应用 X5043/X5045 对 8051 微控制器的管理

X5043/X5045 都有上电复位、低电压复位控制、可编程看门狗定时器、4Kbit 3-WIRE 接口非易失性 EEPROM、仅有 8 个引脚的封装。

上电复位(POR)

当系统上电时，X5043/X5045 的上电复位电路使得 RESER 引脚保持 250ms 激活状态。这防止了微控制器在电源稳定之前的误操作，提高了系统启动的可靠性。

低电压复位(LVR)

工作过程中，低电压复位电路可以检测到供电电压。如果电压低于某一特定值，X5043/X5045 激活 RESET 引脚，停止了微控制器的工作，防止意想不到的操作。如果微控制器工作电压太低，微处理器或外设就会失效，导致系统“锁死”或数据丢失。

看门狗定时器

上电复位(POR)和低电压复位(LVR)电路反之系统出现问题，看门狗定时器帮助系统从问题中恢复出来。计数时间到，看门狗复位系统。作为软件循环的一部分，定时器计时完成前，微处理器复位看门狗定时器。如果有软件问题，如死循环或等待外部器件，看门狗定时到，就会复位微控制器。

硬件电路

如图 1 所示，电路包括手动复位和 X5043 控制复位。R1 作为漏极开路(激活状态为“低”)复位输出的上拉电阻。2N7000 N-MOSFET 管用来转换激活复位信号(低)，可以直接控制 8031 的 RST 引脚。图 2 所示的电路有一个手动复位和 X5045 控制复位。图 2 所示的电路更好一些，因为 X5045 输出极性与 8051 正好一致。

软件设置

X5045/X5045 需要有下列程序接口：

wren_cmd：设置写允许。必须在写 EEPROM 存储器阵列或写状态之前设置。写操作后，WEL 位自动复位

wrdi_cmd：复位写允许(写禁止)

wrsr_cmd：写状态寄存器中的看门狗定时位(WD0, WD1)和块保护位(BP0, BP1)。

rdsr_cmd：读状态寄存器

byte_write：单字节写入到 EEPROM 存储器阵列

byte_read：从 EEPROM 存储器阵列读取单字节

page_write：向 EEPROM 存储器阵列写入 3 个连续字节。可很容易改为写入一页(至多 16 字节)

sequ_read：从 EEPROM 存储器阵列顺序读取字节。很容易改为读任何字节

rst_wdog：复位看门狗定时器

图 1　X5043 与具有手动复位的 8051 微控制器的连接

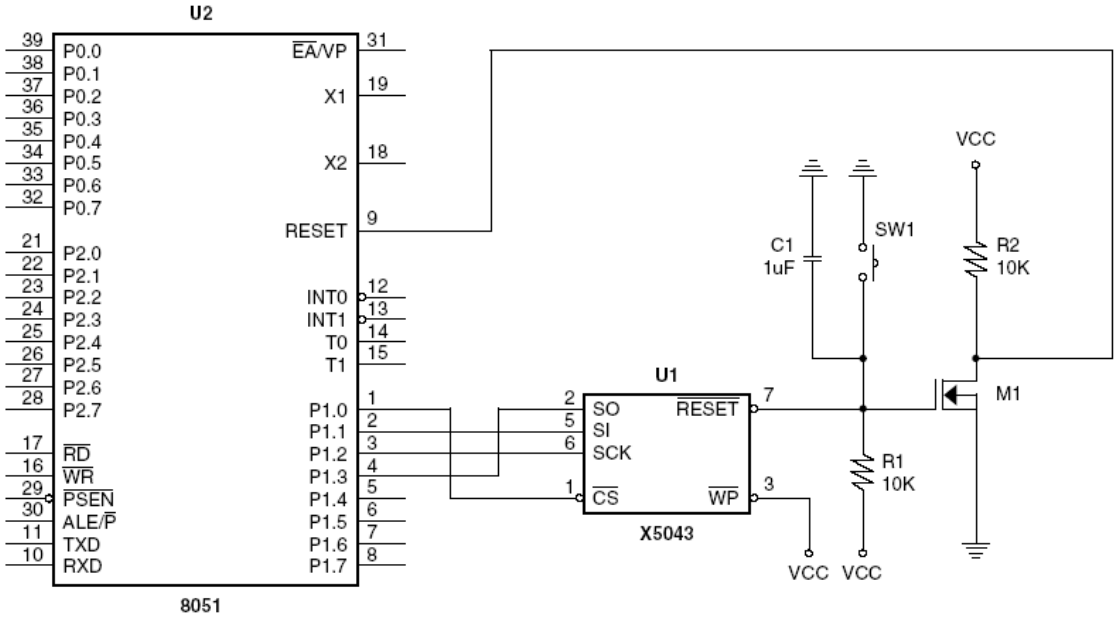
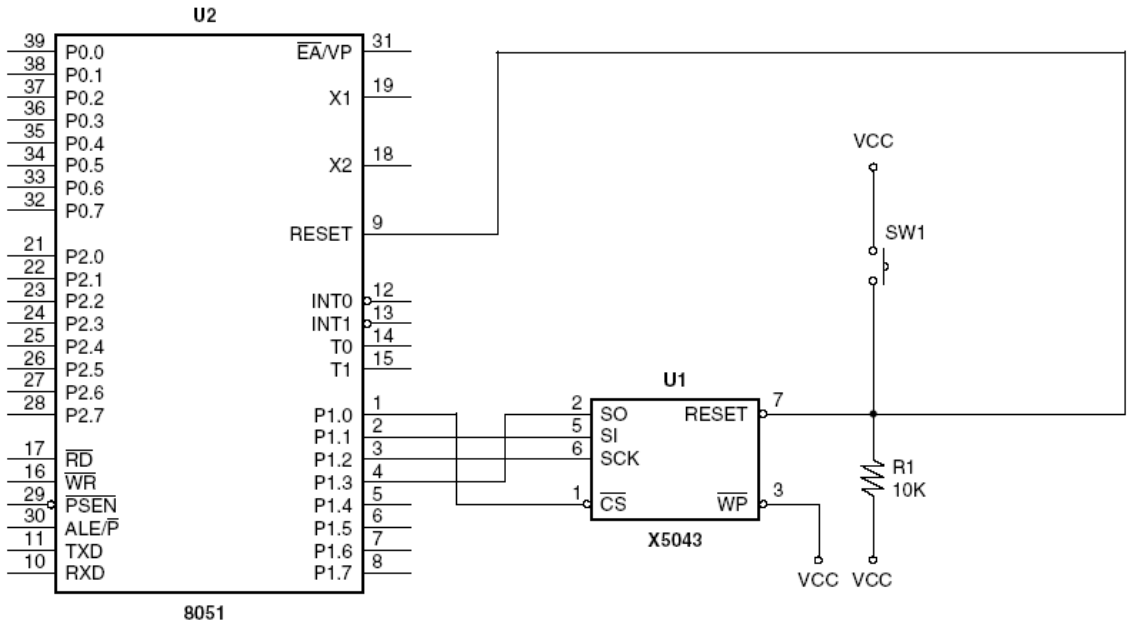


图 2　X5045 与具有手动复位的 8051 微控制器的连接



```
;* 标题(X5043/8031/1.0)
;*****************************************************************************************
;* Copyright (c) 1994 Xicor, Inc.
;* 作者: Richard Downing
;*****************************************************************************************
;* 这段代码为用户提供X5045和8031微控制器之间的一个接口。
;* 这个接口使用通用并行口P1: P1.0接片选端(/CS)；P1.1接串行输入数据(SI)；
;* P1.2接串行时钟(SCK);P1.3接串行输出数据(SO)
;*
;* 所有X5045指令(功能)如下:-
;*
;* 1. 设置写允许
;* 2. 复位写允许(写禁止)
;* 3. 写状态寄存器
;* 4. 读状态寄存器
;* 5. 单字节写
;* 6. 单字节读
;* 7. 页写
;* 8. 顺序读
;* 9. 复位看门狗定时器
;
;* 这段程序将00H写至状态寄存器；读状态寄存器；将11H按字节模式写进地址55H；
;* 从55H单字节读；将22H、33H、44H按页写模式写进1F0H、1F1H和1F2H；
;* 从1F0H、1F1H和1F2H顺序读；复位看门狗定时器。
;* 这段代码也适用于X5045，只是RESET输出极性与X5043不同

;*****************************************************************************************
;* 内容
cs bit P1.0                     ; P1.0脚，用作片选(/CS)
si bit P1.1                     ; P1.1脚，用作串行输入(CI)
sck bit P1.2                    ; P1.2脚，用作串行时钟(SCK)
so bit P1.3                     ; P1.3脚，用作串行输出(SO)
WREN_INST equ 06H               ; 写允许指令(WREN)
WRDI_INST equ 04H               ; 写禁止指令(WRDI)
WRSR_INST equ 01H               ; 写状态寄存器指令(WRSR)
RDSR_INST equ 05H               ; 读状态寄存器指令(RDSR)
WRITE_INST equ 02H              ; 写寄存器指令(WRITE)
READ_INST equ 03H               ; 读寄存器指令(READ)
BYTE_ADDR equ 55H               ; 字节工作模式的地址
BYTE_DATA equ 11H               ; 字节写操作的数据
PAGE_ADDR equ 1F0H              ; 页操作模式的存储器地址
PAGE_DATA1 equ 22H              ; 页写模式的第1个数据
PAGE_DATA2 equ 33H              ; 页写模式的第2个数据
PAGE_DATA3 equ 44H              ; 页写模式的第3个数据
STATUS_REG equ 00H              ; 状态寄存器
MAX_POLL equ 99H                ; Maximum number of polls
INIT_STATE equ 09H              ; 控制端口初始化(P1)
USER equ 030H                   ; 用户代码地址


;*****************************************************************************************
;* 初始化RAM
STACK_TOP equ 060H              ; 设置栈顶

;*****************************************************************************************
;* 代码
        ORG 0000H               ; 复位后从此处进入程序
        ljmp main
        ORG 0100H
main:
        mov SP, #STACK_TOP      ; 初始化栈顶
        clr EA                  ; 中断总关闭
        mov P1, #INIT_STATE     ; 初始化控制线(/CS&SO=1,SCK&SI=0)
        lcall wren_cmd          ; 设置写允许
        lcall wrsr_cmd          ; 写00H到状态寄存器
        lcall wren_cmd          ; 设置写允许
        lcall byte_write        ; 调用字节写(写11H至55H)
        lcall byte_read         ; 调用字节读(从55H读取数据)
        lcall wren_cmd          ; 设置写允许
        lcall page_write        ; 调用页写(写22H/33H/44H 至1F0/1/2H)
        lcall sequ_read         ; 调用顺序读(从1F0/1/2H读取)
        lcall rst_wdog          ; 调用复位看门狗
        jmp USER

;*****************************************************************************************
;* 名称: WREN_CMD
;* 描述: 设置写允许
;* 功能: 这段程序允许写EEPROM存储器阵列或状态寄存器
;* 调用: outbyt
;* 输入: 无
```

```
;* 输出: 无
;* 使用寄存器: A
;*******************************************************************************************
;
wren_cmd:
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #WREN_INST       ; 给A赋写允许指令(0000 0110)
        lcall outbyt            ; 送WREN指令
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        ret

;*******************************************************************************************
;* 名称: WRDI_CMD
;* 描述: 复位写允许
;* 功能: 这段程序禁止写EEPROM存储器阵列或状态寄存器
;* 调用: outbyt
;* 输入: 无
;* 输出: 无
;* 使用寄存器: A
;*******************************************************************************************
;
wrdi_cmd:
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #WRDI_INST       ; 给A赋写禁止指令(0000 0100)
        lcall outbyt            ; 调用outbyt,送出WRDI指令
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        ret

;*******************************************************************************************
;* 名称: WRSR_CMD
;* 描述: 写状态寄存器
;* 功能: 这段程序写WD0、WD1、BP0 and BP1 EEPROM
;* 调用: outbyt, wip_poll
;* 输入: 无
;* 输出: 无
;* 使用寄存器: A
;*******************************************************************************************
;
wrsr_cmd:
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #WRSR_INST       ; 写状态寄存器指令
        lcall outbyt            ; 送出WRSR指令
        mov A, #STATUS_REG      ; 写状态内容
        lcall outbyt            ; 送出状态寄存器内容
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        lcall wip_poll          ; 查写周期完成否(等待写完)
        ret

;*******************************************************************************************
;* 名称: RDSR_CMD
;* 描述: 读状态寄存器
;* 功能: 这段程序读状态寄存器内容
;* 调用: outbyt, inbyt
;* 输入: 无
;* 输出: A = 状态寄存器
;* 使用寄存器: A
;*******************************************************************************************
;
rdsr_cmd:
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #RDSR_INST       ; 读状态寄存器指令
        lcall outbyt            ; 送出RDSR指令
        lcall inbyt             ; 读出状态寄存器
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        ret

;*******************************************************************************************
;* 名称: BYTE_WRITE
;* 描述: 单字节写
;* 功能: 这段程序送出命令，写单个字节到EEPROM存储器阵列
;* 调用: outbyt, wip_poll
;* 输入: 无
;* 输出: 无
;* 使用寄存器: A, B
;*******************************************************************************************
;
byte_write:
```

```
        mov DPTR, #BYTE_ADDR    ; 设置字节写地址
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #WRITE_INST      ; 写指令(A3不确定)
        mov B, DPH
        mov C, B.0
        mov ACC.3, C            ; 确定写哪一页
        lcall outbyt            ; 送出写指令(包括页地址ACC.3)
        mov A, DPL
        lcall outbyt            ; 送出8位地址
        mov A, #BYTE_DATA       ; 数据
        lcall outbyt            ; 送出数据
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        lcall wip_poll          ; 查写周期是否完成(等待写完)
        ret

;*********************************************************************************************
;* 名称: BYTE_READ
;* 描述: 单字节读
;* 功能: 这段程序送出命令，读EEPROM存储器中的一个字节数据
;* 调用: outbyt, inbyt
;* 输入: 无
;* 输出: A = read byte
;* 使用寄存器: A, B
;*********************************************************************************************
byte_read:
        mov DPTR, #BYTE_ADDR    ; 设置读取数据的地址
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #READ_INST       ; 读指令
        mov B, DPH
        mov C, B.0
        mov ACC.3, C            ; 确定读哪一页
        lcall outbyt            ; 送出读指令(包括页地址AA.3)
        mov A, DPL
        lcall outbyt            ; 送出地址(8位)
        lcall inbyt             ; 读取数据
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        ret

;*********************************************************************************************
;* 名称: PAGE_WRITE
;* 描述: 页写
;* 功能: 这段程序送出指令，通过页写方式，向EEPROM存储器写入连续的3个字节
;* 调用: outbyt, wip_poll
;* 输入: 无
;* 输出: 无
;* 使用寄存器: A, B
;*********************************************************************************************
page_write:
        mov DPTR, #PAGE_ADDR    ; 设置第一个要写的字节的地址
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #WRITE_INST
        mov B, DPH
        mov C, B.0
        mov ACC.3, C
        lcall outbyt            ; 写指令，包括页地址
        mov A, DPL
        lcall outbyt            ; 送写地址(8位)
        mov A, #PAGE_DATA1
        lcall outbyt            ; 送出第一个数据字节
        mov A, #PAGE_DATA2
        lcall outbyt            ; 送出第二个数据字节
        mov A, #PAGE_DATA3
        lcall outbyt            ; 送出第三个数据字节
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        lcall wip_poll          ; 查写周期是否完成(等待写完)
        ret

;*********************************************************************************************
;* 名称: SEQU_READ
;* 描述: 顺序读
;* 功能: 这段程序送出读指令，从EEPROM存储器阵列读连续三个字节
;* 调用: outbyt, inbyt
;* 输入: 无
;* 输出: A = last byte read
```

```
;* 使用寄存器: A, B
;*********************************************************************************************
;
sequ_read:
        mov DPTR, #PAGE_ADDR    ; 设置读字节的第一个地址
        clr sck                 ; SCK置低
        clr cs                  ; /CS置低
        mov A, #READ_INST       ; 读指令
        mov B, DPH
        mov C, B.0
        mov ACC.3, C
        lcall outbyt            ; 送出读指令(包括页地址)
        mov A, DPL
        lcall outbyt            ; 送出读地址(8位)
        lcall inbyt             ; 读第一个字节
        lcall inbyt             ; 读第二个字节
        lcall inbyt             ; 读第三个字节
        clr sck                 ; SCK置低
        setb cs                 ; /CS置高
        ret

;*********************************************************************************************
;* 名称: RST_WDOG
;* 描述: 复位看门狗
;* 功能: 这段程序不需要其它指令复位看门狗
;* 调用: 无
;* 输入: 无
;* 输出: 无
;* 使用寄存器: None
;*********************************************************************************************
;
rst_wdog:
        clr cs                  ; /CS置低，复位看门狗
        setb cs                 ; /CS置高
        ret

;*********************************************************************************************
;* 名称: WIP_POLL
;* 描述: 写进度查询
;* 功能: 这段程序通过检测状态寄存器中的WIP位，来查询非易失性写周期是否完成
;* 调用: rdsr_cmd
;* 输入: 无
;* 输出: 无
;* 使用寄存器: R1, A
;*********************************************************************************************
;
wip_poll:
        mov R1, #MAX_POLL       ; 设置查询最大数目
wip_poll1:
        lcall rdsr_cmd          ; 读状态寄存器
        jnb ACC.0, wip_poll2    ; 如果WIP＝'0' , 写完成
        djnz R1, wip_poll1      ; 如果WIP＝ '1' , 继续查询(等待)
wip_poll2:
        ret

;*********************************************************************************************
;* 名称: OUTBYT
;* 描述: 送一个字节给EEPROM
;* 功能: 这段程序送一个字节给EEPROM，高位(MSB)在前
;* 调用: 无
;* 输入: A (要送的字节)
;* 输出: 无
;* 使用寄存器: R0, A
;*********************************************************************************************
;
outbyt:
        mov R0, #08             ; 设置位计数器(8个)
outbyt1:
        clr sck                 ; SCK置低
        rlc A                   ; 带进位C字节左移
        mov si, C               ; 送出移出的位
        setb sck                ; SCK置高
        djnz R0, outbyt1        ; 最后一位，则结束
        clr si                  ; 置SI于已知状态
        ret

;*********************************************************************************************
;* 名称: INBYT
;* 描述: 从EEPROM接收一个字节
;* 功能: 这段程序从EEPROM接收一个字节，高字节(MSB)在前
;* 调用: 无
;* 输入: 无
;* 输出: A(接收到的字节)
;* 使用寄存器: R0, A
```

```
;*******************************************************************************************
inbyt:
        mov R0, #08             ; 设置位计数器(8个)
inbyt1:
        setb sck                ; SCK置高
        clr sck                 ; SCK置低
        mov C, so               ; 接收位并存储在进位C中
        rlc A                   ; 带进位C右移
        djnz R0, inbyt1         ; 最后一位，则结束
        ret

END
```