



QG701 系列中文版说明书

OTP-Based 8-Bit Microcontroller Series



目录

| | |
|---|----|
| 目录 | 2 |
| 1.0、功能特色 | 5 |
| 2.0、功能特性 | 6 |
| 3.0、概述 | 7 |
| 4.0、结构图 | 7 |
| 5.0、管脚图 | 8 |
| 5.1、管脚功能描述 | 8 |
| 6.0、存储器结构 | 9 |
| 6.1、程序存储器..... | 9 |
| 6.2、数据存储器..... | 10 |
| 7.0、功能介绍..... | 12 |
| 7.1、寄存器操作..... | 12 |
| 7.1.1、 INDF (间接寻址寄存器) | 12 |
| 7.1.2、 TMRO (定时/计数器 Time lock/Counter register) | 13 |
| 7.1.3、 PCL (Low Bytes of Program Counter) & Stack | 13 |
| 7.1.4、 STATUS (状态字寄存器) | 15 |
| 7.1.5、 FSR(间接寻址指针) | 16 |
| 7.1.6、 PORTA, PORTB (Port 寄存器) | 16 |
| 7.1.7、 TCXCON (TCX定时器模式控制寄存器) | 16 |
| 7.1.8、 PCON (电源控制寄存器) | 17 |
| 7.1.9、 WUCON (Port B输入改变/唤醒控制寄存器) | 18 |
| 7.1.10、 PCHBUF (PC指针高位缓冲区) | 19 |
| 7.1.11、 PDCON (I/O下拉控制寄存器) | 19 |
| 7.1.12、 ODCON (I/O开漏控制寄存器) | 20 |
| 7.1.13、 PHCON (I/O上拉控制寄存器) | 20 |
| 7.1.14、 INTEN (中断屏蔽寄存器) | 21 |
| 7.1.15、 INTFLAG (中断标志寄存器) | 22 |
| 7.1.16、 E2PCON (E2P控制寄存器) | 22 |
| 7.1.17、 E2PADR (E2P地址寄存器) | 23 |
| 7.1.18、 E2PDIN (E2P数据输入寄存器) | 23 |
| 7.1.19、 E2PDATA (E2P数据寄存器) | 23 |
| 7.1.20、 TC0M (TC0控制寄存器) | 24 |
| 7.1.21、 TC0C (TC0计数寄存器) | 25 |
| 7.1.22、 TC0R (TC0自动装载寄存器) | 25 |
| 7.1.23、 TC1M (TC1控制寄存器) | 25 |
| 7.1.24、 TC1C (TC1计数寄存器) | 26 |
| 7.1.25、 TC1R (TC1自动装载寄存器) | 26 |
| 7.1.26、 TC2M (TC2控制寄存器) | 26 |
| 7.1.27、 TC2C (TC2计数寄存器) | 26 |
| 7.1.28、 TC2R (TC2自动装载寄存器) | 26 |
| 7.1.29、 ACC (Accumulator)累加器 | 27 |
| 7.1.30、 OPTION Register (选项寄存器) | 27 |
| 7.1.31、 IOSTA & IOSTB (I/O口控制寄存器)..... | 28 |
| 7.2、 I/O Ports | 29 |



| | |
|---|----|
| 7.3、Timer0/WDT & Prescaler/TC0/TC1/TC2 | 31 |
| 7.3.1、Timer0 | 31 |
| 7.3.1.1、使用内部时钟: 定时模式 | 31 |
| 7.3.1.2、使用外部时钟: 计数模式 | 31 |
| 7.3.2、看门狗定时器 (WDT) | 31 |
| 7.3.3、Prescaler (预置器) | 31 |
| 7.3.4、TC0 | 32 |
| 7.3.4.1、TC0定时模式 | 32 |
| 7.3.4.2、TC0 BUZZER输出 | 34 |
| 7.3.4.3、TC0 PWM输出 (脉宽调制) | 35 |
| 7.3.5、TC1/TC2 | 36 |
| 7.4、中断方式 | 37 |
| 7.4.1、外部中断 | 37 |
| 7.4.2、Timer0 中断 | 37 |
| 7.4.3、Port B 输入改变中断 | 37 |
| 7.4.4、TC0 中断 | 38 |
| 7.4.5、TC1中断 | 38 |
| 7.4.6、TC2中断 | 38 |
| 7.5、内置EEPROM操作 | 38 |
| 7.5.1、内置EEPROM烧写 | 38 |
| 7.5.2、内置EEPROM的字节擦除 | 39 |
| 7.5.3、内置EEPROM的扇区擦除 | 39 |
| 7.5.4、内置EEPROM的片擦除 | 40 |
| 7.5.5、内置EEPROM的读操作 | 40 |
| 7.6、省电模式 (SLEEP) | 41 |
| 7.6.1、睡眠唤醒 | 41 |
| 7.7、复位 | 42 |
| 7.7.1、上电复位计数器(Power-up Reset Timer PWRT) | 42 |
| 7.7.2、振荡启动计数器(Oscillator Start-up Timer OST) | 42 |
| 7.7.3、复位顺序 | 43 |
| 7.8、十六进制转化为十进制 (Hexadecimal Convert to Decimal HCD) | 45 |
| 7.9、振荡器配置 (Oscillator Configurations) | 46 |
| 7.10、配置选项 | 49 |
| 8.0、指令集合 | 52 |
| 9.0、绝对最大额定值 | 64 |
| 10.0、操作条件 | 64 |
| 11.0、电气特性 | 65 |
| 11.1、QG701电气特性 | 65 |
| 11.2、QG701电气特性表 | 69 |
| 11.2.1、操作频率vs操作电压 (Ta=25°C) | 69 |
| 11.2.2、内部4MHz RC vs 供应电压 (Ta=25°C) | 69 |
| 11.2.3、内部8MHz RC vs 供应电压 (Ta=25°C) | 69 |
| 11.2.4、内部1MHz RC vs 供应电压 (Ta=25°C) | 70 |
| 11.2.5、内部455KHz RC vs 供应电压 (Ta=25°C) | 70 |
| 11.2.6、内部4MHz RC vs 温度 | 70 |
| 11.2.7、内部8MHz RC vs 温度 | 70 |



| | |
|---|----|
| 11.2.8、内部1MHz RC vs温度 | 71 |
| 11.2.9、内部455KH Hz RC vs温度 | 71 |
| 11.2.10、WTD18毫秒复位时间 vs温度 | 71 |
| 11.2.11、WTD4.5毫秒复位时间 vs温度 | 71 |
| 11.2.12、WTD72毫秒复位时间 vs温度 | 72 |
| 11.2.13、WTD288毫秒复位时间 vs温度 | 72 |
| 11.2.14、WTD18毫秒复位时间 vs供应电压 (Ta=25°C) | 72 |
| 11.2.15、WTD4.5毫秒复位时间 vs供应电压 (Ta=25°C) | 72 |
| 11.2.16、WTD72毫秒复位时间 vs供应电压 (Ta=25°C) | 73 |
| 11.2.17、WTD288毫秒复位时间 vs供应电压 (Ta=25°C) | 73 |
| 11.2.18、LVDT2.0V vs温度 | 73 |
| 11.2.19、LVDT3.6V vs温度 | 73 |
| 11.2.20、LVDT1.8V vs温度 | 74 |
| 11.2.21、LVDT2.2V vs温度 | 74 |
| 11.2.22、LVDT2.4V vs温度 | 74 |
| 11.2.23、LVDT2.6V vs温度 | 74 |
| 12.0、封装尺寸 | 75 |
| 12.1、14-PIN PDIP 300mil | 75 |
| 12.2、14-PIN SOP 150mil | 76 |
| 13.0、封装IR回流焊接曲线 | 77 |
| 14.0、订购信息 | 77 |



1.0、功能特色

- 内置 256x8 EEPROM (无需 I2C 协议，寄存器控制擦写)
- 双时钟模式 IRC_RTC (IRC for Fcpu/LF for T0)
- LVD 复用 LVR (通过软件控制)
- 四路独立控制定时器 (T0\TC0\TC1\TC2)
- 三路独立控制 PWM/BUZZER 输出 (PB2\PB1\PB0)
- PB3 口软件可控上拉
- 硬件控制端口强弱驱动电流



2.0、 功能特性

- 只有 42 个单字指令
- 除跳转指令为两个周期指令以外其余为单周期指令
- 13-bit 指令宽度
- GOTO 指令能跳转到所有的 ROM/EPROM 地址空间
- 子程序能返回到所有的 ROM/EPROM 地址空间
- 能处理 8 位数据
- 5 级硬件堆栈
- 运行速度: DC-20 MHz 工作频率
DC-100 ns 指令周期

| 型号 | 管脚# | I/O# | EPROM/ROM 空间 (Byte) | RAM (Byte) |
|-------|-----|------|---------------------|------------|
| QG701 | 14 | 12 | 1K | 48 |

- 支持直接与间接数据寻址方式
- 一个带 8 位预置器的 8 位定时/计数器 (Timer0)，三个带 8 位自动重载功能的 8 位定时/计数器 (TC0, TC1, TC2)
- 内部上电复位
- 内含一个低电压检测电路供掉电复位使用
- 上电复位计数器 (PWRT) 和振荡启动计数器 (Oscillator Start-up Timer OST)
- 内部振荡器集成了一个看门狗保证了可靠的操作同时软件使能看门狗操作
- 两类双向输入输出 I/O 口 IOA 和 IOB
- 三路独立控制可编程 PWM/蜂鸣器输出
- 通过编程控制 I/O 端口的上拉/下拉、开漏等状态
- 内置 256byte 可擦写 EEPROM/ (共 16 个 sector, 每个 sector 含有 8 个 byte)
- 四个内部计数中断源; 两个外部中断源: INT 管脚, PortB 的输入改变
- 通过 INT 管脚或者 PortB 的输入改变来实现睡眠唤醒
- 省电睡眠模式
- 内部有 8MHz, 4MHz, 1MHz, 和 455KHz RC 振荡器
- 有可靠的保证使得程序代码不被读出。
- 内部 RC 振荡器
- 提供以下振荡源的选择:
 - ERC: External Resistor/Capacitor Oscillator (外部的 RC 振荡器)
 - IRC/ERIC: Internal or External Resistor/Internal Capacitor Oscillator - (内部电阻内部的电容 RC 振荡器或外部的电阻内部的电容 RC 振荡器)
 - IRC_RTC: Internal or External Resistor/ Low Frequency Crystal Oscillator For T0 RTC
 - HF: High Frequency Crystal/Resonator Oscillator (高频率的晶体振荡器)
 - LF: Low Frequency Crystal Oscillator (低频率的晶体振荡器)
 - XT: Crystal/Resonator Oscillator (晶体/陶瓷振荡器)



- 工作电压范围: 2.0V - 5.5V
 - ≤4MHZ: 2.4V - 5.5V
 - ≤8MHZ : 2.6V- 5.5V

3.0、概述

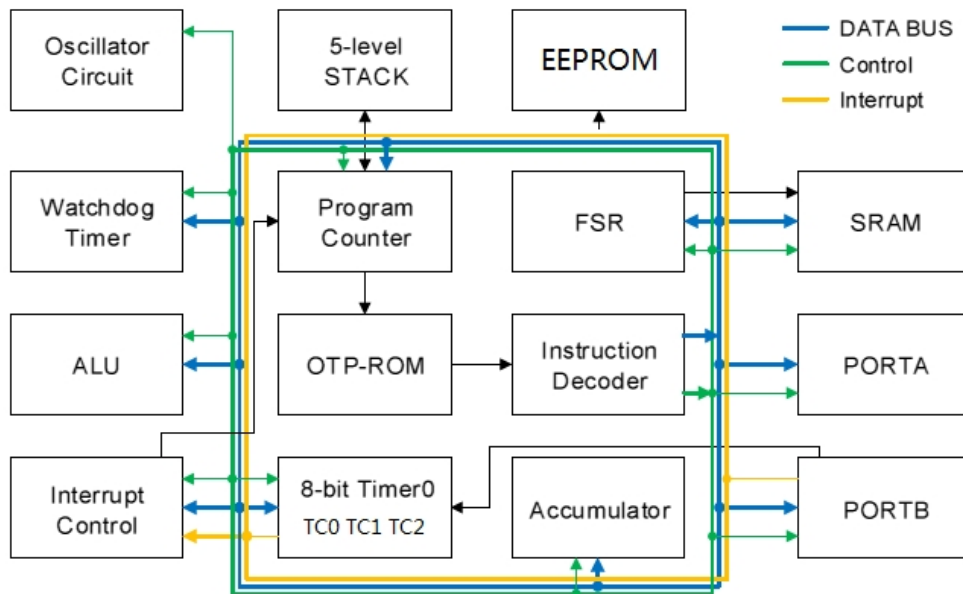
QG701 是一款低功耗，高速，高噪声容限，EPROM/ROM 基于 8 位 CMOS 工艺制造的单片机，采用 RISC 指令集，共有 42 条指令，除分支指令为两个周期指令以外其余为单周期指令。这种易用、易记的指令集大大缩短了开发时间。

QG701 包含了上电复位(Power-on Reset POR)，掉电复位(Brown-out Reset BOR)，上电复位计数器 (Power-up Reset Timer PWRT)，振荡启动计数器 (Oscillator Start-up Timer OST)，看门狗定时器(Watchdog Timer)，EPROM/ROM，EEPROM，SRAM，双向三态 I/O 口，（可以设置为上拉/下拉、开漏），省电睡眠模式，一个带 8 位预置器的 8 位定时/计数器，三个带 8 位自动重载的 8 位定时器/计数器，独立中断，睡眠唤醒模式和可靠的代码保护，有两个振荡源可供用户配置选择，包含省电振荡源和低功耗振荡器。

QG701 可访问 1K×13 的程序存储空间。

QG701 能直接或间接访问寄存器以及数据存储区，所有的特殊功能寄存器分布在数据存储区同时包含特定的程序指针。

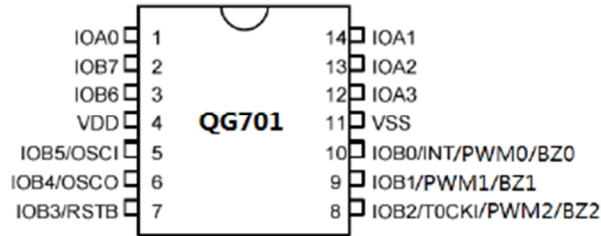
4.0、结构图





5.0、管脚图

PDIP, SOP



5.1、管脚功能描述

| 管脚名称 | I/O | 说明 |
|-----------------|-----|---|
| IOA0 ~ IOA3 | I/O | IOA0 ~ IOA3 双向I/O口 软件可以设置为下拉 |
| IOB0/INT/PWM0 | I/O | 双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部中断输入脚 Buzzer0输出引脚/PWM0输出引脚 |
| IOB1/PWM1 | I/O | 双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 Buzzer1输出引脚/PWM1输出引脚 |
| IOB2/T0CKI/PWM2 | I/O | 双向I/O口同时具有系统唤醒功能 软件设置为上拉/下拉和开漏 外部计数输入脚 Buzzer2输出引脚/PWM2输出引脚 |
| IOB3/RSTB | I/O | 双向I/O口同时具有系统唤醒功能 软件设置为上拉/硬件设置开漏 系统复位脚. 低电平复位. 设置为复位脚时上拉自动开启 |
| IOB4/OSCO | I/O | 双向I/O口同时具有系统唤醒功能 (RCOUT 可选择IRC/ERIC, ERC模式) 软件设置为上拉/开漏 晶体振荡器输出脚 (HF, LF, IRC_RTC模式) 基于指令周期晶体振荡器输出 (RCOUT 可选择IRC/ERIC, ERC模式) |
| IOB5/OSCI | I/O | 双向I/O口同时具有系统唤醒功能 (IRC 模式) 软件设置为上拉/开漏 晶体振荡器输入脚 (HF, LF, IRC_RTC模式) 外部实时时钟输入脚(ERIC, ERC模式) |
| IOB6 ~ IOB7 | I/O | 双向I/O口同时具有系统唤醒功能 软件设置为上拉/开漏 |
| Vdd | — | 电源 |
| Vss | — | 地 |

Legend: I=输入, 0=输出, I/O=输入/输出



6.0、存储器结构

QG701 存储器包含程序存储器和数据存储器。

6.1、程序存储器

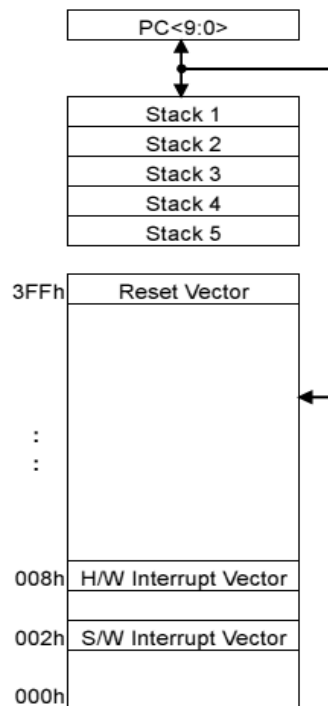
QG701 有一个 10 位 PC 指针能访问 $1K \times 13$ 的存储空间。

QG701 的复位地址为 3FFh。

H/W 中断向量地址 008h.， S/W 中断向量地址 002h。

QG701 的 CALL/GOTO 能指向在同一个程序页面（一个程序页面为 1K）的所有存储空间。

程序存储器分布图和堆栈结构：



QG701



6.2、数据存储器

数据存储器包含特殊功能器组和通用寄存器组，所有通用寄存器可以直接寻址或者通过 FSR 寄存器间接寻址。特殊功能寄存器用来控制 CPU 或外围功能模块的工作。

表 1.1: QG701 寄存器列表

| Address | Description Bank0 | Description Bank1 | Description Bank2 |
|-----------|---------------------------|-------------------|-------------------|
| 00h | INDF | | |
| 01h | TMR0 | | |
| 02h | PCL | | |
| 03h | STATUS | | |
| 04h | FSR | | |
| 05h | PORTA | | |
| 06h | PORTB | | |
| 07h | TCXCON | E2PCON | TCOM |
| 08h | PCON | E2PADR | TCOC |
| 09h | WUCON | E2PDIN | TCOR |
| 0Ah | PCHBUF | E2PDATA | TC1M |
| 0Bh | PDCON | | TC1C |
| 0Ch | ODCON | | TC1R |
| 0Dh | PHCON | | TC2M |
| 0Eh | INTEN | | TC2C |
| 0Fh | INTFLAG | | TC2R |
| 10h ~ 3Fh | General Purpose Registers | | |

| | |
|-----|--------|
| N/A | OPTION |
| 05h | IOSTA |
| 06h | IOSTB |

表 1.2: 通过 OPTION 或 IOST 指令控制的寄存器

| 地址 | 说明 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|--------|------------------|--------|------|------|-----|-----|-----|-----|
| N/A (w) | OPTION | TOTB | INTEDG | TOCS | TOSE | PSA | PS2 | PS1 | PS0 |
| 05h (w) | IOSTA | Port A I/O 控制寄存器 | | | | | | | |
| 06h (w) | IOSTB | Port B I/O 控制寄存器 | | | | | | | |

表 1.3: 寄存器列表

| 地址 | 说明 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|---------------------------|------|----------------------|------|------|------|------|------|
| 00h (r/w) | INDF | 通过 FSR 访问数据区(不是一个实际的物理地址) | | | | | | | |
| 01h (r/w) | TMR0 | 8 位定时/计数器 | | | | | | | |
| 02h (r/w) | PCL | 低 8 位 PC 指针 | | | | | | | |
| 03h (r/w) | STATUS | RST | GP1 | GP0 | /T0 | /PD | Z | DC | C |
| 04h (r/w) | FSR | RP1 | RP0 | 间接地址访问指针 (RAM 选择寄存器) | | | | | |
| 05h (r/w) | PORTA | — | — | — | — | IOA3 | IOA2 | IOA1 | IOA0 |
| 06h (r/w) | PORTB | IOB7 | IOB6 | IOB5 | IOB4 | IOB3 | IOB2 | IOB1 | IOB0 |



| 地址Bank0 | 说明 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|---------|-------|-------|--------|-------|--------|----------|---------------------|----------|
| 07h (r/w) | TCXCON | NG | NG | NG | NG | TC10EN | PWM1INV | TC20EN | PWM2INV |
| 08h (r/w) | PCON | WDTE | EIS | LVDTTE | LVDF | LVDRB | LVDESEL2 | LVDESEL1 | LVDESEL0 |
| 09h (r/w) | WUCON | WUB7 | WUB6 | WUB5 | WUB4 | WUB3 | WUB2 | WUB1 | WUB0 |
| 0Ah (r/w) | PCHBUF | — | — | — | — | — | — | 2 MSBs Buffer of PC | |
| 0Bh (r/w) | PDCON | * | /PDB2 | /PDB1 | /PDB0 | /PDA3 | /PDA2 | /PDA1 | /PDA0 |
| 0Ch (r/w) | ODCON | ODB7 | ODB6 | ODB5 | ODB4 | — | ODB2 | ODB1 | ODB0 |
| 0Dh (r/w) | PHCON | /PHB7 | /PHB6 | /PHB5 | /PHB4 | /PHB3 | /PHB2 | /PHB1 | /PHB0 |
| 0Eh (r/w) | INTEN | GIE | * | TC2IE | TC1IE | TC0IE | INTIE | PBIE | TOIE |
| 0Fh (r/w) | INTFLAG | — | — | TC2IF | TC1IF | TC0IF | INTIF | PBIF | TOIF |

| 地址Bank1 | 说明 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 07h (r/w) | E2PCON | — | — | — | — | E2PCER | E2PSER | E2PBER | E2PWR |
| 08h (r/w) | E2PADR | E2PADR7 | E2PADR6 | E2PADR5 | E2PADR4 | E2PADR3 | E2PADR2 | E2PADR1 | E2PADR0 |
| 09h (r/w) | E2PDIN | E2PDIN7 | E2PDIN6 | E2PDIN5 | E2PDIN4 | E2PDIN3 | E2PDIN2 | E2PDIN1 | E2PDIN0 |
| 0Ah (r) | E2PDATA | E2PDA7 | E2PDA6 | E2PDA5 | E2PDA4 | E2PDA3 | E2PDA2 | E2PDA1 | E2PDA0 |

| 地址Bank2 | 说明 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|--------|--------|--------|---------|--------|--------|--------|
| 07h (r/w) | TCOM | TC0EN | TC0PS2 | TC0PS1 | TC0PS0 | TC0CKS- | ALOAD0 | TC0OUT | PWMOEN |
| 08h (r/w) | TCOC | TCOC7 | TCOC6 | TCOC5 | TCOC4 | TCOC3 | TCOC2 | TCOC1 | TCOC0 |
| 09h (r/w) | TCOR | TCOR7 | TCOR6 | TCOR5 | TCOR4 | TCOR3 | TCOR2 | TCOR1 | TCOR0 |
| 0Ah (r/w) | TC1M | TC1EN | TC1PS2 | TC1PS1 | TC1PS0 | TC1CKS | ALOAD1 | TC1OUT | PWM1EN |
| 0Bh (r/w) | TC1C | TC1C7 | TC1C6 | TC1C5 | TC1C4 | TC1C3 | TC1C2 | TC1C1 | TC1C0 |
| 0Ch (r/w) | TC1R | TC1R7 | TC1R6 | TC1R5 | TC1R4 | TC1R3 | TC1R2 | TC1R1 | TC1R0 |
| 0Dh (r/w) | TC2M | TC2EN | TC2PS2 | TC2PS1 | TC2PS0 | TC2CKS | ALOAD2 | TC2OUT | PWM2EN |
| 0Eh (r/w) | TC2C | TC2C7 | TC2C6 | TC2C5 | TC2C4 | TC2C3 | TC2C2 | TC2C1 | TC2C0 |
| 0Fh (r/w) | TC2R | TC2R7 | TC2R6 | TC2R5 | TC2R4 | TC2R3 | TC2R2 | TC2R1 | TC2R0 |

Legend: - = unimplemented, read as '0', * = unimplemented, read as '1', NG= no used bit



7.0、功能介绍

7.1、寄存器操作

7.1.1、 INDF (间接寻址寄存器)

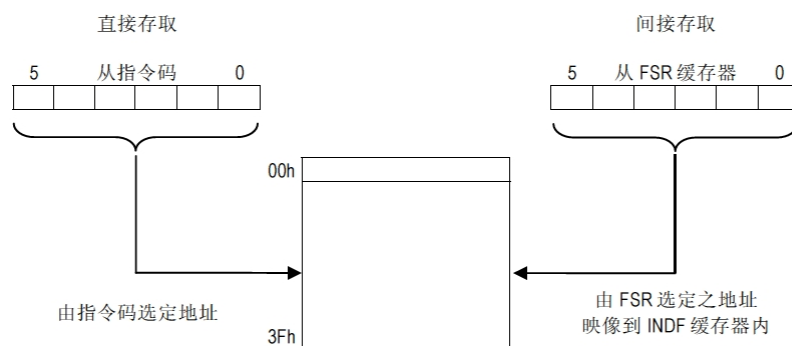
| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|---------------------------|----|----|----|----|----|----|----|
| 00h (r/w) | INDF | 通过 FSR 访问数据区(不是一个实际的物理地址) | | | | | | | |

INDF 不是一个实际的物理地址，间接寻址时 INDF 通过 RAM 选择寄存器（FSR）来访问其所指向的地址。间接寻址读操作直接读地址 00h(FSR=" 0")，间接寻址不能对 INDF 直接进行写操作（尽管有些状态会发生改变）。FSR 的 5-0 位可以用来选择 64 个寄存器（地址:00h ~ 3Fh）。

例: 间接寻址

- 地址 38 内容为 10h
- 地址 39 内容为 0Ah
- 将 38 写入 FSR 中
- 通过 A 读 INDF 返回 10h
- FSR 加 1 (@FSR=39h)
- 通过 A 读 INDF 返回 0A h

图 2. 1: 直接/间接存取





7.1.2、TMRO (定时/计数器 Time lock/Counter register)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-----------|----|----|----|----|----|----|----|
| 01h (r/w) | TMRO | 8 位定时/计数器 | | | | | | | |

TMRO 是一个 8 位定时/计数器寄存器，Timer0 的时钟源可以取值于指令周期或外部实时钟（TOCKI pin），使用外部时钟需要设置 OPTION 的 TOCS(TOCS=5)位为 1。

使用 TMRO 的预置器需要设置 OPTION 的 PSA (PSA =3)位为 0，这种模式下 TMRO 值的改变，预置器被清零。

7.1.3、PCL (Low Bytes of Program Counter) & Stack

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-----|----------|----|----|----|----|----|----|----|
| 02h (r/w) | PCL | PC 低 8 位 | | | | | | | |

QG701 的 PC 指针和堆栈的位数为 10 位，堆栈有 5 级，低位的 PC 指针为 PCL 寄存器，该寄存器时可读写的，高位的 PC 指针为 PCH 寄存器，该寄存器包含 PC<9: 8> 位，该寄存器不能直接读写。PCH 寄存器的改变是通过 PCHBUF 寄存器来实现的。每一条指令执行的时候他的 PC 指针包含下一条指令的操作地址。指令没有改变 PC 内容时候、在每一个指令周期 PC 指针自动加 1。

对于 GOTO 指令有 PC<9: 0>，PCL 映射成 PC<7: 0>，PCHBUF 不变。

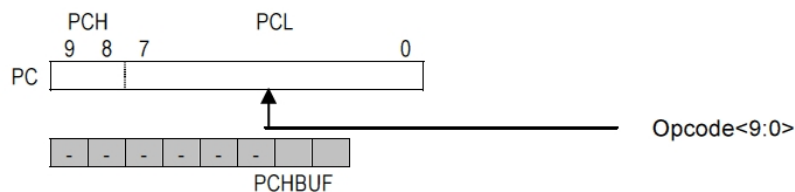
对于 CALL 指令有 PC<9: 0>，下一条指令地址被推进堆栈，PCL 映射成 PC<7: 0>，PCHBUF 不变。

对于 RETIA， RETFIE， RETURN 指令有 PC<9: 0>，PC 的内容更改为出栈信息，PCL 映射成 PC<7: 0>，PCHBUF 不变。

对于其他指令，PCLj 就是目标信息，PC<7: 0>的内容就是指令地址或。不管怎样，PC<9: 8> 来源于 PCHBUF<1: 0> 位 (PCHBUF PCH)。PCHBUF 不会改变，从而 PCH 不会改变。

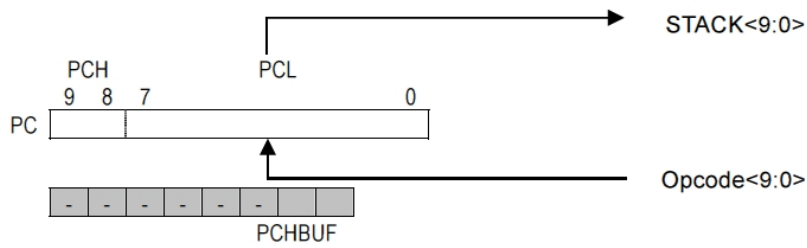
图2. 2: 不同的指令调用PC指针跳转方式

1、GOTO 指令

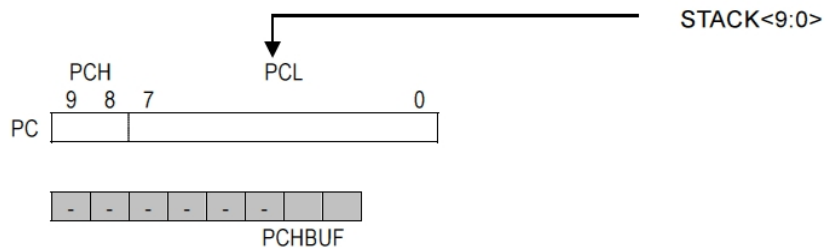




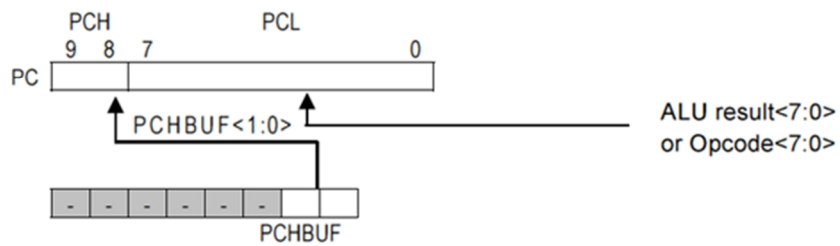
2、CALL 指令



3、RETIA, RETFIE, RETURN 指令



4、以 PCL 为目的的指令



注释 1. PCHBUF 只有在 PCL 内容是目标地址才有效，当 PCL 是运算结果时候，PCHBUF 不起作用。



7.1.4、STATUS (状态字寄存器)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|-----|-----|-----|-----|-----|----|----|----|
| 03h (r/w) | STATUS | RST | GP1 | GP0 | /T0 | /PD | Z | DC | C |

状态字寄存器包含运算标志，结果标志。

指令执行以后可能会影响 STATUS 寄存器的 Z、DC、C 标志位，则不能直接对这三个标志位进行写操作，这些标志位的设置由 MCU 的逻辑自动完成。同时，T0 和 PD 位也是不能通过指令直接改变写操作。因此，与 STATUS 作为目标寄存器的指令后，结果可能会与预期的不同。例如：运行 CLRR STATUS 将把 STATUS 的高三位置零和 Z 标志位置 1 同时该寄存器的内容如下：

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | u | u | 1 | u | u |
|---|---|---|---|---|---|---|---|

u 表示为指令执行前后该位没有发生改变

C : 进位标志

ADDAR, ADDIA

= 1, 有进位

= 0, 无进位

SUBAR, SUBIA

= 1, 无借位

= 0, 有借位

注释：减法是通过将 2 的补第二个操作数的执行。旋转 (RRR, RLR) 指令，该位装载高或低位源寄存器位。

DC : 辅助进位/借位标志. (低四位向高四位进位/借位标志)

ADDAR, ADDIA

= 1, 底 4 位有进位

= 0, 底 4 位无进位

SUBAR, SUBIA

= 1, 底 4 位无借位

= 0, 底 4 位有借位

Z : 零标志位

= 1, 算术或逻辑运算结果为“0”时

= 0, 算术或逻辑运算结果不为“0”时

/PD : 系统休眠标志位

= 1, 当系统上电时或执行“CLRWDT”指令后

= 0, 当执行“SLEEP”指令后

/T0 : 看门狗溢出标志位

= 1, 当系统上电时或执行“CLRWDT”或 SLEEP 指令后

= 0, 看门狗定时器溢出



GP1:GP0 :通用寄存器读/写位

RST :定义系统复位类型位.

- = 1, 唤醒 SLEEP 或 Port B 脚位变化唤醒 SLEEP
- = 0, 其他类型唤醒 SLEEP.

7.1.5、FSR(间接寻址指针)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-----|-----|-----|--------|----|----|----|----|----|
| 04h (r/w) | FSR | RP1 | RP0 | 间接寻址指针 | | | | | |

Bi t5:Bi t0 : 用来选择访问间接寻址时目标寄存器地址. 具体描述见 7.1.1。

Bi t7:Bi t6 : 用来选择控制寄存器的页, 控制寄存器页选择只对地址 07H-0FH 有效, 地址 00H-06H 和控制寄存器页选择无关。

- = 00, 选择 bank0
- = 01, 选择 bank1
- = 10, 选择 bank2
- = 11, 没有使用

7.1.6、PORTA, PORTB (Port 寄存器)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|------|------|------|------|------|------|------|------|
| 05h (r/w) | PORTA | — | — | — | — | I0A3 | I0A2 | I0A1 | I0A0 |
| 06h (r/w) | PORTB | I0B7 | I0B6 | I0B5 | I0B4 | I0B3 | I0B2 | I0B1 | I0B0 |

读端口(PORTA, PORTB 寄存器)的状态依赖于该端口是输入/输出模式, 写端口是向锁存器写数据。
 PORTA 是一个 4 位端口数据寄存器, 只有低 4 位被使用 (PORTA<3:0>). Bi t7~Bi t4 没有使用, 置 0。
 PORTB 是一个 8 位端口数据寄存器。

7.1.7、TCXCON (TCX 定时器模式控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|----|----|----|----|--------|---------|--------|---------|
| 07h (r/w) | TCXCON | NG | NG | NG | NG | TC10EN | PWM1INV | TC20EN | PWM2INV |

PWM2INV: I0B2 口输出 PWM 或者 BUZZER 波形取反
 = 0, I0B2 口输出 PWM2 或 BUZZER2 波形
 = 1, I0B2 口输出 PWM2 反向或者 BUZZER2 反向波形

TC20EN: TC2 定时器使能信号选择
 = 0, TC2 定时器使能信号为 TC2EN
 = 1, TC2 定时器使能信号为 TCOEN



PWM1INV: IOB1 口输出 PWM 或者 BUZZER 波形取反
 = 0, IOB1 口输出 PWM1 或 BUZZER1 波形
 = 1, IOB1 口输出 PWM1 反向或者 BUZZER1 反向波形

TC10EN: TC1 定时器使能信号选择
 = 0, TC1 定时器使能信号为 TC1EN
 = 1, TC1 定时器使能信号为 TCOEN

Bit7:Bit4: 通用读写位

7.1.8、PCON (电源控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|------|-----|-------|------|--------|----------|----------|----------|
| 08h (r/w) | PCON | WDTE | EIS | LVDTE | LVDF | LVDREB | LVDSSEL2 | LVDSSEL1 | LVDSSEL0 |

LVDSSEL2:LVDSSEL0: 当然 LVDREB=1 时, 选择 LVD 检测点
 = 000, NO LVD
 = 001, LVD=2.0V, 睡眠时关闭
 = 010, LVD=2.0V
 = 011, LVD=3.6V
 = 100, LVD=1.8V
 = 101, LVD=2.2V
 = 110, LVD=2.4V
 = 111, LVD=2.6V

LVDREB : LVD 功能复用选择位
 = 0, LVD 作为低压复位点
 = 1, LVD 作为低压检测点

LVDF : LVD 低电压检测标志位
 = 0, VDD 高于设定的低压检测点
 = 1, VDD 低于设定的低压检测点

LVDTE : LVDT (低电压检测) 使能位
 = 0, 关闭 LVDT
 = 1, 使能 LVDT

EIS : 定义管脚 B0/INT 功能位
 = 0, IOB0 (双向 I/O 口) is selected. 屏蔽了 INT 功能.
 = 1, INT (外部中断输入脚), 在这种模式下, PORTB 的 IOB0 必须置“1”. IOB0 作为 I/O 口输入功能通过硬件屏蔽了, 读取 INT 管脚信息的与读 PORTB. 方式相同



WDTE : WDT (watch-dog timer) 使能看门狗定时器
 = 0, 关闭 WDT
 = 1, 使能 WDT

7.1.9、WUCON (Port B 输入改变/唤醒控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|------|------|------|------|------|------|------|------|
| 09h (r/w) | WUCON | WUB7 | WUB6 | WUB5 | WUB4 | WUB3 | WUB2 | WUB1 | WUB0 |

WUB0 : = 0, 禁止 IIOB0 输入改变/唤醒功能
 =1, 使能 IOB0 输入改变/唤醒功能

WUB1 : = 0, 禁止 IIOB1 输入改变/唤醒功能
 =1, 使能 IOB1 输入改变/唤醒功能

WUB2 : = 0, 禁止 IIOB2 输入改变/唤醒功能
 =1, 使能 IOB2 输入改变/唤醒功能

WUB3 : = 0, 禁止 IIOB3 输入改变/唤醒功能
 =1, 使能 IOB3 输入改变/唤醒功能

WUB4 : = 0, 禁止 IIOB4 输入改变/唤醒功能
 =1, 使能 IOB4 输入改变/唤醒功能

WUB5 : = 0, 禁止 IIOB5 输入改变/唤醒功能
 =1, 使能 IOB5 输入改变/唤醒功能

WUB6 : = 0, 禁止 IIOB6 输入改变/唤醒功能
 =1, 使能 IOB6 输入改变/唤醒功能

WUB7 : = 0, 禁止 IIOB7 输入改变/唤醒功能
 =1, 使能 IOB7 输入改变/唤醒功能



7.1.10、PCHBUF (PC 指针高位缓冲区)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|----|----|----|----|----|----|-------------|----|
| 0Ah (r/w) | PCHBUF | — | — | — | — | — | — | 电脑 2MSBs 缓冲 | |

Bit1:Bit0 : 见 7.1.3

Bit7:Bit2 : 没有使用, 置 0

7.1.11、PDCON (I/O 下拉控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|----|-------|-------|-------|-------|-------|-------|-------|
| 0Bh (r/w) | PDCON | * | /PDB2 | /PDB1 | /PDB0 | /PDA3 | /PDA2 | /PDA1 | /PDA0 |

PDA0 : = 0, 使能 IOA0 内部下拉
 = 1, 禁止 IOA0 内部下拉

/PDA1 : = 0, 使能 IOA1 内部下拉
 = 1, 禁止 IOA1 内部下拉

/PDA2 : = 0, 使能 IOA2 内部下拉
 = 1, 禁止 IOA2 内部下拉

/PDA3 : = 0, 使能 IOA3 内部下拉
 = 1, 禁止 IOA3 内部下拉

/PDB0: = 0, 使能 IOB 0 内部下拉
 =1, 禁止 IOB0 内部下拉

/PDB1: = 0, 使能 IOB1 内部下拉
 =1, 禁止 IOB1 内部下拉

/PDB2: = 0, 使能 IOB2 内部下拉
 =1, 禁止 IOB2 内部下拉

Bit7 : 没有使用, 置 1



7.1.12、ODCON (I/O 开漏控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|------|------|------|------|----|------|------|------|
| 0Ch (r/w) | ODCON | ODB7 | ODB6 | ODB5 | ODB4 | — | ODB2 | ODB1 | ODB0 |

ODB0 : = 0, 禁止 IOB0 内部开漏
=1, 使能 IOB0 内部开漏

ODB1 : = 0, 禁止 IOB1 内部开漏
=1, 使能 IOB1 内部开漏

ODB2 : = 0, 禁止 IOB2 内部开漏
=1, 使能 IOB2 内部开漏

Bit3 : 没有使用, 置 0

ODB4 : = 0, 禁止 IOB4 内部开漏
=1, 使能 IOB4 内部开漏

ODB5 : = 0, 禁止 IOB5 内部开漏
=1, 使能 IOB5 内部开漏

ODB6 : = 0, 禁止 IOB6 内部开漏
=1, 使能 IOB6 内部开漏

ODB7 : = 0, 禁止 IOB7 内部开漏
=1, 使能 IOB7 内部开漏

7.1.13、PHCON (I/O 上拉控制寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0Dh (r/w) | PHCON | /PHB7 | /PHB6 | /PHB5 | /PHB4 | /PHB3 | /PHB2 | /PHB1 | /PHB0 |

/PHB0 : = 0, 使能 IOB0 内部上拉.
= 1, 禁止 IOB0 内部上拉

/PHB1 : = 0, 使能 IOB1 内部上拉.
= 1, 禁止 IOB1 内部上拉

/PHB2 : = 0, 使能 IOB2 内部上拉
= 1, 禁止 IOB2 内部上拉



/PHB3: = 0, 使能 IOB3 内部上拉
= 1, 禁止 IOB3 内部上拉

/PHB4 : = 0, 使能 IOB4 内部上拉.
= 1, 禁止 IOB4 内部上拉

/PHB5 : = 0, 使能 IOB5 内部上拉
= 1, 禁止 IOB5 内部上拉

/PHB6 : = 0, 使能 IOB6 内部上拉
= 1, 禁止 IOB6 内部上拉

/PHB7 : = 0, 使能 IOB7 内部上拉
= 1, 禁止 IOB7 内部上拉

7.1.14、INTEN (中断屏蔽寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-------|-----|----|-------|-------|-------|-------|------|------|
| 0Eh (r/w) | INTEN | GIE | * | TC2IE | TC1IE | TC0IE | INTIE | PBIE | TOIE |

TOIE : Timer0 溢出中断屏蔽位。
= 0, 禁止 Timer0 溢出中断
= 1, 使能 Timer0 溢出中断

PBIE : Port B 输入改变中断屏蔽位
= 0, 禁止 Port B 输入改变中
= 1, 使能 Port B 输入改变中

INTIE : 外部中断屏蔽位
= 0, 禁止外部中断.
= 1, 使能外部中断

TC0IE : TC0 溢出中断屏蔽位。
= 0, 禁止 TC0 溢出中断
= 1, 使能 TC0 溢出中断

TC1IE : TC1 溢出中断屏蔽位。
= 0, 禁止 TC1 溢出中断
= 1, 使能 TC1 溢出中断



TC2IE : TC2 溢出中断屏蔽位。
 = 0, 禁止 TC2 溢出中断
 = 1, 使能 TC2 溢出中断

Bit6 : 没有使用, 置 1

GIE : 中断允许总控位
 = 0, 禁止所有中断. 对于睡眠唤醒模式的中断事件, MCU 将执行 SLEEP 后的指令。
 = 1, 使能所有没有屏蔽的中断. 对于睡眠唤醒模式的中断事件, MCU 将跳转到中断地址 (008h)。

注释: 在中断事件发生时, GIEB 被硬件清零并禁止一切中断, 所以 GIE 以及与该中断相关的中断屏蔽位需要重新开启。RETFIE 为退出中断程序并重新设置 GIE =1 允许中断。

7.1.15、INTFLAG (中断标志寄存器)

| 地址Bank0 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|---------|----|----|-------|-------|-------|-------|------|------|
| 0Fh (r/w) | INTFLAG | — | — | TC2IF | TC1IF | TC0IF | INTIF | PBIF | TOIF |

TOIF : 溢出中断标志, 发生 Timer0 溢出中断置 1, 软件设置清零
 PBIF : Port B 输入改变中断标志 interrupt flag. Port B 输入改变时置 1, 软件设置清零
 INTIF : 外部中断标志. 当管脚 INT 上升沿/下降沿 (是上升沿/下降沿由 INTEDG 位 (OPTION<6>)决定)时置 1, 软件设置清零。

TC0IF : 溢出中断标志, 发生 TC0 溢出中断置 1, 软件设置清零
 TC1IF : 溢出中断标志, 发生 TC1 溢出中断置 1, 软件设置清零
 TC2IF : 溢出中断标志, 发生 TC2 溢出中断置 1, 软件设置清零
 Bit7:BIT6 : 没有使用, 置 0

7.1.16、E2PCON(E2P 控制寄存器)

| 地址Bank1 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|----|----|----|----|--------|--------|--------|-------|
| 07h (r/w) | E2PCON | — | — | — | — | E2PCER | E2PSER | E2PBER | E2PWR |

E2PWR : 内置 EEPROM 写使能, 置 1 开始写 EEPROM, 烧写完自动清零
 E2PBER : 内置 EEPROM 字节擦除使能, 置 1 开始按字节擦除 EEPROM, 擦除完自动清零
 E2PSER : 内置 EEPROM 扇区擦除使能, 置 1 开始按扇区擦除 EEPROM, 擦除完自动清零
 E2PCER : 内置 EEPROM 片擦除使能, 置 1 开始按片擦除 EEPROM, 擦除完自动清零
 Bit7:BIT4 : 没有使用, 置 0



7.1.17、E2PADR(E2P 地址寄存器)

| 地址Bank1 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| 08h (r/w) | E2PADR | E2PADR7 | E2PADR6 | E2PADR5 | E2PADR4 | E2PADR3 | E2PADR2 | E2PADR1 | E2PADR0 |

Bit7:BIT0: 内置 EEPROM 地址，地址 00H~FFH 共 256 字节。

7.1.18、E2PDIN(E2P 数据输入寄存器)

| 地址Bank1 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| 09h (r/w) | E2PDIN | E2PDIN7 | E2PDIN6 | E2PDIN5 | E2PDIN4 | E2PDIN3 | E2PDIN2 | E2PDIN1 | E2PDIN0 |

Bit7:BIT0: 内置 EEPROM 数据输入寄存器，在烧写内置 EEPROM 前将要写的数据写入 E2PDIN 寄存器中。

7.1.19、E2PDATA(E2P 数据寄存器)

| 地址Bank1 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0Ah (r) | E2PDATA | E2PDA7 | E2PDA6 | E2PDA5 | E2PDA4 | E2PDA3 | E2PDA2 | E2PDA1 | E2PDA0 |

Bit7:BIT0: 内置 EEPROM 数据读出寄存器，E2PADR 所指地址中的数据，只读寄存器，用 MOVR 指令直接读入 ACC 中。



7.1.20、TCOM(TCO 控制寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|---------|---------|---------|--------|--------|--------|--------|
| 07h (r/w) | TCOM | TCOEN | TCOPOS2 | TCOPOS1 | TCOPOS0 | TCOCKS | ALOAD0 | TCOOUT | PWMOEN |

PWMOEN: PWM0 输出控制

= 0, 禁止 PWM0 输出, IOB0 为 GPIO 引脚

= 1, 使能 PWM0 输出, IOB0 输出 PWM0 信号, 其占空比由 TCOOUT 和 ALOAD0 控制

TCOOUT: TCO 溢出输出信号控制, 仅当 PWMOEN=0 时有效

= 0, 禁止, IOB0 为 GPIO 引脚

= 1, 使能, IOB0 输出 TCOOUT 信号

ALOAD0: 自动装载控制, 仅当 PWMOEN=0 时有效

= 0, 禁止 TCO 自动装载

= 1, 使能 TCO 自动装载

TCOCKS: TCO 时钟选择

= 0, TCO 选择 Fcpu 作为时钟输入

= 1, TCO 选择 Fhosc 作为时钟输入

TCOPS[2:0]: TCO 分频选择位

= 000, Ftc0/128

= 001, Ftc0/64

= 010, Ftc0/32

= 011, Ftc0/16

= 100, Ftc0/8

= 101, Ftc0/4

= 110, Ftc0/2

= 111, Ftc0/1

TCOEN: TCO 启动控制位

= 0, 禁止 TCO 定时器

= 1, 开启 TCO 定时器



7.1.21、TCOC(TC0 计数寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 08h (r/w) | TCOC | TCOC7 | TCOC6 | TCOC5 | TCOC4 | TCOC3 | TCOC2 | TCOC1 | TCOC0 |

8 位计数器 TCOC 溢出时，TCOIF 置 1 并由程序清零，用来控制 TC0 的中断间隔时间。首先须写入正确的值到 TCOC 和 TCOR 寄存器，并使能 TC0 定时器以保证第一个周期正确。TC0 溢出后，TCOR 的值自动装入 TCOC。

TCOC 初始值的计算公式如下：

$$TCOC \text{ 初始值} = N - (TC0 \text{ 中断间隔时间} * TC0 \text{ 时钟 rate})$$

N为TCX二进制计数范围。各模式下参数的设定如下表所示：

| PWMOEN | ALOAD0 | TC0OUT | N | TCOC 范围 | 注释 |
|--------|--------|--------|-----|---------|-------------|
| 0 | x | x | 256 | 00H-FFH | 每计数 256 次溢出 |
| 1 | 0 | 0 | 256 | 00H-FFH | 每计数 256 次溢出 |
| 1 | 0 | 1 | 64 | 00H-3FH | 每计数 64 次溢出 |
| 1 | 1 | 0 | 32 | 00H-1FH | 每计数 32 次溢出 |
| 1 | 1 | 1 | 16 | 00H-0FH | 每计数 16 次溢出 |

7.1.22、TCOR(TC0 自动装载寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 09h (r/w) | TCOR | TCOR7 | TCOR6 | TCOR5 | TCOR4 | TCOR3 | TCOR2 | TCOR1 | TCOR0 |

TC0 内置自动重装功能，TCOR 寄存器存储重装值。TCOC 溢出时，TCOR 的值自动装入 TCOC 中。TC0 定时器工作在计时模式时，要通过修改 TCOR 寄存器来修改 TC0 的间隔时间，而不是通过修改 TCOC 寄存器。在 TC0 定时器溢出后，新的 TCOC 值会被更新，TCOR 会将新的值装载到 TCOC 寄存器中。但在初次设置 TCOM 时，必须要在开启 TC0 定时器前把 TCOC 以及 TCOR 设置成相同的值。

TC0 为双重缓存器结构。若程序对 TCOR 进行了修改，那么修改后的 TCOR 值首先被暂存在 TCOR 的第一个缓存器中，TC0 溢出后，TCOR 的新值就会被存入 TCOR 缓存器中，从而避免 TC0 中断时间出错以及 PWM0 误动作。

7.1.23、TC1M(TC1 控制寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|--------|--------|--------|--------|--------|--------|--------|
| 0Ah (r/w) | TC1M | TC1EN | TC1PS2 | TC1PS1 | TC1PS0 | TC1CKS | ALOAD1 | TC1OUT | PWM1EN |

具体操作请参考 TCOM



7.1.24、TC1C(TC1 计数寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0Bh (r/w) | TC1C | TC1C7 | TC1C6 | TC1C5 | TC1C4 | TC1C3 | TC1C2 | TC1C1 | TC1C0 |

具体操作请参考 TCOC

7.1.25、TC1R(TC1 自动装载寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0Ch (r/w) | TC1R | TC1R7 | TC1R6 | TC1R5 | TC1R4 | TC1R3 | TC1R2 | TC1R1 | TC1R0 |

具体操作请参考 TCOR

7.1.26、TC2M(TC2 控制寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|--------|--------|--------|--------|--------|--------|--------|
| 0Dh (r/w) | TC2M | TC2EN | TC2PS2 | TC2PS1 | TC2PS0 | TC2CKS | ALOAD2 | TC2OUT | PWM2EN |

具体操作请参考 TCOM

7.1.27、TC2C(TC2 计数寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0Eh (r/w) | TC2C | TC2C7 | TC2C6 | TC2C5 | TC2C4 | TC2C3 | TC2C2 | TC2C1 | TC2C0 |

具体操作请参考 TCOC

7.1.28、TC2R(TC2 自动装载寄存器)

| 地址Bank2 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0Fh (r/w) | TC2R | TC2R7 | TC2R6 | TC2R5 | TC2R4 | TC2R3 | TC2R2 | TC2R1 | TC2R0 |

具体操作请参考 TCOR



7.1.29、ACC (Accumulator)累加器

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|-----------|-----|-----|----|----|----|----|----|----|----|
| N/A (r/w) | ACC | 累加器 | | | | | | | |

累加器是一个内部数据转化、指令操作和存放操作结果的存储单元，不能被访问。

7.1.30、OPTION Register (选项寄存器)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|--------|------|--------|------|------|-----|-----|-----|-----|
| N/A (w) | OPTION | TOTB | INTEDG | TOCS | TOSE | PSA | PS2 | PS1 | PS0 |

通过 OPTION 指令访问。

在执行 OPTION 指令时候，该数据单元由 ACC（累加器）转化为选项寄存器（OPTION Register）。

选项寄存器是一个 7 位只写寄存器，它的一些控制位主要用来配置与 Timer0/WDT 分频器，Timer0，外部中断选项相关信息。

除 INTEDG 位以外其他位是只写并可以置 1。

PS2: PS0 : 分频率选择控制位

| PS2: PS0 | Timer0 Rate | WDT Rate |
|----------|-------------|----------|
| 0 0 0 | 1 : 2 | 1 : 1 |
| 0 0 1 | 1 : 4 | 1 : 2 |
| 0 1 0 | 1 : 8 | 1 : 4 |
| 0 1 1 | 1 : 16 | 1 : 8 |
| 1 0 0 | 1 : 32 | 1 : 16 |
| 1 0 1 | 1 : 64 | 1 : 32 |
| 1 1 0 | 1 : 128 | 1 : 64 |
| 1 1 1 | 1 : 256 | 1 : 128 |

PSA : 分频器选择位.

= 1, WDT (看门狗定时器)

= 0, TMRO (Timer0)

TOSE : TMRO 触发方式控制位

= 1, TOCKI 脚下降沿触发计数

= 0, TOCKI 脚上升沿触发计数



TOCS : TMRO 时钟源选择控制位

- = 1, 外部 TOCKI 脚. 当 IOST IOB2 = “0” .时, IOB2/TOCKI 脚设置为输入
- = 0, 内部指令时钟周期或者 RTC 时钟

INTEDG : 中断触发方式控制位.

- = 1, 中断触发方式为 INT 脚上升沿出发
- = 0, 中断触发方式为 INT 脚下降沿出发

TOTB : TO 时钟 RTC 选择位.

- = 1, TO 时钟选择 RTC 时钟 (外部 32768 晶振)
- = 0, TO 时钟选择内部指令时钟

7.1.31、IOSTA & IOSTB (I/O 口控制寄存器)

| 地址 | 名称 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---------|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| N/A (w) | IOSTA | | | | | IOSTA3 | IOSTA2 | IOSTA1 | IOSTA0 |
| N/A (w) | IOSTB | IOSTB7 | IOSTB6 | IOSTB5 | IOSTB4 | IOSTB3 | IOSTB2 | IOSTB1 | IOSTB0 |

通过 IOST 指令访问

通过指令 IOST R (05h-06h)把累加器 A 的内容加载到 I/O 控制寄存器, 按位将 IOSTA, IOSTB 设为 1 表示该脚为输入 (高阻抗)、设为 0 时表示该脚为输出。

IOST 寄存器只写, 系统复位以后设置为输入 (高阻抗)。



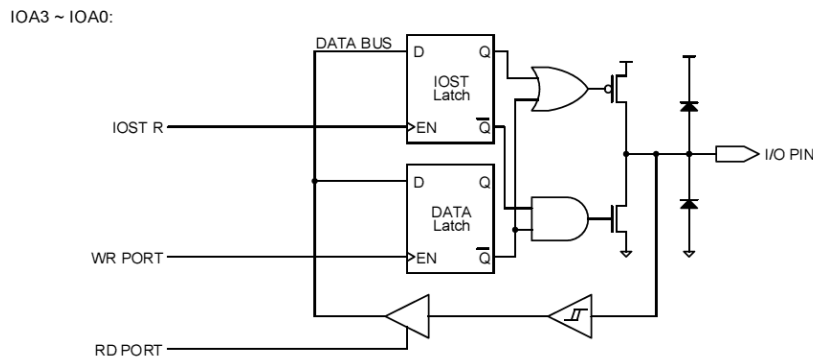
7.2、I/O Ports

Port A 和 PortB 为双向三态 I/O 口。Port A 为 4 脚 I/O 口。Port B 为 8 脚 I/O 口。IOB2 需要通过选项寄存器(Option) 的 TOCS((OPTION<5>))位控制,所有的 I/O 的输入/输出方式由 I/O 控制寄存器(IOSTA, IOSTB) 设置。 IOB<7: 4> 和 IOB<2: 0>有相应的上拉控制位(PHCON 寄存器)来设置使能内部上拉, 如果设置为输出模式, 内部上拉功能会自动关闭。IOA<3: 0>和 IOB<2: 0>有相应的下拉控制位(PDCON 寄存器)来设置使能内部下拉, 如果设置为输出模式, 内部下拉功能会自动关闭。IOB<7: 4>和 IOB<2: 0>有相应的开漏控制位(ODCON 寄存器)来设置使能开漏来设置输出为开漏输出。

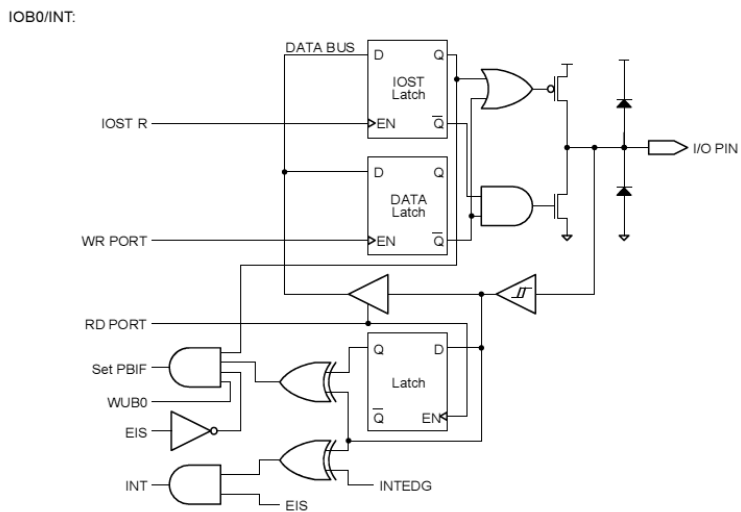
IOB<7: 0> 有输入改变中断/唤醒功能. 它的每个管脚是否具有该功能通过取决于 WUCON 寄存器的相应位。当 EIS(PCON<6>)=1 时, IOB0 作为外部中断输入脚, 在该模式下 IOB0 输入改变中断/唤醒功能被硬件屏蔽, 即使软件已经设置为中断/唤醒功能可用也不可启用该功能。

配置字能交替设置 I/O 口的不同功能, 功能交替设置完以后, 读的 I/O 的值为 0。

图 2.3: I/O 脚的结构图



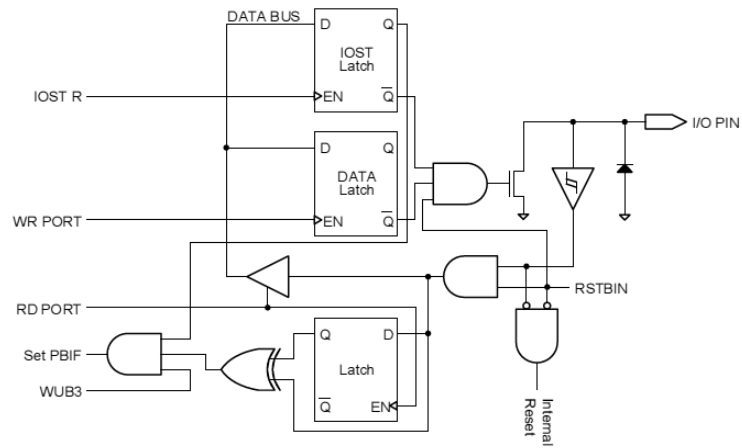
下拉在图中未显示



上拉/下拉和漏极开漏在图中未显示

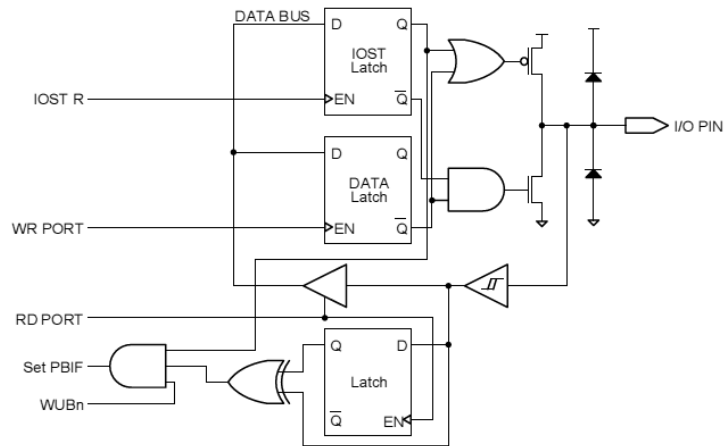


IOB3:



电压在这个引脚禁止超过 VDD

IOB7 ~ IOB4, IOB2 ~ IOB1:



上拉/下拉和漏极开漏在图中未显示



7.3、Timer0/WDT & Prescaler/TC0/TC1/TC2

7.3.1、Timer0

Timer0 为 8 位定时/计数器，Timer0 的时钟源可以是内部或外部时钟源(TOCLKI pin)。

7.3.1.1、使用内部时钟: 定时模式

TOCS(OPTION<5>)=0 为定时模式，定时模式在没有预置器的情况下，定时寄存器每个指令周期自动加 1，设置 TMRO 以后，定时器将在两个时钟周期以后开始自增。

7.3.1.2、使用外部时钟: 计数模式

TOCS(OPTION<5>)=1 为计数模式，是选择通过 TOCK 管脚的上升或下降沿触发 Timer0 寄存器的增加由 TOSE 位 (OPTION<4>)决定，外在时钟要求与内部时钟(Tosc)同步。同步以后，Timer0 实际增加有一个延迟。

在没有预置器的情况下，外部时钟输入同样也可以作为预置器输出；TOCKI 与内部时钟同步时能方便处理在 T2 和 T4 周期上的预分频。因此 TOCKI 为高或低电平必须要保持两个以上时钟周期才有效。

有预置分频时器，外部时钟输入被异步分频器平分，这种常用来计算波形。因此：因此 TOCKI 的一个波形周期至少 4Tosc 才能被预置器平分。

7.3.2、看门狗定时器 (WDT)

看门狗定时器 (WDT) 的运行依赖于芯片里的 RC 振荡器，无需任何额外电路即能工作。不管时钟 OSC1 和 OSC0 管脚是否关闭，它都能运行，如在睡眠模式。在一般操作或睡眠模式情况下，看门狗定时器的溢出都会导致 MCU 复位同时 T0 (STATUS<4>)位被清零。

如 WDTE 位(PCON<7>)清零，看门狗定时器不能工作。

在没有预置器时看门狗的溢出为 18 ms, 4.5ms, 288ms, 72ms 这个时间可以通过 SUT<1:0> 设置。

需要看门狗的 t 溢出周期变长可以通过设置 OPTION 寄存器的看门狗定时器分频大于 1:128., 因此最长的看门狗溢出周期为 36.8 秒。

CLRWDT 指令能使 WDT 和预置器清零，启用看门狗可以防止超时，如果超时 MCU 能复位。

SLEEP 指令重置 WDT 和预置器，启用看门狗就给机器分派了一个最大睡眠时间。

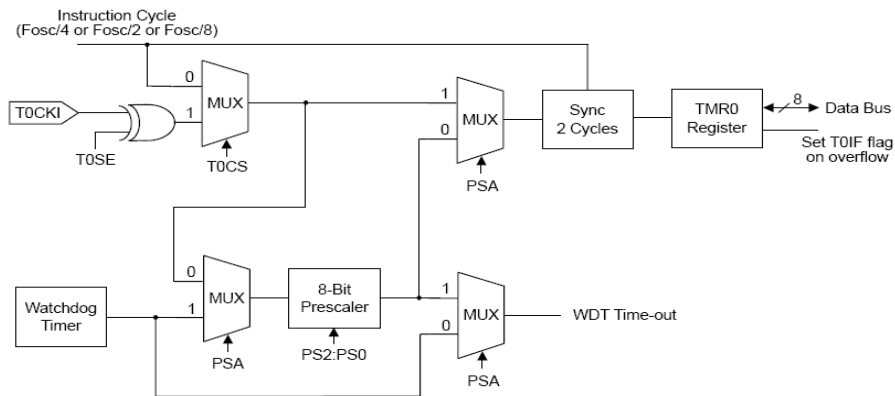
7.3.3、Prescaler (预置器)

有一个 8 位的向下计数器作为 Timer0 和看门狗定时器(WDT)的预置器。注意该预置器只能分配给 Timer0 或 WDT 使用，不能两者同时使用。PSA 位(OPTION<3>) 决定预置器是指派给 Timer0 还是 WDT. PS<2:0> 位(OPTION<2:0>) 配置分频。当作为 Timer0 的预置器的时候，TMRO 会被预置器清零。当作为 WDT 的预置器的时候，CLRWD 指令会清除预置器内容。预置器不能读写，机器复位，预置器各位全为 1。

为了避免机器非正常复位，当 Timer0 或 WDT 的预置器发生改变的时候，需要执行 CLRWDT 或 CLRR TMRO 指令，反之亦然。



图 2.4: Timer0/WDT Prescaler 结构图



7.3.4、TC0

8 位二进制定时/计数器具有基本定时器、Buzzer 和 PWM 功能。基本定时器功能可以支持标志显示 (TC0IF) 和中断操作 (中断向量)。由 TC0M、TC0C、TC0R 寄存器控制 TC0 的中断间隔时间。TC0 还内置 Buzzer 和 PWM 功能，Buzzer 和 PWM 的周期和分辨率由 TC0PS[2:0]、TC0R 寄存器控制，故具有良好性能的 Buzzer 和 PWM 可以处理 IR 载波信号，马达控制和光度调节等。

TC0 的主要用途如下：

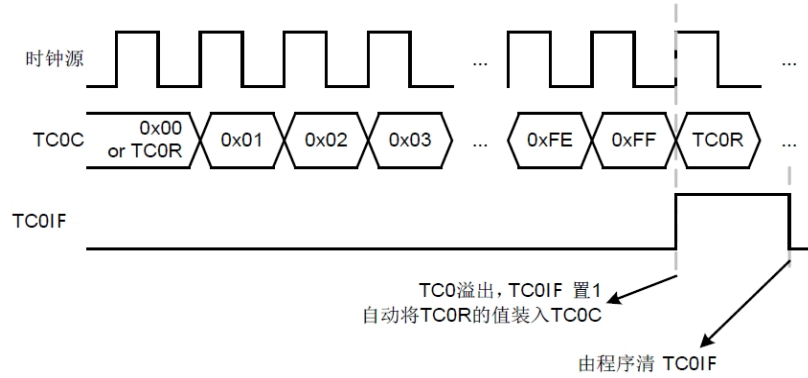
1. 8 位可编程定时器：根据选择的时钟信号，产生周期中断；
2. 中断功能：TC0 定时器支持中断，当 TC0 溢出时，TC0F 置 1，当 TC0IE=1 和 GIE=1 时系统执行中断；
3. PWM 输出：由 TC0rate 和 TC0R 寄存器控制占空比/周期；
4. Buzzer 输出：Buzzer 的输出信号是 TC0 间隔时间的 1/2 个周期；

7.3.4.1、TC0 定时模式

TC0 定时器由 TC0EN 控制。当 TC0EN=0 时，TC0 停止工作；当 TC0EN=1 时，TC0 开始计数。使能 TC0 之前，先要设定好 TC0 的功能模式，如基本定时器、TC0 中断等。TC0C 溢出（从 0FFH 到 00H）时，TC0IF 置 1 以显示溢出状态并由程序清零。在不同的功能模式下，TC0C 不同的值对应不同的操作，若改变 TC0C 的值影响到操作，会导致功能出错。TC0 内置双重缓存器以避免此种状况的发生。在 TC0C 计数的过程中不断的刷新 TC0C，保证将最新的值存入 TC0R（重装缓存器）中，当 TC0 溢出后，TC0R 的值由自动存入 TC0C。进入下一个周期后，TC0 进入新的工作状态。定时器模式下，由 ALOAD0 控制自动重装功能；PWM 模式下，使能 TC0 时，自动使能 TC0 的自动重装功能。如果使能 TC0 中断功能 (TC0IE=1)，在 TC0 溢出时系统执行中断服务程序，在中断时必须由程序清 TC0IF。



图 2.5: TCO 定时溢出示意图



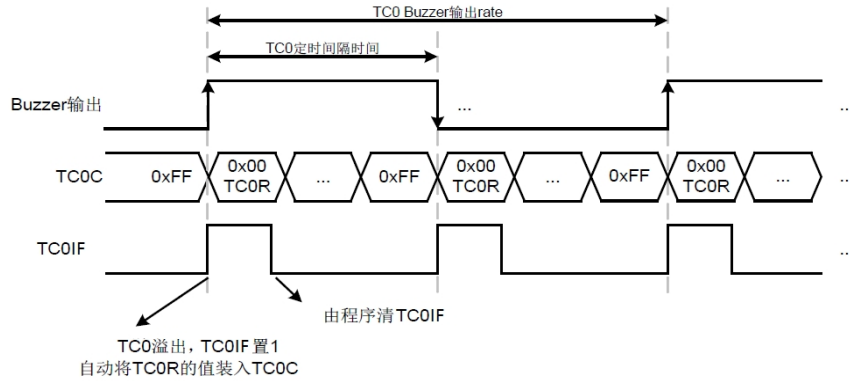
| TCOPS[2:0] | TCO 时钟 | TCO 间隔时间 | | | |
|------------|----------|----------------------------------|----------|---------------------------------|----------|
| | | Fhosc=16MHz Fcpu=Fhosc/4 (4T) | | Fhosc=4Mhz Fcpu=Fhosc/4 (4T) | |
| | | max.(ms) | Unit(us) | max.(ms) | Unit(us) |
| 000b | Ftc0/128 | 8.192 | 32 | 32.768 | 128 |
| 001b | Ftc0/64 | 4.096 | 16 | 16.384 | 64 |
| 010b | Ftc0/32 | 2.048 | 8 | 8.192 | 32 |
| 011b | Ftc0/16 | 1.024 | 4 | 4.096 | 16 |
| 100b | Ftc0/8 | 0.512 | 2 | 2.048 | 8 |
| 101b | Ftc0/4 | 0.256 | 1 | 1.024 | 4 |
| 110b | Ftc0/2 | 0.128 | 0.5 | 0.512 | 2 |
| 111b | Ftc0/1 | 0.064 | 0.25 | 0.256 | 1 |



7.3.4.2、TC0 BUZZER 输出

Buzzer 输出是一个简单的 1/2 占空比信号输出，由 TC0 产生。当 TC0 溢出时，Buzzer 开始输出一个方波，中断间隔时间频率 2 分频后作为 Buzzer 输出的频率。Buzzer 输出的波形图如下所示：

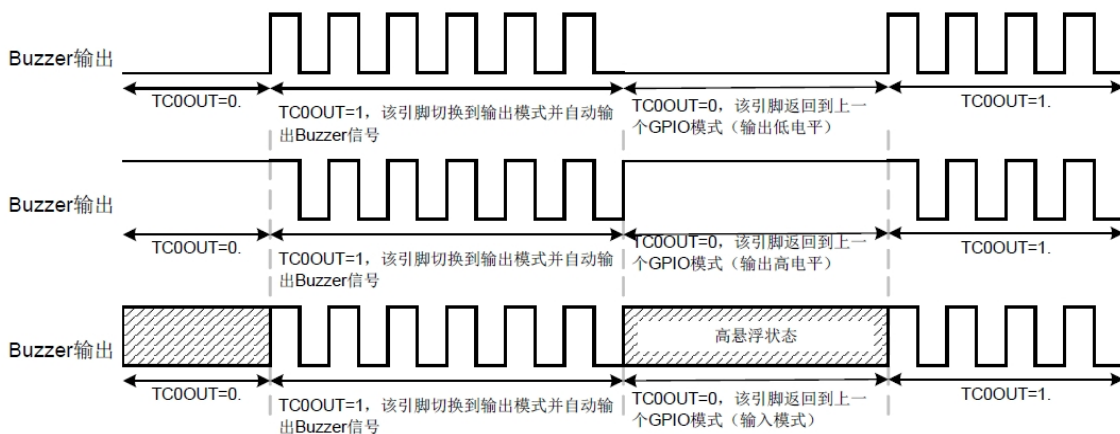
图 2.6: Buzzer 输出示意图



TC0 溢出后，Buzzer 输出时，TC0IF 有效，且当 TC0IE=1 时，使能 TC0 中断功能。但强烈建议小心同时使用 Buzzer 和 TC0 定时器，以确保两种功能都能正常工作。

Buzzer 输出引脚与 GPIO 引脚共用，TC0OUT=1 时，该引脚自动设为 Buzzer 输出引脚。如清 TC0OUT 位以禁止 Buzzer 输出后，该引脚自动返回到最后一个 GPIO 模式。

图 2.7: Buzzer 输出/I/O 切换状态示意图



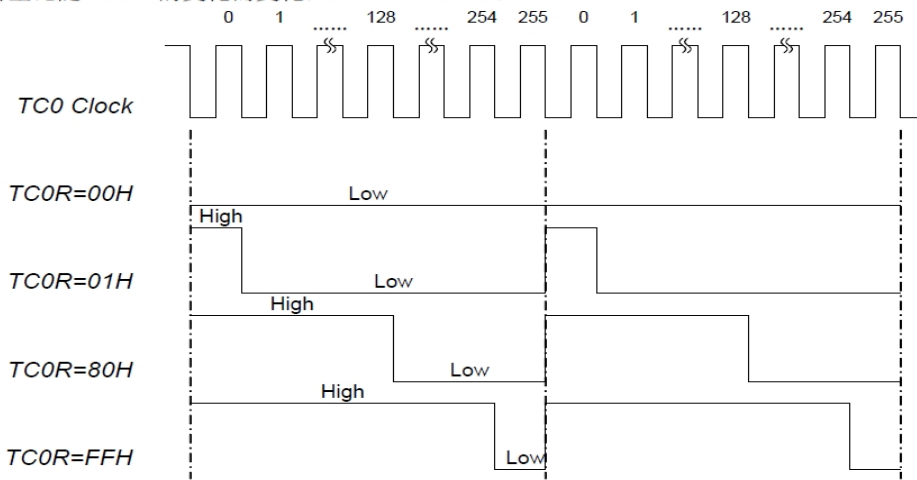


7.3.4.3、TC0 PWM 输出（脉宽调制）

可编程控制占空比/周期的PWM可以提供不同的PWM信号。使能TC0定时器且PMMOEN=1时，由PMMO输出引脚(I0B0)输出PWM信号。PWM首先输出高电平，然后输出低电平。TC0PS[2:0]控制PWM的周期，ALOAD0和TC0OUT决定PWM的分辨率，TCOR寄存器决定PWM的占空比（脉冲高电平的长度）。开启TC0定时器且定时器溢出后，TCOC的初始值为0。当TCOC=TCOR时，PWM输出低电平；TC0溢出时（TCOC的值从0FFH到00H），整个PWM周期完成，并进入下一个周期。TC0溢出时，PWM的一个周期完成。在PWM输出的过程由程序更改PWM的占空比，则在下一个周期开始输出新的占空比的PWM信号。

图2.8: PWM输出示意图

PWM 输出占空比随 TCOR 的变化而变化：0/256~255/256。

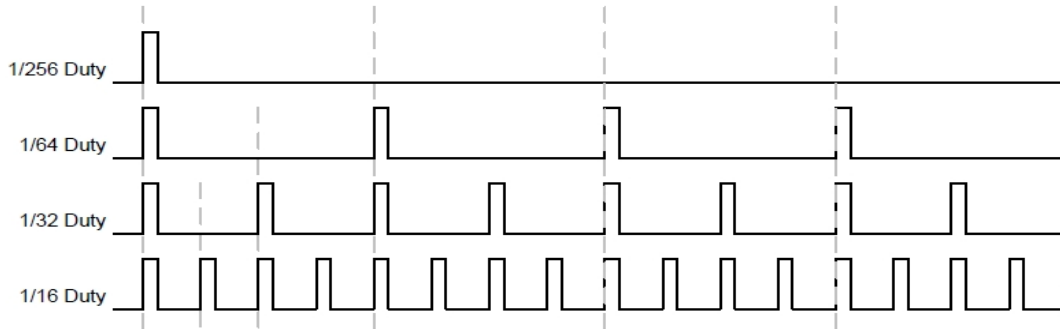


PWM的分辨率由ALOAD0和TC0OUT决定，以实现高速PWM信号。当ALOAD0、TC0OUT = 00时，PWM的分辨率为1/256；ALOAD0、TC0OUT = 01时，PWM的分辨率为1/64；ALOAD0、TC0OUT = 10时，PWM的分辨率为1/32；ALOAD0、TC0OUT = 11时，PWM的分辨率为1/16。若需调制PWM的分辨率，TCOR PWM的占空比控制范围必须调制到一个合适的分辨率。PWM输出过程中，TC0溢出时，TCOIF有效，TCOIE=1时，则使能TC0中断。但强烈建议小心同时使用PWM和TC0定时器功能，保证两种功能都能正常工作。



图 2.9: PWM 周期设置示意图

| ALOAD0 | TC0OUT | PWM 分辨率 | TC0R 有效值 | TC0R 有效值 (二进制) |
|--------|--------|---------|----------|---------------------|
| 0 | 0 | 256 | 00H~FFH | 00000000b~11111111b |
| 0 | 1 | 64 | 00H~3FH | xx000000b~xx111111b |
| 1 | 0 | 32 | 00H~1FH | xxx00000b~xxx11111b |
| 1 | 1 | 16 | 00H~0FH | xxxx0000b~xxxx1111b |



PWM 输出引脚和 GPIO 引脚共用，PWM0OUT=1 时，该引脚自动输出 PWM 信号。如果清 PWM0OUT 位以禁止 PWM 时，该引脚返回到最后一个 GPIO 模式。

7.3.5、TC1/TC2

所有操作具体参考TC0



7.4、中断方式

QG701 系统具备有六种中断方式:

1. INT 管脚的外部中断
2. TMR0 溢出中断
3. Port B 输入改变中断 (IOB7:IOB0 脚)
4. TCO 溢出中断
5. TC1 溢出中断
6. TC2 溢出中断

INTFLAG 为中断标志寄存器, 决定该寄存器机器所发生的中断状态。

中断允许总控位 GIE (INTEN<7>), 能使所有中断被开放 (GIE=1) 或屏蔽所有中断(GIE=0), 每中断能否启用决定 INTEN 寄存器同时保证 GIE=1。

中断发生时 GIE 位 (在中断发生前 GIE 位和该中断相关的中断屏蔽位置 1) 被硬件清零从而禁止进一步中断 (QG701 不区分中断优先级别), 同时下条指令跳到 008h 后开始执行。中断标志位在中断允许总控位 GIE 重新置 1 的时候需要被软件清零以防止重复中断。一个中断标志位 (PBIF 除外的) 会被它的中断事件置 1, 而不管与它相关的中断屏蔽位是否启用。通过 INTFLAG 和 INTEN 的相应中位来判断是否发生中断以及中断类型。当通过 INT 指令发生软中断时, 下条指令跳到 002 后开始执行。

7.4.1、外部中断

外部中断 INT 管脚上升沿还是下降沿触发由 INTEDG 位 (OPTION<6>)决定, 当一个有效的跳变发生时标志位 INTIF 置 1, 如 INTIE 位(INTEN<2>)清零, 该中断被屏蔽。

在睡眠之前 INTIE 位已被置 1, INT 管脚可以作为系统睡眠条件。在睡眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。

7.4.2、Timer0 中断

TMR0 发生溢出 (Ffh→00h)时 TOIF 标志位置 1 (INTFLAG<0>). TOIE 位(INTEN<0>)清零, 该中断被屏蔽。

7.4.3、Port B 输入改变中断

输入改变中断触发时 IOB<7:0> PBIF 标志位置 1 (INTFLAG<1>). PBIE 位(INTEN<1>)清零, 该中断被屏蔽。

在输入改变中断发生之前, 必须读取 port B 信息 与 PortB 的管脚相对应的 WUBn 位 (WUCON<7:0>) i 清零或设置为输出或 IOB0 脚设置为外部中断输入脚 INT 都拥有该功能。PBIE 在睡眠之前置 1, Port B 输入脚改变中断也可以作为睡眠唤醒条件。在睡眠之前 GIE 位已被置 1 机器唤醒以后会执行中断服务程序, 否则会运行睡眠以后的下一条指令。



7.4.4、TC0 中断

TC0 发生溢出 (FFh→00h)时 TC0IF 标志位置 1 (INTFLAG<3>). TC0IE 位(INTEN<3>)清零, 该中断被屏蔽。

7.4.5、TC1 中断

TC1 发生溢出 (FFh→00h)时 TC1IF 标志位置 1 (INTFLAG<4>). TC1IE 位(INTEN<4>)清零, 该中断被屏蔽。

7.4.6、TC2 中断

TC2 发生溢出 (FFh→00h)时 TC2IF 标志位置 1 (INTFLAG<5>). TC2IE 位(INTEN<5>)清零, 该中断被屏蔽。

7.5、内置 EEPROM 操作

内置 EEPROM 共有 256 个字节, 分为 16 个扇区, 每个扇区以及每个字节都可以单独擦除, 也可以整个地址全部擦除。

E2PADR 寄存器为内置 EEPROM 的地址, 通过给 E2PADR 写入不同的值来选择不同的地址。

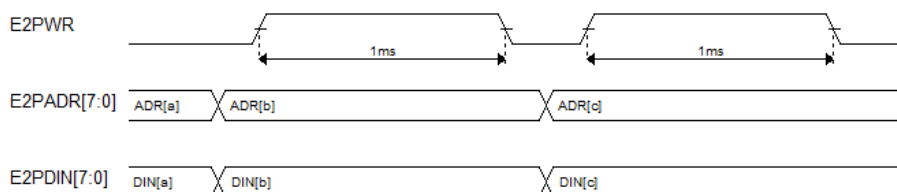
E2PDIN 寄存器为即将要写入内置 EEPROM 的值, 通过给 E2PDIN 写入不同的值来修改将要写入的数据。

内置 EEPROM 只能按字节写入数据, 写入时间大约为 1ms。擦除时间大约为 100ms。

7.5.1、内置 EEPROM 烧写

将需要烧写的地址写入 E2PADR 寄存器, 将需要烧写的的数据写入 E2PDIN 寄存器, E2PCON 控制寄存器中的 E2PWR 位置 1, 内置 EEPROM 开始烧写, 大约 1ms 左右烧写结束, 烧写完毕后 E2PWR 位自动清零。

图 2.10:内置 EEPROM 烧写示意图

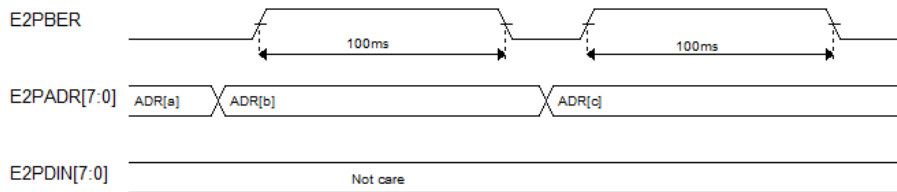




7.5.2、内置 EEPROM 的字节擦除

将需要擦除的地址写入 E2PADR 寄存器，E2PCON 控制寄存器中的 E2PBER 位置 1，内置 EEPROM 开始擦除，大约 100ms 左右擦除结束，擦除完毕后 E2PBER 位自动清零。

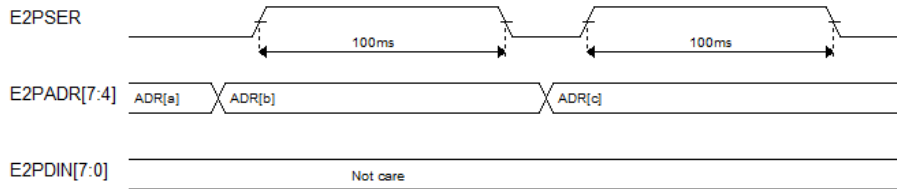
图 2.11:内置 EEPROM 字节擦除示意图



7.5.3、内置 EEPROM 的扇区擦除

将需要擦除的地址写入 E2PADR 寄存器，扇区地址有 E2PADR 高 4 位决定，E2PCON 控制寄存器中的 E2PSEr 位置 1，内置 EEPROM 开始擦除，大约 100ms 左右擦除结束，擦除完毕后 E2PSEr 位自动清零。

图 2.12:内置 EEPROM 扇区擦除示意图

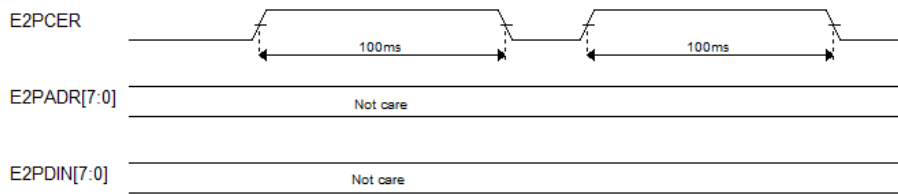




7.5.4、内置 EEPROM 的片擦除

与地址无关，所有地址中的数据将被擦除，E2PCON 控制寄存器中的 E2PCER 位置 1，内置 EEPROM 开始擦除，大约 100ms 左右擦除结束，擦除完毕后 E2PCER 位自动清零。

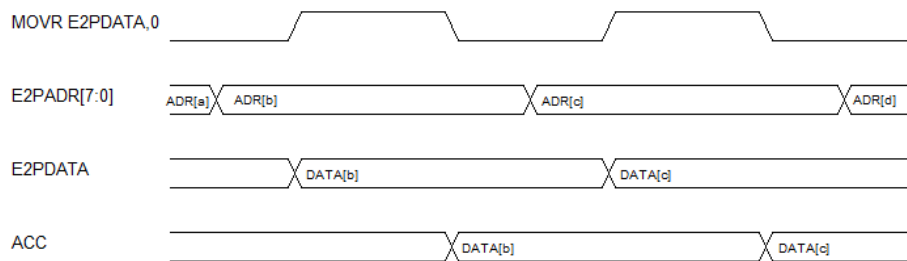
图 2.13:内置 EEPROM 片擦除示意图



7.5.5、内置 EEPROM 的读操作

将需要读出数据的地址写入 E2PADR 寄存器，然后通过 MOVR 指令读出 E2PDATA 寄存器中的数据，数据存入 ACC 中。E2PDATA 是只读寄存器。

图 2.14: 内置 EEPROM 读操作示意图





7.6、省电模式 (SLEEP)

执行 SLEEP 指令以后机器进入省电模式。

执行 SLEEP 指令， /PD 位清零 (STATUS<3>)， /TO 位置 1，看门狗清零同时保持运行状态，晶体停振。

I/O 维持原状

7.6.1、睡眠唤醒

在睡眠状态下，单片机能通过以下方式唤醒：

1. RSTB 管脚复位
2. 看门狗复位 (机器设置了看门狗)。
3. RBO/INT 管脚中断，或 PORTB 输入改变中断。

外部的 RSTB 管脚和看门狗溢出都能使机器复位。通过查看 /PD 和 /TO 位可以检测机器是哪种复位， /PD 位置 1 为上电复位，置 0 为执行 SLEEP， /TO 位置 0 为看门狗溢出复位。

机器通过中断唤醒，该中断屏蔽位置 1，中断唤醒不管 GIE 是否置 1。当 GIE 位被清零，机器唤醒以后执行 SLEEP 指令以后的指令；当 GIE 位被置 1，机器唤醒以后跳转到中断复位地址 (008h)。在高频或低频模式机器复位延迟时间为 18/4.5/288/72ms (该延迟时间由 SUT<1:0>设置) 加上 64 个振荡周期。

在 IRC/ERIC or ERC 模式，机器复位延迟时间为 640us。



7.7、复位

QG701 单片机能通过以下方式复位:

1. 上电复位(POR)
2. 掉电复位(Brown-out Reset BOR)
3. RSTB 管脚复位
4. 看门狗 WDT 溢出复位

一些寄存器在一些复位条件下没有影响, 在上电和其他一些复位情况下它们的状态是未知的。大多数寄存器会回到复位状态在上电复位, RSTB 管脚复位, 看门狗 WDT 溢出复位。

对 Vdd 上升信号检测告之芯片是否加上上电复位脉冲信号。 要使用这个特点, 用户需要把 RSTB 管脚连接到 Vdd。掉电复位作为一种典型应用主要用在 AC 或重载交换的应用上。

芯片上的低电压检测模块到电压低于一个固定的电压也会对使芯片复位, 这样能保证芯片只能在正常电压范围内工作。

RSTB 或 WDT 睡眠唤醒也导致芯片复位, 其复位操作的不会在睡眠之前。

根据不同的复原状态设置对/T0 和/PD 位(STATUS<4 :3>)置 1 或清零。

7.7.1、上电复位计数器(Power-up Reset Timer PWRT)

上电复位计数器提供一个 18/4.5/288/72ms 延迟时间 (该延迟时间由 SUT<1:0>设置) (或 640us, 基于不同的振荡源和复位条件) 在 Power-on Reset (POR), Brown-out Reset (BOR), RSTB Reset 或 看门狗溢出复位。只要 PWRT 在运行, 设备就一直保持的复位状态。Vdd、温度和其他变化而会影响 PWDT 控制的设备延迟时间。

表2.1: PWRT Period

| Oscillator Mode | Power-on Reset Brown-out Reset | RSTB Reset WDT time-out Reset |
|-------------------|-----------------------------------|----------------------------------|
| ERC & IRC/ERIC | 18/4.5/288/72 ms | 640 us |
| HF & LF & IRC_RTC | 18/4.5/288/72 ms | 18/4.5/288/72ms |

7.7.2、振荡启动计数器(Oscillator Start-up Timer OST)

在 HF 或 LF 或 IRC_RTC 振荡模式下在 PWRT 延迟 (18/4.5/288/72ms) 之后振荡启动计数器会再提供一个 64 个 clock 的延迟。这种延迟晶体谐振器能提供稳定的振荡源, 这段时间内只要 OST 在工作, 设备就一直保持的复位状态。在 OSCI 信号的振幅到达振荡器输入最大振幅之后, 该计数器只开始增加。



7.7.3、复位顺序

QG701 复位时序如下:

1. 复位锁存器置 1 ， PWRT & OST 清零。
2. 当内部的 POR， BOR， RSTB 复位或 WDT 溢出复位脉冲加载完成后， PWRT 开始计数。
3. PWRT 溢出以后， OST 开始计数延迟。
4. OST 延迟完成以后， 复位锁存器清零最后芯片得到一个复位信号。

在高频或低频振荡模式机器复位延迟时间为 18/4.5/288/72ms 加上 64 个振荡周期，在 IRC/ERIC ， ERC 振荡模式单片机会在 Power-on Reset (POR)， Brown-out Reset (BOR)， 或 RSTB 复位以后在延迟 640us ，看门狗溢出复位后再延迟 18/4.5/288/72ms 的时间。

图 2.15: 复位电路结构图

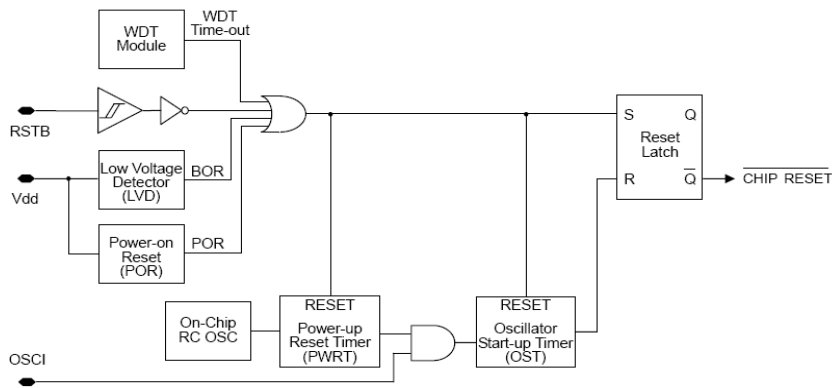


表 2.2: 复位以后各个寄存器状态列表

| 寄存器 | 地址 | 上电复位 掉电复位 | RSTB 复位 WDT 复位 |
|---------------------------|----------|--------------|-------------------|
| ACC | N/A | xxxx xxxx | uuuu uuuu |
| OPTION | N/A | 0011 1111 | 0011 1111 |
| I OSTA | N/A | ---- 1111 | ---- 1111 |
| I OSTB | N/A | 1111 1111 | 1111 1111 |
| INDF | 00h | xxxx xxxx | uuuu uuuu |
| TMRO | 01h | xxxx xxxx | uuuu uuuu |
| PCL | 02h | 1111 1111 | 1111 1111 |
| STATUS | 03h | 0001 1xxx | 000# #uuu |
| FSR | 04h | 00xx xxxx | 00uu uuuu |
| PORTA | 05h | ---- xxxx | ---- uuuu |
| PORTB | 06h | xxxx xxxx | uuuu uuuu |
| General Purpose Registers | 10 ~ 3Fh | Xxxx xxxx | uuuu uuuu |



| Bank0 | 地址 | 上电复位 掉电复位 | RSTB 复位 WDT 复位 |
|---------|-----|--------------|-------------------|
| TCXCON | 07h | 0000 0000 | 0000 0000 |
| PCON | 08h | 1010 0000 | 1010 0000 |
| WUCON | 09h | 0000 0000 | 0000 0000 |
| PCHBUF | 0Ah | ---- --00 | ---- --00 |
| PDCON | 0Bh | 1111 1111 | 1111 1111 |
| ODCON | 0Ch | 0000 0000 | 0000 0000 |
| PHCON | 0Dh | 1111 1111 | 1111 1111 |
| INTEN | 0Eh | 0-00 0000 | 0-00 0000 |
| INTFLAG | 0Fh | --00 0000 | --00 0000 |

| Bank1 | 地址 | 上电复位 掉电复位 | RSTB 复位 WDT 复位 |
|---------|-----|--------------|-------------------|
| E2PCON | 07h | ---- 0000 | ---- 0000 |
| E2PADR | 08h | 0000 0000 | 0000 0000 |
| E2PDIN | 09h | 0000 0000 | 0000 0000 |
| E2PDATA | 0Ah | uuuu uuuu | uuuu uuuu |

| Bank2 | 地址 | 上电复位 掉电复位 | RSTB 复位 WDT 复位 |
|-------|-----|--------------|-------------------|
| TCOM | 07h | 0000 0000 | 0000 0000 |
| TCOC | 08h | 0000 0000 | 0000 0000 |
| TCOR | 09h | 0000 0000 | 0000 0000 |
| TC1M | 0Ah | 0000 0000 | 0000 0000 |
| TC1C | 0Bh | 0000 0000 | 0000 0000 |
| TC1R | 0Ch | 0000 0000 | 0000 0000 |
| TC2M | 0Dh | 0000 0000 | 0000 0000 |
| TC2C | 0Eh | 0000 0000 | 0000 0000 |
| TC2R | 0Fh | 0000 0000 | 0000 0000 |

Legend: u = 不变, x = 未知, - = 不起作用, # = 参见下表的值



表 2.3: RST/ T0 / PD 复位和唤醒后的状态

| RST | /T0 | /PD | 复位方式 |
|-----|-----|-----|------------------------------------|
| 0 | 1 | 1 | Power-on Reset |
| 0 | 1 | 1 | Brown-out reset |
| 0 | u | u | RSTB Reset during normal operation |
| 0 | 1 | 0 | RSTB Reset during SLEEP |
| 0 | 0 | 1 | WDT Reset during normal operation |
| 0 | 0 | 0 | WDT Wake-up during SLEEP |
| 1 | 1 | 0 | Wake-up on pin change during SLEEP |

Legend: u =不变

表: 2.4: /T0 /PD 状态位影响事件

| 事件 | /T0 | /PD |
|--------------------|-----|-----|
| Power-on | 1 | 1 |
| WDT Time-Out | 0 | u |
| SLEEP instruction | 1 | |
| CLRWDT instruction | 1 | 1 |

Legend: u =不变

7.8、十六进制转化为十进制 (Hexadecimal Convert to Decimal HCD)

QG701 具有十进制格式化功能：当一个寄存器里面的内容需要十进制转化的时候，在执行操作 ALU 以后必须把结果进行相应的进制转化。

一个数据在处理过程中进行了转化成了十进制，那么所有对这个数进行的操作（包含存放该数据的 RAM 单元，accumulator (ACC)，立即数，以及所要查表信息）都的进行十进制转化，这样的运算结果才正确。

DAA 指令能在加法运算完成以后将 ACC 里的数据从十六进制转化为十进制重存给 ACC

转换操作在例子 2.2 中被说明。

例 2.2: DAA 转化

| Address | Code |
|---------|---|
| NA | #include <FC701.ASH> |
| n | ... |
| n+1 | MOVIA 0x90 ;Set immediate data = decimal format number "90" (ACC ← 90h) |
| n+2 | MOVAR 0x30 ;Load immediate data "90" to data memory address 30H |
| n+3 | MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h) |
| n+4 | ADDAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-added ;the result loads to the ACC (ACC ← A0h, C ← 0) |
| n+5 | DAA ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "00" and the carry bit C is "1". This represents the ;decimal number "100" |
| n+6 | ... |



DAS 指令能在减法运算完成以后将 ACC 里的数据从十六进制转化为十进制重存给 ACC 转换操作在例子 2.3 中被说明。

例 2.3: DAS 转化

| Address | Code |
|---------|---|
| NA | #include <FC701.ASH> |
| n | ... |
| n+1 | MOVIA 0x10 ;Set immediate data = decimal format number "10" (ACC ← 10h) |
| n+2 | MOVAR 0x30 ;Load immediate data "90" to data memory address 30H |
| n+3 | MOVIA 0x20 ;Set immediate data = decimal format number "20" (ACC ← 20h) |
| n+4 | SUBAR 0x30,A ;Contents of the data memory address 30H and ACC are binary-subtracted ;the result loads to the ACC (ACC ← F0h, C ← 0) |
| n+5 | DAS ;Convert the content of ACC to decimal format, and restored to ACC ;The result in the ACC is "90" and the carry bit C is "0". This represents the ;decimal number "-10" |
| n+6 | ... |

7.9、振荡器配置 (Oscillator Configurations)

QG701 有七种不同的振荡模式，用户可通过编程 Fosc 配置位来选择相应的振荡方式：

- LF: 低频晶体谐振器
- HF: 高频晶体谐振器
- IRC: 内部电阻内部电容振荡器
- ERIC: 外部电阻内部电容振荡器
- ERC: 外部 RC 振荡器
- XT: 晶体/陶瓷振荡器
- IRC_RTC: Fcpu 使用内部电阻电容振荡器/T0 使用低频晶体谐振器

在 LF, XT, HF 或 IRC_RTC 模式下，一个晶体或陶瓷谐振器连接到 OSC1 和 OSC0 管脚建立振荡源。当在 LF, XT 或 HF 模式下，单片机通过 OSC1 脚接入外部时钟源。使用 ERC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，RC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

使用 IRC (IRC_RTC) /ERIC 振荡模式为成本节省主要使用在定时无须精确场合下的应用，单片机具有 4 种不同的振荡频率，8MHz, 4MHz, 1MHz, 和 455KHz, 通过 (RCM<1:0>) 来选择一种，或则用户改变外部电阻来实现。ERIC 振荡器频率取决于电阻和电容 (Cext)，操作温度以及其他过程参数。

图 2.16: HF, XT 和 LF 振荡器模式(晶振振荡器或陶瓷振荡器)

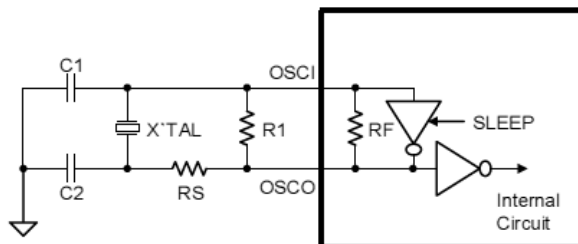




图2.17: HF, XT 和 LF 振荡器模式 (外部时钟输入操作)

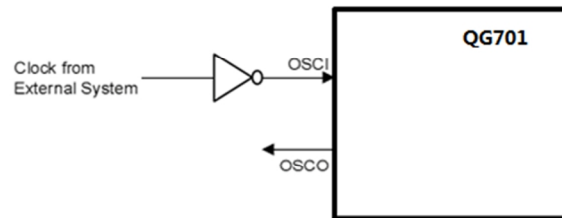


图 2.18: ERC 振荡器模式 (外部电阻电容振荡器)

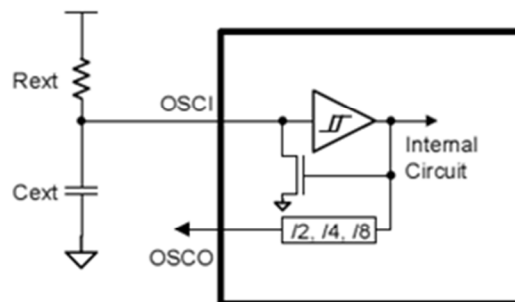


图 2.19: ERIC 振荡器模式 (外部电阻 内部电容 振荡器)

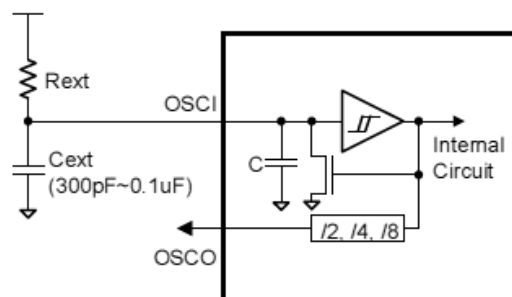
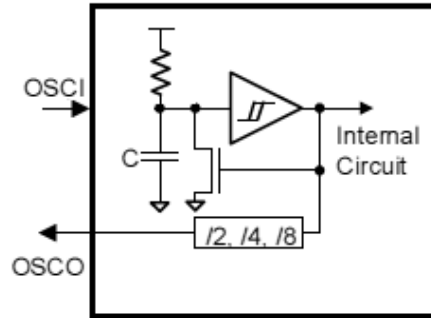




图 2.20: IRC 振荡器模式 (内部电阻电容 振荡器)





7.10、配置选项

表 2.4: 配置选项 0

| 位 | 名称 | 说明 |
|------|------------|---|
| 2~0 | Fosc<2: 0> | 振荡源选择位 = 0 0 0→ERC mode (external R & C) (默认) IOB4/OSCO 管脚为取 OSCOUT 功能 = 1 1 0→HF mode = 1 0 1→XT mode = 1 0 0→LF mode = 0 1 1→IRC mode (internal R & C) IOB4/OSCO 管脚为取 OSCOUT 功能 = 0 0 1→ERIC mode (external R & internal C) IOB4/OSCO 管脚为取 OSCOUT 功能 = 1 1 1→IRC_RTC mode (external R & internal C for Fcpu/LF for T0) |
| 5~3 | LVDT<2: 0> | 低电压检测选择位 = 0 0 0→禁止低电压检测(默认) = 0 0 1→enable, LVDT vol tage = 2.0V, 睡眠模式关闭 = 0 1 0→enable, LVDT vol tage = 2.0V = 0 1 1→enable, LVDT vol tage = 3.6V = 1 0 0→enable, LVDT vol tage = 1.8V = 1 0 1→enable, LVDT vol tage = 2.2V = 1 1 0→enable, LVDT vol tage = 2.4V = 1 1 1→enable, LVDT vol tage = 2.6V |
| 7~6 | RCM<1: 0> | IRC 选择位 = 0 0→4MHz (默认) = 0 1→8MHz = 1 0→1MHz = 1 1→455KHz |
| 10~8 | SUT<2: 0> | PWRT & WDT 计数周期选择位 = 0 0 0→PWRT = WDT prescaler rate = 18ms (默认) = 0 0 1→PWRT = WDT prescaler rate = 4.5ms = 0 1 0→PWRT = WDT prescaler rate = 288ms = 0 1 1→PWRT = WDT prescaler rate = 72ms = 1 0 0→PWRT = 640us, WDT prescaler rate = 18ms = 1 0 1→PWRT = 640us, WDT prescaler rate = 4.5ms = 1 1 0→PWRT = 640us, WDT prescaler rate = 288ms = 1 1 1→PWRT = 640us, WDT prescaler rate = 72ms |
| 11 | OSCOUT | IRC/ERIC/ERC 模式下 IOB4/OSCO 功能选择位置 = 0, IOB4(默认) = 1, OSCO |
| 12 | RSTBIN | IOB3/RSTB 选择位置 = 0, IOB3 (默认) = 1, RSTB |



表 2.5: 配置选项 1

| 位 | 名称 | 说明 |
|------|--------------|--|
| 0 | WDTEN | 看门狗使能位 = 0, 禁止 WDT(默认) = 1, 使能 WDT |
| 1 | PROTECT | 代码保护选择位 = 0→代码不加密 EPROM code protection off (默认) = 1→代码加密 EPROM code protection on |
| 3-2 | OSCD<1:0> | 指令运行周期选择位 = 0 0→4 个振荡周期 (默认) = 1 0→2 个振荡周期 = 1 1→8 个振荡周期 |
| 4 | PBWKEN | I/O 口中断唤醒使能 = 0, I/O 口中断唤醒使能通过 WUCON 寄存器控制, LVDTE 软件控制 (默认) = 1, I/O 口默认打开中断唤醒使能, LVDTE 使能打开 (无法通过软件设置) |
| 5 | RDPORT | I/O 作为输出时, 读端口方式控制位 = 0, 从寄存器读 (默认) = 1, 从管脚读 |
| 6 | SCHMITT | I/O 输入缓冲控制位 = 0, 通过 Schmitt 触发器(默认) = 1, 不通过 Schmitt 触发器 |
| 7 | I/OB3OD- | I/OB3 开漏输出控制位 = 0, I/OB3 口为 I/O 口通过 IOSTB3 控制输入输出(默认) = 1, I/OB3 口作为输出时为开漏输出 |
| 12-8 | OTPBANK<4:0> | OTP bank 选择 (每个 bank 256 条指令, bank 只能按 bank0 到 bank3 的顺序选择; 每个 bank512 条指令, bank 只能按 bank0 到 bank1 的顺序选择) = 0 0 0 0 0, OTP 1K 容量(默认) = 1 1 0 0 0, OTP 选择 bank_256_0 (PC 000-0FF) = 1 1 0 0 1, OTP 选择 bank_256_1 (PC 100-1FF) = 1 1 0 1 1, OTP 选择 bank_256_2 (PC 200-2FF) = 1 1 1 1 1, OTP 选择 bank_256_3 (PC 300-3FF) = 1 0 0 0 0, OTP 选择 bank_512_0 (PC 000-1FF) = 1 0 1 0 0, OTP 选择 bnak_512_1 (PC 200-3FF) |



表 2.6: 配置选项 2

| 位 | 名称 | 说明 |
|-----|-----------------|--|
| 7-0 | 无定义 | — |
| 8 | POWER_SAVE | 455K 节能选择位 = 0, IRC 选择 455K 时降低功耗 (默认) = 1, 关闭 455K 节能模式 |
| 9 | I0_DRIVE_WEAK | I0 口弱驱动选择位 = 0, I0 口标准驱动能力 (默认) = 1, I0 口弱驱动能力 |
| 10 | I0_DRIVE_STRONG | I0 口强驱动选择位 = 0, I0 口标准驱动能力 (默认) = 1, I0 口强驱动能力 |
| 11 | ERASE_TIME_DEC | 内置 E2P 擦除时间选择位 = 0, 120ms 左右 (默认) = 1, 30ms 左右 |

表 2.7: Selection of I0B5/OSCI and I0B4/OSCO Pins

| 振荡方式 | I0B5/OSCI | I0B4/OSCO |
|----------|---------------|----------------------------------|
| IRC/ERIC | I0B5 (OSCI=0) | I0B4/OSCO selected by OSCOUT bit |
| | OSCI (OSCI=1) | I0B4/OSCO selected by OSCOUT bit |
| ERC | OSCI | I0B4/OSCO selected by OSCOUT bit |
| HF | OSCI | OSCO |
| LF | OSCI | OSCO |



8.0、指令集合

| 操作语法 | 说明 | 操作内容 | 指令周期 | 影响标志位 |
|--------------|---|---|--------------------|----------|
| BCR R, bit | Clear bit in R | 0→R | 1 | — |
| BSR R, bit | Set bit in R | 1→R | 1 | — |
| BTRSC R, bit | Test bit in R, Skip if Clear | Skip if R = 0 | 1/2 ⁽¹⁾ | — |
| BTRSS R, bit | Test bit in R, Skip if Set | Skip if R = 1 | 1/2 ⁽¹⁾ | — |
| NOP | No Operation | No operation | 1 | — |
| CLRWDT | Clear Watchdog Timer | 00h→WDT, 00h →WDT prescaler | 1 | /TO, /PD |
| OPTION | Load OPTION register | ACC→OPTION | 1 | — |
| SLEEP | Go into power-down mode | 00h→WDT, 00h→WDT prescaler | 1 | /TO, /PD |
| DAA | Adjust ACC' s data format from HEX to DEC after any addition operation | ACC(hex)→ACC(dec) | 1 | C |
| DAS | Adjust ACC' s data format from HEX to DEC after any subtraction operation | ACC(hex)→ACC(dec) | 1 | — |
| RETURN | Return from subroutine | Top of Stack→PC | 2 | — |
| RETFIE | Return from interrupt, set GIE bit | Top of Stack→PC, 1→GIE | 2 | — |
| INT | S/W interrupt | PC + 1→Top of Stack, 002h→PC | 2 | — |
| IOST R | Load IOST register | ACC→IOST register | 1 | — |
| CLRA | Clear ACC | 00h→ACC | 1 | Z |
| CLRR R | Clear R | 00h→R | 1 | Z |
| MOVAR R | Move ACC to R | ACC→R | 1 | — |
| MOVR R, d | Move R | R→dest | 1 | Z |
| DECR R, d | Decrement R | R - 1→dest | 1 | Z |
| DECRSZ R, d | Decrement R, Skip if 0 | R - 1→dest, Skip if result = 0 | 1/2 ⁽¹⁾ | — |
| INCR R, d | Increment R | R + 1→dest | 1 | Z |
| INCRSZ R, d | Increment R, Skip if 0 | R + 1→dest, Skip if result = 0 | 1/2 ⁽¹⁾ | — |
| ADDAR R, d | Add ACC and R | R + ACC→dest | 1 | C, DC, Z |
| SUBAR R, d | Subtract ACC from R | R - ACC→dest | 1 | C, DC, Z |
| ADCAR R, d | Add ACC and R with Carry | R + ACC + C→dest | 1 | C, DC, Z |
| SBCAR R, d | Subtract ACC from R with Carry | R + ACC + C→dest | 1 | C, DC, Z |
| ANDAR R, d | AND ACC with R | ACC and R→dest | 1 | Z |
| IORAR R, d | Inclusive OR ACC with R | ACC or R→dest | 1 | Z |
| XORAR R, d | Exclusive OR ACC with R | R xor ACC→dest | 1 | Z |
| COMR R, d | Complement R | R→dest | 1 | Z |
| RLR R, d | Rotate left f through Carry | R<7>→C, R<6:0>→dest<7:1>, C→dest<0> | 1 | C |
| RRR R, d | Rotate right f through Carry | C←dest<7>, R<7:1>←dest<6:0>, R<0>←C | 1 | C |
| SWAPR R, d | Swap R | R<3:0>→dest<7:4>, R<7:4>→dest<3:0> | 1 | — |
| MOVIA I | Move Immediate to ACC | I→ACC | 1 | — |
| ADDIA I | Add ACC and Immediate | I + ACC→ACC | 1 | C, DC, Z |
| SUBIA I | Subtract ACC from Immediate | I - ACC→ACC | 1 | C, DC, Z |
| ANDIA I | AND Immediate with ACC | ACC and I→ACC | 1 | Z |
| IORIA I | OR Immediate with ACC | ACC or I→ACC | 1 | Z |



| 操作语法 | 说明 | 操作内容 | 指令周期 | 影响标志位 |
|---------|--------------------------------|----------------------------------|------|-------|
| XORIA I | Exclusive OR Immediate to ACC | ACC xor I → ACC | 1 | Z |
| RETIA I | Return, place Immediate in ACC | I → ACC, Top of Stack → PC | 2 | — |
| CALL I | Call subroutine | PC + 1 → Top of Stack, I → PC | 2 | — |
| GOTO I | Unconditional branch | I → PC | 2 | — |

- 注释: 1. 两周期指令为分支跳转指令
 2. bit : Bit 地址为 8 位寄存器 R 中的某一位
 R : 寄存器地址 (00h to 3Fh)
 I : 立即数
 ACC : 累加器
 d : 目的选择:
 =0 (结果存放在 ACC)
 =1 (结果存放在 R)
 dest : 目的地
 PC : 程序指针
 PCHBUF : 高位缓冲程序指针
 WDT : 看门狗计数器
 GIE : 中断允许总控制位
 TO : 计数溢出位
 PD : 省电模式选择位
 C : 进位/借位标志
 DC : 辅助进位/借位标志. (低四位向高四位进位/借位标志)
 Z : 零标志



| | |
|---------------------|--|
| ADCAR(带进位加法) | Add ACC and R with Carry |
| 语 法 | ADCAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R + ACC + C \rightarrow dest$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 A 寄存器的內含值加上 R 寄存器的內含值（带进位），如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| | |
|---------------------|---|
| ADDAR (加法指令) | ACC and R with Carry |
| 语 法 | ADDAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R + ACC \rightarrow dest$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 A 寄存器的內含值加上 R 寄存器的內含值（不带进位），如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| | |
|--------------|---------------------------------|
| ADDIA | Add ACC and Immediate |
| 语 法 | ADDIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | $ACC + I \rightarrow ACC$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 A 寄存器的內含值加上立即数 I，结果在 ACC 中存放。 |
| 指令执行周期 | 1 |

| | |
|--------------|--|
| ANDAR | AND ACC and R |
| 语 法 | ANDAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $ACC \text{ and } R \rightarrow dest$ |
| 受影响的标志 | Z |
| 说 明 | 將 A 寄存器內含值和 R 寄存器做相与操作，如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| | |
|--------------|---------------------------------------|
| ANDIA | AND Immediate with ACC |
| 语 法 | ANDIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | $ACC \text{ and } I \rightarrow dest$ |
| 受影响的标志 | Z |
| 说 明 | 將 A 寄存器的內含值与立即数 I 做相与操作，结果在 ACC 中存放 |
| 指令执行周期 | 1 |

| | |
|------------|---|
| BCR | Clear Bit in R |
| 语 法 | BCR R, b |
| 操作数 | $0 \leq R \leq 63$ $0 \leq b \leq 7$ |
| 操作内容 | $0 \rightarrow R \langle b \rangle$ |
| 受影响的标志 | 无 |
| 说 明 | R 寄存器的位“b” 被设成 0 |
| 指令执行周期 | 1 |



| | |
|------------|---|
| BSR | Set Bit in R |
| 语 法 | BSR R, b |
| 操作数 | $0 \leq R \leq 63$ $0 \leq b \leq 7$ |
| 操作内容 | $1 \rightarrow R\langle b \rangle$ |
| 受影响的标志 | 无 |
| 说 明 | R 寄存器的位“b” 被设成 1 |
| 指令执行周期 | 1 |

| | |
|--------------|---|
| BTRSC | Test Bit in R, Skip if Clear |
| 语 法 | BTRSC R, b |
| 操作数 | $0 \leq R \leq 63$ $0 \leq b \leq 7$ |
| 操作内容 | 当 $R\langle b \rangle = 0$ 跳过下条指令 |
| 受影响的标志 | 无 |
| 说 明 | $R\langle b \rangle = 0$ 跳过下条指令 $R\langle b \rangle = 0$ 时, 该指令周期中提取的下条指令被丢弃, 并以执行 NOP 操作来替换这条 2 周期指令。 |
| 指令执行周期 | 1(2) |

| | |
|--------------|---|
| BTRSS | Test Bit in R, Skip if Set |
| 语 法 | BTRSS R, b |
| 操作数 | $0 \leq R \leq 63$ $0 \leq b \leq 7$ |
| 操作内容 | 当 $R\langle b \rangle = 1$ 跳过下条指令 |
| 受影响的标志 | 无 |
| 说 明 | $R\langle b \rangle = 1$ 跳过下条指令 $R\langle b \rangle = 1$ 时, 该指令周期中提取的下条指令被丢弃, 并以执行 NOP 操作来替换这条 2 周期指令。 |
| 指令执行周期 | 1(2) |

| | |
|-------------|---|
| CALL | Subroutine Call |
| 语 法 | CALL I |
| 操作数 | $0 \leq I \leq 1023$ |
| 操作内容 | $PC + 1 \rightarrow \text{Top of Stack}$ $I \rightarrow PC$ |
| 受影响的标志 | 无 |
| 说 明 | 子程序调用。 首先下一条指令地址(PC+1)进栈。 10 位立即地址被装载入 PC 指针的位 $\langle 9 : 0 \rangle$ 。 CALL 是二周期指令。 |
| 指令执行周期 | 2 |

| | |
|-------------|--|
| CLRA | Clear ACC |
| 语 法 | CLRA |
| 操作数 | 无 |
| 操作内容 | $00h \rightarrow ACC$ $I \rightarrow Z$ |
| 受影响的标志 | Z |
| 说 明 | ACC 被清零, Z 标志为置 1 |
| 指令执行周期 | 1 |



| | |
|--------|--------------------|
| CLRR | Clear R |
| 语 法 | CLRR R |
| 操作数 | $0 \leq R \leq 63$ |
| 操作内容 | 00h→R 1→Z |
| 受影响的标志 | Z |
| 说 明 | R 被清零, Z 标志为置 1 |
| 指令执行周期 | 1 |

| | |
|--------|---|
| CLRWDT | Clear Watchdog Timer |
| 语 法 | CLRWDT |
| 操作数 | 无 |
| 操作内容 | 00h→WDT: 00h→WDT prescaler (已经设置了 WDT 预置器): 1→T0: 1→PD |
| 受影响的标志 | T0, PD |
| 说 明 | CLRWDT 指令重置 WDT, 如已经设置了 WDT 预置器, 也重置 WDT 预置器; 并把 T0, PD 位置 1 |
| 指令执行周期 | 1 |

| | |
|--------|---|
| COMR | Complement R |
| 语 法 | COMR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | R→dest |
| 受影响的标志 | Z |
| 说 明 | 将 R 内含内容取反, 如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| | |
|--------|---|
| DAA | Adjust ACC' s data format from HEX to DEC |
| 语 法 | DAA |
| 操作数 | 无 |
| 操作内容 | ACC(hex)→ACC(dec) |
| 受影响的标志 | C |
| 说 明 | 在有些加法操作以后把 ACC 内值的十六进制转化十进制, |
| 指令执行周期 | 1 |

| | |
|--------|---|
| DAS | Adjust ACC' s data format from HEX to DEC |
| 语 法 | DAS |
| 操作数 | 无 |
| 操作内容 | ACC(hex)→ACC(dec) |
| 受影响的标志 | C |
| 说 明 | 在有些减法操作以后把 ACC 内值的十六进制转化十进制, |
| 指令执行周期 | 1 |



| DECR | Decrement R |
|--------|---|
| 语 法 | DECR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R - 1 \rightarrow \text{dest}$ |
| 受影响的标志 | Z |
| 说 明 | 递减 R 寄存器的值，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| DECRSZ | Decrement R, Skip if 0 |
|--------|---|
| 语 法 | DECRSZ R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R - 1 \rightarrow \text{dest}$ 如果结果等于 0，跳过下条指令 |
| 受影响的标志 | 无 |
| 说 明 | 递减 R 寄存器的值，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。 如果结果等于 0，该指令周期中提取的下条指令被丢弃，并以执行 NOP 操作来替换这条 2 周期指令。 |
| 指令执行周期 | 1(2) |

| GOTO | Unconditional Branch |
|--------|--|
| 语 法 | GOTO I |
| 操作数 | $0 \leq I \leq 1023$ |
| 操作内容 | $I \rightarrow \text{PC}$ |
| 受影响的标志 | 无 |
| 说 明 | 无条件跳转。10 位立即地址被装载入 PC 指针的位<9:0>。GOTO 是二周期指令。 |
| 指令执行周期 | 2 |

| INCR | Increment R |
|--------|--|
| 语 法 | INCR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R + 1 \rightarrow \text{dest}$ |
| 受影响的标志 | Z |
| 说 明 | 将被指定 R 寄存器的内含值加 1，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |



| | |
|--------|--|
| INCRSZ | Increment R, Skip if 0 |
| 语 法 | INCRSZ R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R + 1 \rightarrow \text{dest}$ 如果结果等于 0, 跳过下条指令 |
| 受影响的标志 | 无 |
| 说 明 | 将被指定 R 寄存器的内含值加 1, 如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 如果结果等于 0, 该指令周期中提取的下条指令被丢弃, 并以执行 NOP 操作来替换这条 2 周期指令。 |
| 指令执行周期 | 1(2) |

| | |
|--------|---|
| INT | S/W Interrupt |
| 语 法 | INT |
| 操作数 | 无 |
| 操作内容 | $PC + 1 \rightarrow \text{Top of Stack}$: $002h \rightarrow PC$ |
| 受影响的标志 | 无 |
| 说 明 | 子程序调用。 首先下一条指令地址(PC+1)进栈。 10 位地址 002h 被装载入 PC 指针的位<9 : 0>. CALL 是二周期指令。 |
| 指令执行周期 | 2 |

| | |
|--------|---|
| IORAR | OR ACC with R |
| 语 法 | IORAR |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $ACC \text{ or } R \rightarrow \text{dest}$ |
| 受影响的标志 | Z |
| 说 明 | 将 A 寄存器内含值和 R 寄存器做或操作, 如果 d 是 0 结果在 ACC 中存放。 如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| | |
|--------|---|
| IORIA | OR Immediate with ACC |
| 语 法 | IORIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | $ACC \text{ or } I \rightarrow \text{dest}$ |
| 受影响的标志 | Z |
| 说 明 | 将 A 寄存器的内含值与立即数 I 做相与操作, 结果在 ACC 中存放 |
| 指令执行周期 | 1 |



| | |
|--------|----------------------------------|
| IOST | Load IOST Register |
| 语 法 | IOST R |
| 操作数 | R = 5 or 6 |
| 操作内容 | ACC→IOST register R |
| 受影响的标志 | 无 |
| 说 明 | 将 A 寄存器的内含值加载到 IOST register R 中 |
| 指令执行周期 | 1 |

| | |
|--------|--------------------|
| MOVAR | Move ACC to R |
| 语 法 | MOVAR R |
| 操作数 | $0 \leq R \leq 63$ |
| 操作内容 | ACC→R |
| 受影响的标志 | 无 |
| 说 明 | 将数据从 ACC 传送到 R |
| 指令执行周期 | 1 |

| | |
|--------|-----------------------|
| MOVIA | Move Immediate to ACC |
| 语 法 | MOVIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | I→ACC |
| 受影响的标志 | 无 |
| 说 明 | 将立即值载入 A 寄存器中 |
| 指令执行周期 | 1 |

| | |
|--------|--|
| MOVR | Move Immediate to ACC |
| 语 法 | MOVR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | R→dest |
| 受影响的标志 | 无 |
| 说 明 | 将 R 寄存器内容载入 R 中, 如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。d 为 1 用来测试该寄存器对标志 Z 是否有影响 |
| 指令执行周期 | 1 |



| | |
|--------|--------------|
| NOP | No Operation |
| 语 法 | NOP |
| 操作数 | 无 |
| 操作内容 | 无操作 |
| 受影响的标志 | 无 |
| 说 明 | 不做任何操作 |
| 指令执行周期 | 1 |


| | |
|--------|----------------------|
| OPTION | Load OPTION Register |
| 语 法 | OPTION |
| 操作数 | 无 |
| 操作内容 | ACC→OPTION |
| 受影响的标志 | 无 |
| 说 明 | 将 A 寄存器内容载入 OPTION 中 |
| 指令执行周期 | 1 |


| | |
|--------|--------------------------------------|
| RETFIE | Return from Interrupt, Set 'GIE' Bit |
| 语 法 | RETFIE |
| 操作数 | 无 |
| 操作内容 | Top of Stack→PC |
| 受影响的标志 | 无 |
| 说 明 | 程序计数器载入堆栈返回地址。 GIE 位被设置到 1。 这是二周期指令。 |
| 指令执行周期 | 2 |

| | |
|--------|------------------------------------|
| RETIA | Return with Immediate in ACC |
| 语 法 | RETIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | I→ACC: Top of Stack→PC |
| 受影响的标志 | 无 |
| 说 明 | 程序计数器载入堆栈返回地址，并把立即数送入 A 中。这是二周期指令。 |
| 指令执行周期 | 2 |

| | |
|--------|------------------------|
| RETURN | Return from Subroutine |
| 语 法 | RETURN |
| 操作数 | 无 |
| 操作内容 | Top of Stack→PC |
| 受影响的标志 | 无 |
| 说 明 | 程序计数器载入堆栈返回地址。这是二周期指令。 |
| 指令执行周期 | 2 |



| | |
|------------|--|
| RLR | Rotate Left f through Carry |
| 语 法 | RLR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R \langle 7 \rangle \rightarrow C$: $R \langle 6:0 \rangle \rightarrow \text{dest} \langle 7:1 \rangle$: $C \rightarrow \text{dest} \langle 0 \rangle$ |
| 受影响的标志 | C |
| 说 明 | R 寄存器的内含值又移 1bit，右移时包含 C(进位标志)，如下图，结果存放由 d 决定，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。  |
| 指令执行周期 | 1 |

| | |
|------------|--|
| RRR | Rotate Right f through Carry |
| 语 法 | RRR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $C \rightarrow \text{dest} \langle 7 \rangle$: $R \langle 7:1 \rangle \rightarrow \text{dest} \langle 6:0 \rangle$: $R \langle 0 \rangle \rightarrow C$ |
| 受影响的标志 | C |
| 说 明 | R 寄存器的内含值又移 1bit，右移时包含 C(进位标志)，如下图，结果存放由 d 决定，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。  |
| 指令执行周期 | 1 |

| | |
|--------------|--|
| SLEEP | SLEEP |
| 语 法 | SLEEP |
| 操作数 | 无 |
| 操作内容 | $00h \rightarrow \text{WDT}$: $00h \rightarrow \text{WDT prescaler}$: $1 \rightarrow \text{T0}$: $0 \rightarrow \text{PD}$ |
| 受影响的标志 | T0, PD |
| 说 明 | T0 位置 1。PD 位清零，WDT 和 WDT 预置器清零 单片机进入睡眠模式 |
| 指令执行周期 | 1 |



| SBCAR (带借位加法) | Subtract ACC from R with Carry |
|---------------|--|
| 语 法 | SBCAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $(R - ACC - C) \rightarrow dest$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 R 寄存器的內含值減去 A 寄存器的內含值 (帶借位), 如果 d 是 0 結果在 ACC 中存放。 如果 d 是 1 結果在 R 中存放。 |
| 指令执行周期 | 1 |

| SUBAR | Subtract ACC from R |
|--------|---|
| 语 法 | SUBAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R - ACC \rightarrow dest$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 R 寄存器的內含值減去 A 寄存器的內含值 (不帶借位), 如果 d 是 0 結果在 ACC 中存放。 如果 d 是 1 結果在 R 中存放。 |
| 指令执行周期 | 1 |

| SUBIA | Subtract ACC from Immediate |
|--------|----------------------------------|
| 语 法 | SUBIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | $ACC - I \rightarrow ACC$ |
| 受影响的标志 | C, DC, Z |
| 说 明 | 將 A 寄存器的內含值減去立即數 I, 結果在 ACC 中存放。 |
| 指令执行周期 | 1 |

| SWAPR | Swap nibbles in R |
|--------|---|
| 语 法 | SWAPR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | $R<3:0> \rightarrow dest<7:4>$ $R<7:4> \rightarrow dest<3:0>$ |
| 受影响的标志 | 无 |
| 说 明 | 將所選定的寄存器, 高 4 位以及低 4 位, 如果 d 是 0 結果在 ACC 中存放。 如果 d 是 1 結果在 R 中存放。 |
| 指令执行周期 | 1 |



| XORAR | Exclusive OR ACC with R |
|--------|--|
| 语 法 | XORAR R, d |
| 操作数 | $0 \leq R \leq 63$ $d \in [0, 1]$ |
| 操作内容 | ACC xor R \rightarrow dest R |
| 受影响的标志 | Z |
| 说 明 | 将 A 寄存器的值和 R 寄存器的值异或在一起，如果 d 是 0 结果在 ACC 中存放。如果 d 是 1 结果在 R 中存放。 |
| 指令执行周期 | 1 |

| XORIA | Exclusive OR Immediate with ACC |
|--------|------------------------------------|
| 语 法 | XORIA I |
| 操作数 | $0 \leq I \leq 255$ |
| 操作内容 | ACC xor I \rightarrow ACC |
| 受影响的标志 | Z |
| 说 明 | 将 A 寄存器的值和立即数 I 异或在一起，结果在 ACC 中存放。 |
| 指令执行周期 | 1 |



9.0、绝对最大额定值

操作环境温度: 0℃到+70℃

存储器额定温度: -65℃到+150℃

DC 电源电压(Vdd): 0V到+6.0V

输入电压(对地电压 (Vss)): -0.3V到(Vdd + 0.3)V

10.0、操作条件

DC 供电电压: +2.0V到+5.5V

操作温度: 0℃到+70℃

*细节详见 11.1



11.0、电气特性

11.1、QG701 电气特性

电气特性是在四时钟指令周期和 WDT & LVDT 禁用情况下

Ta=25°C

| Sym | Description | Conditions | Min. | Typ. | Max. | Unit | |
|--------------------------|-------------------------|-------------------------------|-------|------|------|------|---|
| VDD | Supply voltage | 0Hz~1MHz | | 2.0 | 5.5 | V | |
| | | 1MHz~4MHz | | 2.0 | 5.5 | | |
| | | 4MHz~8MHz | | 2.4 | 5.5 | | |
| | | 8MHz~10MHz | | 2.6 | 5.5 | | |
| | | 10MHz~16MHz | | 3.0 | 5.5 | | |
| | | 16MHz~20MHz | | 3.5 | 5.5 | | |
| TPWR | Power rising time | Vdd=0V to Vdd | 0.8 | | 2.6 | ms/V | |
| FHF | X'tal oscillation range | HF mode, Vdd=5V | 4 | | 20 | MHz | |
| | | HF mode, Vdd=3V | 4 | | 16 | | |
| FXT | X'tal oscillation range | XT mode, Vdd=5V | 0.455 | | 20 | MHz | |
| | | XT mode, Vdd=3V | 0.455 | | 8 | | |
| FLF | X'tal oscillation range | LF mode, Vdd=5V | 32 | | 455 | MHz | |
| | | LF mode, Vdd=3V | 32 | | 455 | | |
| FERC | RC oscillation range | ERC mode, Vdd=5V | DC | | 16 | MHz | |
| | | ERC mode, Vdd=3V | DC | | 13 | | |
| FIRC/ERIC | RC oscillation range | ERIC mode, external R, Vdd=5V | DC | | 16 | MHz | |
| | | ERIC mode, external R, Vdd=3V | DC | | 16 | | |
| | | IRC mode, internal R, Vdd=5V | 0.455 | | 8 | | |
| | | IRC mode, internal R, Vdd=3V | 0.455 | | 8 | | |
| VIH | Input high voltage | With Schmitt-trigger | | | | | V |
| | | I/O ports, Vdd=5V | 2.2 | | VDD | | |
| | | RSTB, TOCKI pins, Vdd=5V | 2.2 | | VDD | | |
| | | I/O ports, Vdd=3V | 1.7 | | VDD | | |
| | | RSTB, TOCKI pins, Vdd=3V | 1.7 | | VDD | | |
| | | Without Schmitt-trigger | | | | | |
| | | I/O ports, Vdd=5V | 2.0 | | VDD | | |
| | | RSTB, TOCKI pins, Vdd=5V | 2.0 | | VDD | | |
| VIL | Input low voltage | With Schmitt-trigger | | | | | V |
| | | I/O ports, Vdd=5V | VSS | | 0.8 | | |
| | | RSTB, TOCKI pins, Vdd=5 | VSS | | 0.8 | | |
| | | I/O ports, Vdd=3V | VSS | | 0.5 | | |
| | | RSTB, TOCKI pins, Vdd=3V | VSS | | 0.5 | | |
| | | Without Schmitt-trigger | | | | | |
| | | I/O ports, Vdd=5V | VSS | | 0.8 | | |
| | | RSTB, TOCKI pins, Vdd=5V | VSS | | 0.8 | | |
| I/O ports, Vdd=3V | VSS | | 0.6 | | | | |
| RSTB, TOCKI pins, Vdd=3V | VSS | | 0.6 | | | | |



| Sym | Description | Conditions | Min. | Typ. | Max. | Unit |
|-------|---------------------|---|------|------|------|------|
| VOH | Output high voltage | I _{OH} =-5.4mA, V _{dd} =5V | 3.6 | | | V |
| VOL | Output low voltage | I _{OL} =8.7mA, V _{dd} =5V | | | 0.6 | V |
| IPH | Pull-high current | Input pin at V _{ss} , V _{dd} =5V | -50 | -66 | -80 | uA |
| IPL | Pull-down current | Input pin at V _{dd} , V _{dd} =5V | 30 | 44 | 60 | uA |
| IWDT | WDT current (18mS) | V _{dd} =5V | | 6 | | uA |
| | | V _{dd} =3V | | 1.5 | | |
| TWDT | WDT period (18mS) | V _{dd} =3V | | 20.2 | | mS |
| | | V _{dd} =4V | | 17.5 | | |
| | | V _{dd} =5V | | 15.9 | | |
| ILVDT | LVDT current | V _{dd} =5V, LVDT=3.6V | | 1.89 | | uA |
| | | V _{dd} =5V, LVDT=2.4V | | 2.4 | | |
| | | V _{dd} =3V, LVDT=2.4V | | 1 | | |
| ISB | Power down current | Sleep mode, V _{dd} =5V, WDT LVDT disable | | 1 | 2 | uA |
| | | Sleep mode, V _{dd} =3V, WDT LVDT disable | | 0.5 | 1 | |
| IDD | Operating current | HF mode, V _{dd} =5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | mA |
| | | 20MHz | | 3.03 | | |
| | | 16MHz | | 2.61 | | |
| | | HF mode, V _{dd} =3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | |
| | | 20MHz | | 1.27 | | |
| IDD | Operating current | HF mode, V _{dd} =5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | mA |
| | | 20MHz | | 4.36 | | |
| | | 16MHz | | 3.41 | | |
| | | HF mode, V _{dd} =3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | |
| | | 20MHz | | 1.94 | | |
| IDD | Operating current | XT mode, V _{dd} =5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | mA |
| | | 20MHz | | 2.13 | | |
| | | 16MHz | | 1.77 | | |
| | | XT mode, V _{dd} =5V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | |
| | | 455KHz | | 0.42 | | |
| IDD | Operating current | XT mode, V _{dd} =5V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | mA |
| | | 32KHz | | — | | |
| | | XT mode, V _{dd} =3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | |
| | | 16MHz | | 0.76 | | |
| | | 8MHz | | 0.43 | | |
| IDD | Operating current | XT mode, V _{dd} =3V, 4 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | mA |
| | | 455KHz | | 0.16 | | |
| | | 32KHz | | 0.10 | | |
| | | XT mode, V _{dd} =5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | |
| | | 20MHz | | 3.12 | | |
| IDD | Operating current | XT mode, V _{dd} =5V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | mA |
| | | 16MHz | | 2.57 | | |
| | | XT mode, V _{dd} =5V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | |
| | | 455KHz | | 0.44 | | |
| | | 32KHz | | — | | |



| Sym | Description | Conditions | | | Min. | Typ. | Max. | Unit |
|-----------|-------------------|--|-------------|-----------|------|-------|------|------|
| IDD | Operating current | XT mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | | | mA |
| | | 16MHz | | | | — | | |
| | | 8MHz | | | | 0.62 | | |
| | | XT mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 100pF | | | | | | |
| | | 455KHz | | | | 0.44 | | |
| IDD | Operating current | 32KHz | | | | — | | uA |
| | | LF mode, Vdd=5V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | | | |
| | | 455KHz | | | | 106.3 | | |
| | | 32KHz | | | | 42.5 | | |
| | | LF mode, Vdd=3V, 4 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | | | |
| IDD | Operating current | 455KHz | | | | 38.7 | | uA |
| | | 32KHz | | | | 12.1 | | |
| | | LF mode, Vdd=5V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | | | |
| | | 455KHz | | | | 130.9 | | |
| | | 32KHz | | | | 44.8 | | |
| IDD | Operating current | LF mode, Vdd=3V, 2 clock instruction, OSCI / OSCO = 20pF / 20pF | | | | | | uA |
| | | 455KHz | | | | 50.4 | | |
| | | 32KHz | | | | 13 | | |
| | | ERC mode, Vdd=5V, 4 clock instruction | | | | | | |
| | | C=3P | R=10Kohm | F=7.4MHz | | 1.2 | | |
| R=3.3Kohm | F=16.4MHz | | | 2.87 | | | | |
| IDD | Operating current | ERC mode, Vdd=3V, 4 clock instruction | | | | | | mA |
| | | C=3P | R=10Kohm | F=7MHz | | 0.61 | | |
| | | | R=3.3Kohm | F=13.6MHz | | 1.41 | | |
| | | ERC mode, Vdd=5V, 2 clock instruction | | | | | | |
| | | C=3P | R=10Kohm | F=7.6MHz | | 1.87 | | |
| R=3.3Kohm | F=15.1MHz | | | 3.94 | | | | |
| IDD | Operating current | ERC mode, Vdd=3V, 2 clock instruction | | | | | | mA |
| | | C=3P | R=10Kohm | F=6.7MHz | | 0.87 | | |
| | | | R=3.3Kohm | F=13.7MHz | | 2 | | |
| | | ERIC mode, external R, Vdd=5V, 4 clock instruction | | | | | | |
| | | F=8MHz | R=103.2Kohm | | 1.11 | | | |
| F=4MHz | R=206.2Kohm | | 0.58 | | | | | |
| IDD | Operating current | ERIC mode, external R, Vdd=3V, 4 clock instruction | | | | | | mA |
| | | F=8MHz | R=99.4Kohm | | 0.56 | | | |
| | | F=4MHz | R=187.6Kohm | | 0.29 | | | |
| | | ERIC mode, external R, Vdd=5V, 2 clock instruction | | | | | | |
| | | F=8MHz | R=103.2Kohm | | 1.78 | | | |
| F=4MHz | R=206.2Kohm | | 0.9 | | | | | |
| IDD | Operating current | ERIC mode, external R, Vdd=3V, 2 clock instruction | | | | | | mA |
| | | F=8MHz | R=99.4Kohm | | 0.89 | | | |
| | | F=4MHz | R=187.6Kohm | | 0.46 | | | |

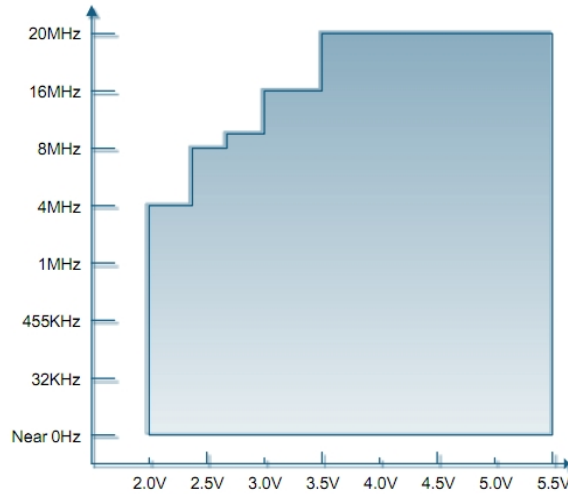


| Sym | Description | Conditions | Min. | Typ. | Max. | Unit | |
|-----|----------------------|---|------|------|------|------|----|
| IDD | Operating current | IRC mode, internal R, Vdd=5V, 4 clock instruction | | | | | mA |
| | | F=8MHz | | 1.77 | | | |
| | | F=4MHz | | 0.93 | | | |
| | | F=1MHz | | 0.32 | | | |
| | | F=455KHz | | 0.21 | | | |
| | | IRC mode, internal R, Vdd=3V, 4 clock instruction | | | | | |
| | | F=8MHz | | 0.97 | | | |
| | | F=4MHz | | 0.55 | | | |
| | | F=1MH | | 0.2 | | | |
| | | F=455KHz | | 0.14 | | | |
| IDD | Operating current | IRC mode, internal R, Vdd=5V, 2 clock instruction | | | | | mA |
| | | F=8MHz | | 1.75 | | | |
| | | F=4MHz | | 0.95 | | | |
| | | F=1MHz | | 0.33 | | | |
| | | F=455KHz | | 0.22 | | | |
| | | IRC mode, internal R, Vdd=3V, 2 clock instruction | | | | | |
| | | F=8MHz | | 0.99 | | | |
| | | F=4MHz | | 0.55 | | | |
| | | F=1MHz | | 0.21 | | | |
| | | F=455KHz | | 0.15 | | | |
| IER | EEPROM erase current | Vdd=5V | | 10 | | mA | |
| IPG | EEPROM prog current | Vdd=5V | | 5 | | mA | |

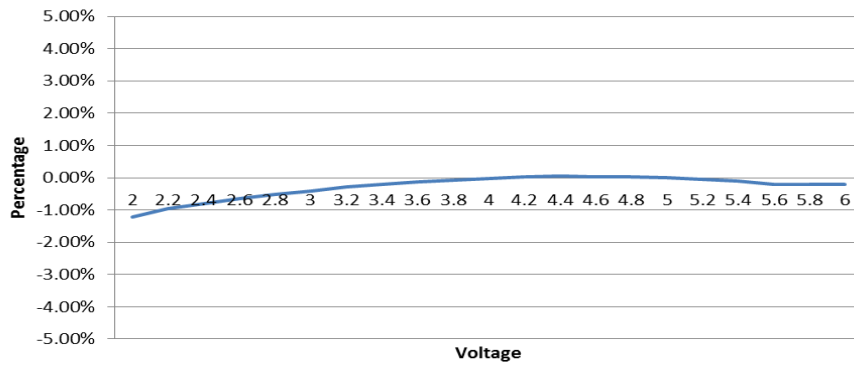


11.2、QG701 电气特性表

11.2.1、操作频率 vs 操作电压 (Ta=25°C)

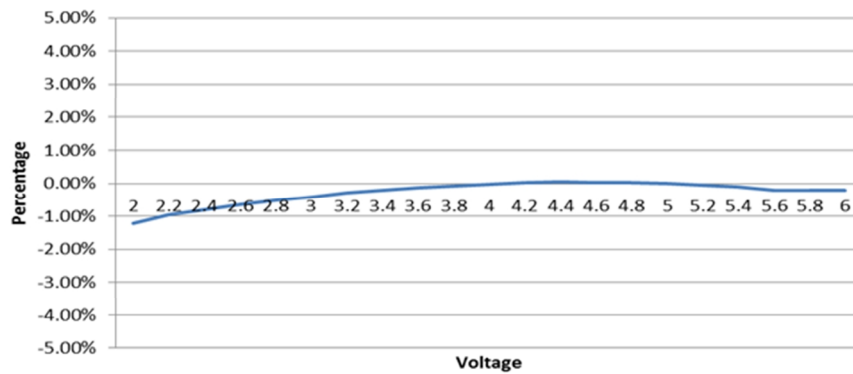


11.2.2、内部 4MHz RC vs 供应电压 (Ta=25°C)



注: 曲线仅供设计参考

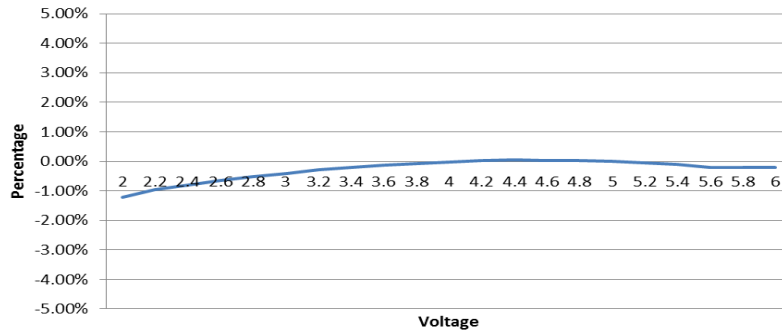
11.2.3、内部 8MHz RC vs 供应电压 (Ta=25°C)



注: 曲线仅供设计参考

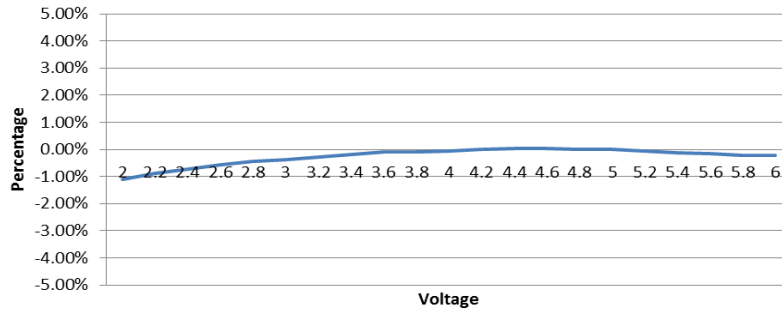


11.2.4、内部 1MHz RC vs 供应电压 (Ta=25°C)



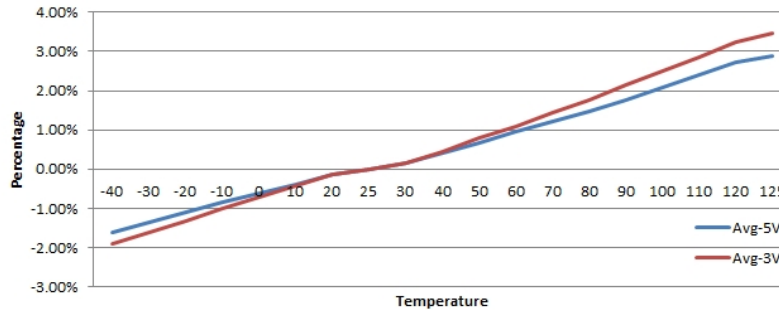
注: 曲线仅供设计参考

11.2.5、内部 455KHz RC vs 供应电压 (Ta=25°C)



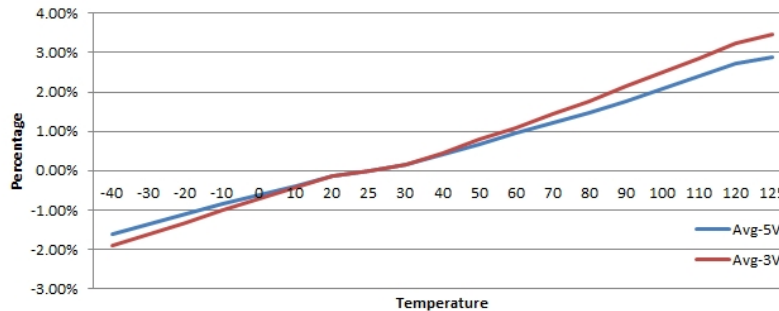
注: 曲线仅供设计参考

11.2.6、内部 4MHz RC vs 温度



注: 曲线仅供设计参考

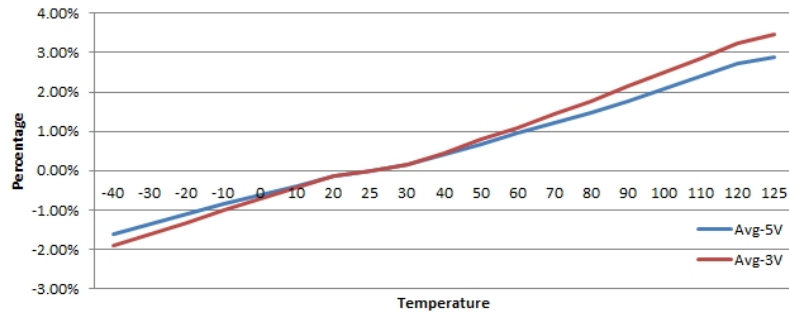
11.2.7、内部 8MHz RC vs 温度



注: 曲线仅供设计参考

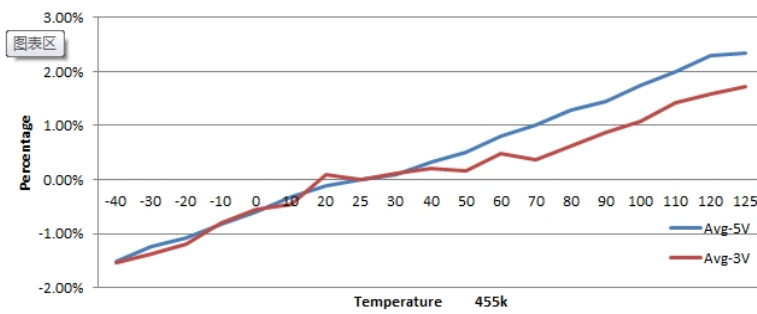


11.2.8、内部 1MHz RC vs 温度



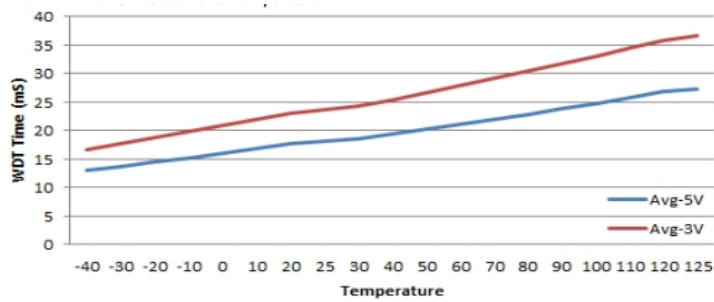
注: 曲线仅供设计参考

11.2.9、内部 455KH Hz RC vs 温度



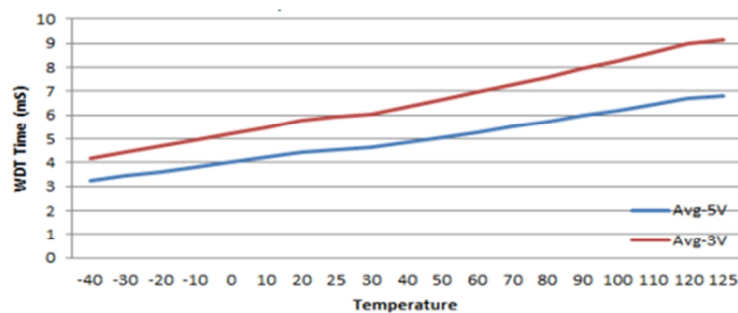
注: 曲线仅供设计参考

11.2.10、WTD18 毫秒复位时间 vs 温度



注: 曲线仅供设计参考

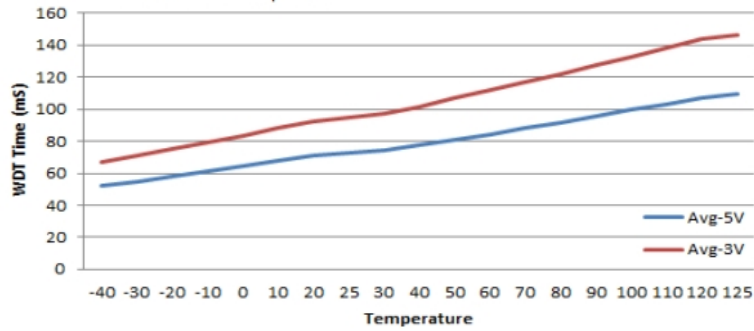
11.2.11、WTD4.5 毫秒复位时间 vs 温度



注: 曲线仅供设计参考

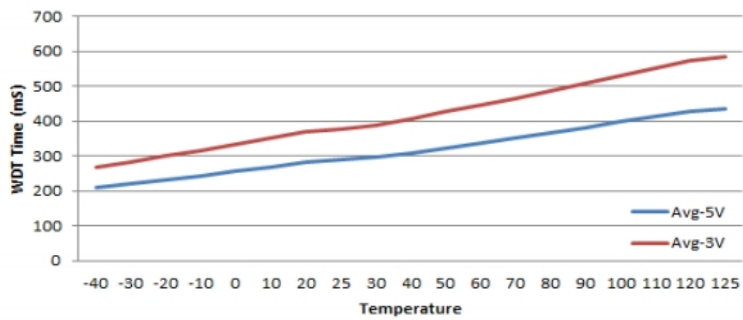


11.2.12、WTD72 毫秒复位时间 vs 温度



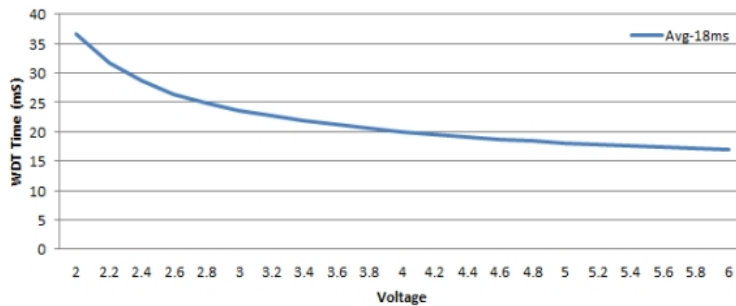
注: 曲线仅供设计参考

11.2.13、WTD288 毫秒复位时间 vs 温度



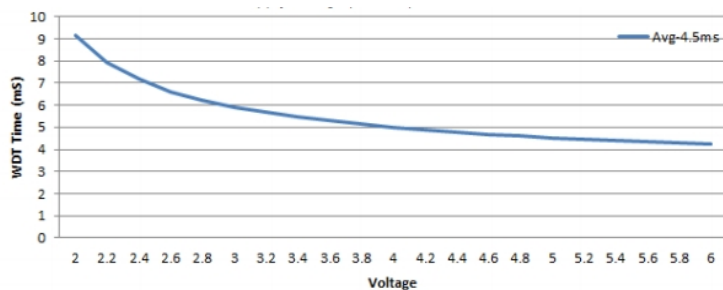
注: 曲线仅供设计参考

11.2.14、WTD18 毫秒复位时间 vs 供应电压 (Ta=25°C)



注: 曲线仅供设计参考

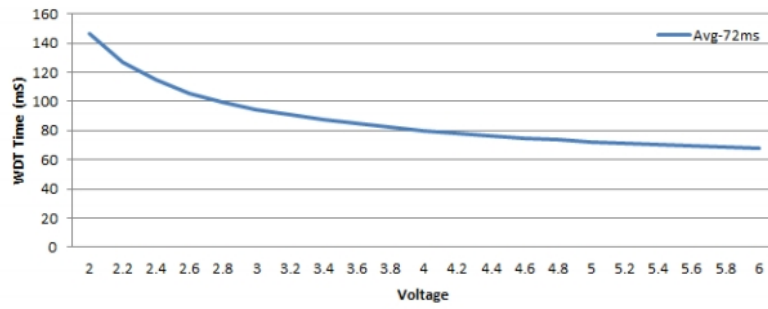
11.2.15、WTD4.5 毫秒复位时间 vs 供应电压 (Ta=25°C)



注: 曲线仅供设计参考

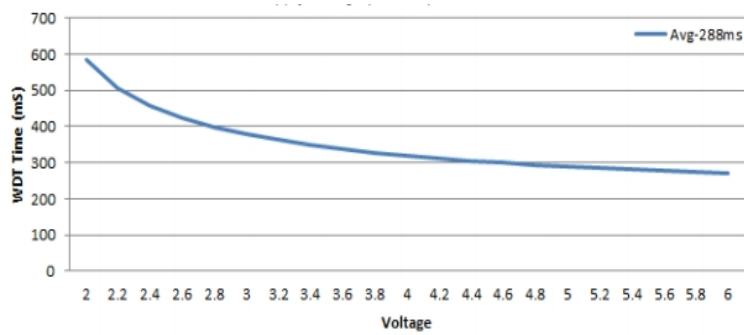


11.2.16、WTD72 毫秒复位时间 vs 供应电压 (Ta=25°C)



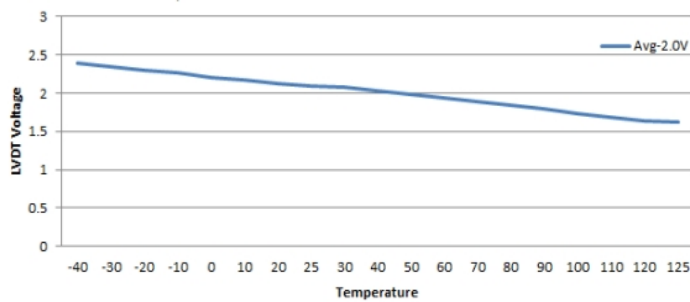
注: 曲线仅供设计参考

11.2.17、WTD288 毫秒复位时间 vs 供应电压 (Ta=25°C)



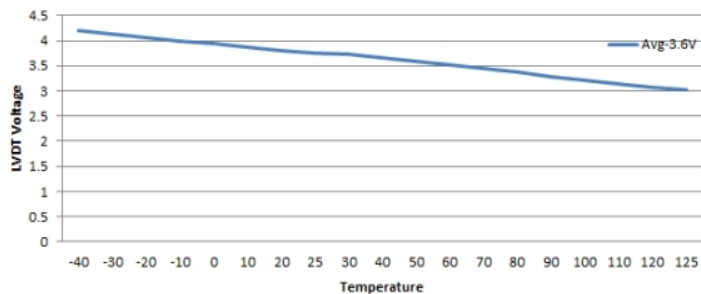
注: 曲线仅供设计参考

11.2.18、LVDT2.0V vs 温度



注: 曲线仅供设计参考

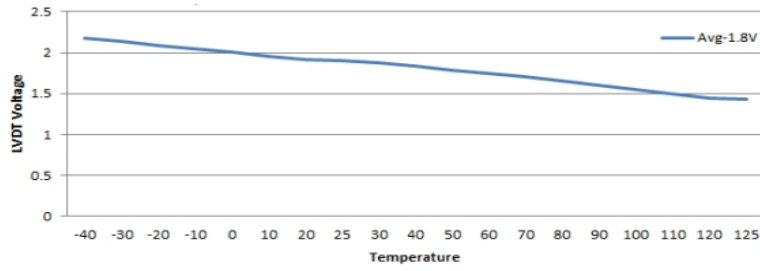
11.2.19、LVDT3.6V vs 温度



注: 曲线仅供设计参考

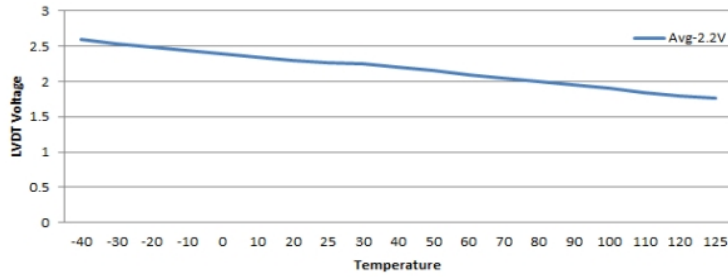


11.2.20、LVDT1.8V vs 温度



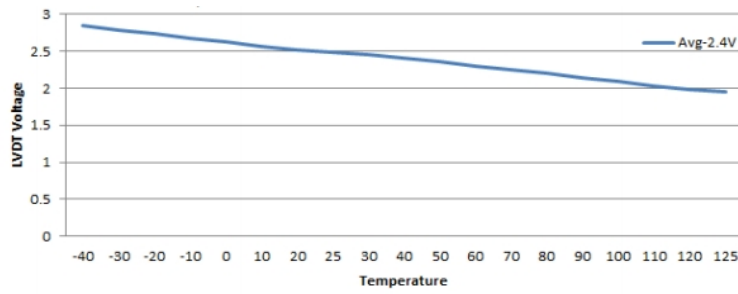
注: 曲线仅供设计参考

11.2.21、LVDT2.2V vs 温度



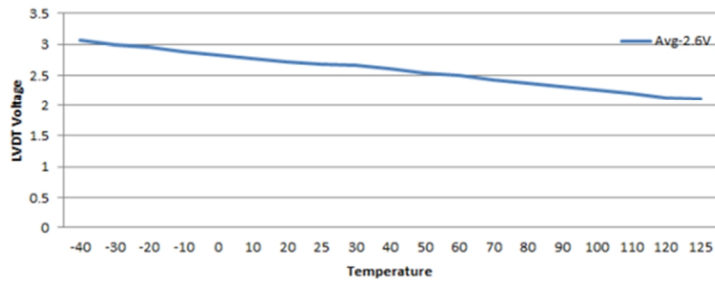
注: 曲线仅供设计参考

11.2.22、LVDT2.4V vs 温度



注: 曲线仅供设计参考

11.2.23、LVDT2.6V vs 温度

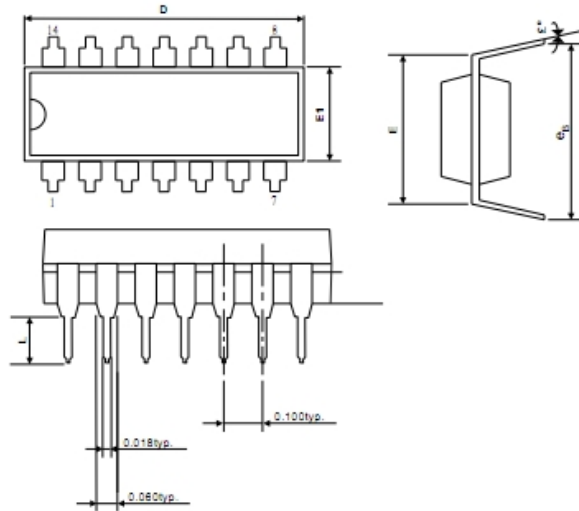


注: 曲线仅供设计参考



12.0、封装尺寸

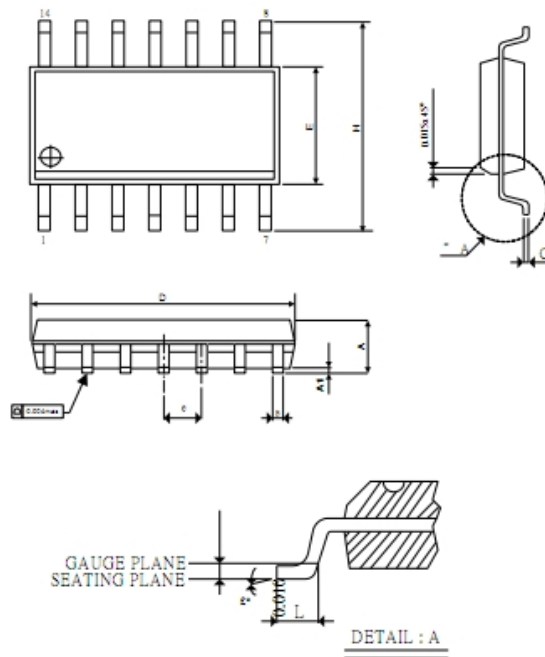
12.1、14-PIN PDIP 300mil



| Symbol s | Dimension In Inches | | |
|----------|---------------------|-------|-------|
| | Min | Nom | Max |
| A | — | — | 0.210 |
| A1 | 0.015 | — | — |
| A2 | 0.125 | 0.130 | 0.135 |
| D | 0.735 | 0.750 | 0.775 |
| E | 0.300 BSC. | | |
| E1 | 0.245 | 0.250 | 0.255 |
| L | 0.115 | 0.130 | 0.150 |
| eB | 0.335 | 0.355 | 0.375 |
| ⊖ ° | 0° | 7° | 15° |



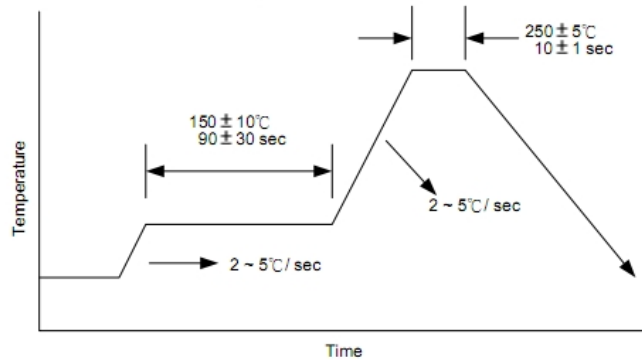
12.2、14-PIN SOP 150mil



| Symbol s | Dimension In Inches | | |
|-----------------|---------------------|-------|--------|
| | Min | Nom | Max |
| A | 0.058 | 0.064 | 0.068 |
| A1 | 0.004 | — | 0.010 |
| B | 0.013 | 0.016 | 0.020 |
| C | 0.0075 | 0.008 | 0.0098 |
| D | 0.336 | 0.341 | 0.344 |
| E | 0.150 | 0.154 | 0.157 |
| e | — | 0.050 | — |
| H | 0.228 | 0.236 | 0.244 |
| L | 0.015 | 0.025 | 0.050 |
| \ominus° | 0° | — | 8° |



13.0、封装 IR 回流焊接曲线



14.0、订购信息

| OTP Type MCU | Package Type | Pin Count | Package Size |
|--------------|--------------|-----------|--------------|
| QG701 | PDIP | 14 | 300 mil |
| QG701 | SOP | 14 | 150 mil |