

# M68HC08 中断及复位系统

## 一.中断概念

计算机系统的“中断”是指中央处理器 CPU 正在处理某件事情的时候，发生了异常事件(如定时器溢出等)产生一个中断请求信号，请求 CPU 迅速去处理。CPU 暂时中断当前的工作，转入处理所发生的事件，处理完以后，再回到原来被中断的地方继续原来的工作，这样的过程称为中断，实现这种功能的部件称为中断系统，产生中断的部件或设备称为中断源。

一个计算机系统一般有多个中断请求源。当多个中断源同时向 CPU 请求中断时，就存在 CPU 优先响应哪一个中断请求源的问题。一般根据中断源(所发生的事件)的轻重缓急，规定中断源的优先级，CPU 优先响应中断优先级高的中断源请求。

当 CPU 正在处理一个中断请求时，又发生了另外的中断请求，如果 CPU 能暂时中止对原中断的处理，转去处理优先级更高的中断请求，待处理完以后，再继续处理原来的中断事件，这样的过程称为中断嵌套。

这样的中断系统称为多级中断系统。而没有中断嵌套功能的系统称为单级中断系统。

## 二.M68HC08 中断系统

M68HC08 的中断源数量也是随产品型号不同而不同的。一般随功能模块的增加而增加，中断源的数量反映了一个微控制器中断处理功能的强弱。

### 1、GP32 的中断源

GP32 中断系统有 24 个中断源，每个中断源都有一个中断标志位、中断允许位、17 个中断向量，表 2-4 列出了 GP32 的中断源。

表 1 MC68HC908GP32 中断源

中断源	标志	屏蔽	INT 寄存器标志	优先级	向量地址
执行 SWI 软件中断	None	None	None	0	\$FFFC-\$FFFD
外部中断 IRQ	IRQF	IMASK1	IF1	1	\$FFFA-\$FFFB
CGM 中断	PLLIF	PLLIE	IF2	2	\$FFF8-\$FFF9
TIM1 通道 0	CHOF	CH0IE	IF3	3	\$FFF6-\$FFF7
TIM1 通道 1	CHIF	CH1IE	IF4	4	\$FFF4-\$FFF5
TIM1 溢出	TOF	TOIE	IF5	5	\$FFF2-\$FFF3
TIM2 通道 1	CHOF	CH0IE	IF6	6	\$FFF0-\$FFF1
TIM2 溢出	CHIF	CH1IE	IF7	7	\$FFE6-\$FFE7
SPI 接收器满	TOF	TOIE	IF8	8	\$FFFC-\$FFED
SPI 溢出	SPRF	SPRIE	IF9	9	\$FFFA-\$FFFB
SPI 方式错	OVRF	ERRIE			
SPI 发送器空	MODF	ERRIE			
SCI 接收器溢出	SPTF	SPTIE	IF10	10	\$FFF8-\$FFF9
SCI 噪声溢出	OR	ORIE	IF11	11	\$FFE6-\$FFE7
SCI 噪声标志	NF	NEIE			
SCI 格式错	FE	FEIE			
SCI 奇偶错	PE	PEIE			
SCI 接收器满	SCRF	SCRIE	IF12	12	\$FFE4-\$FFE5
SCI 输入空闲	IDLE	ILIE			
SCI 发送器空	SCTE	SCTIE	IF13	13	\$FFE2-\$FFE3
SCI 发送完成	TC	TCIE			

键盘输入中断	KEYF	IMASKK	IF14	14	\$FFE0-\$FFE1
ADC 转换完成	COCO	AIEN	IF15	15	\$FFDE-\$FFDF
时基中断	TBIF	TBIE	IF16	16	\$FFDC-\$FFDD

## 2、中断响应过程

CPU 每执行完一条指令，如果条件码寄存器 CCR 中的中断屏蔽位 I=0，按中断优先级次序查询所有的中断标志位，如果查到允许的中断请求源标志为“1”，则响应该中断请求。中断过程如下：

- ①CPU 寄存器 PCL、PCH、X、A、CCR 依次进栈；
- ②置位中断屏蔽位 I(关中)；
- ③从所响应的中断请求源相对应的中断向量地址中取出中断向量(即中断服务程序入口地址)送入 PC；
- ④CPU 从中断入口地址开始执行中断服务程序，直至碰到近回指令 RTI 为止，RTI 指令从堆栈中依次弹出 CLR、A、X、PCH、PCL，使 CPU 回到原来被中断地方继续执行原来的程序；
- ⑤CPU 响应中断执行中断服务程序时，I=1，因此不能响应其他中断请求。如果需要，在执行中断服务程序过程中执行清零 I(CLI)指令，就可以响应其他中断请求以实现中断嵌套。

## 3、中断标志位和中断屏蔽位

HC08 的中断源标志位和屏蔽位分布在各个功能模块的控制状态寄存器中。对于 TIM1、TIM2 中断、SCI 中断、SPI 中断的标志位和屏蔽位，在以后章节中作详细介绍。本章仅对外部中断和键盘中断进一步说明。

### 三. 外部中断 IRQ

外部中断模块产生的中断请求信号是可以屏蔽的中断请求信号。

IRQ 状态控制寄存器 INTSCR(\$1D)

	7	6	5	4	3	2	1	0	
R	—	—	—	—	IRQF	0	IMASK	MODE	复位后 内容为\$0
W					—	ACK			

- MODE: IRQ 触发方式选择位：

MODE=1:  $\overline{IRQ}$  输入负跳变或低电平时产生中断(1—IRQF)；

MODE=0:  $\overline{IRQ}$  输入仅负跳变时产生中断(1—IRQF)。

- IMASK: 外部  $\overline{IRQ}$  中断屏蔽位：

IMASK=1: 禁止 IRQ 中断；

IMASK=0: 允许 IRQ 中断。

- IRQF: IRQ 中断标志位：

IRQF=1 正在向 CPU 请求中断；

IRQF=0 未向 CPU 请求中断。

- ACK: 外部中断请求响应位：

负跳变触发方式时(MODE=0)当 IRQF=1 软件置“1”ACK 时清零 IRQF；

负跳变及低电平触发方式时(MODE=1)当 IRQF=1，并且  $\overline{IRQ}$  引脚为高电平时，软件置位 ACK 才清零 IRQF。

在外部中断服务程序中必须有置“1” ACK 的指令，用以清零中断标志 IRQF。

#### 四. 键盘中断模块 KBI

键盘中断提供多个可屏蔽的外部中断。GP32 的 PTA0—PTA7 既可以作为通用的双向 I/O 口使用，也可以作为键盘输入线(或附加外部中断输入线)按键时产生键盘中断。键盘中断可以唤醒 CPU 退出节电方式(WAIT 或 STOP)回到正常的运行状态,对键盘输入信息进行处理。

##### 1、键盘中断状态和控制寄存器 INTKBSCR(\$1A)

	7	6	5	4	3	2	1	0	
R	—	—	—	—	KEYF	0	IMASK	MODE	复位后 内容为\$0
W					—	ACKK			

• MODEK: 键盘中断触发方式位;

MODEK=1: 键输入线发生负跳变或为低电平时产生中断请求(1—KEYF);

MODEK=0 时: 仅当键输入线发生负跳变时产生中断, 请求(1—KEYF);

• KEYF: 键盘中断标志位;

KEYF=1 键盘正在请求中断;

KEYF=0 无键盘中断请求。

• IMASKK: 键盘中断屏蔽位;

IMASKK=1 禁止键盘发中断请求;

IMASKK=0 允许键盘发中断请求。

• ACKK: 键盘中断响应位;

MODEK=1: 当 KEYF=1, 键输入线都为高电平时, 置“1” ACKK 同时清零 KEYF, MODEK=0 时; 当 KEYF=1, 软件置“1” ACKK 同时清零 KEYF 标志。

##### 2、键盘中断使能寄存器 TKBIER(\$1B)

	7	6	5	4	3	2	1	0
	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0

• KBIEX(X=7-0)=1 时, PTAX(X=7-0)作为键输入线, 使 PTAX 内部具有拉高电阻。当输入有效时(负跳变或低电平时, 1—KEYF, 若 IMASK=0, 则向 CPU 请求中断。

KBIEX(X=7-0)=0 时, PTAX 作为普通 I/O 线, 不产生中断请求。

##### 3、键盘中断模块初始化

为了防止产生不正确的键盘中断, 按下面步骤初始化:

①1→IMASKK; ②1→KBIEX; ③1→ACKK; ④0→IMASKK

### 五.复位功能

复位使单片机从某些不确定状态或混乱状态回到一个确定的初始状态, 并从这个初始状态开始工作。

• MCU 立即停止正在执行的操作;

• MCU 内各种控制状态寄存器置为确定的初始值;

• 选 CGMXCLK(晶体振荡器输出时钟)÷4 作为部总线时钟;

• 从\$FFFE 和\$FFFF 单元取出复位向量地址送程序计数器 PC;

• MCU 从用户定义的复位入口执行主程序。

#### 1.复位信号源

##### 1)外部复位信号

HC08 的  $\overline{RST}$  引脚为外部复位输入端, 当  $\overline{RST}$  输入低电平持续时间大于  $T_{IRL}$ (如

$f_{BUS}=8\text{MHz}$  时为  $125\ \mu\text{s}$ ) 时使 MCU 复位。 $\overline{RST}$  同时又是内部复位输出端, 当产生内部复位时  $\overline{RST}$  输出一个负脉冲。 $\overline{RST}$  内有拉高电阻, 因此一般常态为高电平。

### 2) 内部复位信号

①上电复位: 当主电源输入端  $V_{DD}$  发生正跳变时, MCU 内部产生一个上电复位信号, 使系统复位。

②低电压复位: 当主电源输入端  $V_{DD}$  发生正跳变时, MCU 内部产生一个上电复位信号, 使系统复位。

③非法地址和非法码复位: 当 CPU 访问非法的地址单元(无物理单元的存储空间保留区)或取出非法操作码时产生复位信号, 使系统复位。

④COP 复位: 当“CPU 正常工作监视器”(通常也称 Watchdog)的“COP 计数器”计数溢出时产生内部复位信号, 使系统复位。

### 3) 复位状态寄存器 RSR(\$FE01)

RSR 为系统集成模块 SIM 中的一个状态寄存, 记录发生复位操作的原因。软件根据 RSR 的状态分析复位原因, 对系统作相应的初始化处理。RSR 寄存器格式如下:

7	6	5	4	3	2	1	0
POR	PIN	COP	ILOP	ILAD	—	LVI	—

- POR: 上电复位标志;
- PIN: 外部  $\overline{RST}$  复位标志;
- COP: 正常工作监视器复位标志;
- ILOP: 非法操作码复位标志;
- ILAD: 非法地址复位标志;
- LVI: 低电压复位标志。

## 六. CPU 正常工作监视器 COP

CPU 正常工作监视器 COP 通常称为“Watchdog”。COP 内部有一个自由运行的计数器, 若计数溢出便产生 COP 复位信号, 使系统复位。系统在正常工作时, CPU 运行的用户程序应定时清“0”COP 计数器(如: 对 \$FFFF 单元写任意一个数则清“0”COP 计数器)使 COP 计数器不会产生溢出复位信号。当系统受干扰, 程序执行不正常时, COP 计数器得不到定时清零而产生 COP 复位信号, 使系统重新启动。这样便实现了 CPU 正常工作的监视功能。结构寄存器(\$1FH)CONFIG1.0 为“COP 禁止”位 COPD。当 COPD=1 时, 禁止 COP 复位。COPD=0 允许 COP 产生复位。系统复位以后 COPD 初态为 0, 即允许 COP 复位, 若禁止 COP 必须由用户初始化程序置“1”该位。

结构寄存器 CONFIG1.7 为 COP 溢出周期选择位 COPRS。COPRS=1 时, COP 计数器溢出周期为  $(2^{13}-2^4)/\text{CGMXCLK}$ ; COPRS=0 时, COP 溢出周期位  $(2^{18}-2^4)/\text{CGMXCLK}$ 。COPRS 初态为 0, 也可以由软件置位。

## 七. LVI 低电压禁止模块

### 1、LVI 模块功能

LVI 模块监视主电源  $V_{DD}$  端的输入电压, 当  $V_{DD} < V_{TRIPT}$  时, 禁止系统继续工作, 产生内部复位信号。LVI 的结构框图 1 所示。

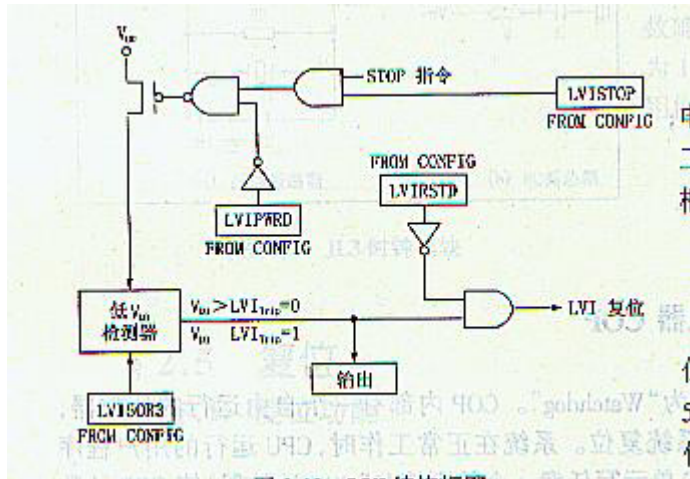


图 1

## 2、LVI 的控制

### 1) 监视 V<sub>DD</sub> 电压阈值 V<sub>TRIPF</sub> 选择

CONFIG1.3 “电源工作方式选择位” (LVI50R3)。LVI50R3=0，工作于 5V 方式，V<sub>TRIPF</sub>=4.3V；LVI50R3=1，工作 3V 方式，V<sub>TRIPF</sub>=2.6V。

系统复位后 LVI50R3=0，工作于 3V 方式，可由软件置“1”，使 MCU 工作于 5V 方式。

### 2) LVI 使能控制

CONFIG1.4 为“LVI 禁止位” (LVID)。LVID=1，禁止 LVI 产生复位信号，LVID=0 允许 LVI 产生复位信号；可以由软件置“1” LVID 以禁止 LVI 产生复位信号。