



深圳富芯电子

RC6T80604x 数据手册

增强型 8 位触控类单片机

Rev. V1.9

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

RC6T80604x (x 代表不同型号) 是一款高性能的 8 位触摸型单片机, 最大支持 25 路触摸通道。内部包含 16k MTP, 64 Byte INFO, 128 Byte EEPROM, 1k SRAM, 3 个 16 位定时器、6 路 10 位 PWM、具有独立时钟的看门狗, 12-bit ADC (最大支持 10 路外部输入和 1 路内部 1/4VDD), 2 路 UART、1 路 I2C 和 1 路 SPI 通讯接口, 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:
 - 2.6MHz, VDD ≥ 2.4V
 - 16MHz, VDD ≥ 4.5V

◆ 工作电压: 2.4V~5.5V

◆ 工作温度: -40°C~85°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, 2.5V~5.5V)
 - 误差 ±2% (-40°C~85°C)
- 内部 32KHz 低速 RC 振荡器 (误差 ±10%)
- 外部 16M 时钟和 32.768K 晶振

◆ 复位:

- 上电复位
- 欠压复位 (2.25V、2.5V、2.8V、3.6V、4.2V)
- 看门狗溢出复位
- ◆ 低电压检测: LVD 共 5 级 (2.3V、2.7V、3.0V、3.8V、4.5V)

◆ 中断 (INT):

- Timer、SCM、VC3、WDT、PWM、CSD、ADC、UART、SPI、I2C、LVD、GPIO 共 18 个中断源, 全部 GPIO 可设上升沿、下降沿、双边沿、低电平中断

◆ 数字外设:

- 3 个 16 位高级定时器, 支持 6 路 PWM 输出功能
 - 支持捕获和刹车功能
 - 支持周期中断和占空比中断
- 1 个 16 位看门狗定时器

- 2 路 UART (支持全双工和半双工)
- 1 路 SPI
- 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: 10 路
- 内部输入: 1 路 (1/4 VDD)
- 参考源 — 外部参考: VDD
内部参考: 1.2/2.4V
- 采样可以通过 PWM 或者管脚的上升沿或者下降沿触发

◆ LCD 驱动:

- 支持 5 路 COM 口, 输出 1/2BIAS 电压

◆ 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 可单独配置上拉 10K 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、引脚中断唤醒
- 深度休眠电流: 3.2uA (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 RC6T80604GA 封装脚位图	6
4.2 RC6T80604GC 封装脚位图	6
4.3 RC6T80604EA 封装脚位图	7
4.4 RC6T80604DA 封装脚位图	7
4.5 引脚功能描述	8
5 端口结构	10
5.1 I/O 口结构框图	10
5.2 配置 GPIO	10
5.3 读-修改-写 指令	12
5.4 GPIO 中断	12
5.5 引脚复用	13
5.6 寄存器定义	13
5.7 GPIO 配置流程	23
5.8 GPIO 操作示例	24
6 CPU 功能简介	25
6.1 内核 SFR 寄存器	25
7 中断控制系统	29
7.1 中断向量表	30
7.2 中断优先级	30
7.3 寄存器定义	30
7.4 外部中断	32
7.5 外部中断配置流程	32
7.6 外部中断操作示例	33
8 存储器系统	34
8.1 存储器系统概述	34
8.2 程序存储器映射	34
8.3 数据存储器映射	34
8.4 MTP 与 SRAM 结构	36
8.5 寄存器映射	36
8.6 MTP 控制器	37
8.7 MTP 操作示例	42
9 时钟控制	43
9.1 时钟概述	43
9.2 时钟信号说明	43
9.3 时钟结构框图	44
9.4 内部主振 (IMO)	45
9.5 内部低速振荡器 (ILO)	45
9.6 外部引脚时钟	45
9.7 外部晶体振荡器	45

9.8 时钟切换	45
9.10 寄存器定义	45
10 外设资源	48
10.1 TIMER	48
10.2 PWM	58
10.3 CHECKSUM	65
10.4 LCD	66
10.5 ADC	68
10.6 UART0/1	75
10.7 SPI	83
10.8 I2C	89
11 系统控制	97
11.1 复位	97
11.2 电源管理	98
11.3 看门狗	100
11.4 系统模式控制寄存器	103
11.5 模拟电路控制	104
12 电气特性	108
12.1 绝对最大额定值	108
12.2 工作条件	108
12.3 电源功耗特性	109
12.4 外部时钟源特性	111
12.5 内部时钟源特性	112
12.6 存储器特性	113
12.7 I/O 引脚特性	113
12.8 ADC 特性	114
12.9 LCD 特性	114
12.10 CSD 触摸特性	114
12.11 EMC 特性	115
12 封装信息	116
12.1 SOP28	116
12.2 TSSOP28	117
12.3 SOP20	118
12.4 SOP16	119
13 订单信息	120
14 开发工具	120
15 版本说明	120

3 模块框图

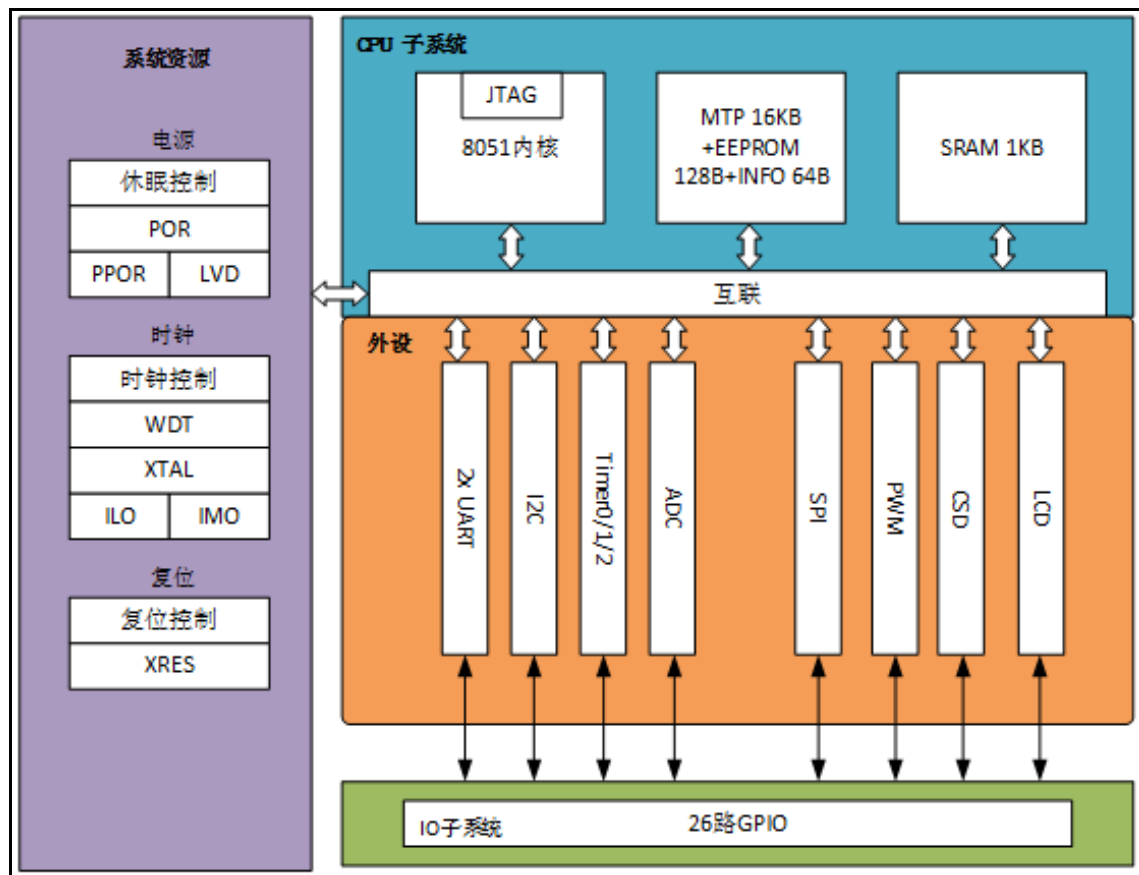


图 3-1 RC6T80604x 模块框图

4 引脚说明

4.1 RC6T80604GA 封装脚位图

VDD	1	28	P0.0/PWM0/COM0/TK0
CMOD/P1.0	2	27	P0.1/PWM1/COM1/TK1
VSS	3	26	P0.2/PWM2/COM2/TK2
TK24/OSCI/P3.1	4	25	P0.3/COM3/TK3
TK23/OSCO/P3.0	5	24	P0.4/COM4/TK4
RSTB/TK22/P1.1	6	23	P0.5/SPI_SCK/[SCL]/ADC_ETR/TK5
TCK/SCL/TK21/RX0/T0/P1.2	7	22	P0.6/T2EX/TK6
TDIO/SDA/TK20/TX0/T1/P1.3	8	21	P0.7/T2/TK7
TK19/TABUS/AIN9/P1.4	9	20	P2.0/MOSI/[SDA]/TX1/TK8
TK18/CLK_EXT/AIN8/P1.5	10	19	P2.1/MISO/RX1/BKIN/TK9
TK17/AIN7/P1.6	11	18	P2.2/AIN0/TK10
TK16/AIN6/P1.7	12	17	P2.3/AIN1/TK11
TK15/PWM5/AIN5/P2.7	13	16	P2.4/AIN2/TK12/TDBUS
TK14/PWM4/AIN4/P2.6	14	15	P2.5/AIN3/ADCREP/PWM3/TK13

RC6T80604GA (SOP28) 封装脚位图

4.2 RC6T80604GC 封装脚位图

VDD	1	28	P0.0/PWM0/COM0/TK0
CMOD/P1.0	2	27	P0.1/PWM1/COM1/TK1
VSS	3	26	P0.2/PWM2/COM2/TK2
TK24/OSCI/P3.1	4	25	P0.3/COM3/TK3
TK23/OSCO/P3.0	5	24	P0.4/COM4/TK4
RSTB/TK22/P1.1	6	23	P0.5/SPI_SCK/[SCL]/ADC_ETR/TK5
TCK/SCL/TK21/RX0/T0/P1.2	7	22	P0.6/T2EX/TK6
TDIO/SDA/TK20/TX0/T1/P1.3	8	21	P0.7/T2/TK7
TK19/TABUS/AIN9/P1.4	9	20	P2.0/MOSI/[SDA]/TX1/TK8
TK18/CLK_EXT/AIN8/P1.5	10	19	P2.1/MISO/RX1/BKIN/TK9
TK17/AIN7/P1.6	11	18	P2.2/AIN0/TK10
TK16/AIN6/P1.7	12	17	P2.3/AIN1/TK11
TK15/PWM5/AIN5/P2.7	13	16	P2.4/AIN2/TK12/TDBUS
TK14/PWM4/AIN4/P2.6	14	15	P2.5/AIN3/ADCREP/PWM3/TK13

RC6T80604GC (TSSOP28) 封装脚位图

4.3 RC6T80604EA 封装脚位图

VDD	1	20	P0.0/PWM0/COM0/TK0
CMOD/P1.0	2	19	P0.1/PWM1/COM1/TK1
VSS	3	18	P0.2/PWM2/COM2/TK2
RSTB/TK22/P1.1	4	17	P0.3/COM3/TK3
TCK /SCL/TK21/RX0/T0/P1.2	5	16	P0.4/COM4/TK4
TDIO /SDA/TK20/TX0/T1/P1.3	6	15	P0.5/SPI_SCK/[SCL]/ADC_ETR/TK5
TK19/TABUS/AIN9/P1.4	7	14	P2.0/MOSI/[SDA]/TX1/TK8
TK18/CLK_EXT/AIN8/P1.5	8	13	P2.1/MISO/RX1/BKIN/TK9
TK15/PWM5/AIN5/P2.7	9	12	P2.4/AIN2/TK12/TDBUS
TK14/PWM4/AIN4/P2.6	10	11	P2.5/AIN3/ADCREP/PWM3/TK13

RC6T80604EA (SOP20) 封装脚位图

4.4 RC6T80604DA 封装脚位图

VDD	1	16	P0.0/PWM0/COM0/TK0
CMOD/P1.0	2	15	P0.1/PWM1/COM1/TK1
VSS	3	14	P0.5/SPI_SCK/[SCL]/ADC_ETR/TK5
RSTB/TK22/P1.1	4	13	P2.0/MOSI/[SDA]/TX1/TK8
TCK /SCL/TK21/RX0/T0/P1.2	5	12	P2.1/MISO/RX1/BKIN/TK9
TDIO /SDA/TK20/TX0/T1/P1.3	6	11	P2.4/AIN2/TK12/TDBUS
TK19/TABUS/AIN9/P1.4	7	10	P2.5/AIN3/ADCREP/PWM3/TK13
TK18/CLK_EXT/AIN8/P1.5	8	9	P2.6/AIN4/PWM4/TK14

RC6T80604DA (SOP16) 封装脚位图

4.5 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
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	支持硬件防水
CMOD	O	触摸外接电容
LCD		
COM0 ~ COM4	O	LCD COM 口
PWM 控制器		
PWM0	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
PWM1	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
PWM2	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
PWM3	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
PWM4	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
PWM5	I/O	10 位 PWM, 两种工作模式, 支持刹车功能
BKIN	I	PWM 刹车输入引脚
UART		
TX0	O	UART0 数据输出引脚
RX0	I	UART0 数据输入引脚
TX1	O	UART1 数据输出引脚
RX1	I	UART1 数据输入引脚
SPI		
MISO	I/O	SPI MISO 引脚
MOSI	I/O	SPI MOSI 引脚
SPI_SCK	I/O	SPI 时钟引脚
I2C		
SCL	I/O	I2C 时钟引脚
SDA	I/O	I2C 数据引脚(注意: 上电 2.1ms 左右, 该脚会输出 50us 左右的低电平)
时钟和电源		
OSCI	I	外接晶振输入
OSCO	O	外接晶振输出
CLK_EXT	I	外部时钟输入
VDD	POWER	电源

VSS	POWER	地
RSTB	RESET	复位引脚
JTAG		
TCK	I	JTAG 时钟引脚
TDIO	I	JTAG 数据传输引脚
ADC		
AIN0 ~ AIN9	I	ADC 外部采样输入通道
ADC_ETR	I	ADC 外部触发采样输入引脚
ADC_REF	I	ADC 外部参考引脚
TIMER		
T0	I	计数器 0 外部输入
T1	I	计数器 1 外部输入
T2EX	I	定时器 2 外部捕获输入
T2	I	定时器 2 外部输入

5 端口结构

5.1 I/O 口结构框图

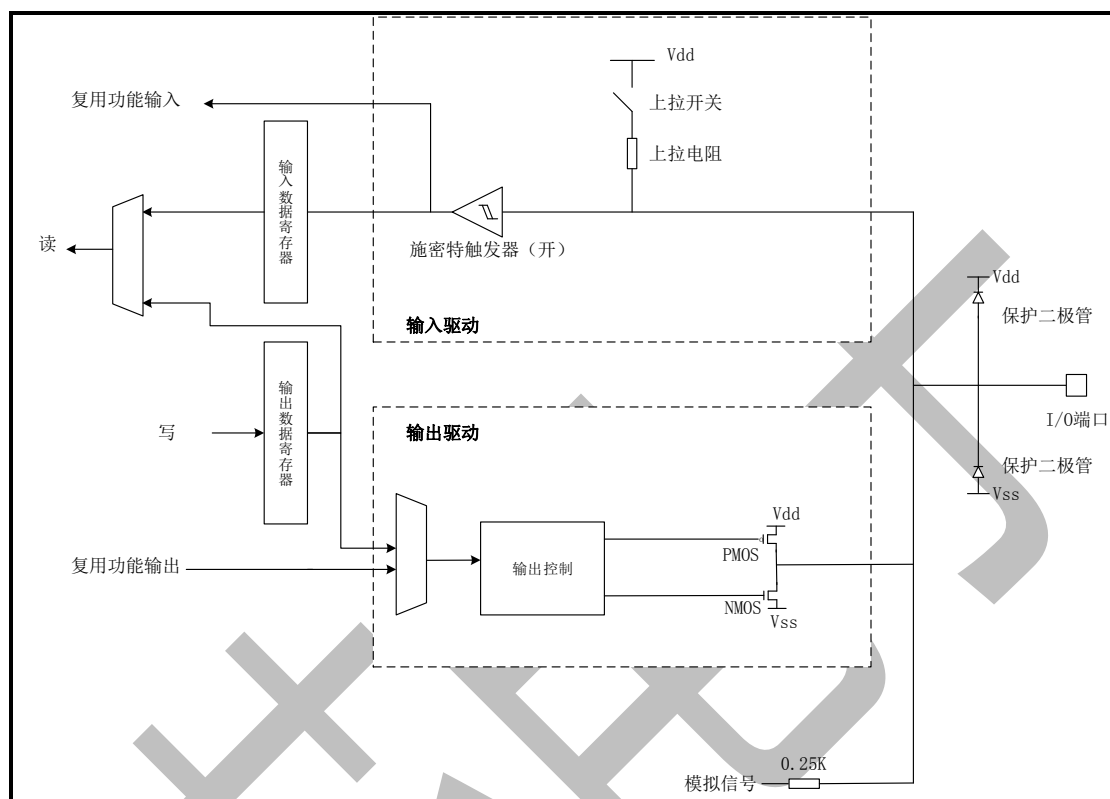


图 5-1 GPIO 结构图

5.2 配置 GPIO

每个 GPIO 的使用三个寄存器 Px_DMx 进行配置模式。

以 P0 口为例，配置 P0 口需要使用 P0_DM0、P0_DM1 和 P0_DM2 三个寄存器进行配置，如下图所示：

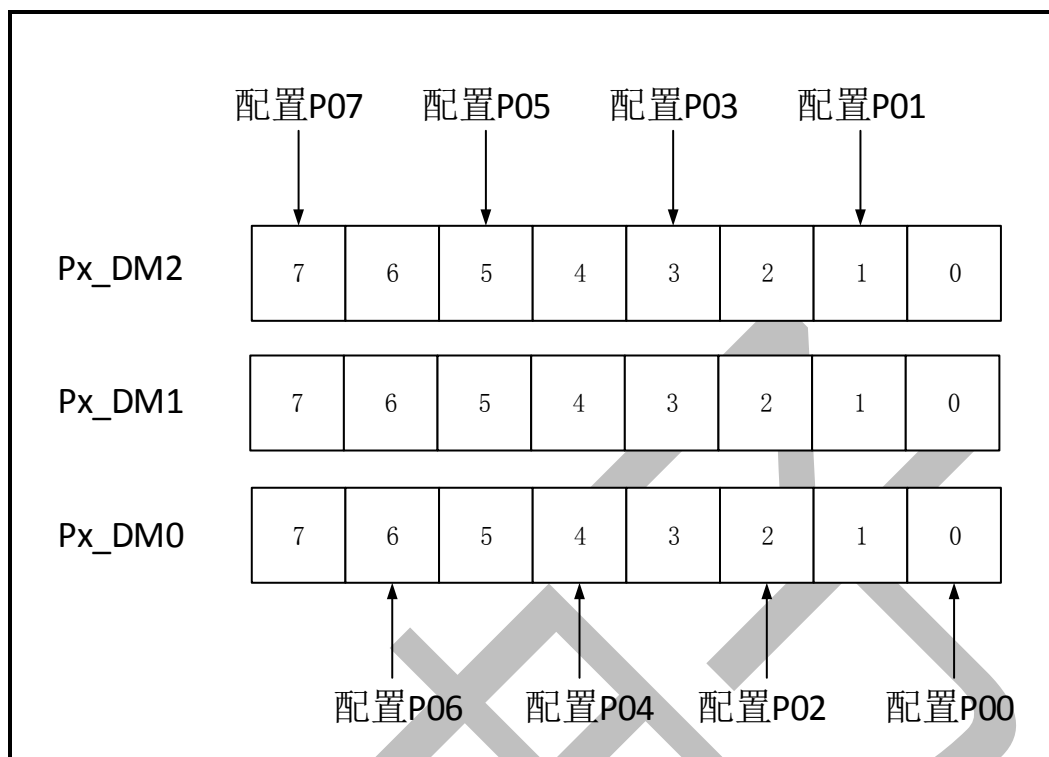


图 5-2 I/O 配置图

GPIO 模式的配置说明如下表所示

表 5-1 GPIO 驱动模式

Px_DM2	Px_DM1	Px_DM0	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、LCD 的 COM 驱动。 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、SPI /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注 1。
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	防水组合 0: P3.1/P3.0/P1.1/P1.2/P1.3/P1.4 防水组合 1: P1.5/P1.6/P1.7/P2.7/P2.6/P2.5 防水组合 2: P2.4/P2.3/P2.2/P2.1/P2.0/P0.7 防水组合 3: P0.6/P0.5/P0.4/P0.3/P0.2/P0.1/P0.0
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 模式下需要配合 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，需要配合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，会有大概 VDD/10K 的电流(比如 5V 电源供电，则会有 500uA 电流)通过。

系统进入 deepsleep 时需要注意开漏的管脚要避免输出低电平。

注 3：配置使能外设时，配置 GPIO 需要遵循的顺序：PT_AFRx→Px_GE→Px_DMx。（P1.2 和 P1.3 管脚默认是 I2C 外设，如果需要在两个管脚上使用其他外设需要先将 I2C 使能关闭），当配置 P1.1 引脚为复位功能时，配置顺序应该 Px_DMx→PT_AFRx→Px_GE。

5.3 读-修改-写 指令

从 SFR 或内部 RAM 读一个字节，修改它，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而不是外部引脚的状态，这种指令读端口 SFR 的值，修改它并写回到 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)

5.4 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。

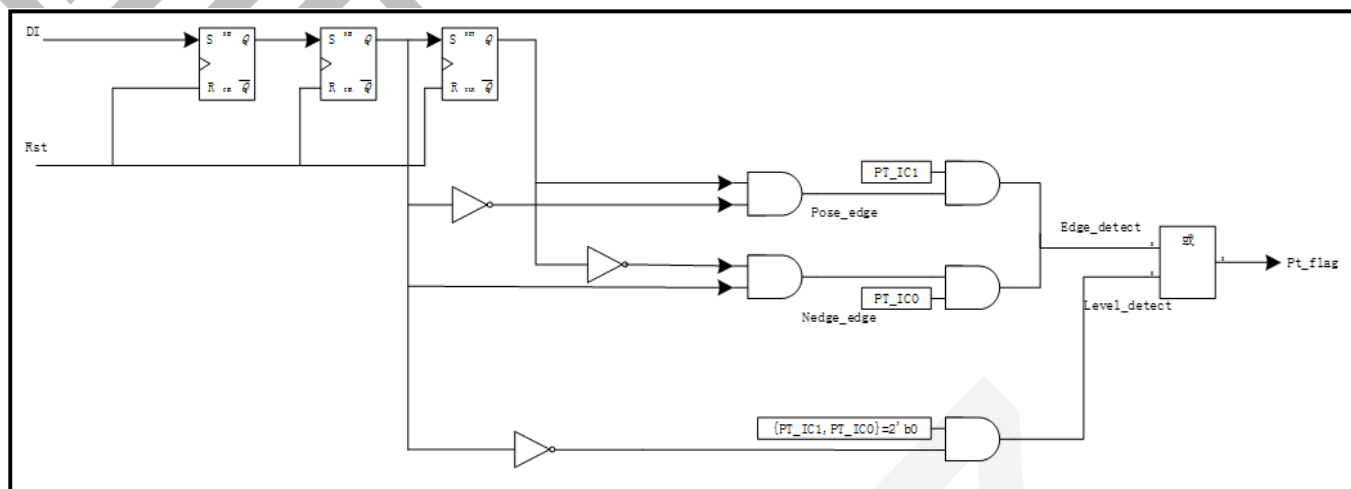


图 5-3 中断触发实现框图

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，需要清除一下对应的 FLAG。

5.5 引脚复用

芯片支持数字外设功能使用不同的管脚位置，通过 PT_AFR, PT_AFR1 寄存器来进行设置。完整的管脚映射见第 4 章中引脚分配。

PT_AFR 和 PT_AFR1 寄存器可以使能部分数字外设的管脚位置，PT_DIG_TEST 寄存器用于数字测试总线，具体见寄存器说明；

P0_GE, P1_GE 和 P2_GE 寄存器用来使能部分数字外设的复用管脚功能；

不允许同一个管脚同时用于多个使能的外设，例如 P05，当 I2C 和 SPI 同时使能时都需要用到 P05，那此时 SPI 和 I2C 的功能都无法正常工作。

5.6 寄存器定义

名字	地址	读写	复位值	描述
PT_AFR	0xFF08	读写	00000000	外设管脚位置使能配置寄存器 0
PT_AFR1	0xFF09	读写	00000010	外设管脚位置使能配置寄存器 1
PT_DIG_TEST	0xFF8F	读写	00000000	数字测试总线控制寄存器
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 0 位
P0_DM1	0xFF41	读写	00000000	端口 0 模式 1 位
P0_DM2	0xFF42	读写	00000000	端口 0 模式 2 位
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_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 0 位
P1_DM1	0xFF49	读写	00001100	端口 1 模式 1 位
P1_DM2	0xFF4A	读写	00000000	端口 1 模式 2 位
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_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 0 位
P2_DM1	0xFF51	读写	00000000	端口 2 模式 1 位
P2_DM2	0xFF52	读写	00000000	端口 2 模式 2 位
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_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 0 位
P3_DM1	0xFF59	读写	00000000	端口 3 模式 1 位
P3_DM2	0xFF5A	读写	00000000	端口 3 模式 2 位
P3_FLAG	0xFF28	读写	00000000	端口 3 中断标志位
P3_PU	0xFF2A	读写	00000000	端口 3 上拉电阻使能寄存器
P3_IE	0xFF2D	读写	00000000	端口 3 中断使能寄存器
P3_IC0	0xFF2E	读写	00000000	端口 3 中断控制 0 位
P3_IC1	0xFF2F	读写	00000011	端口 3 中断控制 1 位

5.6.1 PT_AFR (addr: 0xFF08) 外设管脚位置使能配置寄存器 0

Bit	7	6	5	4	3	2	1	0
Name	SPI_EN	BKIN_EN	ADC_ETR_EN	T1_EN	TO_EN	RZ_EN	UART1_EN	UART0_EN
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	SPI_EN	SPI 管脚复用使能寄存器 0 SPI 管脚位置复用不使能 1 SPI 管脚位置复用使能
6	BKIN_EN	BKIN 刹车管脚复用使能寄存器 0 BKIN 管脚复用不使能 1 BKIN 管脚复用使能
5	ADC_ETR_EN	ADC_ETR 管脚复用使能寄存器 0 ADC_ETR 管脚复用不使能 1 ADC_ETR 管脚复用使能

4	T1_EN	Timer1 输入管脚复用使能寄存器 0 Timer1 管脚位置复用不使能 1 Timer1 管脚位置复用使能
3	T0_EN	Timer0 管脚复用使能寄存器 0 Timer0 管脚位置复用不使能 1 Timer0 管脚位置复用使能
2	RZ_EN	RZ 码输出管脚复用使能寄存器 0 RZ 码管脚位置复用不使能 1 RZ 码管脚位置复用使能
1	UART1_EN	UART1 输出管脚复用使能寄存器 0 UART1 外设管脚复用不使能 UART1 外设管脚复用使能
0	UART0_EN	UART0 输出管脚复用使能寄存器 0 UART0 外设管脚复用不使能 1 UART0 外设管脚复用使能

5.6.2 PT_AFR1 (addr: 0xFF09) 外设管脚位置使能配置寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	I2C_SEL	UART1_SEL	UART0_SEL	UART1_ASYNC_HALF	UART0_ASYNC_HALF	RD_REV_WR_EN	I2C_EN	PRST_EN
Reset	0	0	0	0	0	0	1	0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	I2C_SEL	0 SDA 从管脚 P1.3 位置输出, SCL 从管脚 P1.2 位置输出 1 SDA 从管脚 P2.0 位置输出, SCL 从管脚 P0.5 位置输出
6	UART1_SEL	用于 UART1 配置异步半双工模式时, Rx 和 Tx 管脚位置互换 0 Tx 从管脚 P2.0 位置输出, Rx 从管脚 P2.1 位置输入 1 Tx 从管脚 P2.1 位置输出, Rx 从管脚 P2.0 位置输入
5	UART0_SEL	用于 UART0 配置异步半双工模式时, Rx 和 Tx 管脚位置互换 0 Tx 从管脚 P1.3 位置输出, Rx 从管脚 P1.2 位置输入 1 Tx 从管脚 P1.2 位置输出, Rx 从管脚 P1.3 位置输入
4	UART1_ASYNC_HALF	用于 UART1 配置异步半双工模式时, 使能控制 0 UART1 串口异步半双工不使能 1 UART1 串口异步半双工使能 (P2.1 做异步半双工)
3	UART0_ASYNC_HALF	用于 UART0 配置异步半双工模式时, 使能控制 0 UART0 串口异步半双工不使能 1 UART0 串口异步半双工使能 (P1.2 做异步半双工)
2	RD_REV_WR_EN	0 进行位操作时不使能 1 进行位操作时使能
1	I2C_EN	I2C 管脚位置复用使能寄存器 0 I2C 管脚位置复用不使能 1 I2C 管脚位置复用使能

0	PRST_EN	复位管脚位置复用使能寄存器 0 复位管脚位置复用不使能 1 复位管脚位置复用使能
---	---------	--

5.6.3 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 4.2。

5.6.4 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，如果输入使能则输入电平到 GDI。

5.6.5 P0_DM0/P0_DM1/P0_DM2 端口 0 模式位

Bit	7	6	5	4	3	2	1	0
Name	P0_DMx							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_DMx	P0 模式控制寄存器。

5.6.6 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

5.6.7 P0_IC0/P0_IC1 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	

5.6.8 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

5.6.9 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 4.2。

5.6.10 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	-	-	GE1.5	-	GE1.3	GE1.2	GE1.1	-
Reset	-	-	0	-	1	1	0	-
Type	-	-	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，如果输入使能则输入电平到 GDI

5.6.11 P1_DM0/P1_DM1/P1_DM2 端口 1 模式位

Bit	7	6	5	4	3	2	1	0
Name	P1_DM0							
Reset	0x00							
Type	R/W							

Bit	7	6	5	4	3	2	1	0
Name	P1_DM1							
Reset	0x0C							
Type	R/W							

Bit	7	6	5	4	3	2	1	0
Name	P1_DM2							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P1_DMx	P1 模式控制寄存器。

5.6.12 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

5.6.13 P1_IC0/P1_IC1 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	

5.6.14 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

5.6.15 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, 如果输入使能则输入电平到 GD1。

5.6.16 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 4.2。

5.6.17 P2_DM0/P2_DM1/P2_DM2 端口 2 模式 0 位

Bit	7	6	5	4	3	2	1	0
Name	P2_DMx							
Reset	0x00							

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

Bit	Name	Function
7:0	P2_DMx	P2 模式控制寄存器。

5.6.18 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

5.6.19 P2_IC0/P2_IC1 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	

5.6.20 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

5.6.21 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 4.2。

5.6.22 P3_DM0/P3_DM1/P3_DM2 端口 3 模式位

Bit	7	6	5	4	3	2	1	0
Name	P3_DMx							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P3_DMx	P3 模式控制寄存器。

5.6.23 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

5.6.24 P3_IC0/P3_IC1 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	

5.6.25 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

5.6.26 PX_PU 端口上拉电阻使能寄存器

Bit	7	6	5	4	3	2	1	0
Name	P0_PU							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	P0_PU	上拉使能: 0 关闭上拉 1 打开上拉

Bit	7	6	5	4	3	2	1	0
Name	P1_PU							
Reset	0x0C							
Type	R/W							

Bit	Name	Function
7:0	P1_PU	上拉使能: 0 关闭上拉 1 打开上拉

Bit	7	6	5	4	3	2	1	0
Name	P2_PU							
Reset	0x00							
Type	R/W							

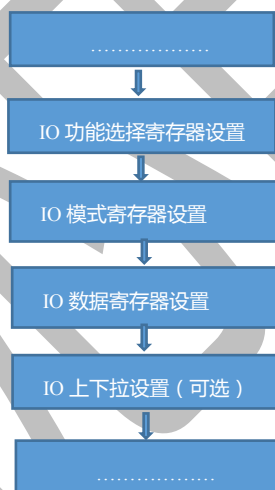
Bit	Name	Function
7:0	P2_PU	上拉使能: 0 关闭上拉 1 打开上拉

Bit	7	6	5	4	3	2	1	0
Name	-						P3_PU	
Reset	-						0x00	
Type	-						R/W	

Bit	Name	Function
7:2	-	-
1:0	P3_PU	上拉使能: 0 关闭上拉 1 打开上拉

5.7 GPIO 配置流程

使用 GPIO 时，可参照如下步骤对相关寄存器进行配置：



GPIO 配置流程图

5.8 GPIO 操作示例

/*P3_0 设置为推挽输出, P3_1 配置为数字输入*/

P3_DM0 = 0x01; //P3.0 Output mode, P3.1 Input mode

P3_DM1 = 0x00;

P3_DM2 = 0x02;

注: 推挽输出一般应用于普通 GPIO 输出或 PWM 输出、SPI/UART 通讯;

数字输入一般应用于普通 GPIO 输入、SPI/UART 通讯。

6 CPU 功能简介

RC6T80604x 系列芯片在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的提升。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	N/A	保留位，读 0
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

RC6T80604x 系列芯片有一个增强的中断控制系统，共有 18 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。

中断控制系统图如下所示：

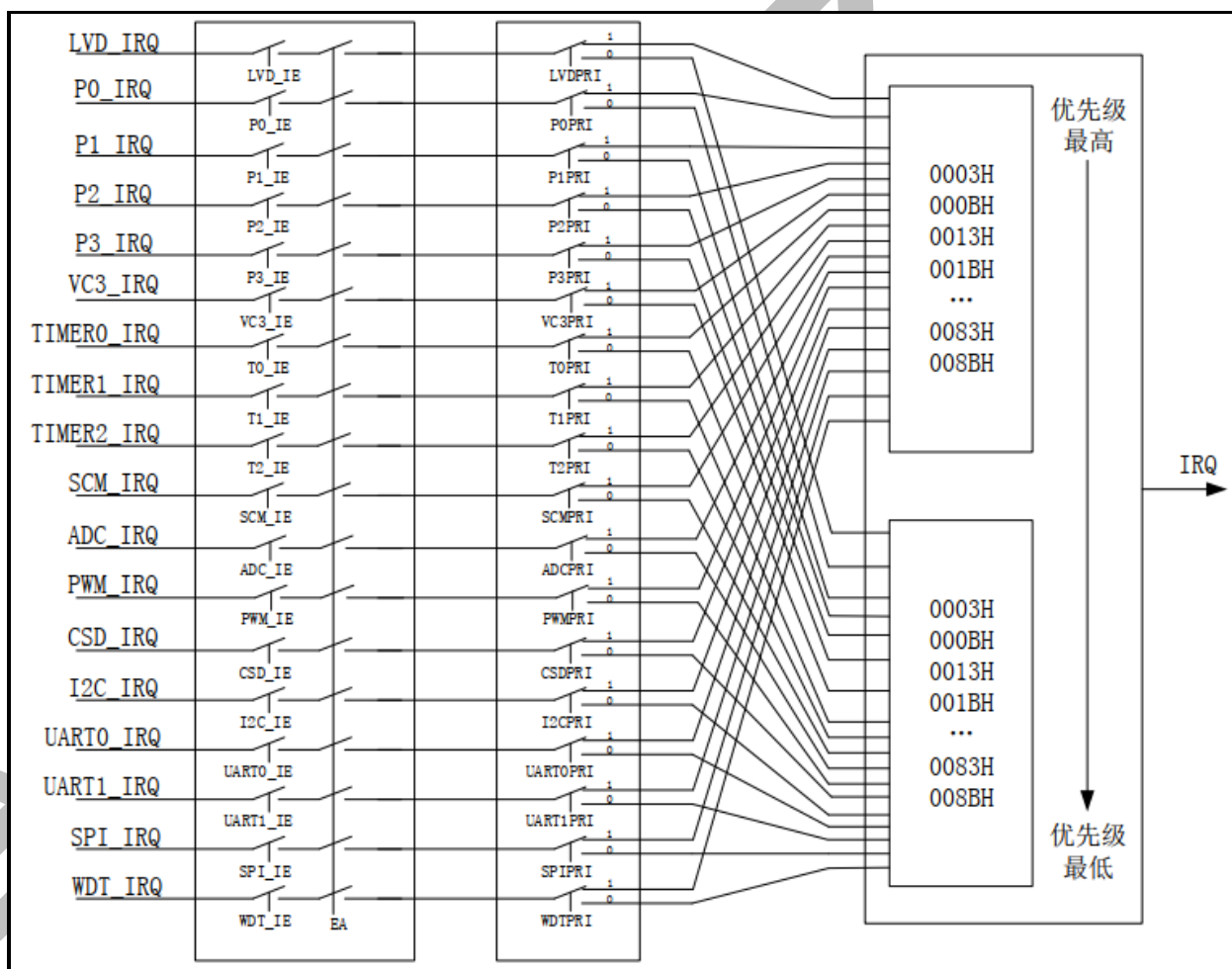


图 7-1 中断控制系统图

7.1 中断向量表

中断控制器支持 18 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	SCM	时钟检测中断	是	否	004BH
10	ADC	ADC 转换完成中断	是	否	0053H
11	PWM	PWM 溢出中断	是	否	005BH
12	CSD	CSD 中断	是	否	0063H
13	I2C	I2C 状态中断	是	否	006BH
14	UART0	UART0 状态中断	是	否	0073H
15	UART1	UART1 状态中断	是	否	007BH
16	SPI	SPI 中断	是	否	0083H
17	WDT	看门狗中断	是	是	008BH

7.2 中断优先级

每个中断有一个唯一的中断号。中断号越小，中断的优先级更高。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	UART1PRI	UARTOPRI	I2CPRI	CSDPRI	PWMPRI	ADCPRI	SCMPRI	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	UART1PRI	0 UART1 中断为低优先级 1 UART1 中断为高优先级
6	UARTOPRI	0 UART0 中断为低优先级 1 UART0 中断为高优先级
5	I2CPRI	0 I2C 中断为低优先级 1 I2C 中断为高优先级
4	CSDPRI	0 CSD 中断为低优先级 1 CSD 中断为高优先级

3	PWMPRI	0 PWM 中断为低优先级 1 PWM 中断为高优先级
2	ADCPRI	0 ADC 中断为低优先级 1 ADC 中断为高优先级
1	SCMPRI	0 SCM 中断为低优先级 1 SMC 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

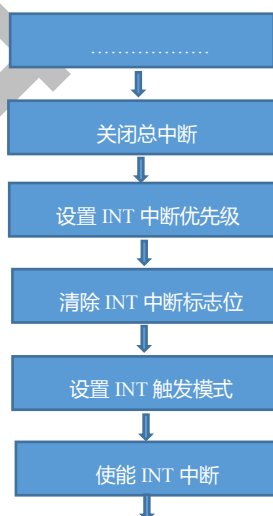
Bit	7	6	5	4	3	2	1	0
Name	-						WDTPRI	SPIPRI
Reset	-						0	0
Type	-						R/W	R/W

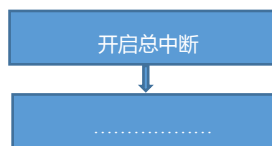
Bit	Name	Function
7:2	-	-
1	WDTPRI	0 WDT 中断为低优先级 1 WDT 中断为高优先级
0	SPIPRI	0 SPI 中断为低优先级 1 SPI 中断为高优先级

7.4 外部中断

RC6T80604x 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

7.6 外部中断操作示例

```

/*将 P0_0 设置为外部中断 0 中断输入引脚，开启外部中断函数，配置下降沿中断*/
P0_GE   = 0x00;
P0_DM0  = 0x00;  //P0_0 口配置数字高阻
P0_DM1  = 0x00;
P0_DM2  = 0x01;
P0_PU   = 0x01;  // P0_0 口配置上拉
P0_IE   = 0x01;  // P0_0 口配置中断使能
P0_IC0  = 0x00;  // P0_0 口配置下降沿中断
P0_IC1  = 0x01;

/*P0 口中断服务函数*/
void P0_IRQHandler(void) interrupt 1
{
    P0_FLAG = 0xFF;
}
  
```

8 存储器系统

8.1 存储器系统概述

RC6T80604x 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。

程序存储只能读不能写，程序存储器大小为 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。

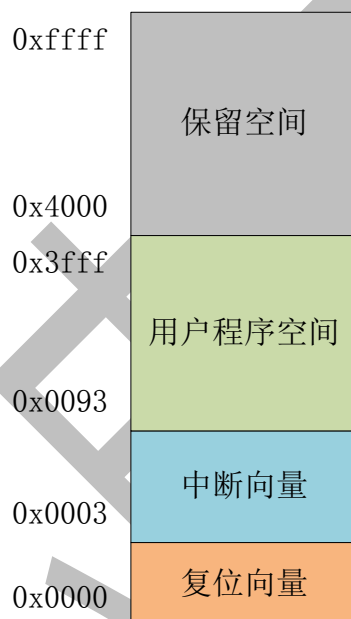


图 8-1 程序存储空间

复位后，MCU 从 0000H 开始执行。从 0003H 开始是中断向量表，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 and 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。

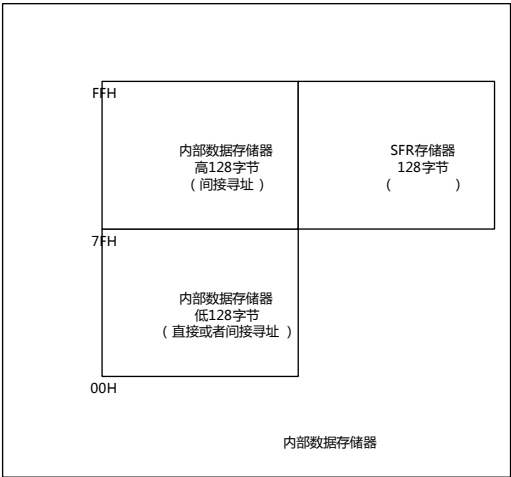


图 8-2 数据存储器

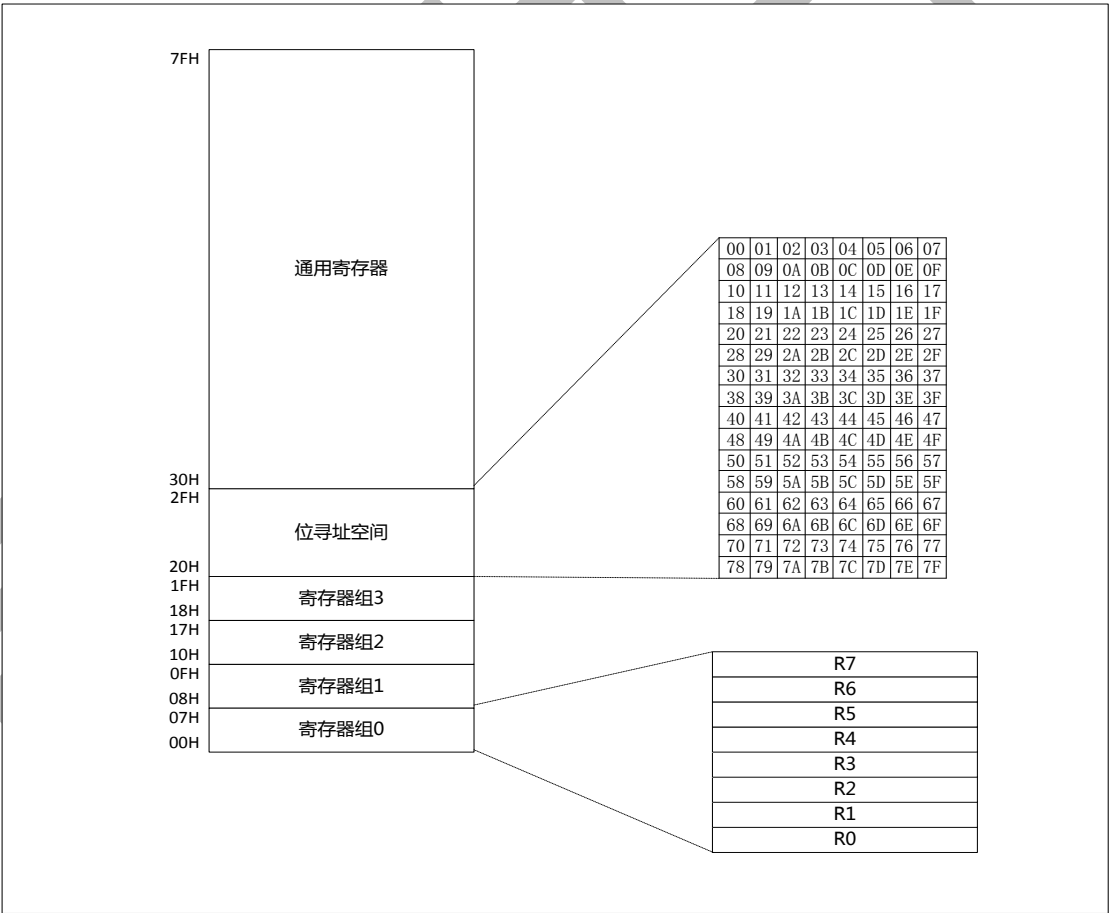


图 8-3 内部低 128 字节数据空间分配

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

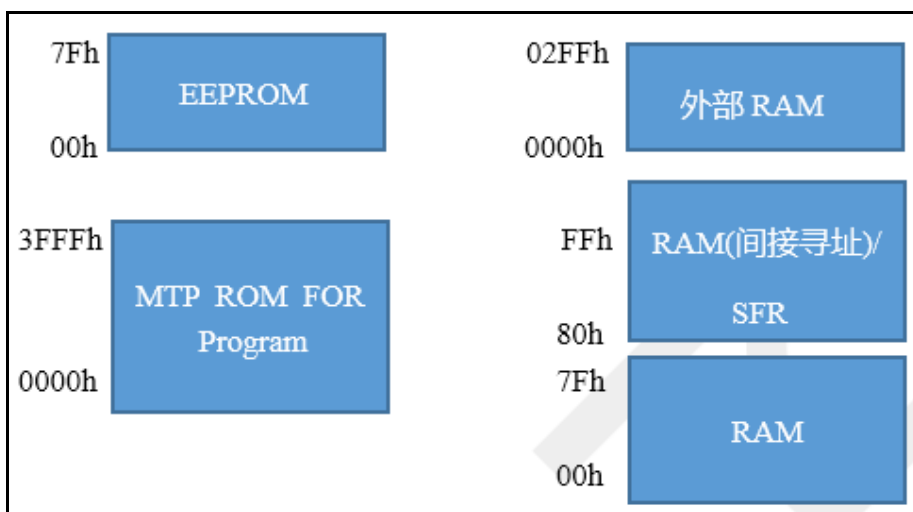


图 8-4 MTP 与 SRAM 结构图

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	SPI_ICR	SPI_DR	SPI_CR	SPI_STAT	SPI_CLK_CR	SPI_WRADDR	
E8h	ADC_CR0	ADC_CR1	ADC_CR2	ADC_CHSEL	ADC_CON	ADC_DLY	ADC_RES1	ADC_RES2
E0h	ACC	-	-	-	-	-	ADC_COMPL	ADC_COMPH
D8h	-	-	PWM_DUTY_2	PWM_DUTY_3	PWM_DUTY_4	PWM_DUTY_5	PWM_DF	PWM_BRAKE
D0h	PSW	PWM_CFG	PWM_CON	PWM_PRD	PWM_CR1	PWM_CR2	PWM_DUTY_0	PWM_DUTY_1
C8h	T2CON	-	RCAP2L	RCAP2H	TL2	TH2	-	-
C0h	-	-	-	-	-	-	-	-
B8h	IP	IP1	IP2	-	-	-	-	-
B0h	P3_DR	-	-	-	-	-	-	-
A8h	IE	-	-	-	-	-	-	-
A0h	P2_DR	I2C_ADDR	I2C_CR	I2C_STAT	I2C_DR	I2C_MCR	-	-
98h	SCON	SBUF	SCON1	SBUF1	-	-	-	-
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	-
80h	P0_DR	SP	DPL0	DPH0	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
FFC0H	-	-	-	-	-	-	-	-
FFB8H	-	-	-	-	-	-	-	-
FFB0H	CHKSUM_CR	CHKSUMH	CHKSUML	-	-	-	-	-
FFA8H	-	-	-	-	-	-	-	-
FFA0H	DISP_CR	DISP_VAL	-	-	-	-	-	-
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	-	-	-	-	-	-	-	-
FF70H	-	-	-	-	-	-	-	-
FF68H	-	-	-	-	-	-	-	-
FF60H	-	-	-	-	-	-	-	-
FF58H	P3_DM0	P3_DM1	P3_DM2	-	-	-	-	-
FF50H	P2_DM0	P2_DM1	P2_DM2	-	-	-	-	-
FF48H	P1_DM0	P1_DM1	P1_DM2	-	-	-	-	-
FF40H	P0_DM0	P0_DM1	P0_DM2	-	-	-	-	-
FF38H	-	-	-	-	-	-	-	-
FF30H	-	-	-	-	-	-	-	-
FF28H	P3_FLAG	-	P3_PU	-	-	P3_IE	P3_IC0	P3_IC1
FF20H	P2_FLAG	P2_GE	P2_PU	-	-	P2_IE	P2_IC0	P2_IC1
FF18H	P1_FLAG	P1_GE	P1_PU	-	-	P1_IE	P1_IC0	P1_IC1
FF10H	P0_FLAG	P0_GE	P0_PU	-	-	P0_IE	P0_IC0	P0_IC1
FF08H	PT_AFR	PT_AFR1-	-	-	-	-	-	-
FF00H	MTP_CR	MTP_CFG	MTP_KEY	MTP_ADL	MTP_ADH	MTP_PBUFL	MTP_PBUFH	MTP_DR

8.6 MTP 控制器

RC6T80604x 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 存储器的读时序和编程器通过编程接口编程 MTP 存储器。

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	FLASH 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	SAVPWR0	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.	Typ.	Max.	Unit
READ Cycle Time @ MTP / Information	Trc	$4.5V \leq VDD \leq 5.5V$	60	-	-	ns
		$1.8V \leq VDD < 4.5V$	300	-	-	ns
READ Cycle Time @ EEPROM		$1.8V \leq VDD \leq 5.5V$	300	-	-	ns
READ Access Time @ MTP / Information	Trac	$4.5V \leq VDD \leq 5.5V$	-	-	60	ns
		$1.8V \leq VDD < 4.5V$	-	-	300	ns
READ Access Time @ EEPROM		$1.8V \leq VDD \leq 5.5V$	-	-	300	ns

Read Operations: Low power mode

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

Read Operations: Power saving mode

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

图 8-1 MTP READ 时序要求

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。

8.7 MTP 操作示例

```

/* MTP 写操作：将地址 0x2FFE 写入 0x00 */
unsigned short MTP_address = 0x2FFE;           //设定操作 MTP 目标地址
unsigned char xdata IAP_MTP_data = 0;
MTP_KEY = 0xCA;                                //打开 MTP 写操作锁
MTP_PBUFL = ((unsigned short)IAP_MTP_data)%256;
MTP_PBUFH = ((unsigned short)IAP_MTP_data)/256;
MTP_ADL = MTP_address;
MTP_ADH = MTP_address >>8;
MTP_CFG = 0x03;                                //配置 MTP 操作周期
MTP_CR = 0xF9;                                 //开启 MTP 写操作
while(MTP_CR == 0x00);
MTP_KEY = 0x00;                                //MTP 上锁

/* MTP 读操作：读取地址 0x2FFE */
unsigned char MTP_rdbuf;
unsigned short MTP_address = 0x2FFE;
MTP_ADL = MTP_address;
MTP_ADH = MTP_address >>8;
MTP_CFG = 0x03;
MTP_CR = 0xC0;
MTP_rdbuf = MTP_DR;

```

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。RC6T80604x 系列芯片时钟源具备以下特性：

- 时钟分频，CPU 可以 1/2/4/8/16/32/64/256 分频
- 无毛刺的时钟切换
- 时钟源
 - ✓ 内部 24MHz IMO 振荡器
 - ✓ 外部晶体振荡器，可配置频率为 2~16MHz 或 32.768KHz
 - ✓ 内部 32KHz ILO 振荡器
 - ✓ 外部引脚时钟
- 复位后，CLK_SYS 使用默认内部的 24MHz RC 振荡器分频的 6MHz 内部时钟，在程序开始运行后用户可以自行修改频率；
- 时钟安全：如果外部主时钟停振，时钟自动切换到内部主时钟。
- 可以配置主时钟输出到 GPIO

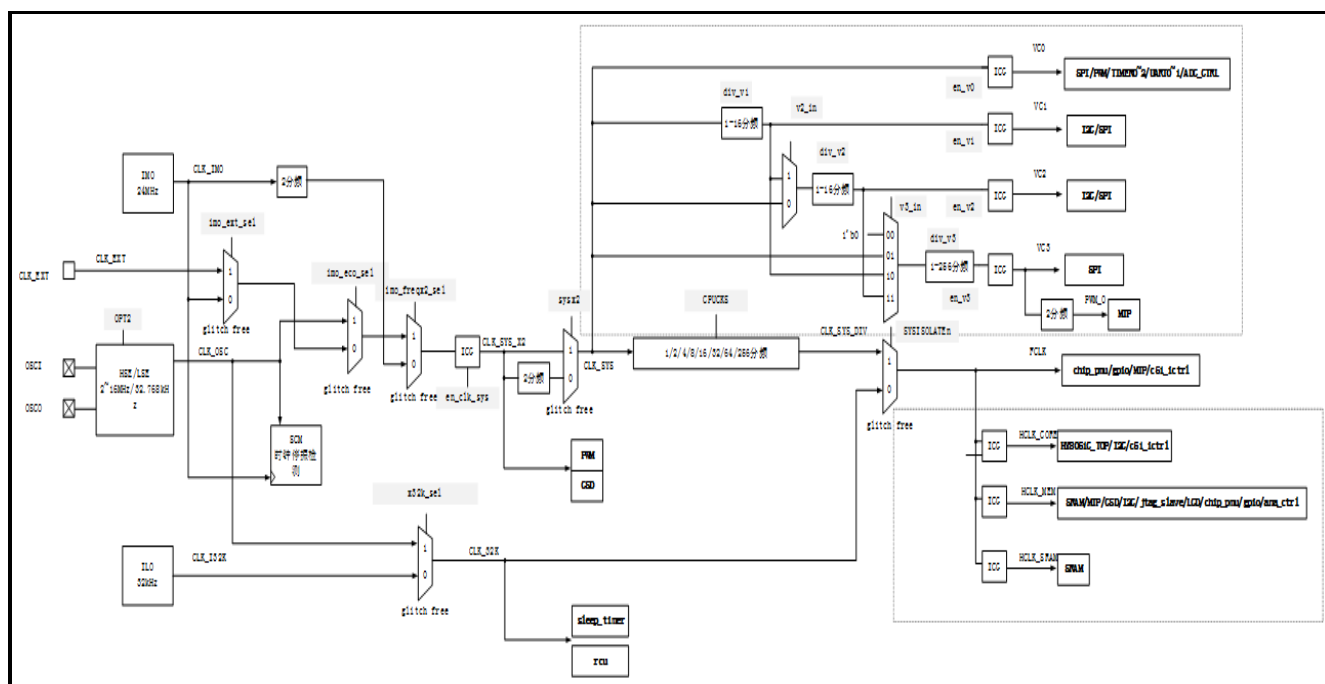
9.2 时钟信号说明

RC6T80604x 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	有三个来源，IMO 的时钟，外部引脚时钟，晶体振荡器的时钟。当使用内部 IMO（24MHz）做主时钟时，CLK_SYS 只能配置为 IMO 的 2 分频即 12MHz。当采用晶体振荡器的做时钟时，CLK_SYS 可以配置等于晶体振荡器的时钟，即可配置为 16MHz。
CLK_SYS_X2	两倍系统时钟，当使用 IMO 的 24MHz 时钟做主时钟时，CLK_SYS_X2 可以输出 24MHz 给 PWM 和 CSD 模块使用。
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，时钟源来自内部 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 打开并且周期约为 160KHz，否则 MTP 的烧录不可靠。

9.3 时钟结构框图



9.4 内部主振（IMO）

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 外部引脚时钟

外部引脚时钟可以用来替换 IMO 的时钟。P1.5 是外部的时钟输入引脚，当 P1.5 做外部时钟时，需要配置其工作模式为高阻数字输入模式。外部时钟频率范围是 2~16MHz，复位后外部时钟没有被选中。切换至外部时钟要注意先有外部时钟输入，再将时钟切换至外部引脚时钟。

9.7 外部晶体振荡器

外部晶体振荡器通过 OSC1 和 OSC0 与外部引脚连接，可以外接 2~16M/32.768kHz 时钟晶体谐振器/陶瓷谐振器。OSC1 和 OSC0 也是普通的 GPIO，当用作外部晶体振荡器时，需要配置这两个引脚的模式为高阻模拟输入模式。

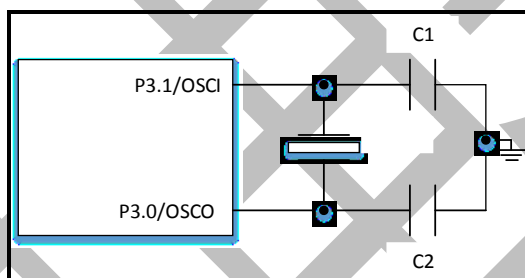


图 9-2 晶体振荡器

2~16M 时钟晶体谐振器/陶瓷谐振器通过 OSC1/OSC0 引入芯片内部，具体 C1/C2 负载电容选择需要参考所选择的谐振器的参数。通过配置 IMO_CR 寄存器可以将外部谐振器作为系统时钟的时钟源。

SCM 晶振时钟检测模块用来监测晶振时钟是否停振。当使用晶振时钟时可以使能 SCM 模块，当 SCM 检测到外部晶振时钟停振时会自动切换到内部 IMO 的四分频，即 FX2_SEL=0 和 SYSX2=0 的情况，同时上报中断。芯片复位后才可恢复时钟选择状态。SCM 使能后不能关闭 IMO，否则检测功能会失效。

9.8 时钟切换

芯片由外部晶体振荡器 16MHz 切换到内部 24MHz 的二分频时钟需要注意按照以下步骤进行：

1. 切换前要先将 IMO_CR 的 SYSX2 配置为 0，选择系统时钟二分频；
2. 延时 4 个系统时钟周期；
3. 配置 EXT_SEL=0，ECO_SEL=0，FX2_SEL=1，选择 24MHz 时钟分频输出；

9.10 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

9.10.1 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

9.10.2 PCLK_CR (addr: 0x95) 外设时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	VC0EN	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	VC0EN	0 禁止 VC0 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

9.10.3 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 $f_{VC2} = f_{CLK_SYS} / (VC2CKS + 1) / (VC1CKS + 1)$

9.10.4 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

RC6T80604x 系列芯片包含 3 个 16 位 Timer (Timer0/1/2), 每一个 Timer 都可以工作为定时器 (以 VC0 (1 2Mhz) 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。

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)
- (4) 2 个 8 位定时器/计数器 (模式 3, 仅 Timer0)

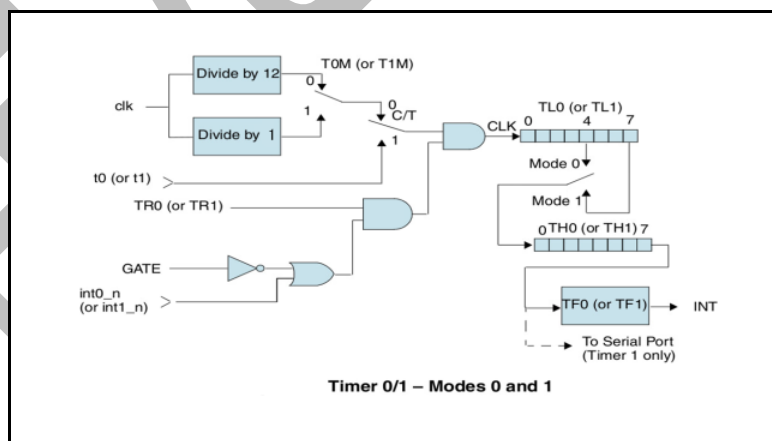
10.1.2 模式 0: 13 位定时器/计数器

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, 使用的是 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, 如若中断使能有效, 则会上报溢出中断至 CPU。

TR0/1 置 1 将打开定时器, 但是不会复位定时器, 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。

10.1.3 模式 1: 16 位定时器/计数器

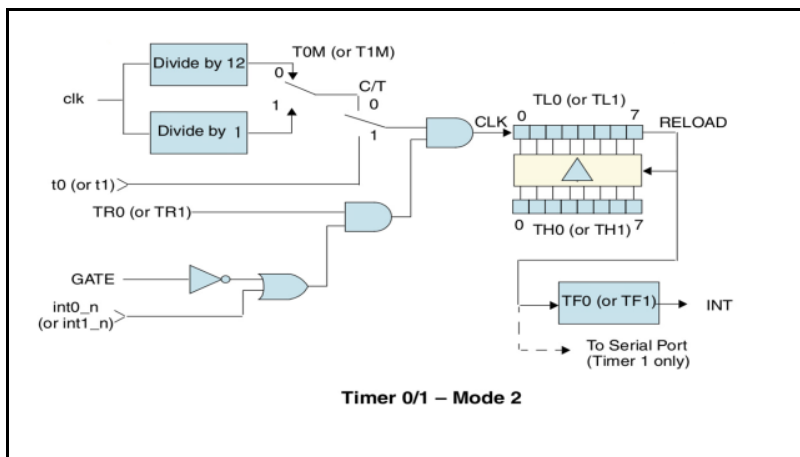
模式 1 的工作示意图如下图所示。Timer0 与 Timer1 都能实现该模式, 除了 TL 位使用了全部 8 位外, 该模式的工作方式与模式 0 相同。



10.1.4 模式 2: 8 位自动重载定时器/计数器

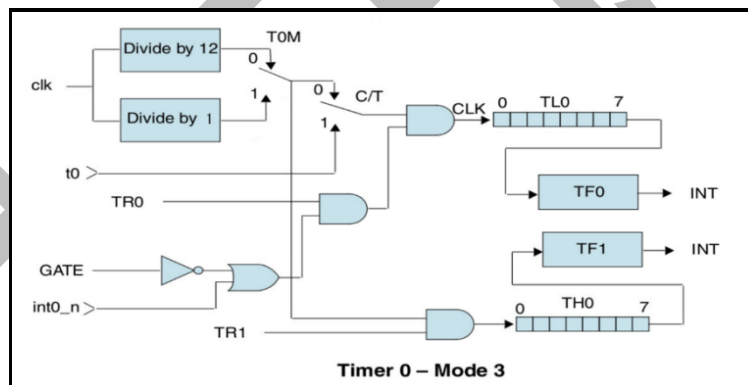
模式 2 的工作示意图如下图所示, Timer0 与 Timer1 都能实现该模式。在模式 2 中, TL 用于存放 Timer 计数值, TH 用于存放重载值, 当 TL 发生溢出时, 硬件会将 TH 中的值自动装载值 TL 中。除了自动重载功能

外，其他工作方式与模式 0/1 相同。



10.1.5 模式 3：2 个 8 位定时器/计数器（仅 Timer0）

模式 3 的工作示意图如下图所示，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。



10.1.6 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 的溢出事件以及外部引脚 T2EX 的下降沿事件（EXEN2 为 1）。外部引脚输入脉冲频率不得大于 Timer 工作时钟的 1/2。Timer2 通过系统时钟或外部引脚 T2EX 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
- 波特率生成器

10.1.7 16 位带捕获功能的定时器/计数器

通过配置 TIMER2_CON 的 CP/RL 位以及 EXEN2 位为 1 使得 Timer2 进入该模式。当外部引脚 T2EX 出现下降沿时，将会触发 Timer2 的捕获功能，硬件将此时的 Timer 值存储在 TIMER2_CAPL 和 TIMER2_CAPH 中。

TR2 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 TR2 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。

10.1.8 16 位自动重载的定时器/计数器

通过配置 TIMER2_CON 的 CP/RL 位为 0 使得 Timer2 进入该模式。在该模式下, Timer2 的溢出事件以及外部引脚 T2EX 出现下降沿时 (EXEN2 位为 1) 将会触发 Timer2 将 TIMER2_CAPL 和 TIMER2_CAPH 的值装载至 Timer2 中。

10.1.9 波特率生成器

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。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。

10.1.10 寄存器定义

名字	地址	读写	复位值	描述
TCON	88H	读写	00H	Timer0/1 控制寄存器
TMOD	89H	读写	00H	Timer0/1 工作模式配置寄存器
TL0	8AH	读写	00H	Timer0 计数值低 8 位
TH0	8BH	读写	00H	Timer0 计数值高 8 位
TL1	8CH	读写	00H	Timer1 计数值低 8 位
TH1	8DH	读写	00H	Timer1 计数值高 8 位
CKCON	8EH	读写	00H	时钟控制寄存器
T2CON	C8H	读写	00H	Timer2 控制寄存器
RCAP2L	CAH	读写	00H	Timer2 捕获/重载值低 8 位
RCAP2H	CBH	读写	00H	Timer2 捕获/重载值高 8 位
TL2	CCH	读写	00H	Timer2 计数值低 8 位
TH2	CDH	读写	00H	Timer2 计数值高 8 位

10.1.10.1 TMOD (addr:0x88) 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 作为计数器, Timer1 计数来源于外部管脚脉冲的下降沿
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 位自动重装载计数器, 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 作为计数器, Timer0 计数来源于外部管脚脉冲的下降沿
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 位自动重装载计数器, 溢出时将 TH0 存放的值自动重装入 TL0 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

10.1.10.2 TCON (addr:0x89) 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
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	T0_IE	Timer0 中断使能: 0 不上报中断 1 上报中断

10.1.10.3 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, 每写一次装载一次 读: 读该寄存器将会读出 Timer0 当前的低 8 位值

10.1.10.4 TH0 (addr:0x8B) 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，每写一次装载一次 读：读该寄存器将会读出 Timer0 当前的高 8 位值

10.1.10.5 TL1 (addr:0x8C) 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，每写一次装载一次 读：读该寄存器将会读出 Timer1 当前的低 8 位值

10.1.10.6 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，每写一次装载一次 读：读该寄存器将会读出 Timer1 当前的高 8 位值

10.1.10.7 CKCON (addr:0x8E) 时钟控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	-		T2M	T1M	T0M	-		
Reset	-		0	0	0	-		
Type	-		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
2:0	-	-

10.1.10.8 T2CON (addr:0xC8) Timer2 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	T2_C/T	CP/RL
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	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	EXF2	Timer2 外部引脚 T2EX 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	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	EXEN2	Timer2 外部输入事件用作捕获/重载触发器允许/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/重载
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	CP/RL	捕获/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 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

10.1.10.9 RCAP2L (addr:0xCA) Timer2 捕获/重载值低 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, 重载模式下, 该寄存器用于存放用于重载值的低 8 位

10.1.10.10 RCAP2H (addr:0xCB) Timer2 捕获/重载值高 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, 重载模式下, 该寄存器用于存放用于重载值的高 8 位

10.1.10.11 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

10.1.10.12 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

10.1.11 TIMER 配置流程



TIMER0/1/2 配置流程图

10.1.12 TIMER 操作示例

```

/*配置定时器 0 定时 1ms 进入中断*/
CKCON = 0x00;           //CLK/12 工作时钟 clk 的 12 分频 (SYS_CLK 需配置为 12MHz)
TMOD = 0x01;           //工作模式选择 16 位定时器/计数器
TL0 = (65536 - 1000)%256; //溢出时间: 时钟为 Fsys, 则 1000*(1/Fsys) =1ms;
TH0 = (65536 - 1000)/256;
TCON = 0x11;           //使能溢出中断, 开启定时器
IE_EA = 1;             //开启总中断
/*定时器 0 中断函数*/
void TIMER0_IRQ() interrupt 6
{
    TCON |= 0x20;        //清溢出标志位
    TL0 = (65536 - 1000)%256; //溢出时间: 时钟为 Fsys, 则 1000*(1/Fsys) =1ms;
    TH0 = (65536 - 1000)/256;
}
    
```

10.2 PWM

10.2.1 PWM 概述

RC6T80604x 系列芯片支持 6 通道 PWM，PWM 周期和占空比可在 10 位范围内任意配置。PWM 具备两种工作模式，支持死区控制和互补输出，PWM 的这种特性是专门针对无刷直流电机驱动设计的。其详细特性如下所示：

- 10 位 PWM 精度；
- 输出可设置正反向；
- 具有 2 种工作模式；
- 独立模式：PWM0~5 周期相同，但每一路 PWM 输出波形的占空比可单独设置；
- 互补模式：可同时输出三组互补、带死区的 PWM 波形；
- 提供 1 个 PWM 溢出中断和 1 个刹车中断；
- 支持刹车功能，刹车输入包括：ADC 输出，外部引脚 BKIN 输入；

10.2.2 功能说明

PWM 模块以 CLK_SYS_X2 作为模块工作时钟 clk。

RC6T80604x 系列芯片的 PWM 通过配置 PWM_PRD[9:0] 来控制六路 PWM 的计数周期，当计数器计至终点时会溢出标志位置 1，如若开启 PWM_IE 则会上报 PWM 溢出中断。通过配置 PWM_DUTY_x[9:0] 可以配置每路 PWM 的输出占空比。

外设提供独立模式和互补模式 2 种工作模式，可通过配置 PWM_CR2 的第 6 位（PWM_MODE）来切换。在独立模式下，使用本外设可用上述配置方法操作，在互补模式下，按 PWM0/3，PWM1/4，PWM2/5 分为 3 组，每组互补输出的占空比分别由寄存器 PWM_DUTY_0~2 进行控制，并可通过寄存器 PWM_DF 设定死区时间。

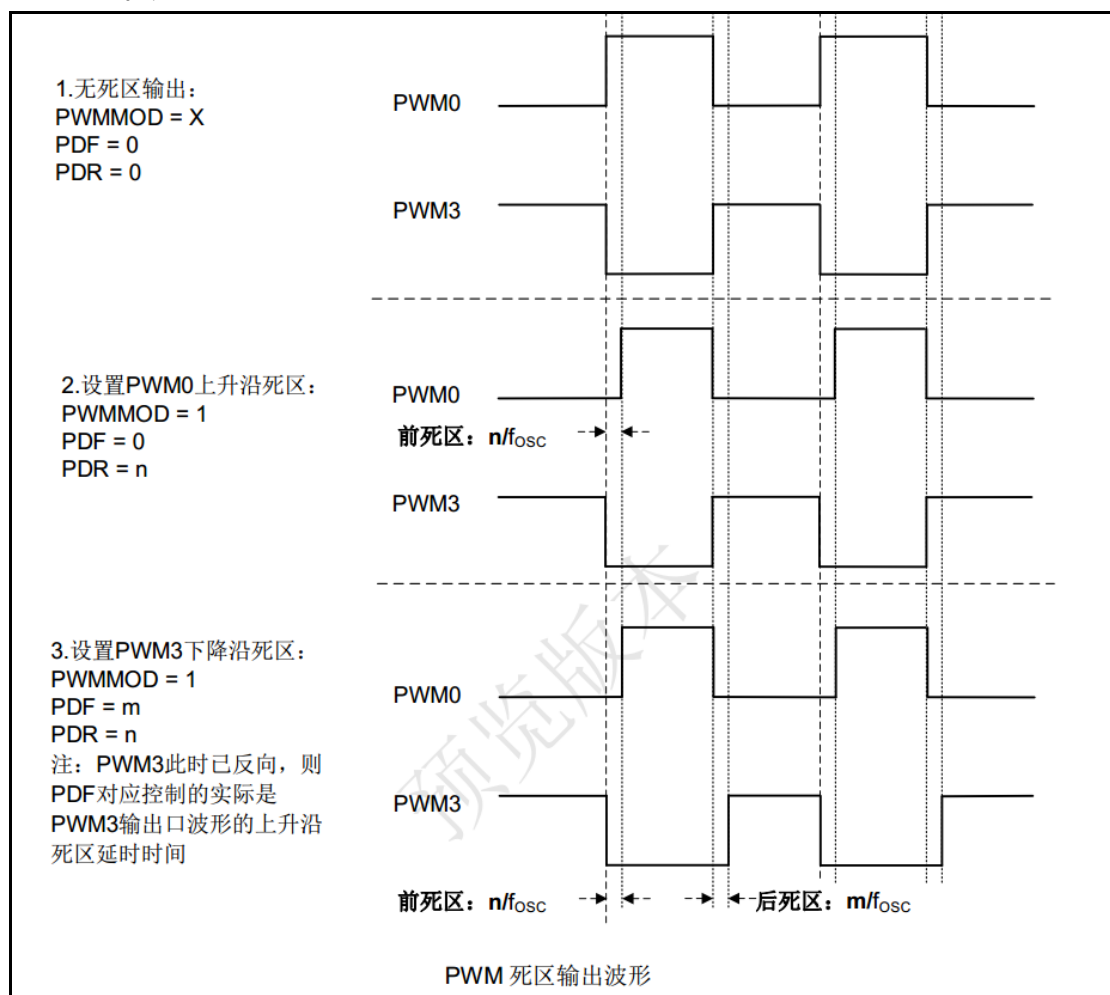
在使用本 PWM 时，建议先配置好所需的 PWM 功能最后再使能 PWM_EN，在计数器工作过程中对 PWM 的周期、占空比、死区进行配置的话，配置值会在下一个计数周期生效。

RC6T80604x 系列芯片的 PWM 功能可以对端口的输出状态进行保护控制。共有 2 类端口输入无效事件（来自 ADC 或外部 BKIN），每个接口上选通的异常状况事件可从刹车控制设定（PWM_CON 寄存器决定刹车使能和刹车事件来源），当这些接口上监测到异常状况时，可以实现对通用 PWM 输出的控制。若开启 PWM 刹车控制中断使能则会在刹车事件产生时触发中断。

对于外部异步刹车输入源 BKIN，有供配置的输入信号滤波功能，若使能滤波功能，则连续 3 个模块工作时钟 clk 周期采集的 BKIN 状态保持稳定则取信其信号状态，否则对其状态跳变进行滤波，不取信其状态变化。

GPIO 端口作为通用 PWM 输出端口在刹车控制异常事件发生时，各端口状态可以变为输出低电平或输出高电平（由 PWM_CON 寄存器对应的 BRAKE_DM5~0 配置决定）。

10.2.3 工作波形



上图是以 PWM0 和 PWM3 在互补模式下的死区时间调整波形图, 为了便于区分, PWM3 已取反。

10.2.4 寄存器定义

名字	地址	读写	复位值	描述
PWM_CFG	D1H	读写	00H	PWM 设置寄存器
PWM_CON	D2H	读写	00H	PWM 控制寄存器
PWM_PRD	D3H	读写	00H	PWM 周期设置寄存器
PWM_CR1	D4H	读写	00H	PWM 占空比设置寄存器 1
PWM_CR2	D5H	读写	00H	PWM 占空比设置寄存器 2
PWM_DUTY_0	D6H	读写	00H	PWM0 占空比设置寄存器
PWM_DUTY_1	D7H	读写	00H	PWM1 占空比设置寄存器
PWM_DUTY_2	DAH	读写	00H	PWM2 占空比设置寄存器
PWM_DUTY_3	DBH	读写	00H	PWM3 占空比设置寄存器
PWM_DUTY_4	DCH	读写	00H	PWM4 占空比设置寄存器
PWM_DUTY_5	DDH	读写	00H	PWM5 占空比设置寄存器

PWD_DF	DEH	读写	00H	PWM 死区周期配置寄存器
PWM_BRAKE	DFH	读写	00H	PWM 刹车控制配置寄存器

10.2.4.1 PWM_CFG (addr:0xD1) PWM 设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	PWM_CLK		INV5	INV4	INV3	INV2	INV1	INV0
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
7:6	PWM_CLK	PWM 内部时钟分频选择: 00 模块工作时钟 clk 01 模块工作时钟 clk/2 10 模块工作时钟 clk/8 11 模块工作时钟 clk/32
5:0	INV (0~5)	PWM0~5 输出反向控制 0 输出不反向 1 输出反向 注: 在互补模式下, 每一组互补输出中需要将其中一路输出配置为反向

10.2.4.2 PWM_CON (addr:0xD2) PWM 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	PWM_EN	PWM_IF	BRAKE_DM5	BRAKE_DM4	BRAKE_DM3	BRAKE_DM2	BRAKE_DM1	BRAKE_DM0
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	PWM_EN	PWM 外设开关 0 关闭 1 使能 PWM 外设 注: 完成 PWM 所需的配置之后再使能 PWM, 将 PWM_EN 配 0 后计数器的值将会清零
6	PWM_IF	PWM 溢出标志位 当计数器计数超过设定的计数值时, 该位会被硬件设定为 1, 对该位写 1 清零
5:0	BRAKE_DM (0~5)	PWM 刹车事件产生的输出状态控制 0 输出低电平 1 输出高电平

10.2.4.3 PWM_PRD (addr:0xD3) PWM 周期设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	PWM_PRD[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	PWM_PRD[9:2]	PWM0~PWM5 共用的周期配置值的高 8 位, 实际输出的 PWM 周期数为 (PWM_PRD[9:0]+1)

10.2.4.4 PWM_CR1 (addr:0xD4) PWM 占空比设置寄存器 1

Bit	7	6	5	4	3	2	1	0
Name	PWM_PRD[1:0]		DUTY_2[1:0]		DUTY_1[1:0]		DUTY_0[1:0]	
Reset	0		0		0		0	
Type	R/W		R/W		R/W		R/W	

Bit	Name	Function
7:6	PWM_PRD[1:0]	PWM0~PWM5 共用的周期配置值的低 2 位
5:4	DUTY_2[1:0]	PWM2 输出占空比配置的低 2 位
3:2	DUTY_1[1:0]	PWM1 输出占空比配置的低 2 位
1:0	DUTY_0[1:0]	PWM0 输出占空比配置的低 2 位

注: 为保证数据的正确生效, 必须遵守先写 DUTY_2/DUTY_1/DUTY_0/PWM_PRD 的高 8 位, 再写低 2 位的要求, 写完低 2 位后才生效。

10.2.4.5 PWM_CR2 (addr:0xD5) PWM 占空比设置寄存器 2

Bit	7	6	5	4	3	2	1	0
Name	PWM_IE	PWM_MODE	DUTY_5[1:0]		DUTY_4[1:0]		DUTY_3[1:0]	
Reset	0	0	0		0		0	
Type	R/W	R/W	R/W		R/W		R/W	

Bit	Name	Function
7	PWM_IE	PWM 中断使能 0 在计数器发生溢出时不上报中断 1 在计数器发生溢出时上报中断
6	PWM_MODE	PWM 工作模式配置 0 独立模式: PWM0~5 使用同一计数周期及各自的占空比设置 1 互补模式: 按 PWM0/3, PWM1/4, PWM2/5 分为 3 组, 同组 PWM 的输出波形互补, 每组波形的占空比分别由 PWM_DUTY_0~2 进行控制, 并可通过寄存器 PWM_DF 设定死区时间
5:4	DUTY_5[1:0]	PWM5 输出占空比配置的低 2 位
3:2	DUTY_4[1:0]	PWM4 输出占空比配置的低 2 位
1:0	DUTY_3[1:0]	PWM3 输出占空比配置的低 2 位

注: 为保证数据的正确生效, 必须遵守先写 DUTY_5/DUTY_4/DUTY_3 的高 8 位, 再写低 2 位的要求, 写完低 2 位后才生效。

10.2.4.6 PWM_DUTY_0 (addr:0xD6) PWM0 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_0[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_0[9:2]	PWM0 输出占空比配置的高 8 位, DUTY_x[9:0]的值为 PWM 高电平持续的周期数

10.2.4.7 PWM_DUTY_1 (addr:0xD7) PWM1 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_1[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_1[9:2]	PWM1 输出占空比配置的高 8 位

10.2.4.8 PWM_DUTY_2 (addr:0xDA) PWM2 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_2[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_2[9:2]	PWM2 输出占空比配置的高 8 位

10.2.4.9 PWM_DUTY_3 (addr:0xDB) PWM3 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_3[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_3[9:2]	PWM3 输出占空比配置的高 8 位

10.2.4.10 PWM_DUTY_4 (addr:0xDC) PWM4 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_4[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_4[9:2]	PWM5 输出占空比配置的高 8 位

10.2.4.11 PWM_DUTY_5 (addr:0xDD) PWM5 占空比设置寄存器

Bit	7	6	5	4	3	2	1	0
Name	DUTY_5[9:2]							
Reset	0x00							
Type	R/W							

Bit	Name	Function
7:0	DUTY_5[9:2]	PWM5 输出占空比配置的高 8 位

10.2.4.12 PWM_DF (addr:0xDE) PWM 死区周期配置寄存器

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

Bit	Name	Function
7	PDF	互补模式下反向输出 PWM 波下降沿死区周期值
6	PDR	互补模式下正向输出 PWM 波上升沿死区周期值

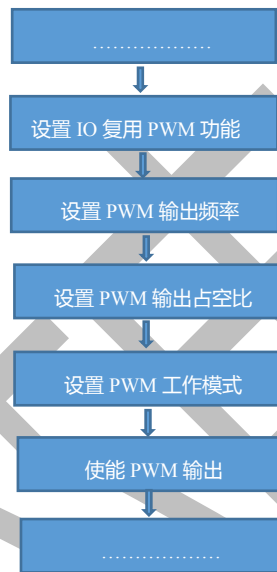
10.2.4.13 PWM_BRAKE (addr:0xDF) PWM 刹车控制配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	BRAKE_IE	BRAKE_IF	BRAKE_FILTER_EN	BKIN_LVL	PWM_MOE	BRAKE_AOE	BRAKE_SEL	BRAKE_EN
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	BRAKE_IE	PWM 刹车中断使能 0 刹车事件不产生中断 1 刹车事件产生中断
6	BRAKE_IF	PWM 刹车中断标志位 当产生刹车事件时, 该位会被硬件设定为 1, 对该位写 1 清零
5	BRAKE_FILTER_EN	PWM 外部 BKIN 管脚输入刹车事件滤波使能 0 BKIN 管脚输入刹车事件滤波不使能 1 BKIN 管脚输入刹车事件滤波使能
4	BKIN_LVL	BKIN 管脚电平选择寄存器 0 BKIN 低电平时刹车有效 1 BKIN 高电平时刹车有效
3	PWM_MOE	PWM 主输出使能 (PWM 输出时, 需要使能该位, 使能该位后只由刹车事件复位置 0, 不可由软件写 0) 刹车事件有效时, 会立即被同步清零。根据 BRAKE_AOE 的选择, 通过软件置 1 或硬件 PWM 计数器溢出自动置 1 1 PWM 主输出使能 0 PWM 主输出关闭
2	BRAKE_AOE	自动输出使能 0 有刹车事件产生时, PWM_MOE 只被软件置 1

		1 有刹车事件产生时，PWM _MOE 可被软件和硬件 PWM 计数器溢出事件置 1
1	BRAKE_SEL	选择 PWM 刹车来源 0 刹车事件选择 ADC 比较输出 1 刹车事件选择 BKIN 输入
0	BRAKE_EN	PWM 刹车功能使能 0 PWM 刹车功能不使能 1 PWM 刹车功能使能

10.2.5 PWM 配置流程



PWM 配置流程示意图

10.2.6 PWM 操作示例

```

/*配置 PWM0 工作独立模式，输出频率为 4K 的方波*/
P0_DM0 = 0x01;    // PWM0 P0.0 配置推挽输出
P0_DM1 = 0x00;
P0_DM2 = 0x00;
P0_GE  = 0x01;

PWM_CFG = 0xC0;    //时钟 CLK/32，输出不反向(系统时钟配置为 12MHz)
PWM_PRD = 0x2E;    //六路 PWM 共用的周期配置值高 8 位
PWM_DUTY_0 = 0x17; //PWM0 输出占空比配置的高 8 位
PWM_CR1 = 0x00;    //PWM0 输出占空比配置第 2 位为 0
PWM_CR2 = 0x80;    //独立模式
PWM_BRAKE = 0x08;  //PWM 主输出使能
PWM_CON |= 0x80;   //使能 PWM 外设
  
```

10.3 CHECKSUM

10.3.1 CHECKSUM 概述

RC6T80604x 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。

10.3.2 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

10.3.2.1 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	-	-
6	CHKSUM_BUSY	CHKSUM 运算状态 1 CHKSUM 模块正在运行 0 CHKSUM 模块没有运行
5	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。

10.3.2.2 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

10.3.2.3 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					

10.4 LCD

10.4.1 LCD 概述

RC6T80604x 系列芯片内置 LCD 驱动模块, 驱动最大可支持 5com * 20seg, 软件 LCD 支持软件控制 COM 口, 可以输出 1/2VDD 电压。支持三种分压电阻, 分别为 12.5K/37.5K/87.5K。

用户在使用软件 LCD 时, 需要先配置 DISP_CR 来使能 LCD 并选择对应分压电阻, 然后配置 DISP_VAL 来显示内容。

注意: 当 GPIO 做 COM 口时必须配置为高阻模拟输入模式。

10.4.2 寄存器定义

名字	地址	读写	复位值	描述
DISP_CR	0xFFA0	读写	00H	LCD 功能配置寄存器
DISP_VAL	0xFFA1	读写	00H	LCD 功能 COM 口开关配置寄存器

10.4.2.1 DISP_CR (addr:0xFFA0) LCD 功能配置寄存器

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

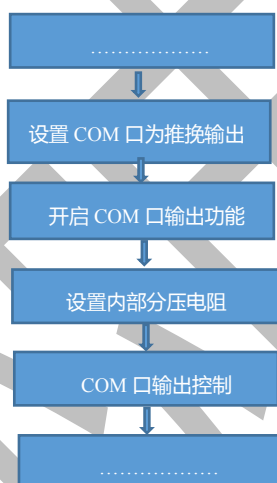
Bit	Name	Function
7:2	-	-
1:0	VLOSEL	LCD 电压输出口分压电阻选择 0 关闭分压电阻 1 分压电阻为 12.5K 2 分压电阻为 37.5K 3 分压电阻为 87.5K

10.4.2.2 DISP_VAL (addr:0xFFA1) LCD 功能 COM 口开关配置寄存器

Bit	7	6	5	4	3	2	1	0
Name	-			COM4	COM3	COM2	COM1	COM0
Reset	-			0	0	0	0	0
Type	-			R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7:5	-	-
4:0	COM4~0	COM 输出选择: 1 COM 口输出 1/2VDD 开关使能 0 COM 口作为普通 I/O 口

10.4.3 LCD 配置流程



LCD 配置流程示意图

10.5 ADC

10.5.1 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 10 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，比较结果可用于触发 PWM 故障刹车。

10.5.2 结构框图

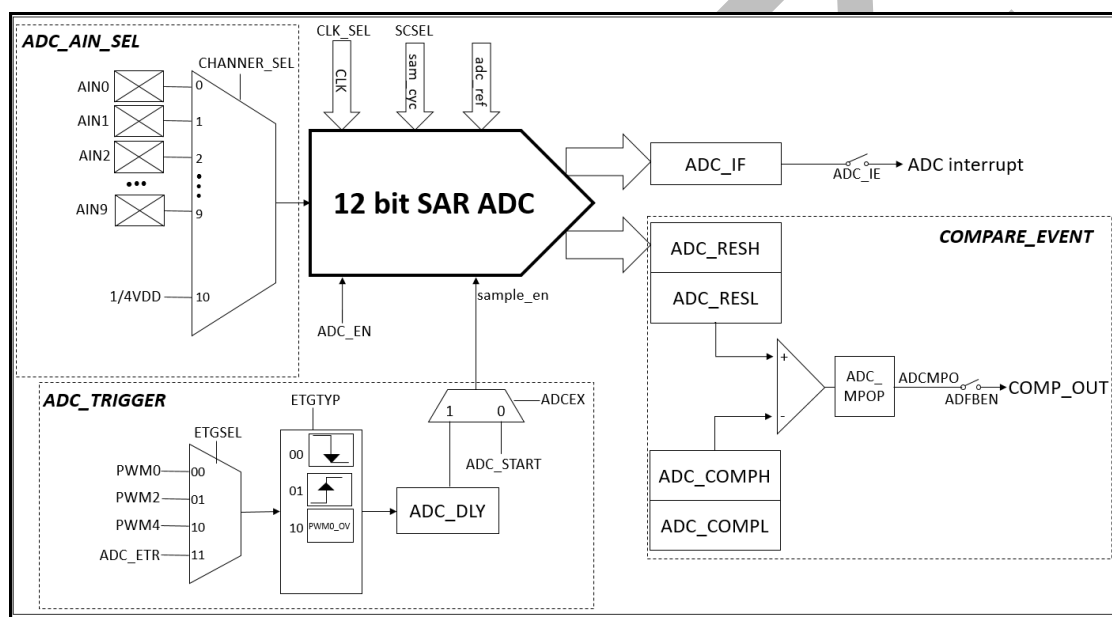


图 10-5-1 ADC 结构框图

10.5.3 ADC 转换时序

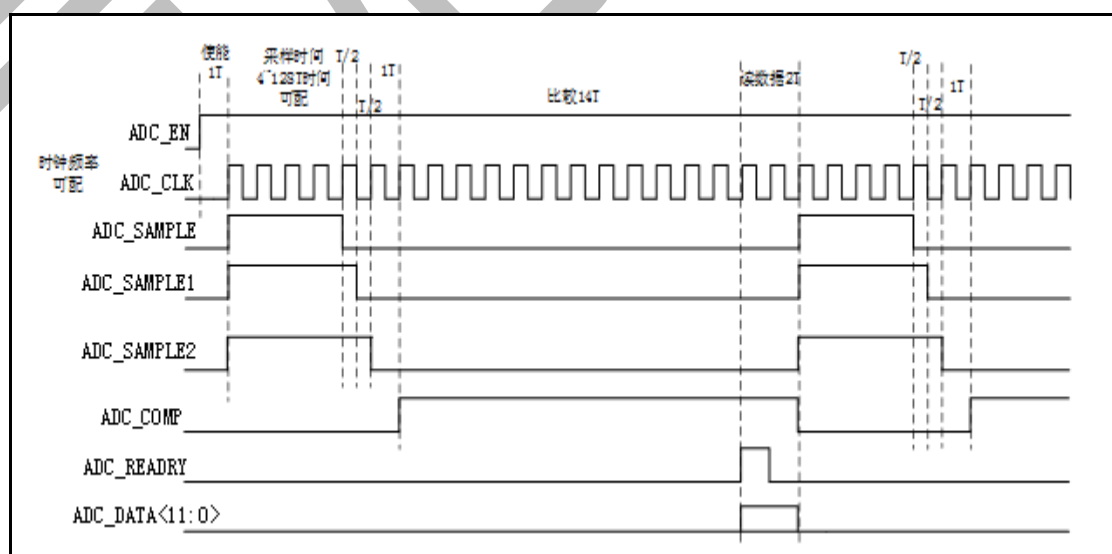


图 10-5-2 ADC 时序图

10.5.4 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	0AH	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_RES_L	0xEE	读	00H	ADC 转换结果低位寄存器
ADC_RES_H	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_COMP_H	0xE7	读写	00H	ADC 比较值高 8 位

10.5.4.1 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	—	ADC_START	ADC_IF	ADC_IE	ADCEX	CLKSEL[1:0]	
Reset	0	—	0	0	0	0	0	
Type	R/W	—	R/W	R/W	R/W	R/W	R/W	

Bit	Name	Function
7	ADC_EN	ADC 使能位： 0 = ADC 转换电路关闭 1 = ADC 转换电路开启
6	—	—
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
1:0	CLKSEL[1:0]	ADC 时钟选择： 00 = 系统时钟的 4 分频

		01 = 系统时钟的 8 分频 10 = 系统时钟的 16 分频 11 = 系统时钟的 32 分频 默认值: 00 注意: 修改 CLK_SEL 寄存器配置必须在 ADC_EN 为 0 的时候进行。
--	--	--

10.5.4.2 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 00 PWM0 触发 01 PWM2 触发 10 PWM4 触发 11 ADC_ETR 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 10 PWM 计数器溢出 (仅在 ETGSEL 配置为 00: PWM0 触发时有效) 11 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。

10.5.4.3 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 增益选择: 0 = 参考 buffer 输出是参考的 2 倍; 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; 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

10.5.4.4 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	-				CHANNEL_SEL			
Reset	-				1	0	1	0
Type	-				R/W	R/W	R/W	R/W

Bit	Name	Function
7:4	-	-
3:0	CHANNEL_SEL	ADC 模拟量输入通道选择: 0000 = 通道 0 0001 = 通道 1 0010 = 通道 2 0011 = 通道 3 0100 = 通道 4 0101 = 通道 5 0110 = 通道 6 0111 = 通道 7 1000 = 通道 8 1001 = 通道 9 1010 = 片内电源输入通道 默认值: 1010 其他值: 保留

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

Bit	7	6	5	4	3	2	1	0
Name	ADFBEN	ADCMPOP	ADCMPEM	ADCMPO	CLEAR	-		ADCPLY8
Reset	0	0	0	0	0	-		0
Type	R/W	R/W	R/W	R/W	R/W	-		R/W

Bit	Name	Function
7	ADFBEN	ADC 比较结果响应故障刹车使能寄存器: 0 = 关闭 1 = ADC 触发故障刹车功能打开 默认值: 0--
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	ADCMPEM	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	ADCPLY8	ADC 外部触发延时计数器数值的高 1 位 默认值为 0

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

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

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

10.5.4.7 ADC_RES (addr:0xEE) ADC 转换结果低位寄存器

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

Bit	Name	Function
7:4	-	-
3:0	ADC_RES	ADC 转换结果低 4 位。

10.5.4.8 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。

10.5.4.9 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			

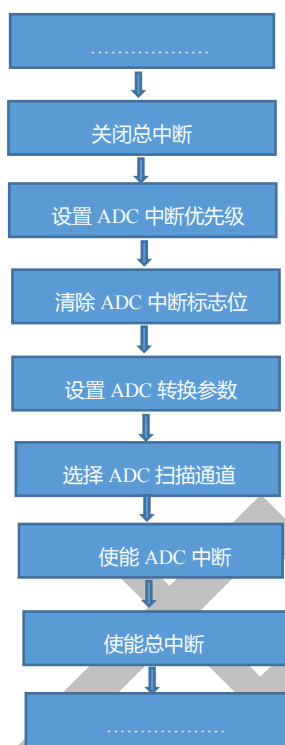
Bit	Name	Function
7:4	-	-
3:0	ADC_COMPL	ADC 比较值低 4 位。

10.5.4.10 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。

10.5.5 ADC 配置流程



ADC 配置流程示意图

10.5.6 ADC 操作示例

```

/*配置模拟输入通道 0 (P2.2)，系统时钟 8 分频*/
P2_DM0 = 0x00;
P2_DM1 = 0x00;
P2_DM2 = 0x00;
ADC_CHSEL = 0x00;    //通道 0 P2.2

ADC_CR0 = 0x81;       //ADC 时钟选择--系统时钟 8 分频，软件触发 ADC 采样
ADC_CR1 = 0x02;       //4 个 ADC 周期
ADC_CR2 = 0x03;       //选择内部 BG 电压 2.4V
ADC_CR0 |= 0x10;      //清除 ADC 中断标志
ADC_CR0 |= 0x08;      //使能 ADC 中断
ADC_CR0 |= 0x20;      //开始 ADC 转换

void ADC_IRQHandler() interrupt 10
{
    ADC_CR0 |= 0x10; //清除采样完成标志, 写 1 清 0
    ADC_CR0 |= 0x20; //开始 ADC 转换
}
    
```

10.6 UART0/1

10.6.1 UART 概述

RC6T80604x 系列产品串口通信模块提供 2 个可选的串口 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。

10.6.2 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：

$$\text{Baud Rate} = \frac{2^{\text{SMODx}}}{32} \times \text{Timer 1 Overflow}$$

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：

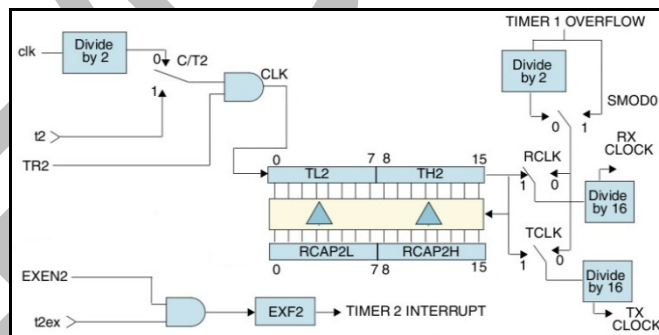
$$\text{TH1} = 256 - \frac{2^{\text{SMODx}} \times \text{clk}}{384 \times \text{Baud Rate}}$$

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：

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

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。



可通过下式获得已知波特率下的 RCAP2H 和 RCAP2L 值：

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

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 份计数器翻转后开始数据发送。UAR

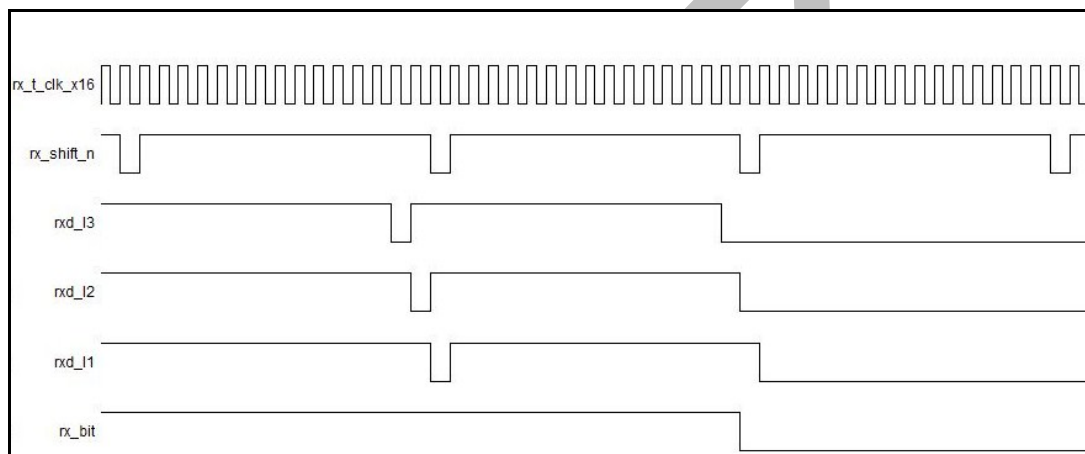
T 根据以下顺序在 txd0（或 txd1）引脚发送数据：起始位，8 位数据位（低位在前），停止位。TI_0（或 TI_1）位在停止位发送完成后的 1 个时钟周期后被置位。

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 份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 位接收时间段中间的三个连续采样，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。

RC6T80604x 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。

10.6.3 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

10.6.3.1 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。

10.6.3.2 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

10.6.3.3 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	RB8_1	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_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。

10.6.3.4 SBUF1 (addr:0x9B) 串口 1 缓存

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

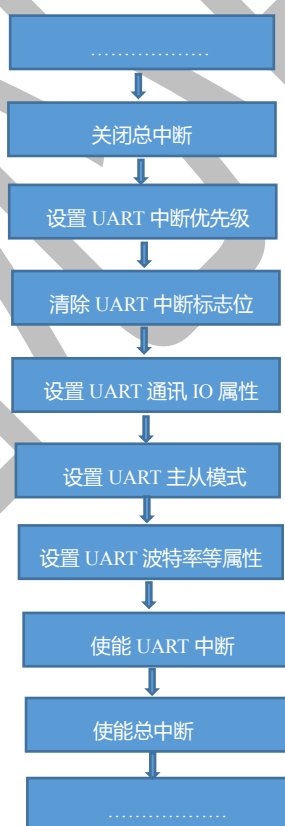
Bit	Name	Function
7:0	SBUF1	串口 1 数据缓存

10.6.3.5 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	SPI_EN	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	BKIN_EN	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

10.6.4 UART 配置流程



UART 配置流程示意图

10.6.5 UART 操作示例

```

/*UART0 波特率 9600, 时钟源 TIMER1, 接收模式*/
PT_AFR  = 0x01;  //UART0 管脚位置使能
PT_AFR1 = 0x00;  //Tx=P1.3, Rx=P1.2
P1_GE   = 0x0C;
P1_DM0  = 0x08;  // P1.3 Output mode, P1.2 Input mode
P1_DM1  = 0x00;
P1_DM2  = 0x04;
P1_PU   = 0x00;

CKCON  = 0x10;  //TIMER1 时钟来源为 clk=12M
TMOD   = 0x20;  //TIMER1 模式 8 位自动重载计数器
SCON0  = 0x50;  //UART0 模式 1 异步全双工 接收使能 Clear Uart0 State Register
PCON   = 0x02;  //打开串口中断使能, 关闭波特率倍增使能
TL1    = 217;
TH1    = 217;  //12M 9600
TCON  |= 0x40;  //启动 Timer1
IE_EA  = 1;    //开启总中断

/*中断处理函数*/
void Uart_Interrupt() interrupt 14
{
    if((SCON0 & 0x02) == 0x02)  //发送完成
    {
        SCON0 &= 0xFD;
    }
    else if((SCON0 & 0x01) == 0x01)  //接收完成
    {
        /*存储接收到的数据*/
        .....
        SCON0 &= 0xFE;
    }
}

```

10.7 SPI

10.7.1 SPI 概述

RC6T80604x 系列芯片提供标准 SPI 串行外设接口，未提供片选信号引脚，其主要特性如下所示：

- 1 最大发送速率：8Mbit/s
- 2 最大接收速率：8Mbit/s
- 3 全双工；
- 4 支持主机模式和从机模式；
- 5 支持可编程的同步时钟极性和相位控制；
- 6 支持数据传输高位优先和低位优先；
- 7 4 级发送和 4 级接收 FIFO；
- 8 主机传输速率为 (SPI 工作时钟/2)；
- 9 支持独立中断；
- 10 支持 RZ 码调制功能。

10.7.2 操作说明

10.7.2.1 SPI 主机

- (1) 配置 GPIO 为 SPI 模式；
- (2) 配置 SCK 时钟分频以及 SPI_CLK_CR 用于配置 SPI 输出时钟的频率；
- (3) 配置 SPI_DR 寄存器准备用于发送的数据；
- (4) 配置 SPI_ICR 寄存器用于选择中断上报事件；
- (5) 配置 SPI_CR 寄存器用于配置 SPI 主机模式；
- (6) 配置 SPI_CR 寄存器使能 SPI 外设。

10.7.2.2 SPI 从机

- (1) 配置 GPIO 为 SPI 模式；
- (2) 配置 SPI_DR 寄存器准备用于发送的数据；
- (3) 配置 SPI_ICR 寄存器用于选择中断上报事件；
- (4) 配置 SPI_CR 寄存器用于配置 SPI 从机模式；
- (5) 配置 SPI_CR 寄存器使能 SPI 外设。

注意事项：在中断触发的 SPI_DR 写操作事件中，在系统时钟配置为 16M 条件下需要至少 4us 才能第一次进行中断服务程序里的写 SPI_DR 操作。若 FIFO 已发空而外部主机时钟继续发送，本 SPI 从机将会循环发出最后 2Byte 数据。

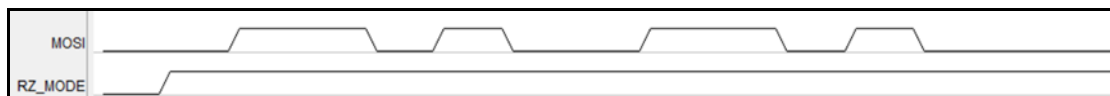
10.7.2.3 RZ 码调制功能

- (1) 配置 GPIO 为 RZ 码模式 (RZ_PEN 为 1)，除 MOSI 端口外，其他 3 个 SPI 端口可作普通 IO 使用；
- (2) 配置 SCK 时钟分频以及 SPI_CLK_CR 用于配置 RZ 码输出时钟的频率；
- (3) 配置 SPI_DR 寄存器准备用于发送的数据；
- (4) 配置 SPI_ICR 寄存器用于选择中断上报事件；

第 83 页 共 120 页

(5) 配置 SPI_CR 寄存器将 SPI 配置为 RZ 输出模式以及使能外设。

注意事项：RZ 码只能在主机模式下使用；RZ 码的输出频率为配置 SPI 工作频率的 1/3，下图为发送数据 A 的 RZ 码波形图。



10.7.3 寄存器定义

名字	地址	读写	复位值	描述
SPI_ICR	0xF1	读	00000000	SPI 中断控制寄存器
SPI_DR	0xF2	读写	00000000	SPI 数据寄存器
SPI_CR	0xF3	读写	00010000	SPI 控制寄存器
SPI_STAT	0xF4	读	00100000	SPI 状态寄存器
SPI_CLR_CR	0xF5	读写	00000000	SPI 突发控制寄存器
SPI_WRADDR	0xF6	读	00000000	SPI 突发地址寄存器

10.7.3.1 SPI_ICR (addr:0xF1) SPI 中断控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	TX_OVER FLOW_EN	TX_EMPTY _EN	TX_EMPTY_EN	RX_UNDER FLOW_EN	RX_OVER FLOW_EN	RX_AFULL _EN	RX_FULL _EN
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	—	—
6	TX_OVER FLOW_EN	发送 FIFO 溢出中断使能 0 不上报中断 1 上报中断
5	TX_EMPTY _EN	发送 FIFO 将空中断使能 0 不上报中断 1 上报中断
4	TX_EMPTY_EN	发送 FIFO 发空中断使能 0 不上报中断 1 上报中断
3	RX_UNDER FLOW_EN	接收 FIFO 下溢中断使能 0 不上报中断 1 上报中断
2	RX_UNDER FLOW_EN	接收 FIFO 溢出中断使能 0 不上报中断 1 上报中断
1	RX_AFULL _EN	接收 FIFO 将满中断使能 0 不上报中断

		1 上报中断
0	RX_FULL _EN	接收 FIFO 收满中断使能 0 不上报中断 1 上报中断

10.7.3.2 SPI_DR (addr:0xF2) SPI 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	SPI 数据寄存器，写入数据为发送数据，读出数据为接收数据

10.7.3.3 SPI_CR (addr:0xF3) SPI 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	–	RZ_MODE	RSV	MASTER	CPHA	CPOL	LSBF	ENABLE
Reset	–	0	0	1	0	0	0	0
Type	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	–	–
6	RZ_MODE	RZ 码调制模式使能 0 禁用 RZ 调制模式 1 使能 RZ 调制模式（本功能仅支持处于 SPI 主机模式下工作）
5	RSV	保留位，该位只能写 0，不能写 1
4	MASTER	主机和从机模式选择 0 从机模式 1 主机模式
3	CPHA	时钟相位选择 0 时钟第一个跳变沿采样 1 时钟第二个跳变沿采样
2	CPOL	时钟极性选择 0 SPI 总线空闲时时钟为低电平 1 SPI 总线空闲时时钟为高电平
1	LSBF	低位优先控制位 0 MSB First 1 LSB First
0	ENABLE	SPI 使能 0 SPI 功能关闭 1 SPI 使能

10.7.3.4 SPI_STAT (addr:0xF4) SPI 状态寄存器

Bit	7	6	5	4	3	2	1	0
Name	–	TX_OVERFLOW	TX_AEMPTY	TX_EMPTY	RX_UNDERFLOW	RX_OVERFLOW	RX_AFULL	RX_FULL
Reset	–	0	1	0	0	0	0	0
Type	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Function
7	–	–
6	TX_OVERFLOW	发送 FIFO 写溢出标志 0 发送 FIFO 未写溢出 1 发送 FIFO 写溢出, 通过对该位写 1 可清零
5	TX_AEMPTY	发送 FIFO 将空标志 0 发送 FIFO 中数据多于 2B 1 发送 FIFO 中数据小于等于 2B, 通过写 SPI_DR 可将该位清零
4	TX_EMPTY	发送 FIFO 发空标志 0 发送 FIFO 未发空 1 发送 FIFO 发空, 通过写 SPI_DR 可将该位清零
3	RX_UNDERFLOW	接收 FIFO 读下溢标志 0 接收 FIFO 未读下溢 1 接收 FIFO 读下溢, 通过对该位写 1 可清零
2	RX_OVERFLOW	接收 FIFO 写溢出标志 0 接收 FIFO 未写溢出 1 接收 FIFO 写溢出, 通过对该位写 1 可清零
1	RX_AFULL	接收 FIFO 将满标志 0 接收 FIFO 中数据少于 2B 1 接收 FIFO 中数据大于等于 2B, 通过读 SPI_DR 可将该位清零
0	RX_FULL	接收 FIFO 满标志 0 接收缓存不满 1 接收缓存满; 对 SPI_DR 进行读操作可以将该位清零

10.7.3.5 SPI_CLK_CR (addr:0xF5) SPI 突发控制寄存器

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

Bit	Name	Function
7: 6	CLKIN	选择 SPI 工作时钟 00 使用 VC1 作为 SPI 工作时钟 01 使用 VC2 作为 SPI 工作时钟 10 使用 VC3 作为 SPI 工作时钟 11 使用 CLK_SYS (VC0) 作为 SPI 工作时钟 该时钟作为主机模式时的工作时钟, 从机模式的工作时钟固定为 CLK_SYS
5:0	–	–

10.7.3.6 SPI_WRADDR (addr:0xF6) SPI 突发地址寄存器

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

Bit	Name	Function
7:2	–	–
1:0	WRADDR	当前缓存的写地址

10.7.4 SPI 配置流程



SPI 配置流程示意图

10.7.5 SPI 操作示例

```

/*配置为 SPI 主机，与 SPI FLASH 25Q127CSIG 通信* /
/*SPI 通信接口配置*/
PT_AFR  |= 0x80;      //SPI 管脚位置使能
PO_GE   = 0x20;      //P2.0 MOSI, P2.1 MISO 复用功能打开
PO_DM0  = 0x60;      //P2.0 推挽输出 P2.1 高阻 P0.5 推挽输出 P06 推挽输出
PO_DM1  = 0x00;
PO_DM2  = 0x00;
PO_6    = 1;

P2_GE   = 0x03;      //P0.5 SCK 复用功能打开 P06 CS
P2_DM0  = 0x01;
P2_DM1  = 0x00;
P2_DM2  = 0x02;

/*SPI 模块初始化*/
PCLK_DIV12 = 0x00;   // 6M
SPI_CR     = 0x10;    //关闭中断，主机模式，模式 00
SPI_CLK_CR = 0x00;    //时钟选择 VC1
SPI_CR     |= 0x01;   //使能 SPI 必须先配置好了再开使能

/*SPI 读取 ID*/
CS = 0;              //CS 位拉低（使用 P06 模拟 CS）
DelayMs(1);
SPI_DR = 0xAB;       //发送指令 AB
SPI_DR = 0x00;
SPI_DR = 0x00;
SPI_DR = 0x00;       //发送 3 个空字节
SPI_DR_Erase(4);
SPI_DR = 0x00;       //发送空字节
DelayMs(1);
CS = 1;              //CS 位拉高
DelayMs(10);
D_ID    = SPI_DR;    //保存数据

```

10.8 I2C

10.8.1 IIC 概述

RC6T80604x 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。

10.8.2 结构框图

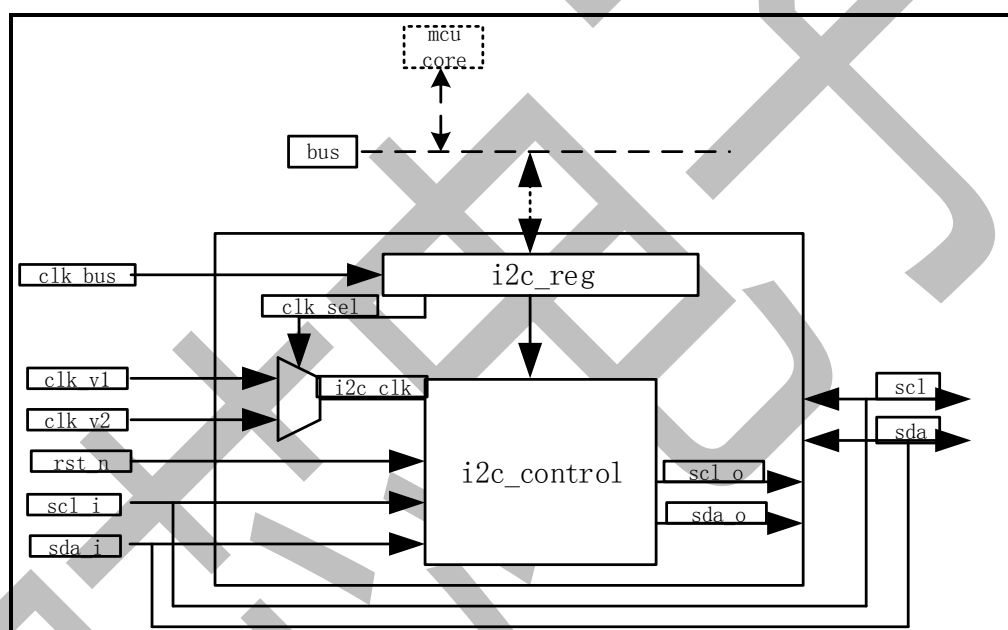


图 10-8-1 I2C 结构框图

10.8.3 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

10.8.4 波特率设置

主机模式下，发送时钟来自时钟源的 17 分频。

10.8.5 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

10.8.5.1 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

10.8.5.2 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	00 主模式关&从模式关

	& Slave	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。

10.8.5.3 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。

10.8.5.4 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；
--	--	---

10.8.5.5 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	N/A	保留位，读 0
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

10.8.6 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。

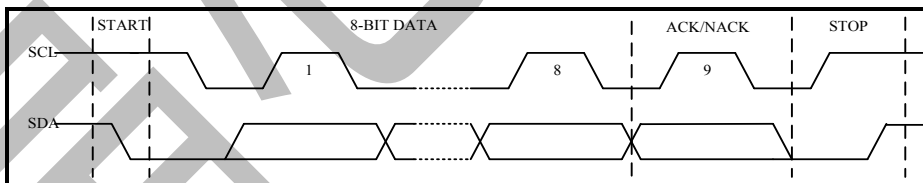


图 10-8-2

10.8.6.1 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 的地址和 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

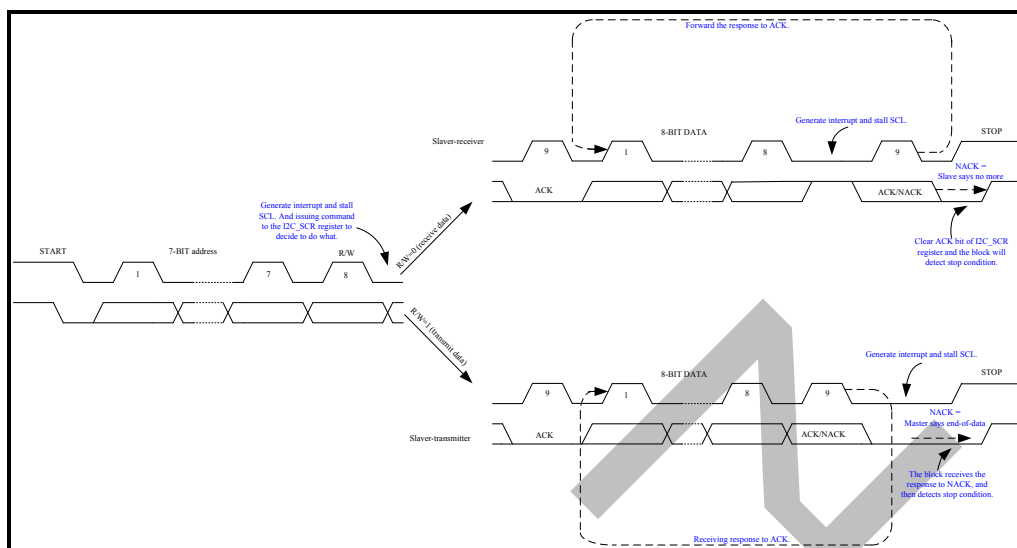


图 10-8-3

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

10.8.6.2 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。

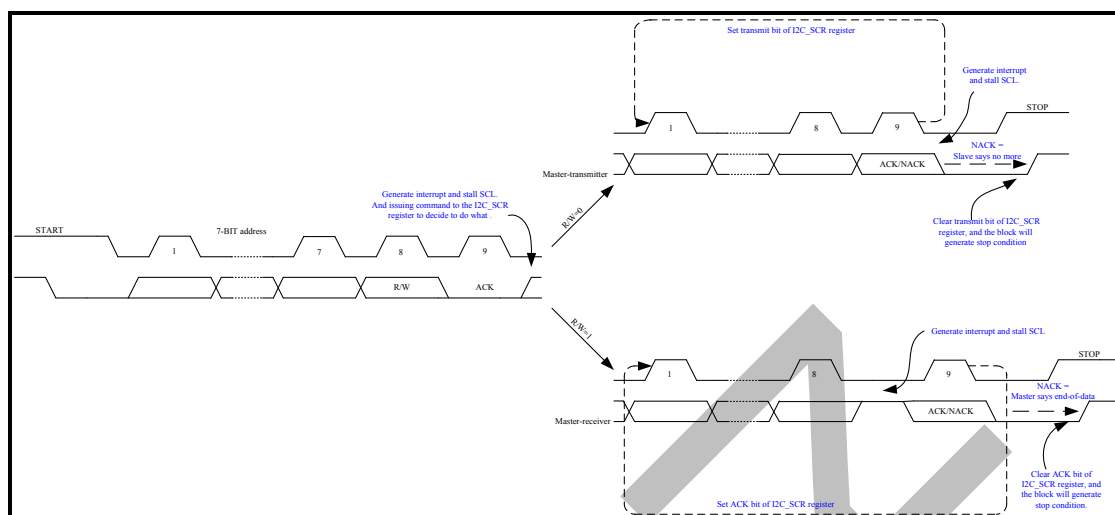


图 10-8-4

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。

10.8.7 IIC 配置流程



IIC 配置流程示意图

10.8.8 IIC 操作示例

```

/*配置 IIC 为主机，向从机写入 2byte 数据*/
PT_AFR1 = 0x02;    //SCL:P1.2, SDA:P1.3
P1_GE = 0x0C;      //P1.3, P1.2 复用打开 //SCL:P1.2, SDA:P1.3
P1_DM0 = 0x00;     //P1.3, P1.2 开漏低输出
P1_DM1 = 0x0C;
P1_DM2 = 0x00;
P1_PU = 0x0C;      //P1.3, P1.2 上拉打开

I2C_CR = 0x00;      // 选择 I2C clock source SCK1
I2C_STAT = 0x00;    // clear state
I2C_CR |= 0x02;     // master mode

I2C_DR = 0x06;      // 从机地址+写到数据寄存器
I2C_MCR = 0x01;     // 发送 STA+SADDR
while(!(I2C_STAT & 0x01)); // 等待从机地址和读写位发送完成
I2C_DR = 0x5A;      // 存储访问存储器地址到发送数据寄存器
I2C_STAT = 0x04;     // 发送 DATA1
while(!(I2C_STAT & 0x01)); // 等待 DATA1 发送完成
I2C_DR = 0xA5;      // 写 DATA1 地址到数据寄存器
I2C_STAT = 0x04;     // 发送 DATA2
while(!(I2C_STAT & 0x01)); // 等到 DATA2 发送完成
I2C_STAT = 0x00;    // 发送停止
    
```

11 系统控制

11.1 复位

11.1.1 复位概述

RC6T80604x 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) 引脚复位，可通过配置 I/O 为复位脚
- (2) 看门狗复位
- (3) 上电复位
- (4) 欠压复位，5 档可配
- (5) 软复位

复位系统结构图如下所示：

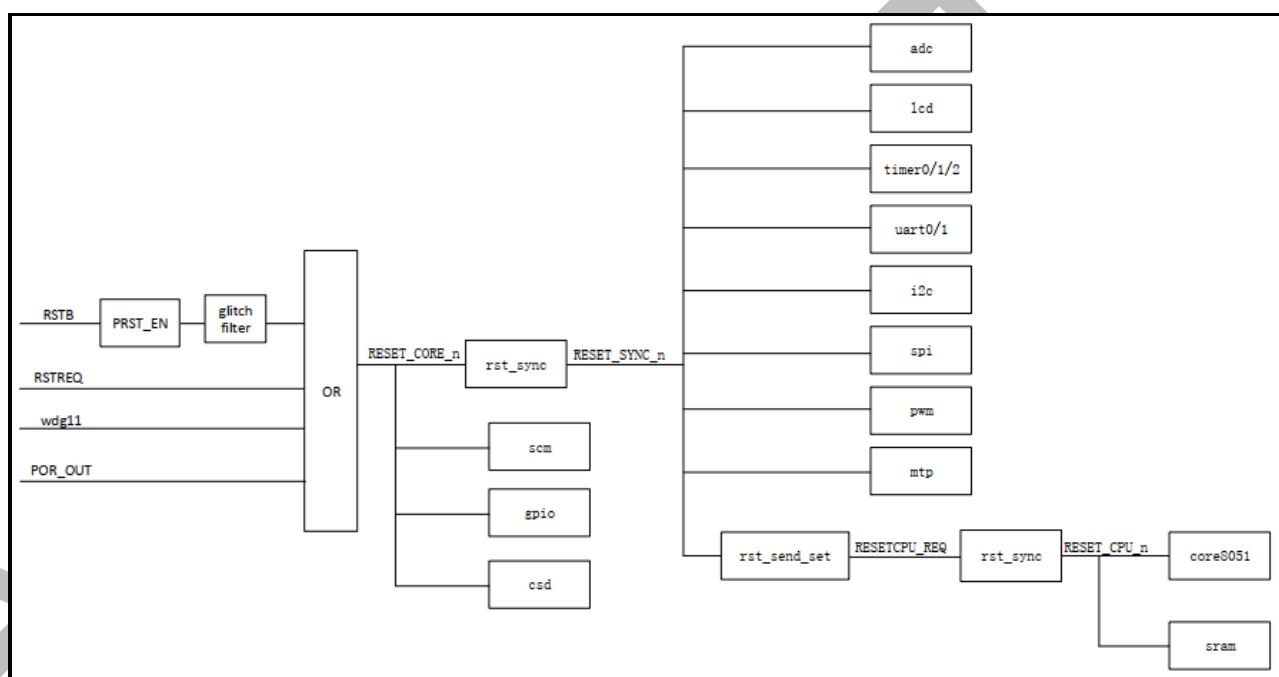


图 11-1

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：

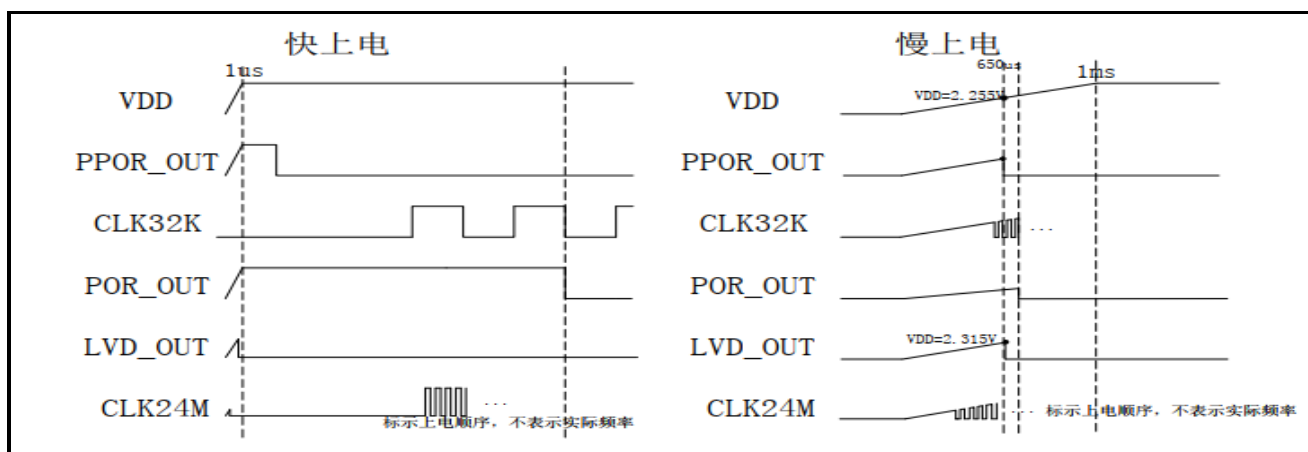


图 11-2

11.1.3 外部引脚复位

外部引脚复位 (RSTB) 通过把引脚拉低触发。GPIO 内部有上拉电阻，如需要使用引脚复位需将复位引脚配置为数字高阻模式，开启复位管脚位置复用使能，将内部上拉打开。在引脚复位期间，IMO 会关闭。一旦 RSTB 释放，IMO 开始工作。

复位引脚功能初始化程序如下：

```
PT_AFR1 |= 0x01;    //使能复位管脚
P1_GE   |= 0x02;    //开启 I/O 复位功能
P1_DM0  |= 0x00;    //P1.1 复位引脚 低电平有效，设置数字高阻模式
P1_DM1  |= 0x00;
P1_DM2  |= 0x02;
P1_PU   |= 0x02;    //开启上拉
```

11.1.4 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。

11.1.5 软复位

RC6T80604x 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

RC6T80604x 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 3.2μA。

- 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 0~3	运行	停止
ADC	运行	停止
SPI	运行	停止
比较器	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	引脚复位、看门狗复位，所有中断	引脚复位，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。

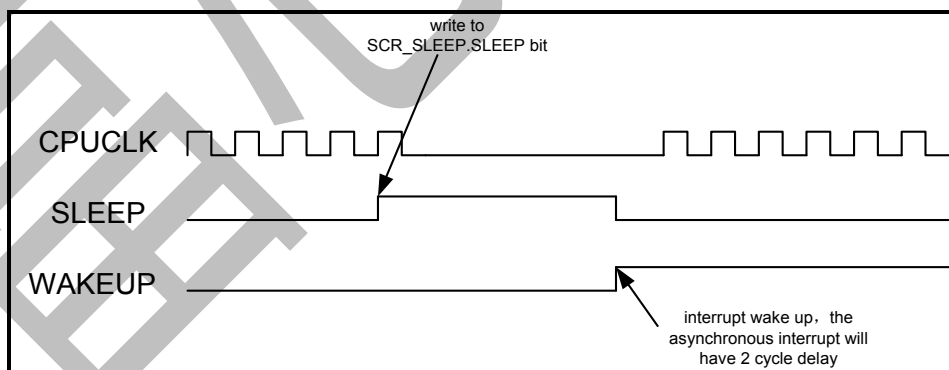


图 11-3 睡眠模式休眠和唤醒时序图

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（IMO）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。下面的图描述了深度休眠唤醒的时序。

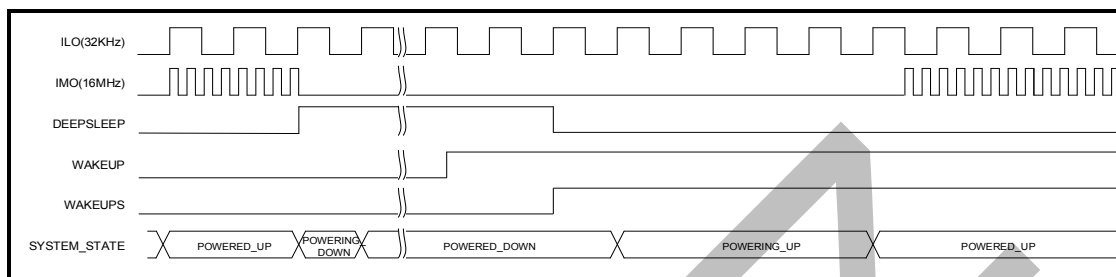


图 11-4 深度休眠唤醒时序

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。RC6T80604x 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	-	RO	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	00000011	系统配置寄存器
SCR_SLEEP	0x92	读写	00000000	休眠寄存器

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_XRES	CALI_SYSRSTREQ	—	—	RSTREQ	BOOT_SHADOW	BOOT
Reset	0	0	0	—	—	0	1	1
Type	R/W1C	R/W1C	R	—	—	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	CALI_XRES	引脚复位标志： 0 没有发生引脚复位 1 发生引脚复位
5	CALI_SYSRSTREQ	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。
4:3	—	—
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

11.5 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	—	—	—	BG_EN_N
Reset	—	—	—	0	—	—	—	0
Type	—	—	—	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
3:1	—	—
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 2.25V (默认值) 001 2.6V 010 2.9V 011 3.7V 100 4.34V 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 2.3V (默认值) 001 2.75V 010 3.0V 011 3.8V 100 4.5V 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	EXT_SEL	ECO_SEL	FX2_SEL	RESET_CTRL	-	EN_BYPASS	SYSX2	IMO_EN
Reset	0	0	0	0	-	0	0	1
Type	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Bit	Name	Function
7	EXT_SEL	外部时钟选择 0 选择 IMO 时钟 1 选择外部引脚时钟
6	ECO_SEL	外部晶体振荡器时钟选择 0 选择 IMO 时钟或者外部引脚时钟 1 选择外部晶体振荡器时钟
5	FX2_SEL	2 倍主频选择 0 选择内部 IMO 的 2 分频 1 选择外部晶体振荡器、引脚时钟或者内部 IMO 时钟 注: 停振保护生效时该位会自动清零。
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。 注：停振保护生效时该位会自动清零。
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。

11.5.1.5 XTAL_CR (addr:0xFF8D) XTAL 控制寄存器

Bit	7	6	5	4	3	2	1	0
Name	XTAL_EN	SCM_IE	SCM_STAT	SCM_EN	—		XTAL_OPT	
Reset	0	0	0	0	—		0	0
Type	R/W	R/W	R/W	R/W	—		R/W	R/W

Bit	Name	Function
7	XTAL_EN	0 禁止外部晶振振荡器 1 使能外部晶振振荡器 注：当检测到外部晶振时钟停振后该位会硬件清零
6	SCM_IE	时钟检测电路中断使能 0 关闭 SCM 中断 1 使能 SCM 中断
5	SCM_STAT	读操作： 0 外部晶振时钟正常 1 外部晶振时钟停振 写操作： 0 无效 1 将该标志位清零
4	SCM_EN	0 禁止外部晶振时钟检测电路 1 使能外部晶振时钟检测电路 注：当检测到外部晶振时钟停振后该位会硬件清零
3:2	—	—
1:0	XTAL_OPT	外部晶体振荡器配置 XTAL_OPT[0] 0 低驱动强度（默认值） 1 高驱动模式 XTAL_OPT[1] 0 32.768kHz 模式（默认值） 1 2~16MHz 模式

12 电气特性

12.1 绝对最大额定值

表 1 电压参数

符号	参数	最小值	最大值	单位
$V_{DD}-V_{SS}$	供电电压	-0.3	6	V
V_{IN}	引脚输入电压	$V_{SS}-0.3$	$V_{DD}+0.3$	V
V_{ESD}	ESD 放电电压		4000	V

表 2 电流参数

符号	参数	最大值	单位
I_{VDD}	VDD 电源上的总拉电流	100	mA
I_{VSS}	VSS 地上的总灌电流	80	mA

表 3 温度参数

符号	参数	最大值	单位
TSTG	存储器温度	-65~150	°C
TJ	最大结温	150	°C

12.2 工作条件

芯片使用的工作条件。

表 4 工作条件

符号	参数	条件	最小值	最大值	单位
f_{CPU}	内部 CPU 时钟频率	-	0	16	MHz
V_{DD}	标准工作电压	-	2.4	5.5	V
T_A	环境温度	最大功率散热	-40	85	°C
T_J	结温		-40	105	°C

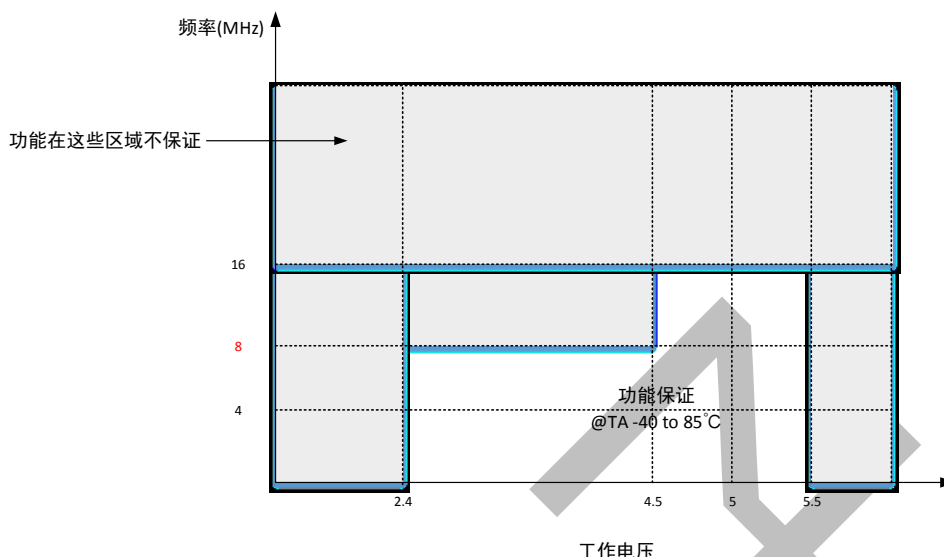


图 1 工作频率及电压对应关系

注意：供电电压在 2.6V 以下 EEPROM 的写功能不能保证。

表 5 上下电的工作条件

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
$t_{TEMPSYS}$	系统复位释放延时	—	—	16	—	ms
V_{IT+}	上电复位阈值	—	—	1.761	—	V
V_{IT-}	欠压复位阈值	BOR_VSEL=000	—	2.25	—	
		BOR_VSEL=001	—	2.6	—	
		BOR_VSEL=010	—	2.9	—	
		BOR_VSEL=011	—	3.7	—	
		BOR_VSEL=100	—	4.34	—	
$V_{HYS(BOR)}$	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 。

表 6 VDD=5V 工作模式下芯片的总工作电流

符号	参数	条件		典型值	最大值	单位
$I_{DD(RUN)}$	工作模式，程序从 MTP 中访问	$f_{CPU}=f_{MASTER}=12MHz$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	—	7.5	mA
		$f_{CPU}=f_{MASTER}=16MHz$	EMO (16MHz)	—	7.5	mA

表 7 VDD=3.3V 工作模式下芯片的总工作电流

符号	参数	条件		典型值	最大值	单位
$I_{DD(RUN)}$	工作模式，程序从 MTP 中访问	$f_{CPU}=f_{MASTER}=12MHz$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	—	7.5	mA
		$f_{CPU}=f_{MASTER}=16MHz$	EMO (16MHz)	—	7.5	mA

表 8 VDD=5V SLEEP 模式下芯片的总工作电流

符号	参数	条件		典型值	最大值	单位
$I_{DD(SLEEP)}$	工作模式，程序从 MTP 中访问	$f_{CPU}=f_{MASTER}=12MHz$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	—	1.5	mA
		$f_{CPU}=f_{MASTER}=16MHz$	EMO (16MHz)	—	1.5	mA

表 9 VDD=3.3V SLEEP 模式下芯片的总工作电流

符号	参数	条件		典型值	最大值	单位
$I_{DD(SLEEP)}$	工作模式，程序从 MTP 中访问	$f_{CPU}=f_{MASTER}=12MHz$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	—	1.5	mA
		$f_{CPU}=f_{MASTER}=16MHz$	EMO (16MHz)	—	1.5	mA

表 10 VDD=5V SLEEP 模式下芯片的总工作电流

符号	参数	条件		典型值	最大值	单位
$I_{DD(SLEEP)}$	SLEEP 模式	$f_{CPU}=f_{MASTER}=12MHz$	IMO (24MHz)，系统时钟采用 IMO 的 2 分频。	—	1.5	mA
		$f_{CPU}=f_{MASTER}=16MHz$	EMO (16MHz)	—	1.5	mA

注：所有外设处于关闭状态。

表 11 VDD=5V DEEPSLEEP 模式下芯片的总工作电流

符号	参数	条件	典型值	最大值	单位
$I_{DD(DEEPSLEEP)}$		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。	3.2	—	μA

表 12 VDD=3.3V DEEPSLEEP 模式下芯片的总工作电流

符号	参数	条件	典型值	最大值	单位
$I_{DD}(\text{DEEPSLEEP})$		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。	3.2	—	μA

表 13 DEEPSLEEP 唤醒时间

符号	参数	条件	典型值	最大值	单位
t_{WUDS}	从 DEEPSLEEP 模式下唤醒的时间		267	—	μs

表 14 外设电流

符号	参数	典型值	最大值	单位
$I_{DD}(\text{ADC})$	ADC 模块工作功耗	600	—	μA
$I_{DD}(\text{CSD})$	CSD 模块工作功耗	500	—	μA

12.4 外部时钟源特性

表 15 外部时钟特性

符号	参数	条件	最小值	典型值	最大值	单位
f_{EMO}	外部高速晶体主振	—	2	—	16	MHz
f_{RTC}	外部低速晶体主振			32.768		kHz
R_F	反馈电阻	$V_{DD}=5\text{V}$, 25°C , $f_{\text{OSC}}=2\sim 16\text{MHz}$	—	8	—	$\text{M}\Omega$
C_{load}	推荐负载电容	—	—	24	—	pF
$I_{DD}(\text{EMO})$	功耗	$C=24\text{pF}$ $f_{\text{OSC}}=16\text{MHz}$, 配置 XTAL_OPT=2' b11	0.302	2	3.58	mA
g_m	晶体传导率	2~16MHz		2.63	—	mA/V
		32K		45.82		$\mu\text{A/V}$
$t_{\text{SU}}(\text{EMO})$	建立时间	V_{DD} 稳定, 频率 16MHz, 配置 XTAL_OP T=2' b11	—	026	—	ms

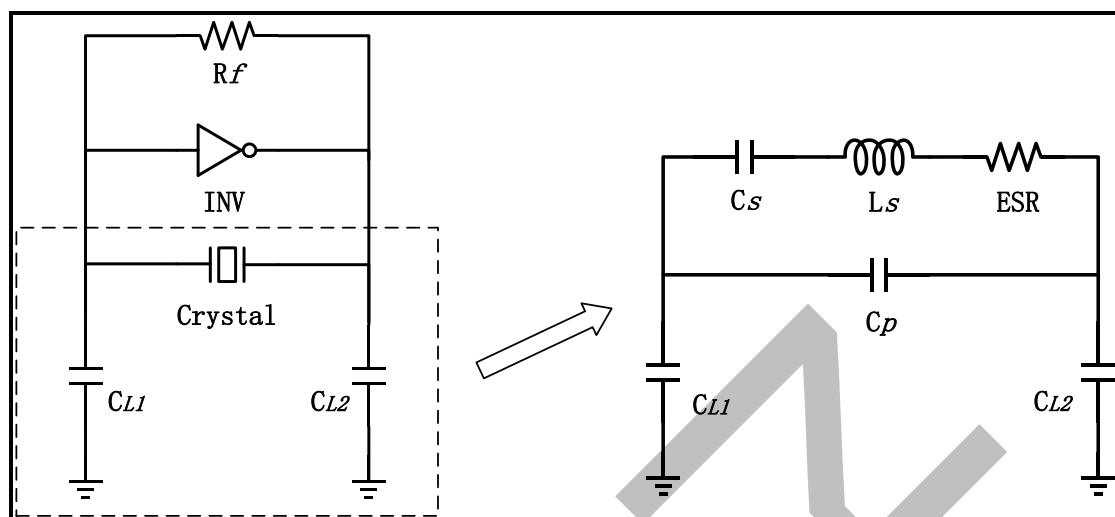


图 晶振整体电路（左）外部等效电路（右）

仿真模型

	32K	2M	8M	16M
C_S	2.4f	3f	7.08f	12.75f
L_S	9.829K	2.1	55.9m	7.76m
ESR	30K	150	80	40
C_P	1.3p	7p	7p	7p
C_{L1} 、 C_{L2}	24p	24p	24p	24p

12.5 内部时钟源特性

表 16 内部主时钟

符号	参数	条件	最小值	典型值	最大值	单位
f_{IMO}	频率	—	—	24	—	MHz
ACC_{IMO}	内部时钟精度 (工厂校准)	$V_{DD}=5V$, $T_A=25^{\circ}C$	-2	—	2	%
$t_{SU(IMO)}$	IMO 唤醒时间, 包括校准	—	—	—	20	μs
$I_{DD(IMO)}$	IMO 功耗	$V_{DD}=5V$, $T_A=25^{\circ}C$, 24MHz	—	316	—	μA

表 17 内部低速时钟

符号	参数	条件	最小值	典型值	最大值	单位
f_{ILO}	频率	—	—	32	—	kHz
ACC_{IMO}	内部时钟精度 (工厂校准)	$V_{DD}=5V$, $-40^{\circ}C \leq T_A \leq 85^{\circ}C$	-10	—	10	%
$I_{DD(ILO)}$	ILO 功耗	—	—	1	—	μA

12.6 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（VDD=5V/1000T 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（VDD=5V/1000T 周期后）
- 4、可编程 50000 次

12.7 I/O 引脚特性

表 18 I/O 静态参数

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	$V_{DD}=5V$	-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	$V_{DD}=5V, V_{IN}=V_{SS}$	-	10	-	k Ω
t_R, t_F	上升和下降时间 (10%-90%)	快速 I/Os 负载 50pF	-	15	-	ns
		标准 I/Os 负载 50pF	-	15	-	ns
I_{Ikg}	数字输入漏电流	$V_{SS} \leq V_{IN} \leq V_{DD}$	-	-	± 1	μA
$I_{Ikganalog}$	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

表 19 输出驱动电流

符号	参数	条件	最小值	最大值	单位
V_{OL}	输出低电平	$I_{IO}=70mA, V_{DD}=5V$	-	0.8	V
		$I_{IO}=50mA, V_{DD}=3.3V$	-	0.8	
V_{OH}	输出高电平	$I_{IO}=17mA, V_{DD}=5V$	4.3	-	
		$I_{IO}=6mA, V_{DD}=3.3V$	3	-	

12.8 ADC 特性

($T_A=25^{\circ}\text{C}$, 除非另有说明)

符号	参数	条件	最小值	典型值	最大值	单位
	供电电压		2.6	5	5.5	V
	工作温度		-40	25	85	$^{\circ}\text{C}$
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	$f_{\text{ADC}}=4\text{M}$			± 8	LSB
E_g	GAIN_ERROR	$f_{\text{ADC}}=4\text{M}$			± 8	LSB
T_{ADC1}	ADC 转换时间 1	$f_{\text{ADC}}=1\text{M}$, ADC 采样周期=2, 总共 20 个时钟周期			20	μs
T_{ADC2}	ADC 转换时间 2	$f_{\text{ADC}}=4\text{M}$, ADC 采样周期=2, 总共 20 个时钟周期			5	μs

12.9 LCD 特性

符号	参数	条件	最小值	典型值	最大值	单位
IQ	静态电流	$\text{VLOSEL_SEL}\langle 1:0 \rangle = 00$	-	-	-	μA
		$\text{VLOSEL_SEL}\langle 1:0 \rangle = 01$	94	199	220	
		$\text{VLOSEL_SEL}\langle 1:0 \rangle = 10$	32	67	74	
		$\text{VLOSEL_SEL}\langle 1:0 \rangle = 11$	14	29	32	

12.10 CSD 触摸特性

符号	参数	测试条件	最小值	典型值	最大值	单位
C_{mod}	CMOD 口外挂电容大小			10		nF
f_{PH}	电容驱动时钟频率	不使用伪随机时钟		0.5	1	MHz
f_{SIN}	计数时钟频率			12	24	MHz
	分辨率		8		16	bit
ISCAN	普通扫描功耗			420		μA
IWPF	防水扫描功耗	4 个防水同时工作		12		mA

TSCAN	扫描时间			2^N (N=分辨率) * 计数时钟周期		
-------	------	--	--	------------------------	--	--

12.11 EMC 特性

12.11.1 EMS

暂无

12.11.2 EMI

测试方法通过产品（通过 I/O 翻转两个 LED 灯）运行一个简单的应用程序，测试产品发射电磁干扰。测试软件，测试 layout 和引脚负载情况参考标准 IEC61967-2 标准。

符号	参数	测试条件			最大值	单位
		通用条件	检测频带	最打 fHSE/fCPU		
				16MHz/ 8MHz	16MHz/ 16MHz	
SEMI	峰值等级	VDD=5V TA=25℃ XX 封装参照 IEC619672 标准	0.1 MHz 到 30MHz	5	5	dB μ V
			30MHz 到 130MHz	4	5	
			130MHz 到 1GHz	5	5	
	EMI 等级		-	2.5	2.5	-

12.11.3 ESD

符号	参数	测试条件	最大值	单位
V _{ESD(HBM)}	人体放电模式	TA=25°C, 遵循 JESD22-A114	4000	V
V _{ESD(MM)}	机器放电模式	TA=25°C,	400	
V _{ESD(CDM)}	元件充电模式	TA=25°C, 遵循 JESD22-C101	1000	V

12.11.4 LU

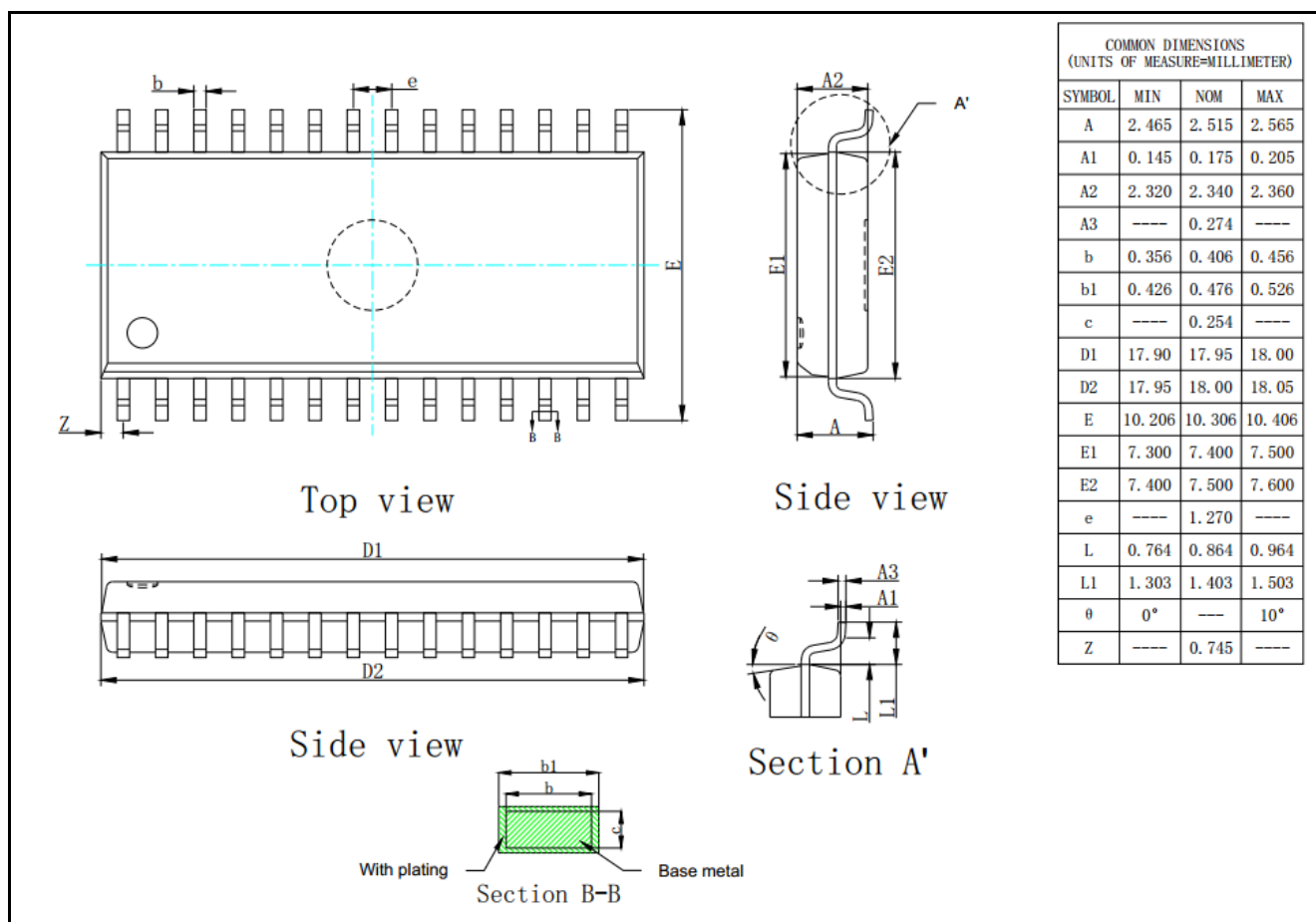
参考标准 EIA/JESD 78 IC 门锁标准。

表 20 电气敏感度

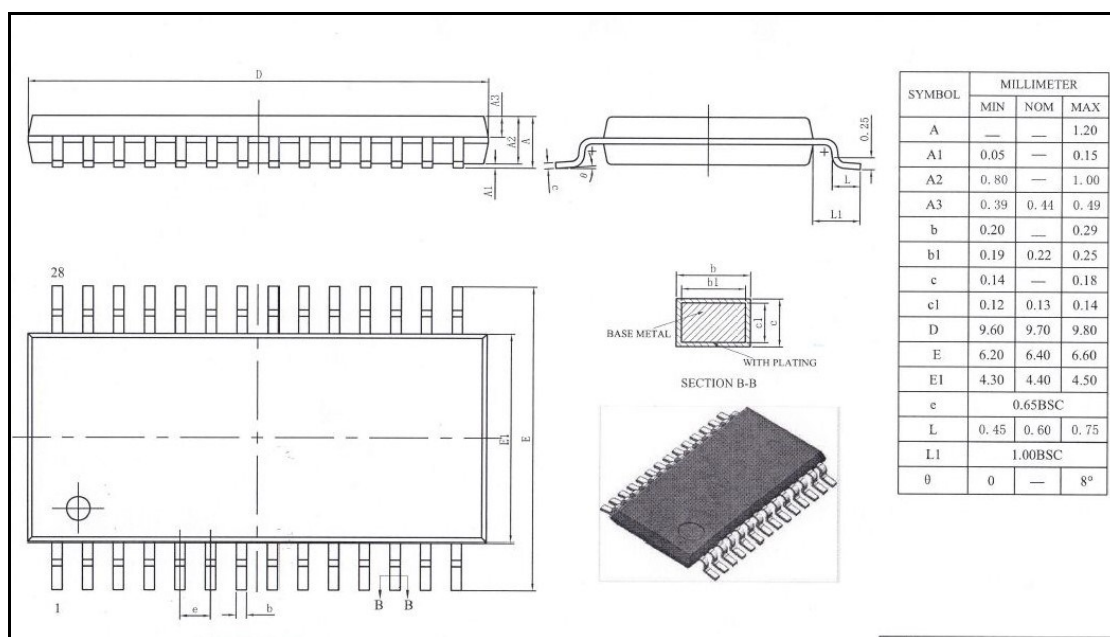
符号	参数	测试条件	等级
LU	静态门锁等级	TA=25°C	A
		TA=85°C	A

12 封装信息

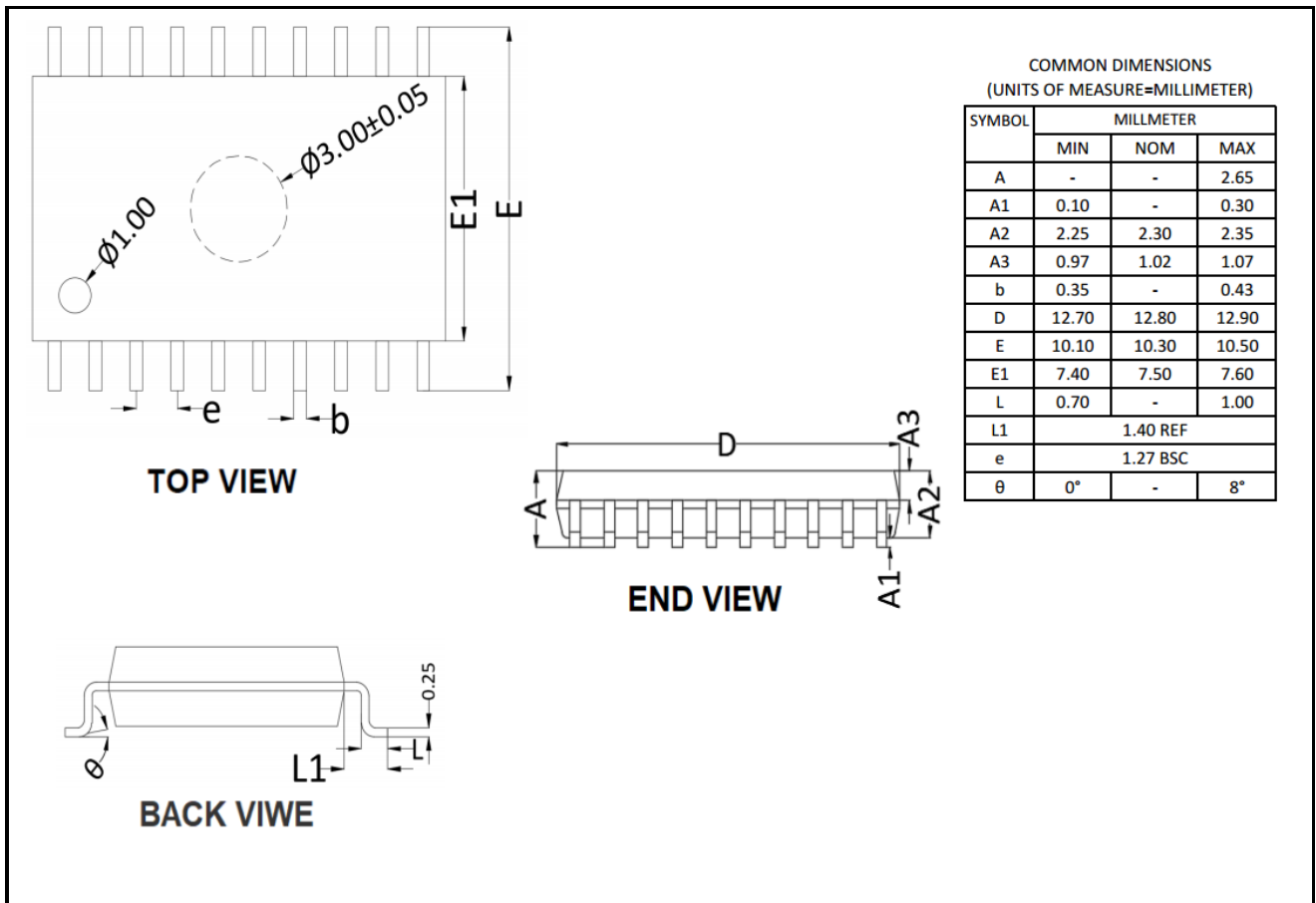
12.1 SOP28



12.2 TSSOP28

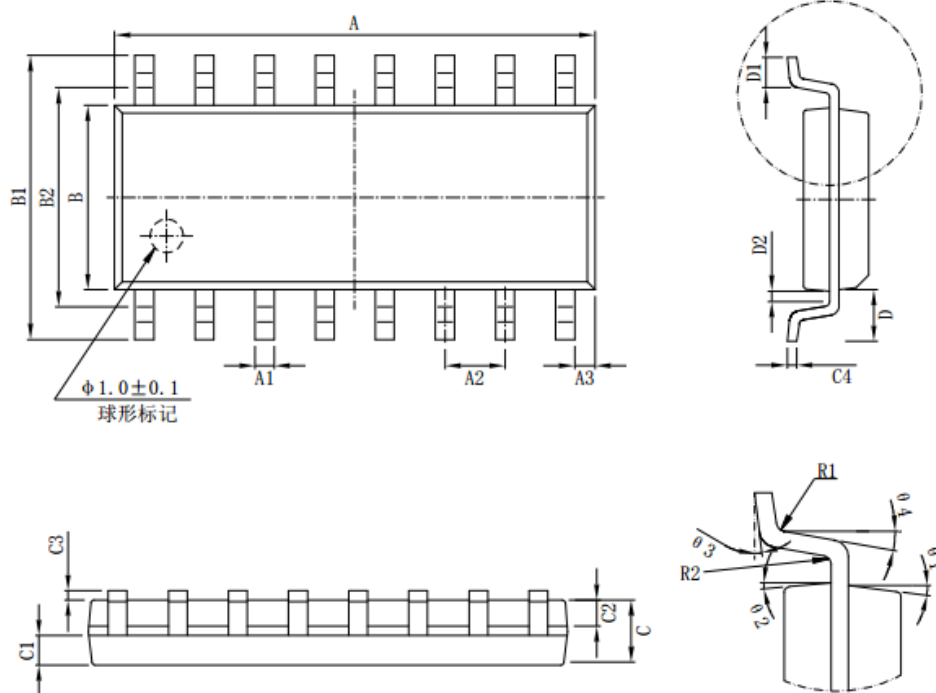


12.3 SOP20



12.4 SOP16

标注	尺寸	最小(mm)	最大(mm)	标注	尺寸	最小(mm)	最大(mm)
A		9.80	10.00	C4		0.203	0.233
A1		0.356	0.456	D		1.05TYP	
A2		1.27TYP		D1		0.40	0.70
A3		0.302TYP		D2		0.15	0.25
B		3.85	3.95	R1		0.20TYP	
B1		5.84	6.24	R2		0.20TYP	
B2		5.00TYP		θ 1		8° ~ 12° TYP4	
C		1.40	1.60	θ 2		8° ~ 12° TYP4	
C1		0.61	0.71	θ 3		0° ~ 8°	
C2		0.54	0.64	θ 4		4° ~ 12°	
C3		0.05	0.25				



13 订单信息

订单型号	封装	丝印标记	包装
RC6T80604GA	SOP28	80604GA	管装
RC6T80604GC	TSSOP28	80604GC	管装
RC6T80604EA	SOP20	80604EA	管装
RC6T80604DA	SOP16	80604DA	管装

14 开发工具

R-Link Pro
调试工具
量产编程器

15 版本说明

版本号	修改时间	修订人	修改内容
V1.0	2022.11.20	FX	初版
V1.1	2023.3.5	FX	1、将芯片特性说明放置目录前
V1.2	2023.3.22	FX	1、删除封装名称后缀
V1.3	2023.5.8	FX	1、纠正 IO 口模拟高阻模式应用说明描述。 2、调整寄存器描述格式。 3、增加 PWM 功能说明。
V1.4	2023.5.15	FX	1、将公司网址添加至页脚处。
V1.5	2023.7.15	FX	1、调整寄存器描述（增加名称说明）。 2、增加外设使用流程说明。 3、调整表格格式。 4、调整 CMOD 口外挂电容大小典型值
V1.6	2023.8.1	FX	1、删除 CSD 模块描述
V1.7	2023.10.10	CYH	1、优化首页及页眉 2、增加部分外设工作原理图 3、增加外设代码示例 4、优化部分外设工作原理描述 5、更新部分性能参数
V1.8	2024.5.17	CYH	1、修正部分笔误（地址映射） 2、修正 SCR_CFG/XTAL_CR/IMO_CR 寄存器默认值 3、删除 TSSOP20 封装相关描述
V1.9	2025.8.25	CYH	1、优化引脚图描述，标识烧录引脚