

MQ6801 仿真说明事项

V1.0

目錄

1.變更歷程 Change History	3
2.仿真 MQ6801 注意事項	4
3.調試注意事項	5
3.1 調試代碼注意事項	5
3.2 MQ6801 調試注意事項	5

1. 變更歷程

Version	Approved Date	Description	Issuer
V0.9	2016/01/08	新建立	彭兆銓
V1.0	2016/02/19	修改仿真注意事項	彭兆銓

2. 仿真 MQ6801 注意事項

MQ6801 (MQ6811) 仿真注意事項			
項次	功能	注意事項	參照說明
1	調試腳(P40,P41,RESETB)注意事項	無法使用(調試占用)	章節 3.1 (1)~(3)
2	單步運行	單步運行會全速運行連續的單周期指令	章節 3.2(1)
3	程序暫停(斷點觸發或暫停)	要注意設置時機與處理方式	章節 3.2(2)~(5)
4	WUC	無法仿真	章節 3.2 (6)
5	ADC	當程序暫停後，需再次讀取 ADCDRH 來允許下一輪轉換數值填入 ADCDRL/H 以及 INTADC 中斷觸發	章節 3.2(7)~(8)
6	ADC	程序暫停影響 ADC 轉換數值	章節 3.2 (7)~(8)
7	低速時鐘控制寄存器 FSCTRL	b6,b7 為包裝型態設置，在調試模式下能被設置	章節 3.2 (9)

3. 调试注意事项

3.1 调试代码注意事项

- (1) P10 RESET 功能不能停用，避免 IDE 在执行芯片复位时出现失去控制情形。
- (2) OCDCK ,OCDIO 占用 P40 P41，不要使用这两管脚。
- (3) 若要使用 MQLink 飞线到板上调试 MQ6801，板上芯片的 P10,P40,P41 脚不要连接额外负载。

3.2 MQ6801 调试注意事项

(1)当单步执行遇到单周期指令时，请注意:

当使用单步运行时，若该指令非以下单周期(instruction cycle)指令，则可正常单步运行。若为单周期指令，则会运行到下一个非单周期指令运行完为止。下列为单周期指令:

NOP ,SET CF ,CLR CF ,CPL CF ,DEC r ,INC r ,LD A,r ,LD r,A

注:r = W,A,B,C,D,E,H,L

(2)若断点前为两个 PUSH 指令，程序会在(PC 断点-1)地址触发断点。

设置断点后，检查反汇编窗口，避免断点前为两个PUSH指令；或设置断点时，增加3个NOP代码，并设置断点于中间，例:

```
__ASM("NOP");
```

```
__ASM("NOP");<= 於此設置斷點
```

```
__ASM("NOP");
```

(3)若断点前一个指令为跳跃指令，当跳跃发生时，程序会暂停在跳跃目标地址

设置断点后，检查反汇编窗口，避免斷點前為跳躍指令；或设置断点时，增加3个NOP代码，并设置断点于中间，例:

```
__ASM("NOP");
```

```
__ASM("NOP");<= 於此設置斷點
```

```
__ASM("NOP");
```

(4)当中断发生时，若后+5 地址内，恰好有断点设置，则中断触发取消

在仿真、测试中断功能(例TC0 定时器模式、INTx外部中断..)时，请勿于等待中断发生的循环中设置断点。

(5)两个断点设置地址需间隔 2 以上(包含 2)

断点可以设置于两相邻指令，但地址需间隔2以上(前面指令不能为单字节)。

(6)仿真时 WUC 功能无法由软件开启

调试时，WUC不会发出INTWUC中断，请使改用循环来达到延时功能。下列为范例函数代码，参照下表调用WUC_DelayLoop(time)来仿真WUC延时功能

```
void WUC_DelayLoop(unsigned char time)
{
    unsigned int counter;
    counter = ((unsigned int)(time))<<3;
    for(;counter>0; counter-)
    {
        __ASM("NOP");
        __ASM("NOP");
    }
}
```

WUC_DelayLoop(time)	不同系统时钟时，WUC_DelayLoop延迟时间			
	SysClk=8Mhz	SysClk=4Mhz	SysClk=2MHz	SysClk=8192Hz
参数time数值				
time = 0x19	~208 us	~416 us	~820 us	~200ms
time = 0x33	~416 us	~820 us	~1640 us	~400ms
time = 0x66	~820 us	~1640 us	~3280 us	~800ms
time = 0xCC	~1640 us	~3280 us	~6560 us	~1600ms

(7)当程序暂停后，需再次读取 ADCDRH 来允许下一轮转换数值填入 ADCDRL/H 以 INTADC 中断触发

在 ADC 重复模式下，当转换完成后，转换数据会保存在 ADCDRL, ADCDRH 内，直到读取 ADCDRH 后，才允许下一轮转换数据填入。当程序暂停(断点触发或暂停)后，ADCDRL, ADCDRH 数据会再次保护状态，需要再一次读取 ADCDRH 来允许下一轮转换数据以及 ADC 中断触发。

(8)程序暂停影响 ADC 转换数值

在程序暂停(断点触发或暂停)后，若 ADC 还处于开启状态(转换中或转换完)，此时 ADC 转换数值(尚未填入至 ADCDRL/H 的数值)会开始上升直到 0x3FF。

为了避免(7)和(8)的问题，请确认转换完后(读取 ADCCR2<EOCF>为 1)才触发断点\暂停，

且程序暫停(斷點觸發或暫停)前，確認 ADC 已關閉(設置 ADCCR1<AMD>為 0)。

(9)低速時鐘控制寄存器 FSCTRL

因 b6, b7 為包裝設置，且在調試模式下可被更動，在切換內外部低速時鐘時，注意不能直接讀取後寫回(b6 b7 讀為 0)，會造成包裝設置改變，需從信息塊讀取正確包裝設置值後，一併寫回，下列為範例代碼:

```
//設置 FSCTRL bit0 為 1，使用外部低速時鐘
```

```
FSCTRL.byte =( FSCTRL.byte & 0xFE) | ( (*(unsigned char*)(0x7e04))<<2)&0xC0 | 0x01;
```

```
//設置 FSCTRL bit0 為 0，使用內部低速時鐘
```

```
FSCTRL.byte =( FSCTRL.byte & 0xFE) | ( (*(unsigned char*)(0x7e04))<<2)&0xC0 | 0x00;
```