 CHECKSUM 概述

RC6T8X634 系列芯片提供 checksum 功能，用户可通过该功能验证烧录程序的正确性。在 checksum 操作前将 checksum 理论值写入 MTP 空间最后两个地址空间：0x3FFE~0x3FFF。CHKSUM 功能用于计算基于 MTP 空间的地址 0x0000~0x3FFD（以每个地址 8 位数据计算）存储的数据计算得出 16 位 checksum 值。具体计算算法为：将地址 0x0000~0x3FFD 的数据逐一累加（checksum=checksum+Data[i]，i=[0x0000:0x3FFD]）。CHKSUM_START 和 CHKSUM_BUSY 寄存器都可以用来查看当前 CHKSUM 的完成状态。在 checksum 操作完成后读取 CHKSUM_SUM 寄存器的计算值与 MTP 中存储的理论值进行对比校验。

10.2.2 寄存器定义

名字	地址	读写	复位值	描述
CHKSUM_CR	0xFFB0	读写	00H	CHKSUM 控制寄存器
CHKSUM_SUMH	0xFFB1	读写	00H	CHKSUM 结果寄存器高 8 位
CHKSUM_SUML	0xFFB2	读写	00H	CHKSUM 结果寄存器低 8 位

10.2.2.1 CHKSUM_CR (addr:0xFFB0) CHKSUM 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	-						CHKSUM_BUSY	CHKSUM_START
Reset	-						0	0
Type	-						R/W	R/W

Bit	Name	Function
7:2	-	-
1	CHKSUM_BUSY	CHKSUM 运算状态 1 CHKSUM 模块正在运行 0 CHKSUM 模块没有运行
0	CHKSUM_START	CHKSUM 运算开始, 该位写 1 启动 CHKSUM 计算, 写 0 没有影响, 读始终为 0。

10.2.2.2 CHKSUM_SUMH (addr:0xFFB1) CHKSUM 结果寄存器高 8 位

Bit	7	6	5	4	3	2	1	0
Name	CHKSUM_SUM[15:8]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	CHKSUM_SUM[15:8]	CHKSUM 运算结果高 8 位

10.2.2.3 CHKSUM_SUML (addr:0xFFB2) CHKSUM 结果寄存器低 8 位

Bit	7	6	5	4	3	2	1	0
Name	CHKSUM_SUM[7:0]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	CHKSUM_SUM[7:0]	CHKSUM 运算结果低 8 位

10.3 ADC

10.3.1 ADC 概述

芯片内部集成了一个 12 位高精度，高转换速率的逐次逼近型模数转换器 (SAR ADC) 模块。提供多达 25 路外部输入通道，一路片内电压输入通道。ADC 时钟源是系统时钟，可设置时钟预分频。ADC 提供多种参考电压源可选，提供 ADC 转换结果比较器，可以配置管脚边沿触发采样。

10.3.2 结构框图

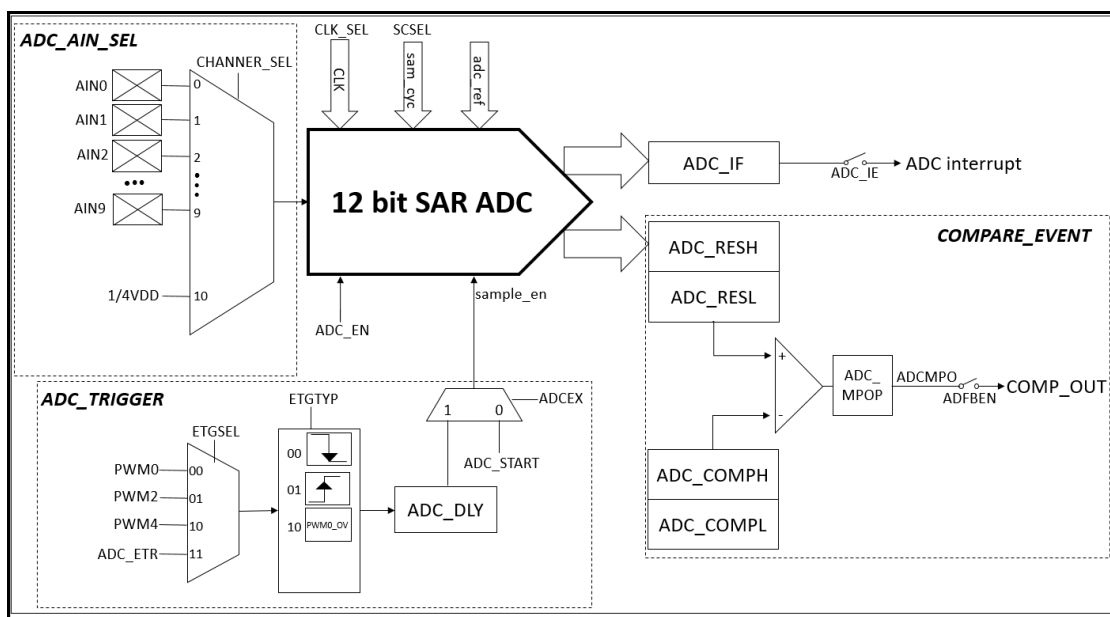


图 10-7 ADC 结构框图

10.3.3 ADC 转换时序

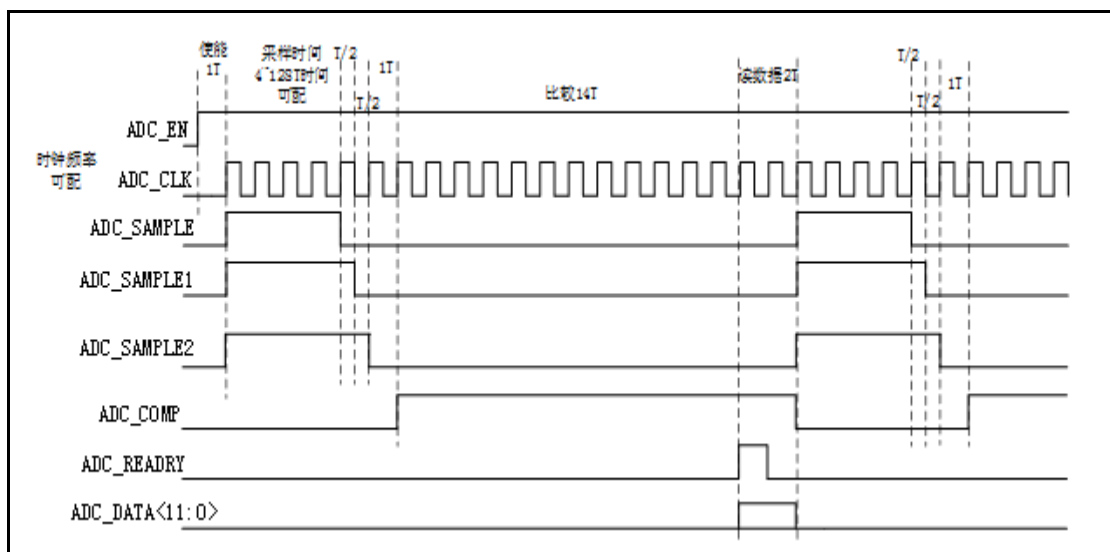


图 10-8 ADC 时序图

10.3.4 寄存器定义

名字	地址	读写	复位值	描述
ADC_CR0	0xE8	读写	00H	ADC 转换控制寄存器 0
ADC_CR1	0xE9	读写	01H	ADC 转换控制寄存器 1
ADC_CR2	0xEA	读写	03H	ADC 转换控制寄存器 2
ADC_CHEL	0xEB	读写	19H	ADC 模拟量输入通道选择寄存器
ADC_CON	0xEC	读写	00H	ADC 配置寄存器
ADC_DLY	0xED	读写	F0H	ADC 触发延迟配置寄存器
ADC_RESL	0xEE	读	00H	ADC 转换结果低位寄存器
ADC_RES	0xEF	读	00H	ADC 转换结果高位寄存器
ADC_COMPL	0xE6	读写	00H	ADC 比较值低 4 位
ADC_COMPH	0xE7	读写	00H	ADC 比较值高 8 位

10.3.4.1 ADC_CR0 (addr:0xE8) ADC 转换控制寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	ADC_EN	CLKSEL2	ADC_START	ADC_IF	ADC_IE	ADCEX	CLKSEL[1:0]	
Reset	0	0	0	0	0	0	0	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Name	Function
7	ADC_EN	ADC 使能位: 0 = ADC 转换电路关闭 1 = ADC 转换电路开启
5	ADC_START	ADC 软件触发采样控制位。ADC 使能后, 该位写 1 开始 ADC 转换, 转换完成后硬件自动将此位清零, ADC 使能之前, 该位无法写入。 0 = 无影响。即使 ADC 已经开始转换工作, 写 0 也不会停止 A/D 转换。 1 = 开始 ADC 转换, 转换完成或关闭 ADC 后硬件自动将此位清零。
4	ADC_IF	ADC 转换结束标志。当 ADC 完成一次转换后, 硬件会自动将此位置 1, 并向 CPU 发出中断请求。此标志位必须软件 (写 1) 清零或复位清零。
3	ADC_IE	ADC 中断使能: 0 = 关闭 ADC 中断 1 = 使能 ADC 中断 默认值: 0
2	ADCEX	该位决定启动 ADC 的触发条件: 0 = 软件触发 1 = 硬件触发 默认值: 0
6、1:0	CLKSEL2: CLKSEL[1:0]	ADC 时钟选择 (以 VCO 为模块工作时钟 clk) 000 工作时钟 clk 的 4 分频 001 工作时钟 clk 的 8 分频 010 工作时钟 clk 的 16 分频 011 工作时钟 clk 的 32 分频 100 工作时钟 clk 的 6 分频 101 工作时钟 clk 的 12 分频

		110 工作时钟 clk 的 24 分频 111 工作时钟 clk 的 48 分频 默认值: 000 注意: 修改 CLK_SEL 寄存器配置必须在 ADC_EN 为 0 的时候进行。
--	--	---

10.3.4.2 ADC_CR1 (addr: 0xE9) ADC 转换控制寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	ETGSEL		—	ETGTYP		SCSEL		
Reset	0		—	0	0	0	0	1
Type	R/W		—	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	ETGSEL	外部触发源选择 当 ADCEX 为 1 时, 该位选择外部触发 ADC 的来源 11 ADC_ETR 其他 保留 默认值: 00
5	—	—
4:3	ETGTYP	外部触发信号类型选择 当 ADCEX 置 1 时该位决定响应外部触发的类型 00 下降沿触发 01 上升沿触发 其他 保留 默认值: 00
2:0	SCSEL	ADC 采样时间周期选择寄存器: 000 = 4 个 ADC 时钟周期 001 = 8 个 ADC 时钟周期 010 = 16 个 ADC 时钟周期 011 = 32 个 ADC 时钟周期 100 = 64 个 ADC 时钟周期 101 = 128 个 ADC 时钟周期 默认值: 001 其他值: 保留 片外很高的输入阻抗时, 增加采样时间, 提高转换精度。

10.3.4.3 ADC_CR2 (addr: 0xEA) ADC 转换控制寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	—		CTRL[5:0]					
Reset	—		0	0	0	0	1	1
Type	—		R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	—	—
5:0	CTRL[5:0]	[5] 参考低噪声使能配置 0 正常工作模式 1 参考噪声减低 默认: 0

		[4] 参考测试模式 0 正常模式 1 测试模式 默认值: 0 [3] 参考 buffer 输出电压选择 0 参考 buffer 输出是参考的 2 倍 (选电源电压参考时不适用); 1 参考 buffer 输出是参考的 1 倍; 默认值: 0 [2] 参考 buffer 输入选择 0 选择内部参考 Bandgap 电压; 1 选择外部参考电压 默认值: 0 [1:0] ADC 参考电压选择 00 选择片外电压不通过 buffer, 直接做 ADC 参考电压; 01 保留 10 选择 VDD, 做 ADC 参考电压; 11 选择 buffer 输出做 ADC 参考电压。 默认值: 11
--	--	---

10.3.4.4 ADC_CHSEL (addr:0xEB) ADC 模拟量输入通道选择寄存器

Bit	7	6	5	4	3	2	1	0
Name	-			ADC_CH				
Reset	-			1	1	0	0	1
Type	-			R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:5	-	-
4:0	ADC_CH	ADC 模拟量输入通道选择 ADC_CH<4:0>=00H, 选择输入 AN<0>, P3. 1; ADC_CH<4:0>=01H, 选择输入 AN<1>, P3. 0; ADC_CH<4:0>=02H, 选择输入 AN<2>, P1. 1 ADC_CH<4:0>=03H, 选择输入 AN<3>, P1. 2 ADC_CH<4:0>=04H, 选择输入 AN<4>, P1. 3 ADC_CH<4:0>=05H, 选择输入 AN<5>, P1. 4 ADC_CH<4:0>=06H, 选择输入 AN<6>, P1. 5 ADC_CH<4:0>=07H, 选择输入 AN<7>, P1. 6; ADC_CH<4:0>=08H, 选择输入 AN<8>, P1. 7; ADC_CH<4:0>=09H, 选择输入 AN<9>, P2. 7; ADC_CH<4:0>=0aH, 选择输入 AN<10>, P2. 6 ADC_CH<4:0>=0bH, 选择输入 AN<11>, P2. 5 ADC_CH<4:0>=0cH, 选择输入 AN<12>, P2. 4 ADC_CH<4:0>=0dH, 选择输入 AN<13>, P2. 3 ADC_CH<4:0>=0eH, 选择输入 AN<14>, P2. 2; ADC_CH<4:0>=0fH, 选择输入 AN<15>, P2. 1; ADC_CH<4:0>=10H, 选择输入 AN<16>, P2. 0; ADC_CH<4:0>=11H, 选择输入 AN<17>, P0. 7 ADC_CH<4:0>=12H, 选择输入 AN<18>, P0. 6

		ADC_CH<4:0>=13H, 选择输入 AN<19>, P0. 5 ADC_CH<4:0>=14H, 选择输入 AN<20>, P0. 4 ADC_CH<4:0>=15H, 选择输入 AN<21>, P0. 3 ADC_CH<4:0>=16H, 选择输入 AN<22>, P0. 2 ADC_CH<4:0>=17H, 选择输入 AN<23>, P0. 1 ADC_CH<4:0>=18H, 选择输入 AN<24>, P0. 0 ADC_CH<4:0>=19H, 选择内部 1/4VDD 做为 ADC 输入; default
--	--	---

10.3.4.5 ADC_CON (addr:0xEC) ADC 配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	–	ADCMPOP	ADCM PEN	ADCMPO	CLEAR	–	–	ADC DLY8
Reset	–	0	0	0	0	–	–	0
Type	–	R/W	R/W	R/W	R/W	–	–	R/W

Bit	Name	Function
7	–	–
6	ADCMPOP	ADC 比较器输出极性选择位: 0 = 若 ADC 输出值大于或等于设定的比较值, 则 ADCMP0 为 1 1 = 若 ADC 输出值小于设定的比较值, 则 ADCMP0 为 1 默认值: 0
5	ADCM PEN	ADC 结果比较使能位: 0 = ADC 结果比较功能关闭 1 = ADC 结果比较功能打开 默认值: 0
4	ADCMPO	ADC 比较结果输出位, 每次 AD 转换结束都会更新输出
3	CLEAR	比较结果清零位, 写 1 清
2:1	–	–
0	ADC DLY8	ADC 外部触发延时计数器数值的高 1 位 默认值为 0

10.3.4.6 ADC_DLY (addr:0xED) ADC 触发延迟配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	ADC DLY							
Reset	0xF0							
Type	R/W							

Bit	Name	Function
7:0	ADC DLY	ADC 外部触发启动延迟计数器的低 8 位, 默认值为 F0。

10.3.4.7 ADC_RES_L (addr:0xEE) ADC 转换结果低位寄存器

Bit	7	6	5	4	3	2	1	0
Name	–				ADC_RES_L			
Reset	–				0x00			
Type	–				R			

Bit	Name	Function
-----	------	----------

7:4	—	—
3:0	ADC_RESL	ADC 转换结果低 4 位。

10.3.4.8 ADC_RES (addr:0xEF) ADC 转换结果高位寄存器

Bit	7	6	5	4	3	2	1	0
Name	ADC_RES							
Reset	0x00							
Type	R							

Bit	Name	Function
7:0	ADC_RES	ADC 转换结果高 8 位。

10.3.4.9 ADC_COMPL (addr:0xE6) ADC 比较值低 4 位

Bit	7	6	5	4	3	2	1	0
Name	—				ADC_COMPL			
Reset	—				0x00			
Type	—				R/W			

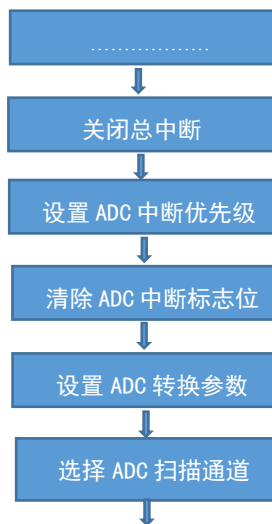
Bit	Name	Function
7:4	—	—
3:0	ADC_COMPL	ADC 比较值低 4 位。

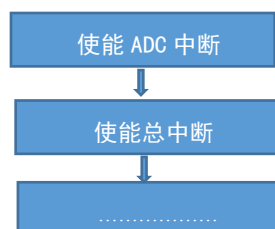
10.3.4.10 ADC_COMP (addr:0xE7) ADC 比较值高 8 位

Bit	7	6	5	4	3	2	1	0
Name	ADC_COMP							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	ADC_COMP	ADC 比较值高 8 位。

10.3.5 ADC 配置流程





ADC 配置流程示意图

10.4 UART0/1

10.4.1 UART 概述

RC6T8X634 系列产品串口通信模块提供 2 个可选择的串口 UART0/1。串口 0 与标准的 8051 串口功能一致。Timer2 仅供串口 0 使用。串口 1 除不可以使用 Timer2 作为波特率发生器之外与串口 0 一致。

每个串口都可以工作在以下所示 4 种模式之一：

模式	同步/异步	波特率时钟	数据位	起始/停止位	第 9 bit 位功能
0	同步	clk 或 clk/12	8	无	无
1	异步	Timer1 或 Timer2 (0)	8	1 起始, 1 停止	无
3	异步	Timer1 或 Timer2 (0)	9	1 起始, 1 停止	0, 1, 奇偶校验

每个串口都可以工作在同步或异步模式。在同步模式下，串口通信模块生成串口时钟，串口工作在半双工模式。在异步模式下，串口工作在全双工模式。在所有模式下，串口通信模块的缓存从保持寄存器中接收数据，以使 UART 能够在软件读取完上一个数据值前就接收即将到来的下一个数据帧。

支持异步半双工模式，通过配置 GPIO 寄存器使用。

本模块选取 VCO 为模块工作时钟 clk。

10.4.2 功能描述

模式 0

串口模式 0 提供同步半双工串行通信。对于串口 0 而言，串行数据由 rxd0_out 发送，由 rxd0_in 接收，txd0 提供数据发送和接收时的移位时钟。对于串口 1 而言，相关引脚信号为 rxd1_out, rxd1_in 和 txd1。

串口模式 0 下的波特率为 clk/12 或 clk 之一，取决于 SM2_0 位的状态（对于串口 1 则对应 SM2_1 位状态）。当 SM2_0 为 0，波特率为 clk/12，当 SM2_0 为 1，波特率为 clk。

模式 0 下与标准 8051 操作一致。在通过 SFR 总线向 SBUF0（或 SBUF1）写入指令时开始数据发送。UART 将数据以选定波特率移位输出，低位在前，直至 8bit 数据全部移位发送范围。

模式 0 下数据接收在相关的 SCON SFR 寄存器的 REN_0（或 REN_1）位置位且 RI_0（或 RI_1）位被清除时开始。移位时钟激活，UART 在每个移位时钟的上升沿将数据进行移位操作，直至 8bit 数据全部接收完成。在 8bit 数据全部移位接收完成后的一个机器周期，RI_0（或 RI_1）位置位，数据接收操作在软件清零 RI 位后结束。

模式 1

模式 1 提供标准异步全双工通信，共使用 10 bit 位：1 个起始位，8 个数据位和 1 个停止位。在接收操作时，停止位缓存在 RB8_0（或 RB8_1）。数据接收和发送都是低位在前。

模式 1 波特率

模式 1 下的波特率发生是 Timer 定时器溢出的功能之一。串口 0 可以使用 Timer1 或者 Timer2 产生波特

率。串口 1 仅能使用 Timer1。两个串口可以在都使用 Timer1 产生波特率时工作在相同波特率，或者在串口 0 使用 Timer2 和串口 1 使用 Timer1 时工作在不同的波特率。

每次 Timer 定时器累计到其最大值 (Timer1: FFh, Timer2: FFFFh) 时, 会发送 1 个时钟至波特率生成电路。该时钟被 16 分频后用来产生波特率。

使用 Timer1 时，通过 SMOD0（或 SMOD1）位可选择是否将 Timer 的翻转率除 2。因此，在使用 Timer1 时，波特率由以下表达式表示：

$$\text{Baud Rate} = \frac{\text{SMODx2}}{32} \times \text{Timer 1 Overflow}$$

其中 SMOD0 和 SMOD1 是 SFR 寄存器 PCON 的控制位。

在使用 Timer2 时，波特率由以下表达式表示：

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow}}{16}$$

在使用 Timer1 作为波特率发生器时，尽管所有计数器模式都可用，最好还是使用 Timer1 的模式 2（8 bit 自动重载）。Timer1 的重载数值缓存在 TH1 寄存器中，使得 Timer1 可表示为如下完整的表达式：

$$\text{Baud Rate} = \frac{2^{\text{SMODx}}}{32} \times \frac{\text{clk}}{12 \times (256 - \text{TH1})}$$

上式分母中的参数“12”可通过设置 SFR 寄存器 CKCON 中的 T1M 位修改为“1”。在已知波特率时（当 T1M 位为 0），可通过如下公式计算获得 TH1 值：

$$TH1 = 256 - \frac{2 \times \text{SMODx} \times \text{clk}}{384 \times \text{Baud Rate}}$$

可以通过使能 Timer1 中断,配置 Timer1 为模式 1,使用 Timer1 中断来初始化一个 16 位软件重载的方式,由 Timer1 产生一个非常低速的串口波特率。

将 Timer2 配置为自动重载模式，并且设置 T2CON SFR 寄存器中的 TCLK 和 RCLK 位为 1，即可使用 Timer2 作为波特率发生器。TCLK 选择 Timer2 作为发送器的波特率产生器；RCLK 选择 Timer2 作为接收器的波特率发生器。Timer2 的 16 位自动重载值缓存在 RCAP2L 和 RCAP2H SFR 寄存器中，使得 Timer2 波特率的公式可表示如下：

$$\text{Baud Rate} = \frac{\text{clk}}{32 \times (65536 - \text{RCAP2H.RCAP2L})}$$

式中的 RCAP2H, RCAP2L 为 RCAP2H 和 RCAP2L 组成的 16 位无符号数。式中的“32”为时钟信号 2 分频和 Timer2 溢出的 16 分频的结果。设置 TCLK 或 RCLK 为 1 将自动造成如下图所示的时钟信号 2 分频, 与 CKCON SFR 寄存器中的 T2M 位所确定的 1 或 12 分频作用一样。

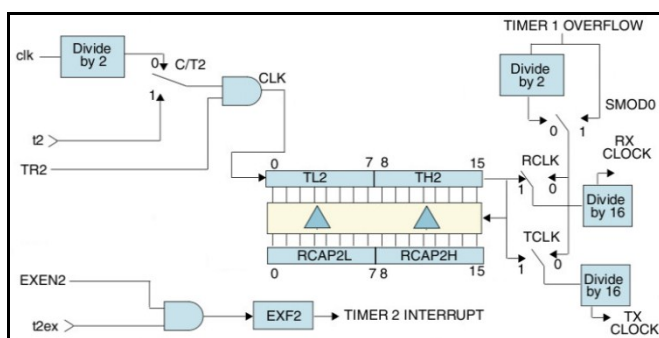


图 10-9

可通过下式获得已知波特率下的 RCAP2H 和 RCAP2L 值:

$$RCAP2H, RCAP2L = 65536 - \frac{clk}{32 \times \text{Baud Rate}}$$

当 RCLK 或 TCLK 任一置位，则 TF2 标识位不会在 Timer2 翻转时置位，并且外部引脚 t2ex 重载触发被禁用。

模式 1 发送

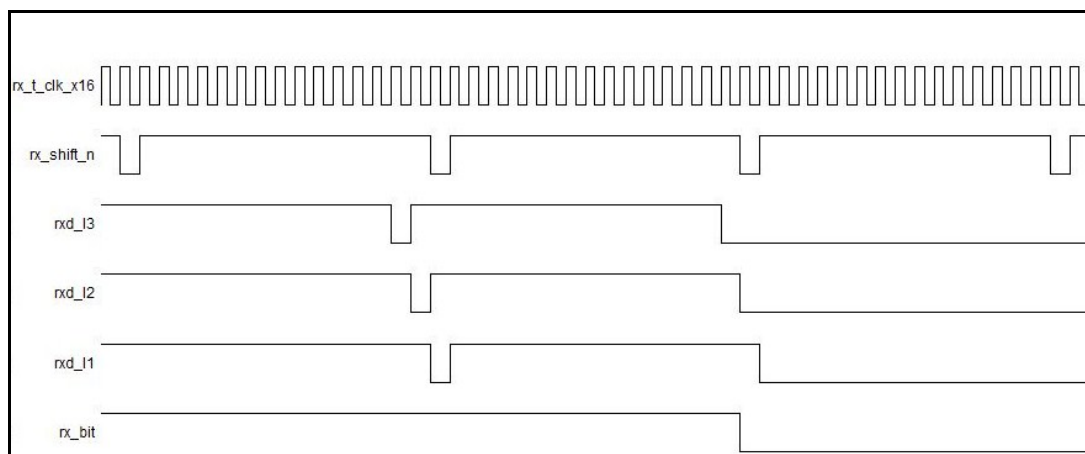
在模式 1 下，UART 在软件写 SBUF0（或 SBUF1）寄存器后的第一个 16 位计数器翻转后开始数据发送。UART 根据以下顺序在 txd0（或 txd1）引脚发送数据：起始位，8 位数据位（低位在前），停止位。TI_0（或 TI_1）位在停止位发送完成后的 1 个时钟周期后被置位。

模式 1 接收

REN_0（或 REN_1）位使能后，接收过程将在检测到 rxd0_in（或 rxd1_in）引脚上的起始位下降沿开始。为达到这个目的，rxd0_in（或 rxd1_in）在任何波特率下都会将每个 bit 位进行 16 次采样。当检测到起始位下降沿时，用来产生接收时钟的 16 位计数器将被复位以使得计数器的翻转与 bit 位的边界对齐一致。

噪声抑制：串口会根据每个 bit 位接收时间段中间三个连续采样，以多数决定法来创建每个接收 bit 位的具体缓存值。这对起始位而言是最正确有效的。如果 rxd0_in（或 rxd1_in）引脚上的下降沿无法通过三个连续采样（低电平）的多数决定法判断确认，则该串口的接收过程将停止并等待对应引脚上的下一个下降沿。

采样判断时序图示例如下：



在停止位接收过程中间，串口将检查以下条件：

RI_0（或 RI_1）=0

若 SM2_0（或 SM2_1）=1，停止位状态为 1

（若 SM2_0（或 SM2_1）=0，停止位状态无关）

如果上述条件满足，串口将接收的数据字节写入 SBUF0（或 SBUF1）寄存器，将停止位状态装载到 RB8_0（或 RB8_1），并置位 RI_0（或 RI_1）位。如果上述条件不满足，接收的数据将被丢弃，SBUF 寄存器和 RB8 位将不被装载写入，RI 比特位也将不被置位。

在停止位接收过程中间过后，串口等待 rxd0_in 或 rxd1_in 引脚的下一个高电平到低电平的状态跳转。

当 Timer1 和 Timer2 工作在 clk/12 频率时钟（默认）下时，模式 1 功能操作上和标准 8051 一致。

模式 3

模式 3 提供异步全双工通信，使用共计 11 个 bit 位：

- 1bit 起始位
- 8bit 数据位
- 1bit 可编程第 9 位
- 1bit 停止位

模式 3 的波特率产生与模式 1 一致。当 Timer1 和 Timer2 使用 clk/12 频率（默认）作为工作时钟时，模式 3 的功能操作与标准 8051 一致。

对于数据发送，第 9 位由 TB8_0（或 TB8_1）位的值决定。

当使用第 9 位作为数据位时，由软件向该位写入第 9 位数据。

当使用第 9 位作奇偶校验位时，可先将待发送数据写入 ACC 寄存器，此时 PSW 寄存器的 P 位即为 ACC 寄存器数据的偶校验位结果。可根据 P 位偶校验结果和约定奇/偶校验模式，决定最终写入 SCON 寄存器 TB8_0（或 TB8_1）位的校验结果，随后再将待发送数据写入 SBUF0/1 中进行发送。

多处理器通信

部分支持多处理器通信模式：在模式 3 下，当 SCON SFR 寄存器中的 SM2 位被置位时（串口 0：SM2_0，串口 1：SM2_1），多处理器通信特征被使能。在多处理器通信模式下，第 9bit 位在接收后被缓存在 RB8_0（或 RB8_1）位，并且在停止位被接收完成后，仅在 RB8_0（或 RB8_1）=1 时激活串口中断。

一个典型的多处理器通信特征的用例就是当一个主机想要发送一段数据给几个从机其中之一。主机首先发送一个地址字节用以识别目标从机。当发送地址字节时，主机将第 9bit 位置位为 1；在发送数据字节时，第 9 bit 位为 0。

当 SM2_0（或 SM2_1）=1 时，不会有从机被其接收的数据字节产生中断。然而，地址字节会中断所有从机以使每个从机能够检测其接收到的地址字节以判定其是否为被选中的地址。软件需要在中断服务程序器件完成地址解码操作。目标地址对应的从机清零其 SM2_0（或 SM2_1）位并准备接收即将到来的数据字节。非目标地址对应的从机保持 SM2_0（或 SM2_1）置位并且忽略即将到来的数据字节。

RC6T8X634 系列产品在多处理器通信模式下不支持硬件从机地址自动匹配。

10.4.3 寄存器定义

名字	地址	读/写	复位值	描述
SCON0	0x98	读/写	00H	串口 0 控制
SBUF0	0x99	读/写	00H	串口 0 缓存
SCON1	0x9A	读/写	00H	串口 1 控制
SBUF1	0x9B	读/写	00H	串口 1 缓存
PCON	0x87	读/写	00H	Power Saving Modes 节能模式控制

10.4.3.1 SCON0（addr:0x98）串口 0 控制

Bit	7	6	5	4	3	2	1	0
Name	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7: 6	SM0_0, SM1_0	串口 0 模式配置 00 模式 0 01 模式 1 10 保留 11 模式 3
5	SM2_0	多处理器通信使能 在模式 3 下，使能多处理器通信功能。此位为 1 时，若接收的第 9bit 位为 0，则 RI_0 不会被激活。 在模式 1 下，此位为 1 时，RI_0 仅会在接收到一个有效的停止位时被激活。 在模式 0 下，根据此位建立波特率：此位为 0 时波特率为 clk/12；此位为 1 时波特率为 clk。

4	REN_0	接收使能 0 不使能 1 使能
3	TB8_0	在模式 3 下，定义被发送的第 9 位数据位状态
2	RB8_0	在模式 3 下，表示接收到的第 9 位数据位状态。 在模式 1 下，表示接收到的停止位 在模式 0 下，无效。
1	TI_0	发送中断标志位，表示发送数据帧已经发送完成。此位必须被软件复位。 在模式 0 下，TI_0 在第 8 位数据发送完时置位。 在其他模式下，TI_0 在停止位被发送时置位。
0	RI_0	接收中断标志位，表示接收串行数据帧已经接收完成。此位必须被软件复位。 在模式 0 下，RI_0 在第 8 位数据接收完时置位。 在模式 1 下，RI_0 在停止位接收完成后置位，受 SM2_0 位影响。 在模式 3 下，RI_0 在 RB8_0 位接收完成时置位。

10.4.3.2 SBUF0 (addr:0x99) 串口 0 缓存

Bit	7	6	5	4	3	2	1	0
Name	SBUF0							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	SBUF0	串口 0 数据缓存

10.4.3.3 SCON1 (addr:0x9A) 串口 1 控制

Bit	7	6	5	4	3	2	1	0
Name	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:6	SM0_1, SM1_1	串口 1 模式配置 00 模式 0 01 模式 1 10 保留 11 模式 3
5	SM2_1	多处理器通信使能 在模式 3 下，使能多处理器通信功能。此位为 1 时，若接收的第 9bit 位为 0，则 RI_1 不会被激活。 在模式 1 下，此位为 1 时，RI_1 仅会在接收到一个有效的停止位时被激活。 在模式 0 下，根据此位建立波特率：此位为 0 时波特率为 clk/12；此位为 1 时波特率为 clk。
4	REN_1	接收使能 0 不使能 1 使能
3	TB8_1	在模式 3 下，定义被发送的第 9 位数据位状态
2		在模式 3 下，表示接收到的第 9 位数据位状态。

	RB8_1	在模式 1 下，表示接收到的停止位 在模式 0 下，无效。
1	TI_1	发送中断标志位，表示发送数据帧已经发送完成。此位必须被软件复位。 在模式 0 下，TI_1 在第 8 位数据发送完时置位。 在其他模式下，TI_1 在停止位被发送时置位。
0	RI_1	接收中断标志位，表示接收串行数据帧已经接收完成。此位必须被软件复位。 在模式 0 下，RI_1 在第 8 位数据接收完时置位。 在模式 1 下，RI_1 在停止位接收完成后置位，受 SM2_1 位影响。 在模式 3 下，RI_1 在 RB8_1 位接收完成时置位。

10.4.3.4 SBUF1 (addr:0x9B) 串口 1 缓存

Bit	7	6	5	4	3	2	1	0
Name	SBUF1							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	SBUF1	串口 1 数据缓存

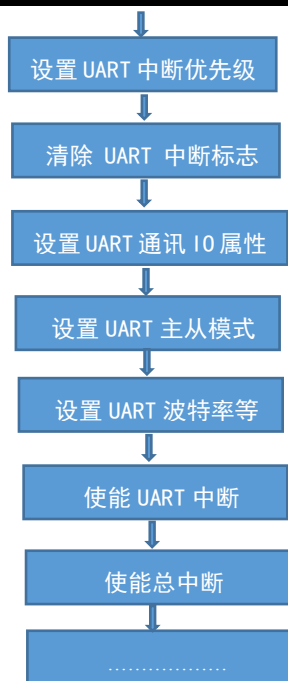
10.4.3.5 PCON (addr:0x87) Power Saving Modes 节能模式控制

Bit	7	6	5	4	3	2	1	0
Name	SMOD0	SMOD1	—				UART0_IE	UART1_IE
Reset	0	0	—				0	0
Type	R/W	R/W	—				R/W	R/W

Bit	Name	Function
7	SMOD0	串口 0 波特率倍增使能（仅针对 Timer1） 0 波特率不倍增（Timer 溢出翻转率除 2） 1 波特率倍增为 2 倍（Timer 溢出翻转率不除 2）
6	SMOD1	串口 1 波特率倍增使能（仅针对 Timer1） 0 波特率不倍增（Timer 溢出翻转率除 2） 1 波特率倍增为 2 倍（Timer 溢出翻转率不除 2）
5:2	—	—
1	UART0_IE	串口 0 中断使能 0 不使能 1 使能
0	UART1_IE	串口 1 中断使能 0 不使能 1 使能

10.4.4 UART 配置流程





UART 配置流程示意图

10.5 I2C

10.5.1 IIC 概述

RC6T8X634 系列芯片提供 I2C 模块，该模块支持芯片与外围 I2C 器件以标准 I2C 协议进行串行数据传输，可以设置为主机或从机模式，具有七位从机地址。支持多主机仲裁以及中断功能。可通过合理配置，设置多种通讯速度。

10.5.2 结构框图

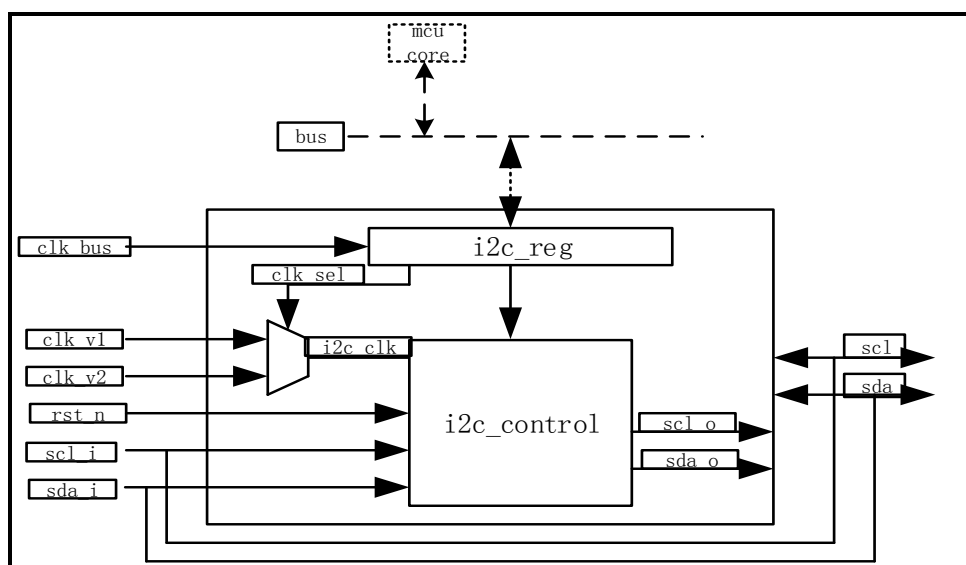


图 10-10 I2C 结构框图

10.5.3 中断

I2C 提供 3 种类型的中断：

- 总线错误中断
- 停止中断
- 传输完成中断

10.5.4 波特率设置

主机模式下，发送时钟来自时钟源的 17 分频。

10.5.5 寄存器定义

名字	地址	读写	复位值	描述
I2C_ADDR	0xA1	读写	01100110	I2C 从机地址寄存器
I2C_CR	0xA2	读写	00000001	I2C 控制寄存器
I2C_STAT	0xA3	读写	00000000	I2C 状态寄存器
I2C_DR	0xA4	读写	00000000	I2C 数据寄存器
I2C_MCR	0xA5	读写	00000000	I2C 主机控制寄存器

10.5.5.1 I2C_ADDR (addr:0xA1) I2C 从机地址寄存器

Bit	7	6	5	4	3	2	1	0
Name	HwAddrEn	Slave Address[6:0]						
Reset	0	0x66						
Type	R/W	R/W						

Bit	Name	Function
7	HwAddrEn	1: 打开地址比较功能 0: 关掉地址比较功能 只用于从模式下。 I2C_ADDR[6:0]为当前 I2C 设备号， HwAddrEn 为 1，收到请求后，会比较收到的地址是否与 Slave Address 一致，如果一致， 则响应请求，不一致则不响应； HwAddrEn 为 0，会响应收到的所有请求
6:0	Slave Address[6:0]	只用于从模式，当前设备的地址

10.5.5.2 I2C_CR (addr:0xA2) I2C 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	I2C IE	–	BusError IE	Stop IE	–	Clk_sel	Enable Master	EnableSlave
Reset	0	–	0	0	–	0	0	1
Type	R/W	–	R/W	R/W	–	R/W	R/W	R/W

Bit	Name	Function
-----	------	----------

7	I2C IE	1 = 打开 I2C 全部中断 0 = 关闭 I2C 全部中断
6	-	-
5	Bus Error IE	1 = 打开 Bus Error 中断 0 = 关闭 Bus Error 中断.
4	Stop IE	1 = 打开结束中断 0 = 关闭结束中断
3	-	-
2	Clk_sel	0 = SCK1 1 = SCK2
1:0	Enable Master & Slave	00 主模式关&从模式关 01 主模式关&从模式开 10 主模式开&从模式关 11 主模式开&从模式开

注意：在 I2C 时钟源切换时，需要先关闭 I2C 使能再切换时钟源，切换完成时钟源后再使能 I2C，切换动作需要分 3 个步骤完成。

10.5.5.3 I2C_STAT (addr:0xA3) I2C 状态寄存器

Bit	7	6	5	4	3	2	1	0
Name	BusError	Lost Arb	StopStatus	ACK	Address	Transmit	LRB	TransComplete
Reset	0	0	0	0	0	0	0	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	Bus Error (状态位)	只用于主模式，数据传送过程中检测到总线上有开始或结束条件时置 1。 只能通过写 0 清除。 注意：若发生了 Bus Error，则需要配置成非主机模式或关掉 I2C。
6	Lost Arb (状态位)	只用于主模式，失去对总线的控制权时置 1 可以通过写 0 清除 每次检测到开始信号都会自动清零。 注意：若主机失去对总线控制，则需要配置成非主机模式或关掉 I2C。
5	Stop Status (状态位)	检测到结束状态时置 1 只能通过写 0 清除
4	ACK (控制位)	1 = 发送 ack 0 = 不发送 ack (nack)
3	Address (状态位)	收到一个地址时置 1 只能通过写 0 清除
2	Transmit (控制位)	1 = 发送模式 0 = 接收模式
1	LRB (状态位)	1 = 最后收到的 bit 是 NACK 0 = 最后收到的 bit 是 ACK 写 0 清除或者检测到 START 信号清除。
0	Trans Complete (状态位)	单字节方式： 1：接收完成 发送模式：8bits 数据传送完成并收到响应 (ACK 或者 NACK)。 接收模式：8bits 数据接收完成。

		写 0 清除或者检测到 START 信号清除。
--	--	-------------------------

10.5.5.4 I2C_DR (addr:0xA4) I2C 数据寄存器

Bit	7	6	5	4	3	2	1	0
Name	Data							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	Data	主从模式接收，保存收到的数据，只读； 主模式产生开始信号前，需写入要发送到总线上的地址； 主从模式开始发送数据前，需写入要发送到客户端的数据；

10.5.5.5 I2C_MCR (addr:0xA5) I2C 主机控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	–	–	–	–	Bus Busy	Master Mode	Restart Gen	StartGen
Reset	–	–	–	–	0	0	0	0
Type	–	–	–	–	R	R	R/W	R/W

Bit	Name	Function
7:4	–	–
3	Bus Busy	检测到开始信号，状态置为 1 检测到结束信号，状态置为 0
2	Master Mode	产生开始信号，状态置为 1 产生结束信号，状态置为 0
1	Restart Gen	1 传送过程中收到响应为 NACK，重启传送过程，重新传送
0	Start Gen	1 产生开始信号并发送地址到 i2c 总线上 传送完成后清零

10.5.6 应用描述

I2C 支持主从模式下的数据发送和接收。基本数据传输方式如下：

主器件产生传输用的时钟（SCL）信号，开始信号（START）和结束信号（STOP）。

数据（SDA）必须在时钟的低电平时改变，并在高电平时保持。

SCL 为高时，检测到 SDA 上有由高到低的跳变，为 START；

SCL 为高时，检测到 SDA 上有由低到高的跳变，为 STOP。

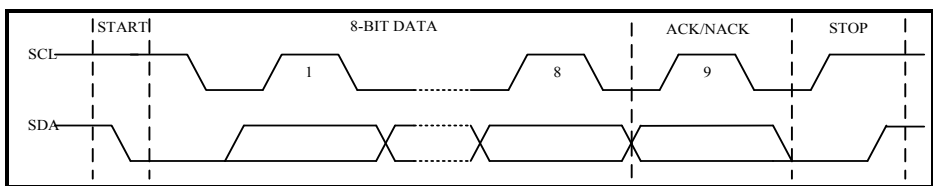


图 10-11

10.5.6.1 从模式 (slave)

从模式下，I2C 的时钟必须配置大于两倍的 I2C 的波特率。

从模式下，会持续监听总线上是否有 START 信号。当监听到 START，会收到 8bit 的数据，其中包括 7 bit 的 address 和 1bit 的 R/W 标志，从器件会根据收到的地址来确认是否响应主器件的读写请求。

如果地址正确，确认响应主器件的请求，从器件会根据 R/W 标志确认是传输数据还是接收数据，过程如图所示

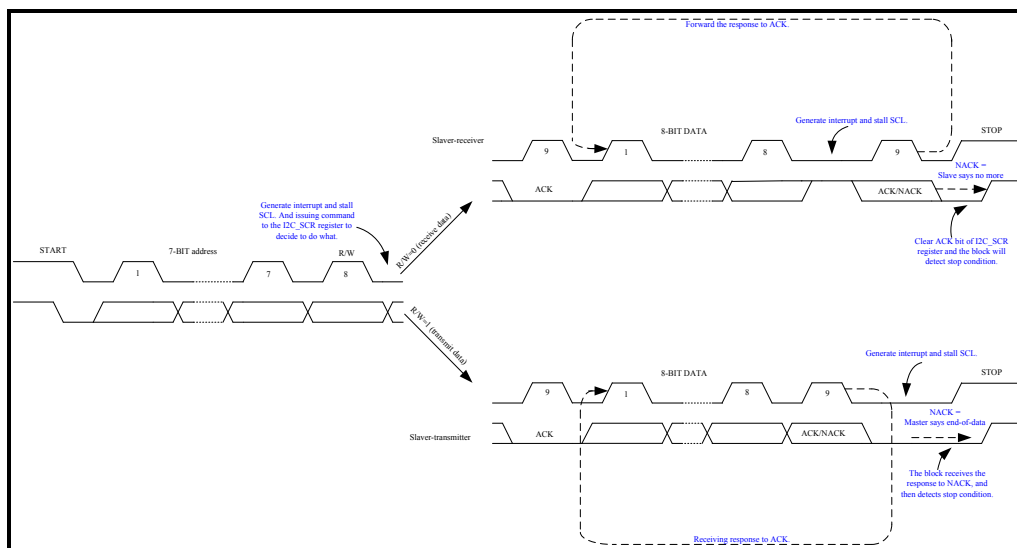


图 10-12

从器件成功发送 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态.
- 2) 打开从模式 (I2C_CR)，处于监听状态.
收到 8-bit data (slave address) 后产生中断.
- 3) 将要发送的数据写入 I2C_DR
- 4) ACK bit 和 transmit bit 置 1 (I2C_STAT) .
- 5) Byte Complete bit 置 1 (I2C_STAT) .
收到 8-bit data 和响应后产生中断.
- 6) 检查 LRB bit (I2C_STAT) .
重复步骤 3~6，可以发送多 byte 数据

从器件成功接收 1byte 数据过程如下：

- 1) 确认寄存器都在初始状态.
- 2) 打开从模式 (I2C_CR)，处于监听状态.
收到 8-bit data (slave address) 后产生中断.
- 3) ACK bit 置 1, transmit bit 清 0 (I2C_STAT) .
- 4) Byte Complete bit 置 1 (I2C_STAT) .
收到 8-bit data 后产生中断..
- 5) ACK bit 清 0 (I2C_STAT) .
重复步骤 3~4，可以接收多 byte 数据

10.5.6.2 主模式 (master)

主模式下，发起一个传送请求前，主设备必须先判断总线是否处于空闲状态。当总线上有设备在传输数据时，总线忙状态位 (Bus Busy) 会一直置为 1，直到检测到一个 STOP 信号，此时，当前设备获得总线使用权，启动一个读/写过程。

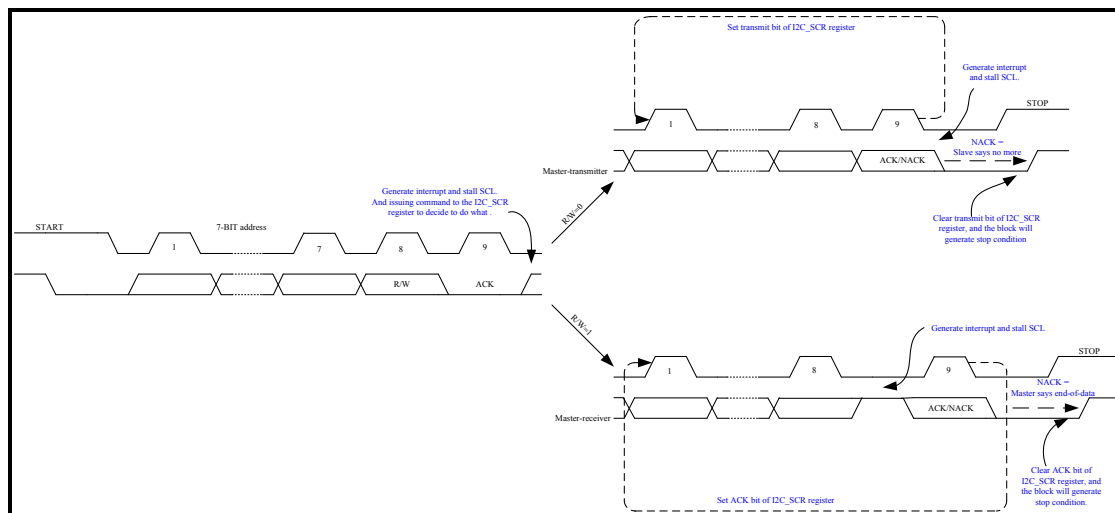


图 10-13

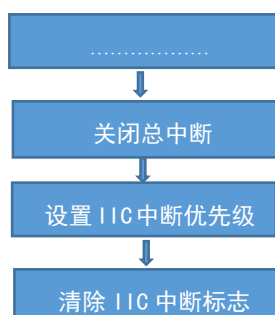
主器件成功发送 1byte 数据过程如下:

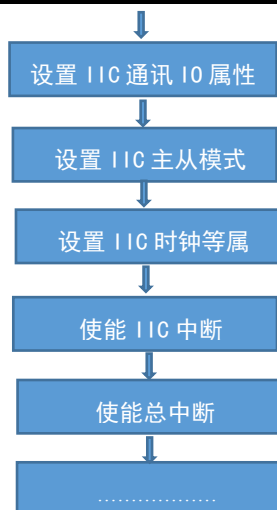
- 1) 确认寄存器都在初始状态.
- 2) 打开主模式 (I2C_CR).
- 3) 将数据(slave address+W)写入 I2C_DR.
- 4) Start Gen bit 置 1 (I2C_MCR).
主设备发送完 8bit 数据并收到 ACK, 产生中断.
- 5) 将要发送数据写入 I2C_DR.
- 6) Transmit bit 置 1 (I2C_STAT).
主设备发送完 8bit 数据并收到 ACK, 产生中断.
- 7) 发送完成, Transmit bit 清零 (I2C_STAT register).
重复步骤 5~6, 可以发送多 byte 数据.

主器件成功接收 1byte 数据过程如下:

- 1) 确认寄存器都在初始状态.
- 2) 打开主模式 (I2C_CR).
- 3) 将数据(slave address+W)写入 I2C_DR.
- 4) Start Gen bit 置 1 (I2C_MCR).
主设备发送完 8bit 数据并收到 ACK, 产生中断.
- 5) Transmit bit 清 0 (I2C_STAT).
主设备收到 8bit 数据, 产生中断.
- 6) 如果需要接收更多数据, ACK bit 置 1, 接收完成 ACK bit 置 0 .
重复步骤 5~6, 能接收多 byte 数据.

10.5.7 IIC 配置流程





IIC 配置流程示意图

10.6 CSD

10.6.1 功能概述

电容检测模块实现了通道转换电路，包括电流源、电压跟踪滤波器，以及时钟产生、伪随机序列生成等。最大支持 25 个触摸按键通道。支持防水功能。

10.6.2 I/O 模式设置

为了使能电容检测功能，被测电容所在管脚必须设置成电容检测模式。

10.6.3 触摸调试

CSD 调试过程可阅读 R-Link Pro III 使用说明文档。

10.7 PI_MUX

10.7.1 功能概述

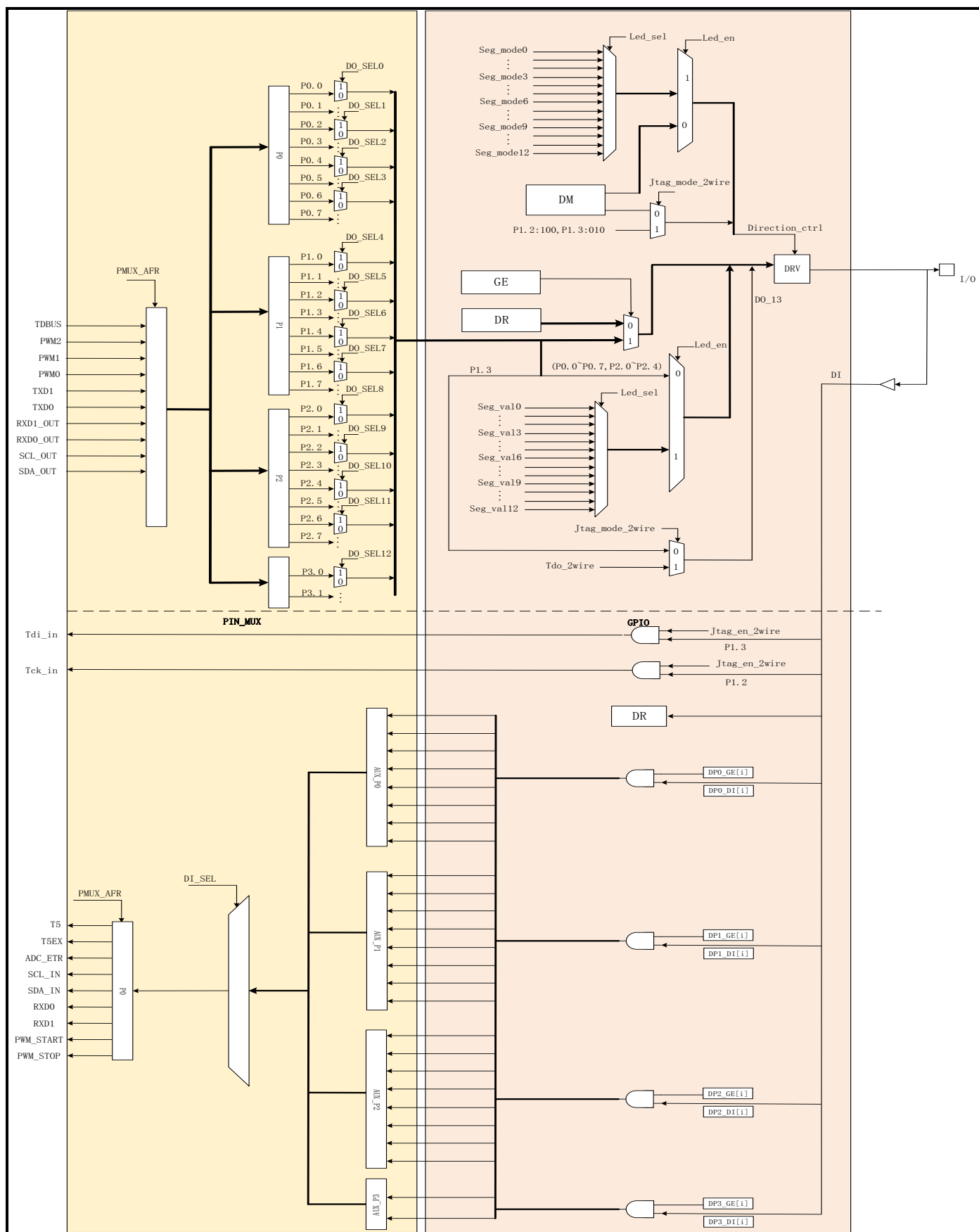
芯片支持 UART、I2C、TIMER 等外设数字输入输出功能 IO 全映射。注意，P1.2、P1.3 默认为调试功能脚，其他脚默认为 GPIO。

如图所示，功能框架图分为 GPIO 和 PIN_MUX 两部分；该文档主要描述 PIN_MUX 部分的内容，PAD 的方向由 DM 寄存器决定，虚线以上为输出，虚线以下为输入；

当 GPIO 作为输出时，并且 GE 关闭时，PAD 的输出由 DR 寄存器的值决定；当 GE 打开时，PAD 的输出由系统外设决定，外设的使能可通过寄存器 PMUX_AFR 配置。输出的外设信号可通过配置寄存器 DO_SELx 进行选择由任意 PAD 输出；JTAG 的 TDO 只能由 P1.3 输出；

当 GPIO 作为输入时，PAD 的输入一条支路被写入到 DR 寄存器中，另一条支路流向外设；并且只有当 GE 打

开时，PAD 的值才能输入到外设，外设的输入使能由寄存器 PMUX_AFR 控制。输入的外设信号可通过配置寄存器 DI_SELx 进行选择从任意 PAD 输入到对应的外设；JTAG 的 TCK 和 TDI 只能由 P1.2 和 P1.3 输入。



10.7.2 寄存器定义

寄存器名称	地址	读写	复位值	描述
DI_SEL0	0xFF60	R/W	0x00	TIMER5 外部计数器输入 PAD 选择
DI_SEL1	0xFF61	R/W	0x00	TIMER5 外部捕获输入 PAD 选择
DI_SEL2	0xFF62	R/W	0x00	ADC_ETR 输入 PAD 选择
DI_SEL3	0xFF63	R/W	0x0A	I2C 的时钟输入 PAD 选择
DI_SEL4	0xFF64	R/W	0x0B	I2C 的数据输入 PAD 选择
DI_SEL5	0xFF65	R/W	0x00	UART0 的 RX0 输入 PAD 选择
DI_SEL6	0xFF66	R/W	0x00	UART1 的 RX1 输入 PAD 选择
DI_SEL7	0xFF67	R/W	0x00	PWM 开始的输入 PAD 选择
DI_SEL8	0xFF68	R/W	0x00	PWM 结束的输入 PAD 选择
PMUX_AFR	0xFF6F	R/W	0x04	复用功能使能寄存器
DO_SEL0	0xFF70	R/W	0x00	P0.0, P0.1 输出数字功能选择
DO_SEL1	0xFF71	R/W	0x00	P0.2, P0.3 输出数字功能选择
DO_SEL2	0xFF72	R/W	0x00	P0.4, P0.5 输出数字功能选择
DO_SEL3	0xFF73	R/W	0x00	P0.6, P0.7 输出数字功能选择
DO_SEL4	0xFF74	R/W	0x00	P1.0, P1.1 输出数字功能选择
DO_SEL5	0xFF75	R/W	0x01	P1.2, P1.3 输出数字功能选择
DO_SEL6	0xFF76	R/W	0x00	P1.4, P1.5 输出数字功能选择
DO_SEL7	0xFF77	R/W	0x00	P1.6, P1.7 输出数字功能选择
DO_SEL8	0xFF78	R/W	0x00	P2.0, P2.1 输出数字功能选择
DO_SEL9	0xFF79	R/W	0x00	P2.2, P2.3 输出数字功能选择
DO_SEL10	0xFF7A	R/W	0x00	P2.4, P2.5 输出数字功能选择
DO_SEL11	0xFF7B	R/W	0x00	P2.6, P2.7 输出数字功能选择
DO_SEL12	0xFF7C	R/W	0x00	P3.0, P3.1 输出数字功能选择

注：在进行复用功能 PAD 的选择时，不能同时配置不同的外设从相同的 PAD 输入；

在使用 PWM 功能时应该避免配置 PWM 功能输入开启和停止时从相同的 PAD 输出；

10.7.2.1 DI_SEL0 (addr: 0xFF60) TIMER5 外部输入计数器选择输入 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL0				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL0	TIMER5 外部输入计数器选择输入 PAD

	0	P00
	1	P01
	2	P02
	3	P03
	4	P04
	5	P05
	6	P06
	7	P07
	8	P10
	9	P11
	10	P12
	11	P13
	12	P14
	13	P15
	14	P16
	15	P17
	16	P20
	17	P21
	18	P22
	19	P23
	20	P24
	21	P25
	22	P26
	23	P27
	24	P30
	25	P31
	其他	保留

10.7.2.2 DI_SEL1 (addr: 0xFF61) TIMER5 外部捕获 T5EX 输入选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL1				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL1	TIMER5 外部捕获 T5EX 选择输入 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11

		10 P12
		11 P13
		12 P14
		13 P15
		14 P16
		15 P17
		16 P20
		17 P21
		18 P22
		19 P23
		20 P24
		21 P25
		22 P26
		23 P27
		24 P30
		25 P31
		其他 保留

10.7.2.3 DI_SEL2 (addr: 0xFF62) ADC_ETR 输入引脚选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL2				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL2	ADC_ETR 输入引脚选择 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12 11 P13 12 P14 13 P15 14 P16 15 P17 16 P20 17 P21 18 P22 19 P23

		20 P24
		21 P25
		22 P26
		23 P27
		24 P30
		25 P31
		其他 保留

10.7.2.4 DI_SEL3 (addr: 0xFF63) I2C 时钟输入引脚选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL3				
Reset	-			0x0A				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL3	I2C 时钟输入引脚选择 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12 11 P13 12 P14 13 P15 14 P16 15 P17 16 P20 17 P21 18 P22 19 P23 20 P24 21 P25 22 P26 23 P27 24 P30 25 P31 其他 保留

10.7.2.5 DI_SEL4 (addr: 0xFF64) I2C 数据输入引脚选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL4				
Reset	-			0x0B				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL4	I2C 数据输入引脚选择 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12 11 P13 12 P14 13 P15 14 P16 15 P17 16 P20 17 P21 18 P22 19 P23 20 P24 21 P25 22 P26 23 P27 24 P30 25 P31 其他 保留

10.7.2.6 DI_SEL5 (addr: 0xFF65) UART0 输入 RX0 选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL5				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL5	UART0 输入 RX0 选择 PAD 0 P00

		1 P01
		2 P02
		3 P03
		4 P04
		5 P05
		6 P06
		7 P07
		8 P10
		9 P11
		10 P12
		11 P13
		12 P14
		13 P15
		14 P16
		15 P17
		16 P20
		17 P21
		18 P22
		19 P23
		20 P24
		21 P25
		22 P26
		23 P27
		24 P30
		25 P31
		其他 保留

10.7.2.7 DI_SEL6 (addr: 0xFF66) UART1 输入 RX1 选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL6				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL6	UART1 输入 RX1 选择 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12

		11 P13
		12 P14
		13 P15
		14 P16
		15 P17
		16 P20
		17 P21
		18 P22
		19 P23
		20 P24
		21 P25
		22 P26
		23 P27
		24 P30
		25 P31
		其他 保留

10.7.2.8 DI_SEL7 (addr: 0xFF67) PWM 启停输入引脚 PIN0 选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL7				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL7	PWM 启停输入引脚 PIN0 选择 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12 11 P13 12 P14 13 P15 14 P16 15 P17 16 P20 17 P21 18 P22 19 P23 20 P24

		21 P25
		22 P26
		23 P27
		24 P30
		25 P31
		其他 保留

10.7.2.9 DI_SEL8 (addr: 0xFF68) PWM 启停输入引脚 PIN1 选择 PAD

Bit	7	6	5	4	3	2	1	0
Name	-			DI_SEL8				
Reset	-			0				
Type	-			R/W				

Bit	Name	Function
7:5	-	-
4:0	DI_SEL8	TIMER5 外部捕获 T5EX 选择输入 PAD 0 P00 1 P01 2 P02 3 P03 4 P04 5 P05 6 P06 7 P07 8 P10 9 P11 10 P12 11 P13 12 P14 13 P15 14 P16 15 P17 16 P20 17 P21 18 P22 19 P23 20 P24 21 P25 22 P26 23 P27 24 P30 25 P31 其他 保留

10.7.2.10 PMUX_AFR (addr: 0xFF6F) I/O 复用功能使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	UART1_ASYN_HALF	UART0_ASYN_HALF	-		RD_REV_WR_EN	I2C_EN	UART1_EN	UART0_EN
Reset	0	0	-		0	1	0	0
Type	R/W	R/W	-		R/W	R/W	R/W	R/W

Bit	Name	Function
7	UART1_ASYN_HALF	用于 UART1 配置异步半双工模式时分时控制发送、接收 0 UART1 串口异步半双工接收 1 UART1 串口异步半双工发送 注：串口使用异步半双工时，串口配置为模式 1 或 3，接收和发送引脚配置为同一个，IO 设置为开漏模式。若串口工作在全双工模式，该位必须配置为 0
6	UART0_ASYN_HALF	用于 UART0 配置异步半双工模式时分时控制发送、接收 0 UART0 串口异步半双工接收 1 UART0 串口异步半双工发送 注：串口使用异步半双工时，串口配置为模式 1 或 3，接收和发送引脚配置为同一个，IO 设置为开漏模式。若串口工作在全双工模式，该位必须配置为 0
5: 4	-	-
3	RD_REV_WR_EN	0 进行位操作时不使能 1 进行位操作时使能 注：该使能位用于避免进行端口操作时引起其他端口的位也发生变化，建议使能该功能。
2	I2C_EN	I2C_SCL 输出管脚复用使能寄存器 0 I2C_SCL 管脚复用不使能 1 I2C_SCL 管脚复用使能
1:0	UART1_EN	UART1 管脚复用使能寄存器 0 UART1 管脚复用不使能 1 UART1 管脚复用使能
	UART0_EN	UART0 管脚复用使能寄存器 0 UART0 管脚复用不使能 1 UART0 管脚复用使能

10.7.2.11 DO_SEL0 (addr: 0xFF70) 数字输出 PAD 选择寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL01				DO_SEL00			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL01[3:0]	P0.1 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWMO 0111 PWM1

		1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL00[3:0]	P0.0 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.12 DO_SEL1 (addr: 0xFF71) 数字输出 PAD 选择寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL03				DO_SEL02			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL03[3:0]	P0.3 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL02[3:0]	P0.2 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS

		其他	保留
--	--	----	----

10.7.2.13 DO_SEL2 (addr: 0xFF72) 数字输出 PAD 选择寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL05				DO_SEL04			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL05[3:0]	P0.5 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT
		0100 UART0_TXD
		0101 UART1_TXD
		0110 PWM0
		0111 PWM1
		1000 PWM2
		1111 TDBUS
		其他 保留
3:0	DO_SEL04[3:0]	P0.4 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT
		0100 UART0_TXD
		0101 UART1_TXD
		0110 PWM0
		0111 PWM1
		1000 PWM2
		1111 TDBUS
		其他 保留

10.7.2.14 DO_SEL3 (addr: 0xFF73) 数字输出 PAD 选择寄存器 3

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL07				DO_SEL06			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4		P0.7 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT

	DO_SEL07[3:0]	0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL06[3:0]	P0.6 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.15 DO_SEL4 (addr: 0xFF74) 数据输出 PAD 选择寄存器 4

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL11				DO_SEL10			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL11[3:0]	P1.1 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL10[3:0]	P1.0 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD

		0110	PWM0
		0111	PWM1
		1000	PWM2
		1111	TDBUS
		其他	保留

10.7.2.16 DO_SEL5 (addr: 0xFF75) 数据输出 PAD 选择寄存器 5

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL13				DO_SEL12			
Reset	0				0x01			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL13[3:0]	P1.3 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL12[3:0]	P1.2 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.17 DO_SEL6 (addr: 0xFF76) 数据输出 PAD 选择寄存器 6

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL15				DO_SEL14			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4		P1.5 输出数字功能选择

		0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	D0_SEL14[3:0]	P1.4 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.18 D0_SEL7 (addr: 0xFF77) 数据输出 PAD 选择寄存器 7

Bit	7	6	5	4	3	2	1	0
Name	D0_SEL17				D0_SEL16			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	D0_SEL17[3:0]	P1.7 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0		P1.6 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT

	DO_SEL16[3:0]	0010	UART0_RXD_OUT
		0011	UART1_RXD_OUT
		0100	UART0_TXD
		0101	UART1_TXD
		0110	PWM0
		0111	PWM1
		1000	PWM2
		1111	TDBUS
		其他	保留

10.7.2.19 DO_SEL8 (addr: 0xFF78) 数据输出 PAD 选择寄存器 8

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL21				DO_SEL20			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL21[3:0]	P2.1 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL20[3:0]	P2.0 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.20 DO_SEL9 (addr: 0xFF79) 数据输出 PAD 选择寄存器 9

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL23				DO_SEL22			
Reset	0				0			

Type	R/W		R/W
------	-----	--	-----

Bit	Name	Function
7:4	DO_SEL23[3:0]	P2. 3 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT
		0100 UART0_TXD
		0101 UART1_TXD
		0110 PWM0
		0111 PWM1
		1000 PWM2
		1111 TDBUS
		其他 保留
3:0	DO_SEL22[3:0]	P2. 2 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT
		0100 UART0_TXD
		0101 UART1_TXD
		0110 PWM0
		0111 PWM1
		1000 PWM2
		1111 TDBUS
		其他 保留

10. 7. 2. 21 DO_SEL10 (addr: 0xFF7A) 数据输出 PAD 选择寄存器 10

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL25				DO_SEL24			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL25[3:0]	P2. 5 输出数字功能选择
		0000 SDA_OUT
		0001 SCL_OUT
		0010 UART0_RXD_OUT
		0011 UART1_RXD_OUT
		0100 UART0_TXD
		0101 UART1_TXD
		0110 PWM0
		0111 PWM1
		1000 PWM2
		1111 TDBUS

		其他	保留
3:0	DO_SEL24[3:0]	P2. 4 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留	

10. 7. 2. 22 DO_SEL11 (addr: 0xFF7B) 数据输出 PAD 选择寄存器 11

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL27				DO_SEL26			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL27[3:0]	P2. 7 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL26[3:0]	P2. 6 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

10.7.2.23 DO_SEL12 (addr: 0xFF7C) 数据输出 PAD 选择寄存器 12

Bit	7	6	5	4	3	2	1	0
Name	DO_SEL31				DO_SEL30			
Reset	0				0			
Type	R/W				R/W			

Bit	Name	Function
7:4	DO_SEL31[3:0]	P3.1 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留
3:0	DO_SEL30[3:0]	P3.0 输出数字功能选择 0000 SDA_OUT 0001 SCL_OUT 0010 UART0_RXD_OUT 0011 UART1_RXD_OUT 0100 UART0_TXD 0101 UART1_TXD 0110 PWM0 0111 PWM1 1000 PWM2 1111 TDBUS 其他 保留

11 系统控制

11.1 复位

11.1.1 复位概述

RC6T8X634 系列芯片有多个内部和外部复位源，复位来源于以下几个部分：

- (1) 看门狗复位
- (2) 上电复位
- (3) 欠压复位，5 档可配
- (4) 软复位
- (5) 序列匹配复位

11.1.2 上电复位

系统上电呈现逐渐上升的曲线形式，需要一定的时间才能达到正常的工作电压。上电复位（POR）在电源电压低于判断电压的时候触发，当电源电压大于触发点的时候 POR 释放。上电复位电路能够保证芯片在上电过程中处于复位状态，芯片上电后能够从一个已知的稳定状态开始运行。

芯片上电过程图如下：

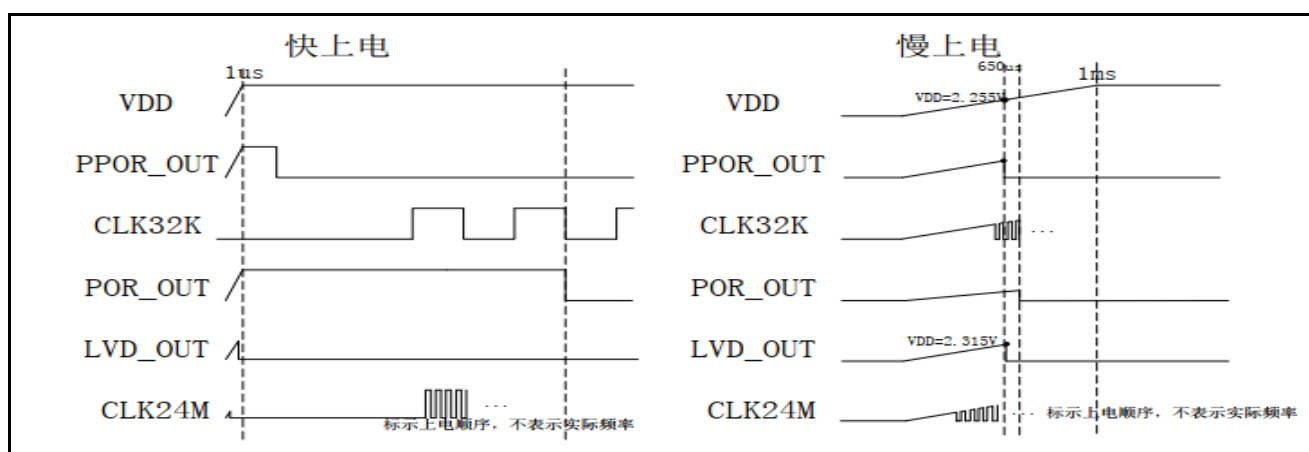


图 11-1

11.1.3 看门狗定时器复位

看门狗定时器负责监控处理器执行指令的情况，通过合适的配置，如果看门狗定时器在特定时间内未被刷新，则可以产生复位信号，防止程序跑飞。看门狗定时器默认是关闭的，用户可通过写 SLPTIM_CR 的 WDT_EN 位使能看门狗定时器。

11.1.4 软复位

RC6T8X634 系列芯片可以在程序控制下执行软复位。通过对 SCR_CFG 寄存器的 RSTREQ 位写 1，CPU 可以发出复位指令。软件复位后，程序将从 B00T 配置指向的位置开始运行。

11.2 电源管理

11.2.1 电源管理概述

RC6T8X634 系列芯片具有两种低功耗模式：SLEEP 模式、DEEPSLEEP 模式。SLEEP 模式系统功耗小于

1. 5mA, DEEPSLEEP 模式系统功耗典型值为 6.5uA (5V 供电)。

- SLEEP 模式：CPU 停止工作，外设可以继续工作。可以通过外设中断或者外部引脚中断唤醒或复位唤醒。
- DEEPSLEEP 模式：深度休眠模式，CPU 停止工作，CPU 和外设时钟都停止。只有 32KHz 时钟工作。
- 这两种模式下程序都停止运行。

两种睡眠模式下各外设工作情况如下表所示：

外设	睡眠模式	深度睡眠模式
CPU	停止	停止
RAM	保持	保持
睡眠定时器	运行	运行
看门狗	运行	运行
定时器	运行	停止
ADC	运行	停止
UART	运行	停止
I2C	运行	停止
内部 24MHz 振荡器	运行	停止
内部 32KHz 振荡器	运行	运行
I/O 口	保持	保持
其他外设	运行	停止
唤醒条件	看门狗复位，所有中断	看门狗复位，引脚中断，睡眠定时器中断

11.2.2 睡眠模式

写 SCR 寄存器 SLEEP=1 且 SLEEPDEEP=0 进入到睡眠模式。该模式下，内部 24MHz 晶振保持工作。同时继续给外设提供时钟，但是 CPU 时钟停止。该模式可以通过复位和中断唤醒。如果使用复位唤醒，那么整个系统会复位而初始化。

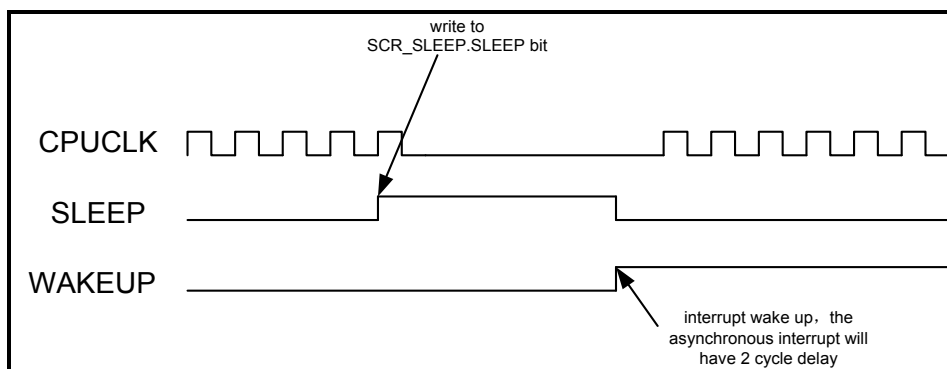


图 11-2 睡眠模式休眠和唤醒时序图

11.2.3 深度睡眠模式

深度睡眠模式通过写 SCR 的 SLEEP=1 且 SLEEPDEEP=1 进入。该模式下，24MHz 主振荡器停止工作，32 KHz 低功耗振荡器继续工作。系统时钟和外设时钟停止，但是睡眠定时器和看门狗继续工作。

注意：如果使用的外部晶体振荡器做主时钟，当进入深度休眠模式后外部晶体振荡器不会自动关闭，需要先切换到内部主振（IM0）后再进入深度休眠模式。

11.2.4 深度休眠模式唤醒

深度睡眠模式可以通过复位和中断唤醒。复位重新初始化所有的控制寄存器，所以重新工作。振荡器的重新工作需要一定时间的延时。下图描述了深度休眠唤醒的时序。

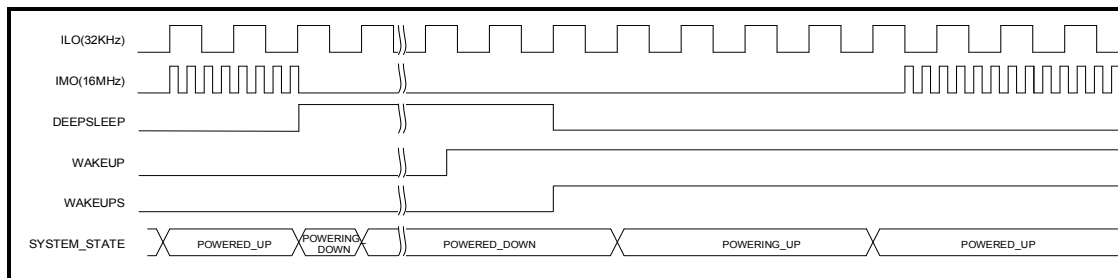


图 11-3 深度休眠唤醒时序

11.3 看门狗

11.3.1 看门狗概述

看门狗定时器是一个可选时钟源的计数器，看门狗主要用于监控系统，避免 CPU 因为外界干扰出现死机。如果软件不能在溢出前刷新看门狗定时器，看门狗将产生内部复位。写 0x38 到 SLPTIM_CLR 寄存器可清除看门狗。RC6T8X634 系列芯片看门狗具备如下特性：

- 16 位睡眠定时器，可用作定时中断
- 2 位看门狗定时器，计数到 3 后溢出
- 看门狗定时器计数时间可以配置为 4/8/16/32/256/512/1024/2048ms
- 看门狗定时器计数时间可自由配置
- 看门狗定时器有单独中断，可用于单独定时

11.3.2 寄存器定义

名字	地址	读写	复位值	描述
SLPTIM_CR	0xF8	读写	00000100	睡眠计数器控制寄存器
SLPTIM_SR	0xF9	读写	00000000	睡眠计数状态
SLPTIM_CLR	0xFA	写	00000000	看门狗清除寄存器
SLPTIM_WDT	0xFB	读写	00000000	看门狗计数器状态
SLPTIM_PRDL	0xFE	读写	00000000	睡眠计数器溢出值低 8 位
SLPTIM_PRDH	0xFF	读写	00000000	睡眠计数器溢出值高 3 位

11.3.2.1 SLPTIM_CR (addr:0xF8) 睡眠计数器控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	SLPIE	—	WDTEN	X32K_SEL	SLEEPDIS	SLPINTS[2:0]		
Reset	0	—	0	0	0	1	0	0
Type	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	SLPIE	0 = 睡眠定时器中断禁止 1 = 睡眠定时器中断使能

6	–	–
5	WDTEN	0 = 看门狗定时器禁止 1 = 看门狗定时器使能
4	X32K_SEL	32K 时钟源选择: 0 = 32K 时钟使用内部振荡器 1 = 32K 时钟使用 XTAL 晶振输入 注: 当检测到外部 RTC 时钟停振后该位会硬件清零, 使用内部 32K 时钟
3	SLEEPDIS	0 = 使能睡眠定时器 1 = 禁止睡眠定时器
2:0	SLPINTS[2:0]	睡眠定时器溢出时间: 000 4ms 001 8ms 010 16ms 011 32ms 100 256ms 101 512ms 110 1024ms 111 2048ms 备注: 实际定时时间会比上面描述的时间多一个 32K cycle 即 30us。

11.3.2.2 SLPTIM_SR (addr:0xF9) 睡眠计数状态

Bit	7	6	5	4	3	2	1	0
Name	SLPEV	–			RSV1	–		SLP_DFT
Reset	0	–			0	–		0
Type	R/W	–			R	–		R/W

Bit	Name	Function
7	SLPEV	0 = 睡眠计数器没有溢出 1 = 睡眠计数器溢出 写 0 清除该位。
6:4	–	–
3	RSV1	该保留位只能写 0, 读为 0。
2:1	–	–
0	SLP_DFT	0 = 睡眠计数器正常模式 1 = 睡眠计数器 DFT 模式 睡眠定时器在 DFT 模式下采用系统时钟进行定时。

11.3.2.3 SLPTIM_CLR (addr:0xFA) 看门狗清除寄存器

Bit	7	6	5	4	3	2	1	0
Name	SLPTIM_CLR							
Reset	–							
Type	WO							

Bit	Name	Function
7:0	SLPTIM_CLR	写 0x38 到该寄存器清除看门狗定时器, 同时会清除睡眠定时器。读该寄存器得到 0。

11.3.2.4 SLPTIM_WDT (addr:0xFB) 看门狗计数器状态

Bit	7	6	5	4	3	2	1	0
Name	–	WDTOV	WDTCNTR		–			
Reset	–	0	0		–			
Type	–	R0	R/W		–			

Bit	Name	Function
7	N/A	保留位, 读 0
6	WDTOV	看门狗溢出标志: 0 = 看门狗没有溢出 1 = 看门狗溢出
5:4	WDTCNTR	比特看门狗计数器计数值
3:0	–	–

11.3.2.5 SLPTIM_PRDL (addr:0xFE) 睡眠计数器溢出值低 8 位

Bit	7	6	5	4	3	2	1	0
Name	PRDRL							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	PRDRL	睡眠定时器溢出值低 8 位。

11.3.2.6 SLPTIM_PRDH (addr:0xFF) 睡眠计数器溢出值高 3 位

Bit	7	6	5	4	3	2	1	0
Name	ACCSEL	–				ACCPDRH		
Reset	0	–				0	0	0
Type	R/W	–				R/W	R/W	R/W

Bit	Name	Function
7	ACCSEL	0 = 选择睡眠定时器溢出值为固定值 1 = 选择睡眠定时器溢出值为 11 位可配置
6:3	–	–
2:0	ACCPDRH	睡眠定时器溢出值高 3 位。

11.3.3 应用描述

对睡眠定时器或者看门狗定时器的操作应遵循以下操作顺序:

1. 配置 SLPTIM_CR 的 SLEEPDIS 位为 1, 关闭睡眠定时器和看门狗定时器的时钟;
2. 进行睡眠定时器和看门狗定时器的时钟选择或者配置寄存器 SLPTIM_PRD_L、SLPTIM_PRD_H 和 SLPTIM_CR;
3. 写寄存器 SLPTIM_CLR 值 0x38, 清除睡眠定时器和看门狗定时器;
4. 配置 SLPTIM_CR 的 SLEEPDIS 位为 0, 使能睡眠定时器和看门狗定时器的时钟;

注意: 在执行完上述操作后芯片不能立即进入休眠模式, 需要等待 $125\mu s + 4 * HCLK_MEM$ 周期后才可进入休眠模式。

11.4 系统模式控制寄存器

11.4.1 寄存器定义

名字	地址	读写	复位值	描述
SCR_CFG	0x91	读写	00001011	系统配置寄存器
SCR_SLEEP	0x92	读写	00000000	休眠寄存器
CHIP_ID0	0x9C	只读	00000110	芯片标识 ID0
CHIP_ID1	0x9D	只读	00100100	芯片标识 ID1

11.4.1.1 SCR_CFG (addr:0x91) 系统配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	CALI_WDR	—	CALI_SYSRSTREQ	—	SEQ_MATCH_EN	RSTREQ	BOOT_SHADOW	BOOT
Reset	0	—	0	—	1	0	1	1
Type	R/W1C	—	R	—	R/W	W	R	R

Bit	Name	Function
7	CALI_WDR	看门狗复位标志： 0 没有看门狗复位（该寄存器的清零可以通过外部、POR、BOR、写 1 来实现） 1 看门狗复位发生 写 1 清清零 CALI_SYSRSTREQ, CALI_WDR
6	—	—
5	CALI_SYSRSTREQ	软件复位标志： 0 没有发生软件复位 1 发生软件复位 注意：该位只能由硬件置位，通过写 1 CALI_WDR 将该位清零
4	—	—
3	SEQ_MATCH_EN	带电复位模块使能 0 不使能 1 使能 使能此位可启用带电复位功能。
2	RSTREQ	软件复位使能： 0 不复位系统 1 复位系统 注意：RSTREQ 会将 CPU 复位，以及 CPU 相关的中断控制器、SRAM 和 MTP 等，外设不会被复位。
1	BOOT_SHADOW	系统复位标志 1 系统发生复位 0 系统复位完成 该标志可以写 1 清除
0	BOOT	系统复位标志： 1 系统发生复位 0 系统复位完成 该标志可以写 1 清，清除该标志时会一同清除掉 BOOT_SHADOW 标志。

11.4.1.2 SCR_SLEEP (addr: 0x92) 休眠寄存器

Bit	7	6	5	4	3	2	1	0
Name	MTP_BUSY	CHKSUM_BUSY					SLEEPDEEP	SLEEP
Reset	0	0					0	0
Type	R	R					R/W	R/W

Bit	Name	Function
7	MTP_BUSY	MTP 编程读模式下 MTP_BUSY 的值表示: 0 MTP 编程完成 1 MTP 编程没有完成
6	CHKSUM_BUSY	CHKSUM 模块 BUSY 的状态表示: 0 CHKSUM 没有运行 1 CHKSUM 正在运行
5:2	—	—
1	SLEEPDEEP	深度休眠模式控制: 0 深度休眠模式关闭 1 深度休眠模式打开
0	SLEEP	休眠模式控制: 0 正常工作模式 1 休眠模式

11.4.1.3 CHIP_ID0 (addr: 0x9C) 芯片 ID0

Bit	7	6	5	4	3	2	1	0
Name	CHIP_ID0							
Reset	0x06							
Type	R							

Bit	Name	Function
7:0	CHIP_ID0	芯片 ID0, 只读

11.4.1.4 CHIP_ID0 (addr: 0x9D) 芯片 ID1

Bit	7	6	5	4	3	2	1	0
Name	CHIP_ID1							
Reset	0x24							
Type	R							

Bit	Name	Function
7:0	CHIP_ID1	芯片 ID1, 只读

11.5 模拟电路控制

11.5.1 寄存器定义

名字	地址	读写	复位值	描述
BG_CR	0xFF80	读写	00000000	Bandgap 使能寄存器
BORLVD_CR	0xFF85	读写	00010001	BORLVD 控制寄存器

BORLVD_STAT	0xFF86	读写	00000000	BORLVD 状态寄存器
IMO_CR	0xFF88	读写	00000001	IMO 控制寄存器
XTAL_CR	0xFF8D	读写	00000000	XTAL 控制寄存器

11.5.1.1 BG_CR (addr:0xFF80) Bandgap 使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	BG_VON_N	—	—	FLT_BYPASS	BG_EN_N
Reset	—	—	—	0	—	—	0	0
Type	—	—	—	R/W	—	—	R/W	R/W

Bit	Name	Function
7:5	—	—
4	BG_VON_N	Bandgap 输出控制位: 0 = 输出 Bandgap 1 = Bandgap 工作但不输出
3:2	—	—
1	FLT_BYPASS	电源滤波控制位: 0 开启电源滤波 1 不开启电源滤波
0	BG_EN_N	Bandgap 使能控制位: 0 = 使能 Bandgap 1 = 关闭 Bandgap

11.5.1.2 BORLVD_CR (addr:0xFF85) BORLVD 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	BOR_VSEL			BOR_EN	LVD_VSEL			LVD_EN
Reset	0			1	R/W			1
Type	R/W			R/W	R/W			R/W

Bit	Name	Function
7:5	BOR_VSEL	BOR 电压点选择 000 2.40V (默认值) 001 2.73V 010 3.10V 011 3.81V 100 4.46V 其它 保留
4	BOR_EN	BOR 控制位: 0 关闭 BOR 1 使能 BOR
3:1	LVD_VSEL	LVD 电压点选择 000 2.55V (默认值) 001 2.98V 010 3.25V 011 4.06V

		100 4.82V 其它 保留
0	LVD_EN	LVD 控制位: 0 关闭 LVD 1 使能 LVD

11.5.1.3 BORLVD_STAT (addr:0xFF86) BORLVD 状态寄存器

Bit	7	6	5	4	3	2	1	0
Name	-			IE_LVD	STAT_BOR	-		STAT_LVD
Reset	-			0	0	-		0
Type	-			R/W	R	-		R

Bit	Name	Function
7:5	-	-
4	IE_LVD	LVD 中断使能位: 0 禁止 LVD 中断 1 使能 LVD 中断
3	STAT_BOR	BOR 输出状态: 0 BOR 没有发生 1 BOR 发生
2:1	-	-
0	STAT_LVD	LVD 输出状态: 0 没有 LVD 事件发生 1 检测到 LVD 事件 该标志位只能读, 不能清除。

11.5.1.4 IMO_CR (addr:0xFF88) IMO 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	FX2_SEL	RESET_CTRL	-	EN_BYPASS	SYSX2	IMO_EN
Reset	-	-	0	0	-	0	0	1
Type	-	-	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7:6	-	-
5	FX2_SEL	2 倍主频选择 0 选择内部 IMO 的 2 分频 1 选择内部时钟 IMO
4	RESET_CTRL	IMO 模拟电路在深睡模式下复位使能控制 0 深睡模式下 IMO 模拟电路不复位 1 深睡模式下 IMO 模拟电路复位
3	-	-
2	EN_BYPASS	IMO_EN=1, EN_BYPASS=0, 2p2v LDO enable IMO_EN=1, EN_BYPASS=1, LDO disable, vd5 bypass to out IMO_EN=0, EN_BYPASS=0, LDO disable, vd5 bypass disable IMO_EN=0, EN_BYPASS=1, LDO disable, vd5 bypass to out 默认值: 0

1	SYSX2	0 选择系统时钟 2 分频 1 选择系统时钟不分频，只用于当系统时钟源为外部晶体振荡器时钟
0	IMO_EN	写该寄存器 0 使能 IMO 1 关闭 IMO 读该寄存器 0 IMO 关闭 1 IMO 使能 备注：关闭 IMO 需要分两个步骤，先写一次 IMOCR 将系统时钟源切换成外部时钟并保持 IMO 使能，再写一次 IMOCR 将 IMO 关闭并保持外部时钟作为系统时钟。

12 电气特性

12.1 绝对最大额定值

符号	参数	最小值	最大值	单位
$V_{DD}-V_{SS}$	供电电压	-0.3	5.5	V
V_{IN}	引脚输入电压	$V_{SS}-0.3$	$V_{DD}+0.3$	V
TSTG	存储器温度	-	-65~150	°C
TJ	最大结温	-	150	°C

12.2 推荐工作条件

符号	参数	最小值	最大值	单位	系统时钟频率
Vdd	工作电压	4.5	5.5	V	12MHz
Vdd	工作电压	2.6	5.5	V	小于 12MHz
T_A	环境温度	-40	85	°C	-

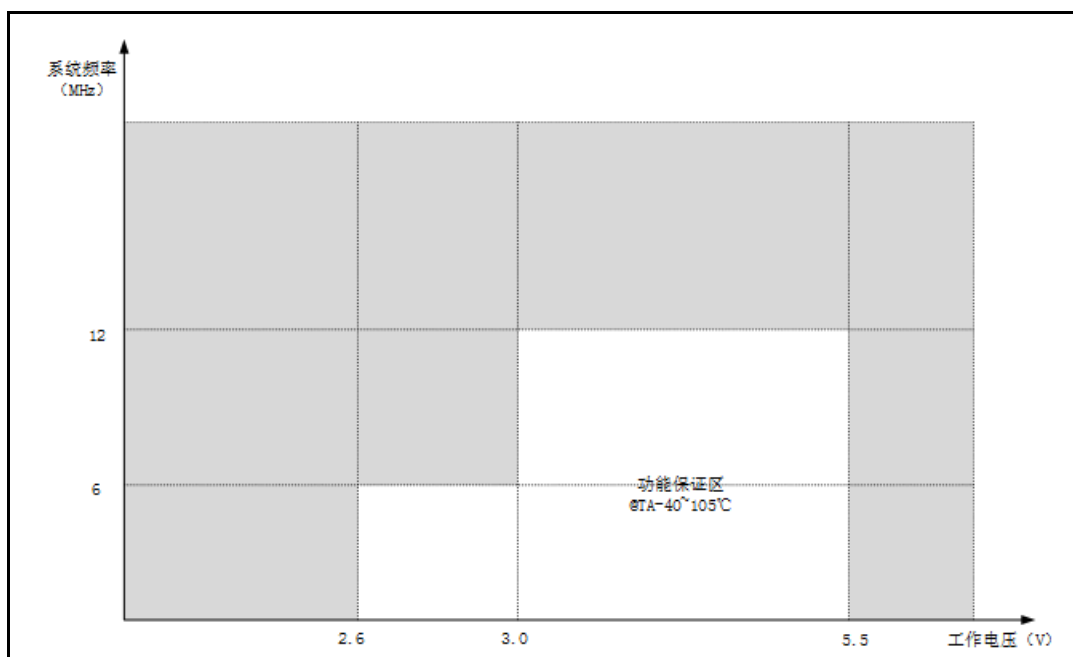


图 12-1 工作频率与电压对应关系

注意：供电电压在 2.6V 以下 EEPROM 的写功能不能保证。

上下电的工作条件如下表所示：

符号	参数	条件	最小值	典型值	最大值	单位
t_{VDD}	VDD 上升时间范围	-	2	-	∞	$\mu s/V$
	VDD 下降时间范围	-	2	-	∞	
t_{TEMP}	POR 复位释放延时	-	-	60	280	μs

$t_{\text{TEMP SYS}}$	系统复位释放延时			16		ms
$V_{\text{IT+}}$	上电复位阈值	—	—	1.761		V
$V_{\text{HYS (BOR)}}$	欠压复位迟滞	—	—	18	—	mV

12.3 电源功耗特性

MCU 工作模式下总电流消耗处于以下的状态：

- 所有 I/O 引脚在输入模式，且处于静态 VDD 或者 VSS（无负载），所有外设处于关闭状态。
- 所有外设关闭（通过外设时钟门控关闭）除非特殊说明测试条件遵循 V_{DD} 和 T_{A} （常温 25°C）。

符号	参数	条件		典型值	最大值	单位
$I_{\text{DD (RUN)}}$	工作模式，程序从 MTP 中访问	$f_{\text{CPU}}=f_{\text{MASTER}}=12\text{MHz}$, $V_{\text{DD}}=5\text{V}$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	6.3	—	mA
$I_{\text{DD (RUN)}}$	工作模式，程序从 MTP 中访问	$f_{\text{CPU}}=f_{\text{MASTER}}=12\text{MHz}$, $V_{\text{DD}}=3.3\text{V}$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	4.5	—	mA
$I_{\text{DD (SLEEP)}}$	工作模式，程序从 MTP 中访问	$f_{\text{CPU}}=f_{\text{MASTER}}=12\text{MHz}$, $V_{\text{DD}}=5\text{V}$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	2.9	—	mA
$I_{\text{DD (SLEEP)}}$	工作模式，程序从 MTP 中访问	$f_{\text{CPU}}=f_{\text{MASTER}}=12\text{MHz}$, $V_{\text{DD}}=3.3\text{V}$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	2.0	—	mA
$I_{\text{DD (DEEPSLEEP)}}$		只有 32kHz 时钟工作和 POR 电路工作，其他模拟电路不工作，MTP 处于静态。看门狗关闭， $V_{\text{DD}}=5\text{V}$		6.5	—	μA
$I_{\text{DD (DEEPSLEEP)}}$		只有 32kHz 时钟工作和 POR 电路工作，其他模拟电路不工作，MTP 处于静态。看门狗关闭， $V_{\text{DD}}=3.3\text{V}$		3.7	—	μA
t_{WUDS}	从 DEEPSLEEP 模式下唤醒的时间			267		μs

12.4 内部时钟源特性

符号	参数	条件	最小值	典型值	最大值	单位
f_{IMO} （内部高速时钟）	频率	—	—	24	—	MHz
ACCIMO	校准后内部时钟精度	$2.6 \leq V_{\text{DD}} \leq 5.5\text{V}$, $-40^{\circ}\text{C} \leq T_{\text{A}} \leq 85^{\circ}\text{C}$	-2	—	2	%
		$2.6 \leq V_{\text{DD}} \leq 5.5\text{V}$,	-3	—	3	%

		$-40^{\circ}\text{C} \leq T_A \leq 105^{\circ}\text{C}$				
TSU (IMO)	IMO 唤醒时间, 包括校准	—	—	—	20	μs
fILO (内部低速时钟)	频率	—	—	32	—	kHz
ACCIL0	校准后内部时钟精度	$2.6 \leq V_{DD} \leq 5.5\text{V}$, $-40^{\circ}\text{C} \leq T_A \leq 105^{\circ}\text{C}$	-10	—	10	%
ID0 (ILO)	ILO 功耗	$V_{DD}=5\text{V}$, $T_A=25^{\circ}\text{C}$, 32KHz	—	1	—	μA

12.5 存储器特性

MTP 和 64 字节信息区存储器特性

- 1、编程电压：4.5~5.5V
- 2、16 位数据改变典型编程时间 0.3ms, 16 位数据不改变最小编程时间 20us
- 3、85°C10 年 ($V_{DD}=5\text{V}/1000\text{T}$ 周期后)
- 4、可编程 1000 次

EEPROM 存储器特性

- 1、编程电压 2.6~5.5V
- 2、数据改变字节典型编程时间 0.7ms, 数据不变字节最小编程时间 80us;
- 3、85°C10 年 ($V_{DD}=5\text{V}/1000\text{T}$ 周期后)
- 4、可编程 50000 次

12.6 I/O 引脚特性

符号	参数	条件	最小值	典型值	最大值	单位
V_{IL}	输入低电平电压	$V_{DD}=5\text{V}$	-0.3		$0.3V_{DD}$	V
V_{IH}	输入高电平电压		$0.7V_{DD}$		$V_{DD}+0.3$	V
V_{hys}	迟滞		—	700	—	mV
R_{PU}	上拉电阻	$V_{DD}=5\text{V}$, $V_{IN}=V_{SS}$	—	10	—	k Ω
t_R , t_F	上升和下降时间 (10%~90%)	快速 I/Os 负载 50pF	—	18	—	ns
		标准 I/Os 负载 50pF	—	12	—	ns
I_{IKG}	数字输入漏电流	$V_{SS} \leq V_{IN} \leq V_{DD}$	—	—	± 1	μA
I_{IKGANA}	TK 输入漏电流	$V_{SS} \leq V_{IN} \leq V_{DD}$	—	—	± 250	nA
I_{IKGTK}	模拟模式输入漏电流	$V_{SS} \leq V_{IN} \leq V_{DD}$	—	—	± 250	nA
V_{OL}	输出低电平 (普通模式)	$V_{DD}=5\text{V}$, 输出低电平, I/O 口灌 0.8V 电压	—	27	86 (P10)	mA

		$V_{DD}=3.3V$, 输出低电平, I0 口灌 0.8 V 电压	—	18.8	60 (P10)	mA
V_{OH}	输出高电平 (普通模式)	$V_{DD}=5V$, 输出高电平, I0 口拉 4.3V 电压	—	10	21 (P10)	mA
		$V_{DD}=3.3V$, 输出高电平, I0 口拉 3.0 V 电压	—	3.1	6.6 (P10)	mA
V_{OL}	输出低电平 (大电流驱动)	$V_{DD}=5V$, 输出低电平, I0 口灌 0.8V 电压	—	85	—	mA
		$V_{DD}=3.3V$, 输出低电平, I0 口灌 0.8 V 电压	—	60	—	mA
V_{OH}	输出高电平 (大电流驱动)	$V_{DD}=5V$, 输出高电平, I0 口拉 4.3V 电压	—	20	—	mA
		$V_{DD}=3.3V$, 输出低电平, I0 口拉 3.0 V 电压	—	6	—	mA

12.7 ADC 特性

($T_A=25^{\circ}C$, 除非另有说明)

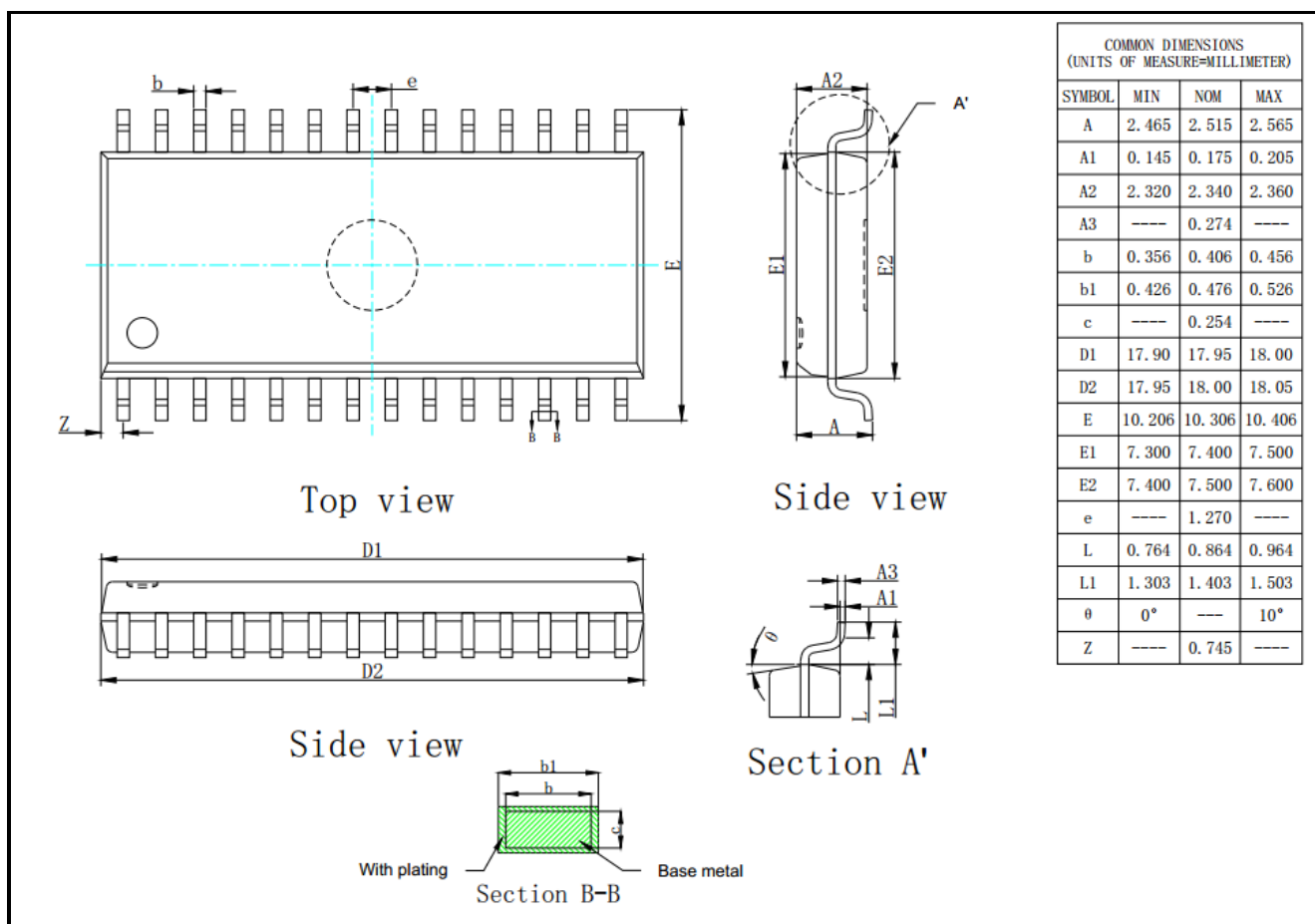
符号	参数	条件	最小值	典型值	最大值	单位
V	供电电压		2.6	5	5.5	V
T	工作温度		-40	25	105	$^{\circ}C$
f_{ADC}	时钟频率		0.5	2	4	M
P_{ADC}	功耗	包括静态动态功耗		300	500	μA
V_{AIN}	ADC 输入电压		GND		VREF	V
C_{ADC}	采样电容			12.8		pF
DNL	微分非线性误差	采用外部 2.4V 参考		± 0.5	± 3	LSB
INL	积分非线性误差	采用外部 2.4V 参考		± 1	± 4	LSB
E_o	OFFSET_ERROR	$f_{ADC}=4M$			± 8	LSB
E_g	GAIN_ERROR	$f_{ADC}=4M$			± 8	LSB
T_{ADC1}	ADC 转换时间 1	$f_{ADC}=1M$, ADC 采样周期=2, 总共 20 个时钟周期			20	μs
T_{ADC2}	ADC 转换时间 2	$f_{ADC}=4M$, ADC 采样周期=2, 总共 20 个时钟周期			5	μs

12.8 ESD 特性

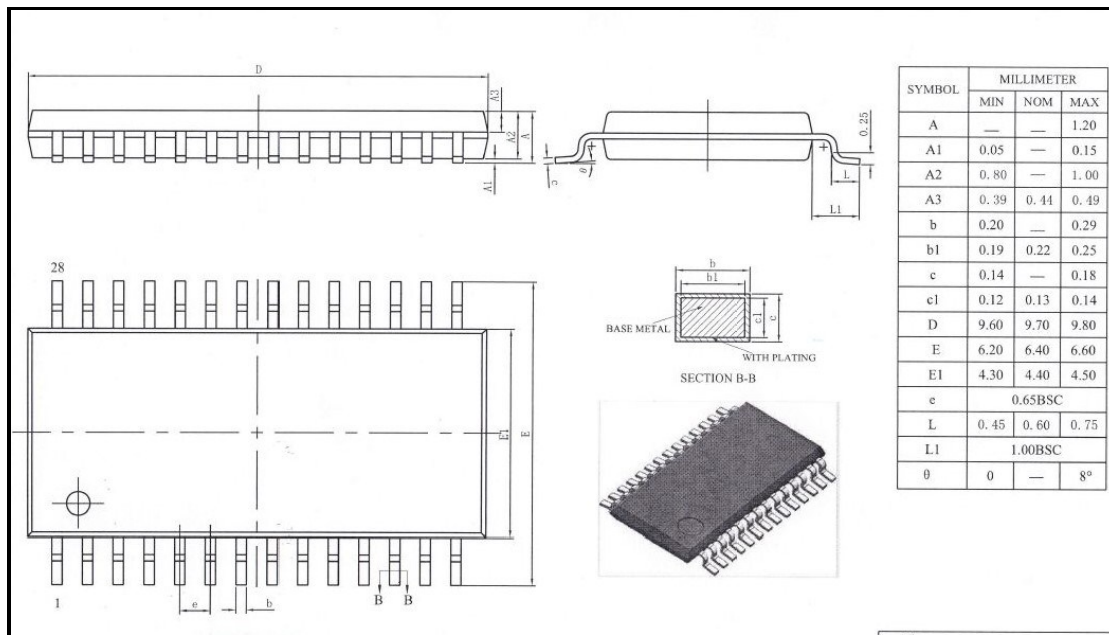
符号	参数	测试条件	最大值	单位
VESD (HBM)	人体放电模式	TA=25°C, 遵循 JESD22-A114	6000	V
VESD (MM)	机器放电模式	TA=25°C,	400	V
VESD (CDM)	元件充电模式	TA=25°C, 遵循 JESD22-C101	2000	V
Latch-up	静态门锁	TA=25°C	200	mA

13 封装信息

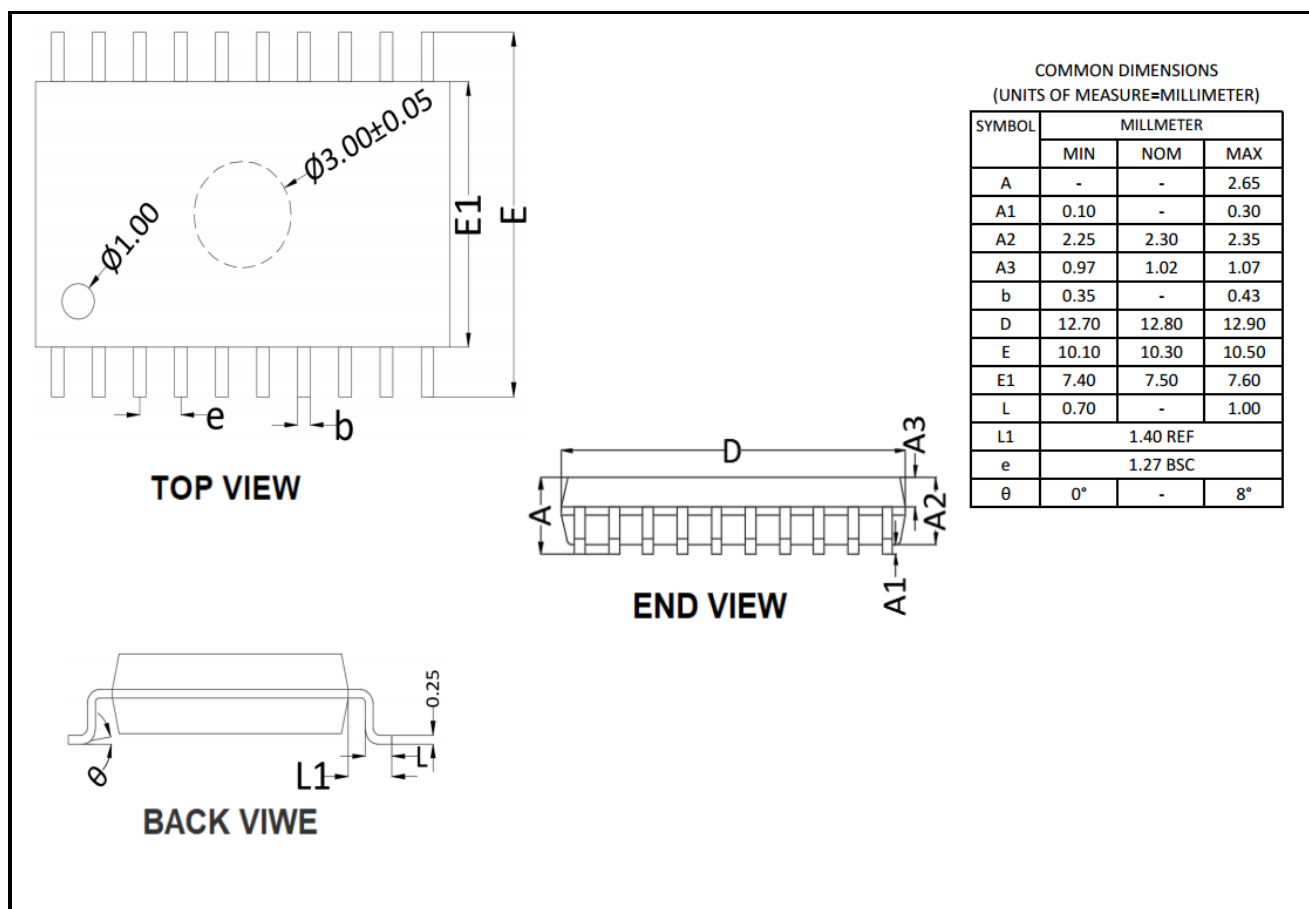
13.1 SOP28



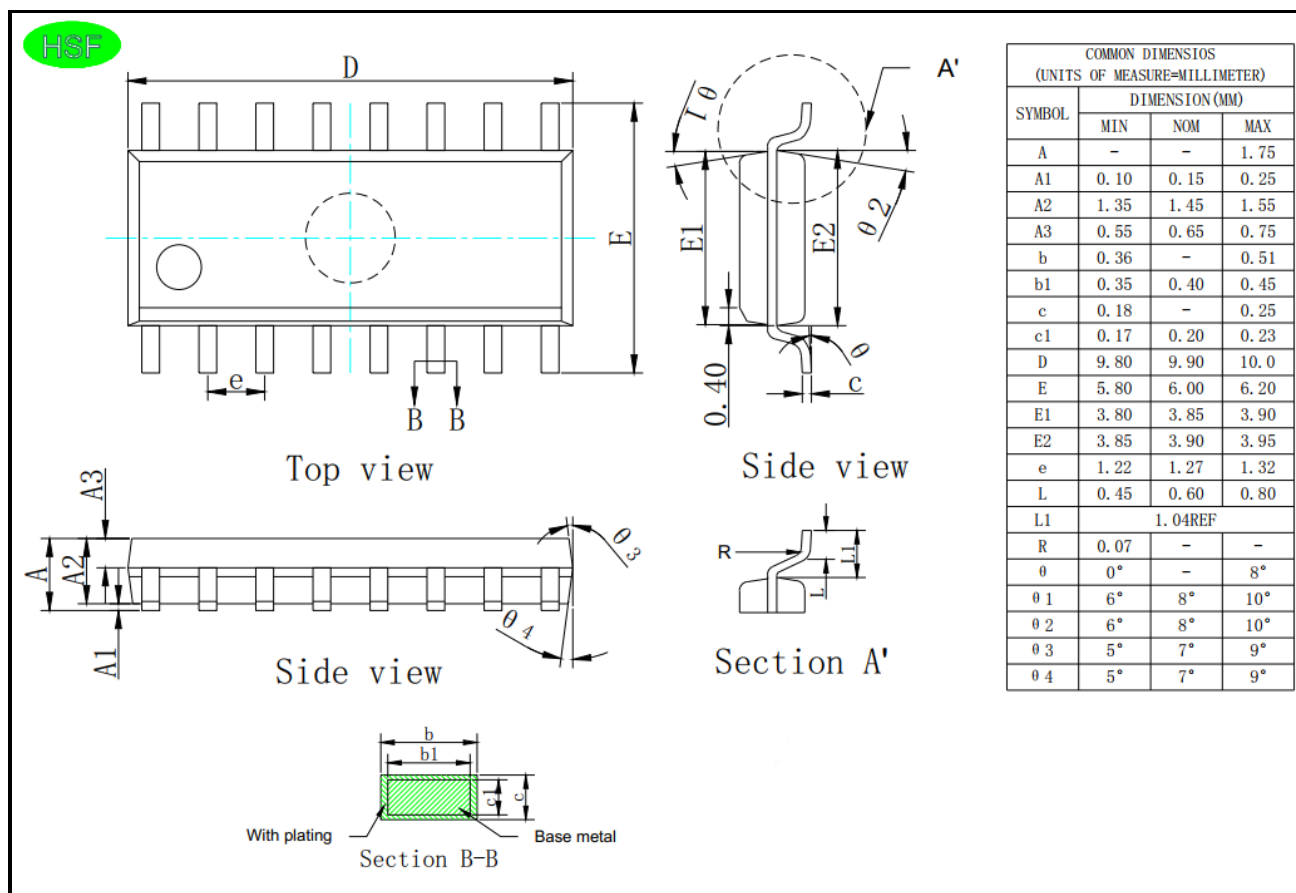
13.2 TSSOP28



13.3 SOP20



13.4 SOP16



14 命名规则

名称	RC6	T8	063	4	G	A	TY	7	A	T
序号	1	2	3	4	5	6	7	8	9	10

序号	含义
1	RC6 = MCU 系列
2	T8 = 8 位触摸
3	0: RC 默认引脚排布 6: 系列号 3: 迭代版本号
4	ROM SIZE: 0 为 1K, 1 为 2K, 2 为 4K, 3 为 8K, 4 为 16K, 5 为 32K, 6 为 64K, 7 为 128K...
5	引脚数: A 为 8 引脚, B 为 10 引脚, C 为 14 引脚, D 为 16 引脚, E 为 20 引脚, F 为 24 引脚, G 为 28 引脚, K 为 32 引脚, T 为 36 引脚...
6	封装类型: A 为 SOP, B 为 QFN, C 为 TSSOP, D 为 DFN, E 为 SSOP, F 为 QFP, G 为 DIP...
7	应用领域: TY 为通用产品, 该位可供客户定制 (仅在订单型号体现, 丝印不体现该信息)
8	温度范围: 4 为 0~70℃, 6 为 -40~85℃, 7 为 -40~105℃, 8 为 -40~125℃... (仅在订单型号体现, 丝印不体现该信息)
9	引脚间距 (见表 14-1) (仅在订单型号体现, 丝印不体现该信息)
10	包装形式: R 为编带卷装, T 为管装, Y 为托盘装 (仅在订单型号体现, 丝印不体现该信息)

表 14-1 封装间距详情表

SOP	A	1.27mm	TSSOP	A	0.65mm
	-	-		-	-

15 订单信息

订单型号	丝印标记	封装	适用温度 (°C)	引脚间距 (mm)	包装
RC6T80634GATY7AT	RC6T80634GA	SOP28	-40~105℃	1.27	管装
RC6T80634GCTY7AT	RC6T80634GC	TSSOP28	-40~105℃	0.65	管装
RC6T80634EATY7AT	RC6T80634EA	SOP20	-40~105℃	1.27	管装
RC6T81634EATY7AT	RC6T81634EA	SOP20	-40~105℃	1.27	管装
RC6T81634DATY7AT	RC6T81634DA	SOP16	-40~105℃	1.27	管装

16 开发工具

R-Link Pro 三合一调试工具

R-Link Pro 烧录调试上位机

17 版本说明

版本号	修改时间	修订人	修改内容
V1.0	2024.4.10	FX	初版
V1.1	2024.8.10	FX	1. 修正 SOP16 封装引脚描述错误。
V1.2	2024.8.29	FX	1. 修正 ESD 特性参数。 2. 修正 MTP_CR 寄存器描述错误。 3. 优化 ADC 结构图
V1.3	2024.11.4	FX	1. 优化防水扫描描述。 2. 增加 RC6T81634EA-SOP20、RC6T81634DA-SOP16 封装、删除 RC6T80634DA-SOP16 封装，删除 RC6T82634xx 封装图。 3. 优化开发工具描述。 4. 修正 SCR_CFG 寄存器描述错误。 5. 更新订单信息。
V1.4	2025.1.13	FX	1. 修正 GPIO 配置描述错误。 2. 优化部分表格排版，删除原 10.7.2 章节引脚功能描述。 3. 增加命名规则和更新订单信息。
V1.5	2025.7.13	FX	1. 修正 I/O 口特性描述错误。
V1.6	2025.8.25	FX	1. 优化引脚图描述，标识烧录引脚。