



<http://www.Microcontrol.cn> 微控设计网

中国 MSP430 单片机专业网站

MSP430F2 系列 16 位超低功耗单片机模块原理

第 5 章 Flash 块控制器

版本: 1.2

日期: 2007.6.

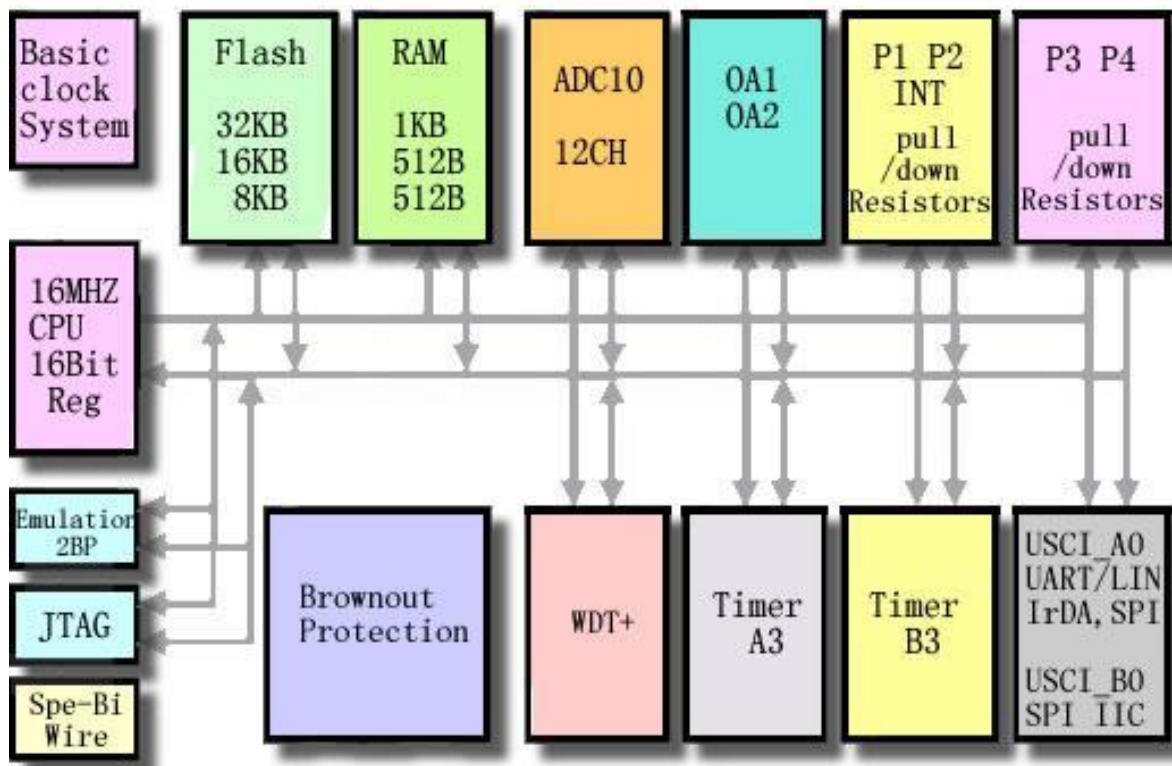
原文: TI MSP430x2xxfamily.pdf

翻译: 余川

编辑: DC 微控技术论坛总版主

注: 以下文章是翻译 TI MSP430x2xxfamily.pdf 文件中的部分内容。由于我们翻译水平有限, 有整理过程中难免有所不足或错误; 所以以下内容只供参考. 一切以原文为准。

详情请密切留意微控技术论坛。



第五章 Flash 存储控制器

本章介绍了 MSP430x2xx 系列单片机 Flash 存储控制器的操作。

- 5.1 Flash 存储器的介绍
- 5.2 Flash 存储器的分段结构
- 5.3 Flash 存储器的操作
- 5.4 Flash 存储器的控制寄存器

5.1 Flash 存储器的介绍

MSP430 的 Flash 存储器是可位/字节/字寻址和编程的存储器。该模块由一个集成控制器来控制编程和擦除的操作。控制器包括三个寄存器，一个时序发生器及一个提供编程/擦除电压的电压发生器。

MSP430 的 Flash 存储器的特点有：

- 产生内部编程电压
- 可位/字节/字编程
- 超低功耗操作
- 支持段擦除和多段模块擦除

Flash 存储器和控制器的结构框图如图 5-1 所示。

注意：Flash 写入和擦除操作期间的最小电压值 V_{CC} 应为 2.2V。如果在操作期间 V_{CC} 低于 2.2V，写入或擦除的结果将是不确定的。

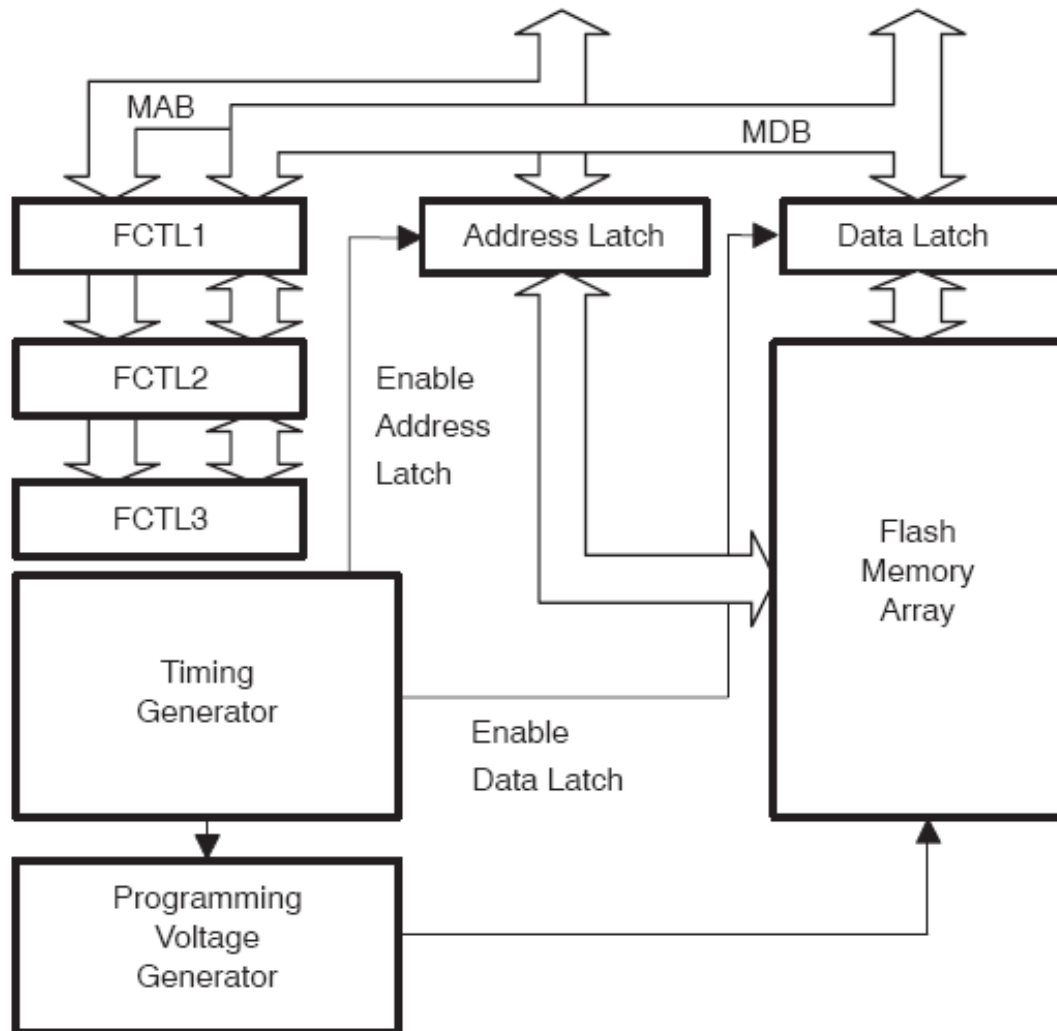


图 5-1 Flash 存储器框图

5.2 Flash 存储器的分段结构

MSP430 Flash 存储器分成多个段。可对其进行单个位/字节/字的写入，但是最小的擦除单位是段。

Flash 存储器分为主存储器和信息存储器两部分，在操作上两者没有什么区别，程序代码和数据可以存储于任意部分。两部分的区别在于段的大小和物理地址。信息存储器有四个 64 字节的段，主存储器有两个或更多的 512 字节的段。不同的芯片可参见各自数据手册的整体存储空间图。

每一段又进一步划分为 64 字节的块，起始地址分别为 0xx00h, 0xx40h, 0xx80h, 0xxC0h, 结束地址分别为 0xx3Fh, 0xx7Fh, 0xxBFh, 0xxFFh。

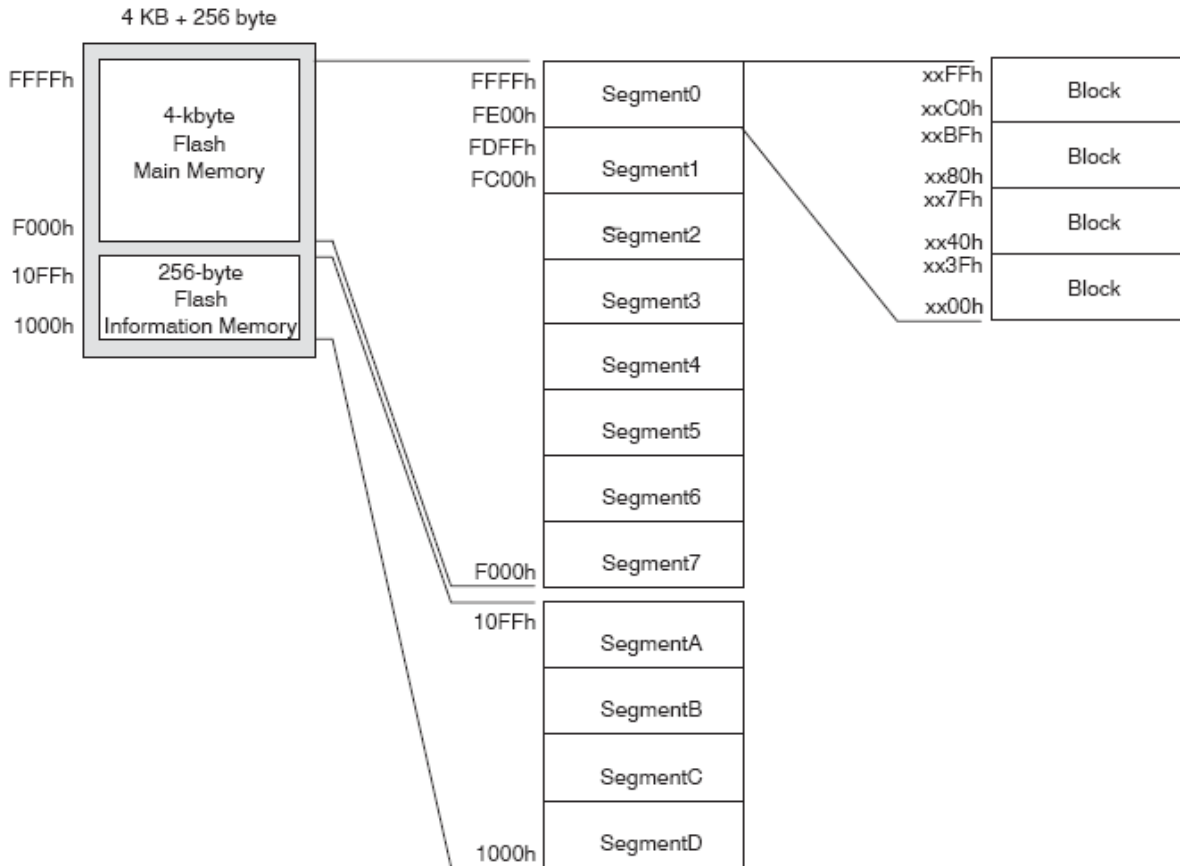


图5-2 Flash存储器分段（以4KB为例）

5.2.1 Segment A

信息存储器中的Segment A可以通过LOCKA位的设置进行锁定，从而与其他段的操作区分开。当LOCKA=1时，Segment A不能写入或擦除，其他的信息存储器在整个模块擦除时被保护起来。当LOCKA=0时，Segment A可以像其他Flash存储器一样进行擦除和写入的操作，同时其他的信息存储器在整个模块擦除时擦除。Segment B,C,D在段擦除时均能被擦除，与LOCKA位的状态无关。

当 LOCKA 写入为 1 时，LOCKA 的状态发生翻转（即如果 LOCKA 原来为 1，对它写入 1 则变为 0；如果 LOCKA 原来为 0，对它写 1 则它变为 1）；当 LOCKA 写入为 0 时，LOCKA 的状态不受影响。这样就能实现当前 flash 编程程序的固定。

； 解锁 SegmentA子程序

```

BIT #LOCKA,&FCTL3           ; 测试LOCKA位，判断是否锁定Segment A
JZ SEGA_UNLOCKED           ; 解锁了吗？
MOV #FWKEY+LOCKA,&FCTL3     ; 没有，解锁
SEGA_UNLOCKED              ; 是的，已经解锁了，继续
； 锁定SegmentA子程序
    BIT #LOCKA,&FCTL3           ; 测试LOCKA位，锁定Segment A 了吗
    JNZ SEGALOCKED           ; 锁定Segment A 了吗
    MOV #FWKEY+LOCKA,&FCTL3     ; 没有，锁定Segment A
    SEGA_LOCKED              ; 锁定了，继续执行
    
```

5.3 Flash 存储器的操作

Flash 存储器的默认工作模式是读模式，在该模式下，存储器无法进行擦除和写入操作，同时时序发生器和电压发生器关闭。此时的操作与 ROM 的操作类似。

MSP430 Flash存储器能在没有外加电压的条件下实现在系统编程（ISP），CPU能对它自带的Flash进行编程。Flash存储器的写入/擦除模式通过BLKWRT，WRT，MERAS和 ERASE位选择，能完成：

- 单字节/单字写入
- 块写入
- 段擦除
- 多段擦除（针对所有主存储器的段）
- 整体擦除（针对所有段）

在编程或擦除期间禁止对Flash存储器进行读或写操作。在写入或擦除是需要用到的CPU程序应该在RAM中运行。对Flash的任何更新都开始于Flash存储器或RAM中。

5.3.1 Flash存储器的时序发生器

时序发生器用于控制Flash存储器的编程和擦除操作，如图5-3所示。时序发生器的输入频率 f_{FTG} 必须在257 kHz 到 476 kHz内。（请参看详细的芯片数据手册）

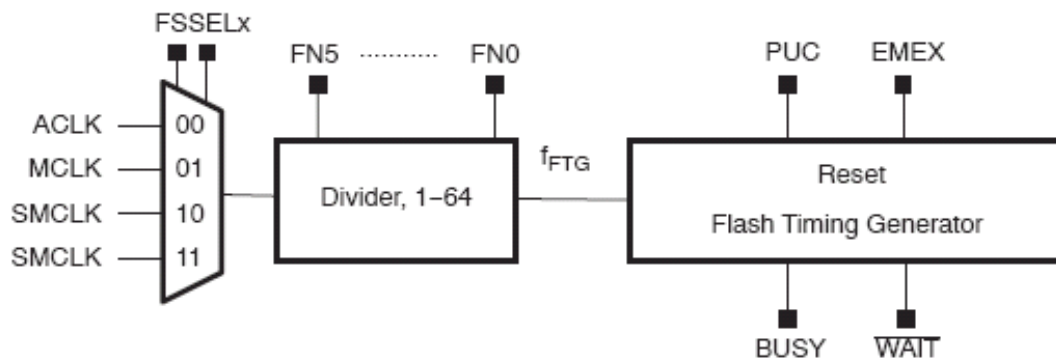


图5-3 时序发生器框图

Flash时序发生器时钟源的选择：

时序发生器可以选择3个时钟源之一：ACLK，SMCLK或 MCLK。选择的时钟信号可通过FNx位的进行分频以适应时序发生器输入频率 f_{FTG} 的要求。如果在写入或擦除操作过程中， f_{FTG} 不符合要求，写入或擦除的结果将不确定，或者Flash被迫工作于正常工作状态的边缘（即只要再超过一点就是非正常操作状态了，也可以翻译为：被迫工作于临界状态）。

如果在写入或擦除操作过程中检测到时钟信号失效，操作将终止，置位FAIL标志位，操作的结果将不确定。当写入或擦除操作启动了时钟源的选择，不会在MSP430进入低功耗模式时无效，将一直保持到写入或擦除操作完成。

5.3.2 Flash存储器的擦除

Flash存储器各位的缺省值为“1”，每一位都可以单独的编程为“0”，但只有擦除操作才能将它恢复成“1”。擦除操作的最小单位是段。通过ERASE和 MERAS位的设置可选择3种擦除模式，如表5-1所示。

表5-1 擦除模式

MERAS	ERASE	擦除模式
0	1	段擦除
1	0	多段擦除（所有主存储器的段）

1	1	整体擦除 (LOCKA=0时, 擦除所有主存储器和信息存储器的段; 主存储器的段只当LOCKA=0时可以擦除)
---	---	---

任何擦除操作开始于对擦除的地址范围内任意位置作一次空写入。空写入的目的是启动时序发生器和擦除操作, 如图5-4所示为擦除周期。在空写入操作之后, BUSY位立即置位, 并保持到擦除周期结束。BUSY, MERAS和ERASE位在擦除周期结束后自动复位。MSP430Fx2xx系列芯片的擦除周期均相等, 与片内Flash存储器的数量无关。

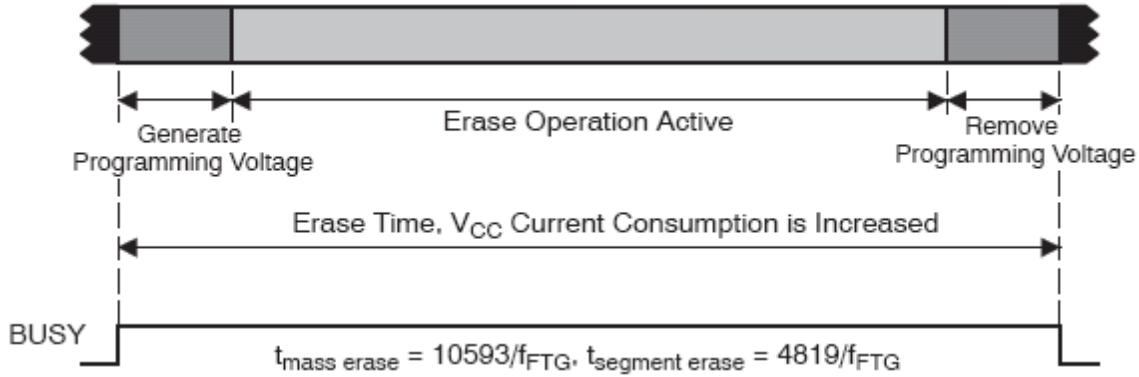


图 5-4 Flash 存储器擦除周期

如果空写入操作的地址在不能执行擦除操作的段地址范围内, 则写入操作不起作用, 对Flash存储器无影响, 也不会有任何标志指示这一状态, 错误的空写入将被忽略。

从 Flash 存储器启动的擦除操作:

任何擦除周期可以从片内 Flash 存储器或 RAM 存储器开始, 如果段擦除操作开始于片内 Flash 存储器内, 所有的定时都由 Flash 控制器控制, CPU 在擦除周期结束前将停止运行, 状态保持不变。在擦除周期结束后, CPU 将重新按照空写入之后的规定执行程序代码。

开始于 Flash 存储器内的段擦除操作, 同样可以把擦除之后要执行的程序代码清除掉, 一旦发生这样的情况, CPU 在擦除周期之后的操作将不确定。

由 Flash 存储器启动擦除操作流程如图 5-5 所示。

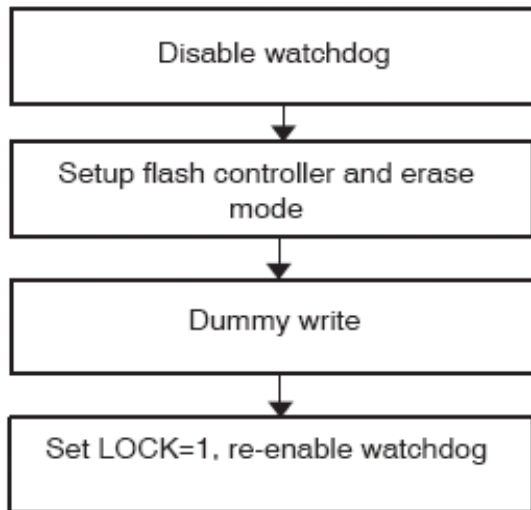


图 5-5 Flash 存储器启动擦除操作流程

; 从Flash存储器启动擦除操作, $514 \text{ kHz} < \text{SMCLK} < 952 \text{ kHz}$

; 假设 $\text{ACCVIE} = \text{NMIE} = \text{OFIE} = 0$

```

MOV    #WDTPW+WDTHOLD, &WDTCTL    ; 停止看门狗WDT
MOV    #FWKEY+FSSEL1+FN0, &FCTL2   ; SMCLK/2
MOV    #FWKEY,&FCTL3               ; 清除LOCK位
MOV    #FWKEY+ERASE,&FCTL1         ; 使能段操作
CLR    &0FC10h                     ; 空写入, 清除S1
MOV    #FWKEY+LOCK,&FCTL3          ; 空写入完成后, 置位LOCK
.....                               ; 重新使能看门狗WDT

```

从 RAM 存储器初始化的擦除操作:

任何擦除周期可以从RAM存储器开始, 在这种情况下, CPU不会停止运行, 能继续的执行RAM存储器中的程序代码。在CPU可以重新访问任何Flash地址前, 必须查询BUSY位以判断擦除周期的结束。如果进入Flash地址发生在BUSY=1时, 这个进入操作是非法的, ACCVIFG(非法访问中断标志)置位, 擦除结果将不确定。

由 RAM 存储器启动擦除操作流程如图 5-6 所示。

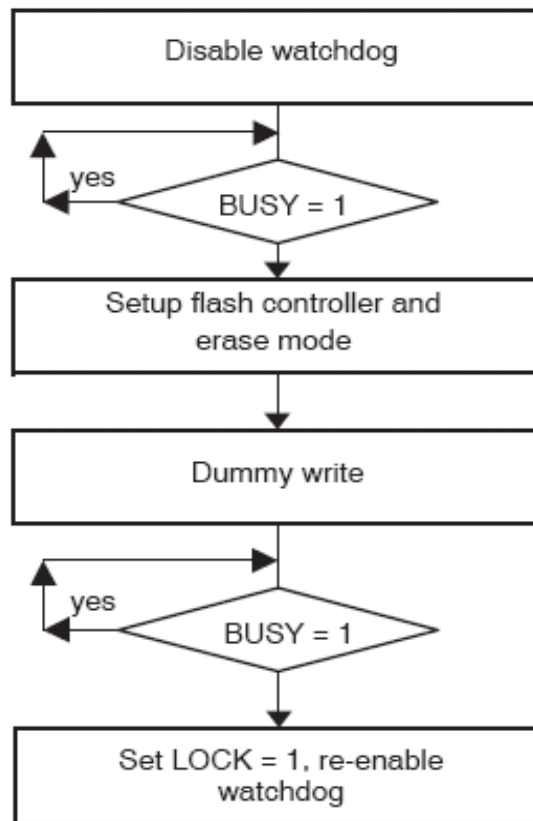


图 5-6 RAM 存储器启动擦除操作流程

; 从RAM存储器启动擦除操作, $514 \text{ kHz} < \text{SMCLK} < 952 \text{ kHz}$

; 假设 $\text{ACCVIE} = \text{NMIE} = \text{OFIE} = 0$

```

MOV    #WDTPW+WDTHOLD,&WDTCTL    ; 停止看门狗WDT
L1    BIT    #BUSY,&FCTL3         ; 测试BUSY位
JNZ    L1                          ; BUSY=1, 即忙, 则循环执行L1
MOV    #FWKEY+FSSEL1+FN0,&FCTL2   ; SMCLK/2

```

```

MOV    #FWKEY,&FCTL3           ; 清除LOCK位
MOV    #FWKEY+ERASE,&FCTL1     ; 使能擦除功能
CLR    &0FC10h                 ; 空写入, 清除S1
L2    BIT    #BUSY,&FCTL3      ; 测试BUSY位
      JNZ    L2                 ; BUSY=1, 即忙, 则循环执行L2
MOV    #FWKEY+LOCK,&FCTL3      ; 空写入完成后, 置位LOCK
      .....                    ; 重新使能看门狗WDT
    
```

5.3.3 Flash存储器的写入

写入模式由WRT和BLKWRT位进行设置, 如表5-2所示。

表5-2 写入模式

BLKWRT	WRT	写入模式
0	1	单字节/单字写入
1	1	块写入

所有的写入模式使用一系列特有的写入命令, 采用块写入的速度大约是单字节/单字写入的2倍, 因为电压发生器在块写入完成期间均能保持。对于这两种写入模式, 任何能修改目的操作数的指令均能用于修改地址。一个Flash字(低+高字节)不能在擦除期间进行两次以上的写入。否则, FLASH可能被损坏。

当启动写入操作时BUSY位置位, 写入结束时复位。如果写入操作由RAM存储器启动, 当BUSY=1时CPU无法进入Flash。否则, 非法进入将产生, ACCVIFG(非法访问中断标志)置位, 写入结果将不确定。

单字节/单字写入

单字节/单字写入可以在片内Flash存储器或RAM存储器执行, 如果段擦除操作开始于Flash存储器内, 所有的定时都由Flash控制器控制, CPU在写入结束前将停止运行, 状态保持不变。在写入结束后, CPU将重新按照写入之后的规定执行程序代码。单字节/单字写入时间周期如图5-7所示。

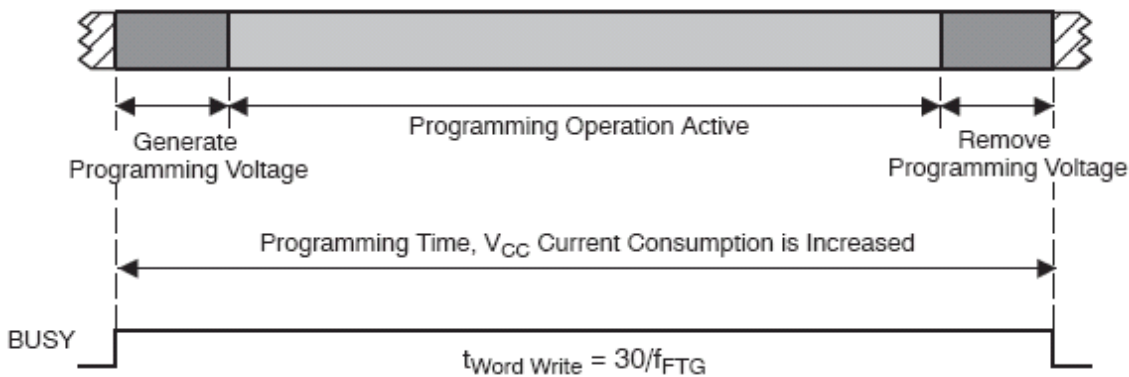


图5-7 单字节/单字写入周期

当是从RAM存储器启动时, CPU能继续的执行RAM存储器中的程序代码。在CPU可以重新进入任何Flash地址前, BUSY位必须复位。否则这个进入操作是非法的, ACCVIFG(非法访问中断标志)置位, 写入结果将不确定。

在单字节/单字写入模式下，内部产生的编程电压在整个64-byte的块编程时间段内都供电，每次写一个字节或字，每次需要写27个 f_{FTG} 周期（见图5-7，即写一个字或字节需要30个周期，其中只有中间27个是编程周期，头尾3个是电压产生和消除周期；一个段（segment）有256个字节，分为4个块，每块64字节，所以它说加电加在“64byte的块”）。在每个字或字节写入时，编程电压所加载那一块（block）的时间一直在累加。累加时间 t_{CPT} ，不能超过任何一块所允许的编程总时间。如果累加编程时间到了，块必须在对该块的写操作之前执行擦除操作。（请参看详细的芯片数据手册）

从Flash存储器启动单字节/单字写入：

从Flash存储器启动单字节/单字写入的操作流程如图5-8所示。

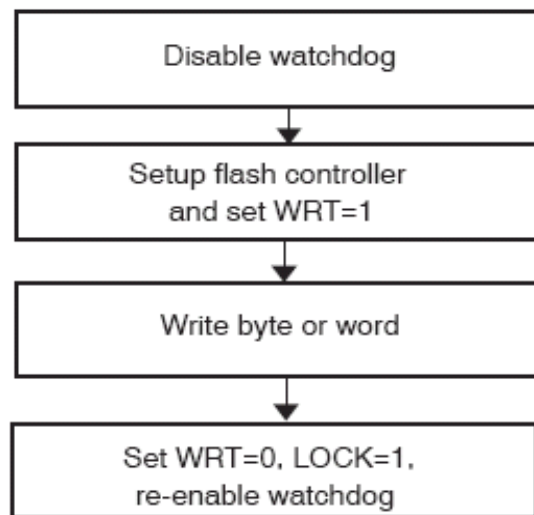


图5-8 Flash存储器启动单字节/单字写入流程

；由Flash启动单字节/单字写入， $514\text{ kHz} < \text{SMCLK} < 952\text{ kHz}$

；假设地址0FF1Eh已经擦除

；假设 $\text{ACCVIE} = \text{NMIE} = \text{OFIE} = 0$

MOV	#WDTPW+WDTHOLD, &WDTCTL	； 停止看门狗WDT
MOV	#FWKEY+FSSEL1+FN0, &FCTL2	； SMCLK/2
MOV	#FWKEY, &FCTL3	； 清除LOCK位
MOV	#FWKEY+WRT, &FCTL1	； 使能写入功能
MOV	#0123h, &0FF1Eh	； 0123h -> 0FF1Eh
MOV	#FWKEY, &FCTL1	； 完成后清除WRT位
MOV	#FWKEY+LOCK, &FCTL3	； 置位LOCK位
	； 重新使能看门狗WDT

从RAM存储器启动单字节/单字写入：

从RAM存储器启动单字节/单字写入的操作流程如图5-9所示。

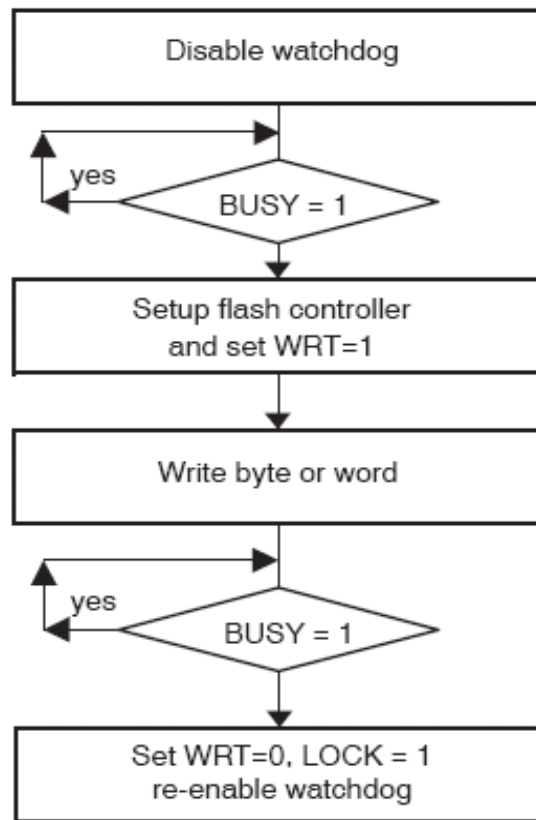


图5-9 RAM存储器启动单字节/单字写入流程

- ; 从RAM存储器启动单字节/单字写入的操作， $514\text{ kHz} < \text{SMCLK} < 952\text{ kHz}$
- ; 假设地址0FF1Eh已经擦除
- ; 假设 $\text{ACCVIE} = \text{NMIE} = \text{OFIE} = 0$.

```

MOV    # WDTPW+WDTHOLD, &WDTCTL    ; 停止看门狗WDT
L1     BIT    # BUSY, &FCTL3        ; 测试BUSY位
      JNZ    L1                    ; BUSY=1, 即忙, 则循环执行L1
MOV    # FWKEY+FSSEL1+FN0, &FCTL2   ; SMCLK/2
MOV    # FWKEY, &FCTL3             ; 清除LOCK位
MOV    # FWKEY+WRT,&FCTL1          ; 使能写入功能
MOV    # 0123h, &0FF1Eh           ; 0123h -> 0FF1Eh
L2     BIT    # BUSY, &FCTL3        ; 测试BUSY位
      JNZ    L2                    ; BUSY=1, 即忙, 则循环执行L2
MOV    # FWKEY, &FCTL1             ; 清除WRT位
MOV    # FWKEY+LOCK, &FCTL3       ; 置位LOCK
      .....
      ; 重新使能看门狗WDT
  
```

块写入

当有连续的字节或字需要进行编程时，块写入的方式能提高Flash写入的速度。在对64字节的块进行写入操作的过程中，Flash编程电压将会一直保持。累加时间 t_{CPT} ，不能超过该块所允许的编程总时间。

块写入周期不能在片内Flash中开始，必须从RAM存储器开始。在块写入期间，BUSY位保

持置位，在写入每个字节或字之间需要检测WAIT位的值，如果WAIT位置位表明下一个字节或字可以写入了。对连续的块进行写操作时，在当前块写入完成时，BLKWRT位必须复位。在FLASH的恢复时间 t_{end} 之后，BLKWRT必须置位以初始化下块的写操作。BUSY位在每个块写入操作完成时清零，以指示下一个块可以写入了。块写入时间周期如图5-10所示。

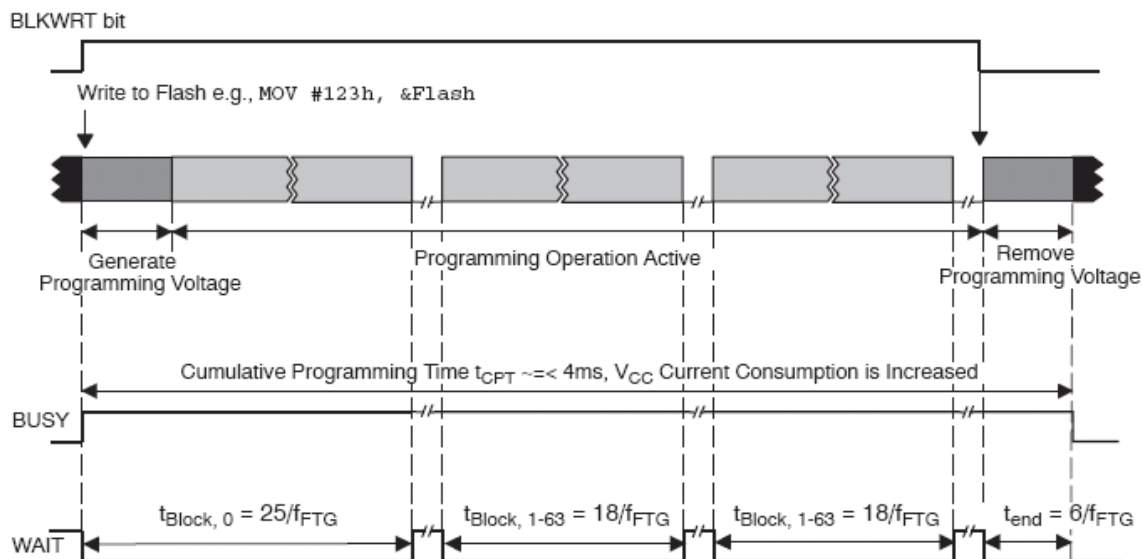


图5-10 Flash存储器块写入周期

块写入流程及举例

块写入流程如图5-11所示，下面为应用举例。

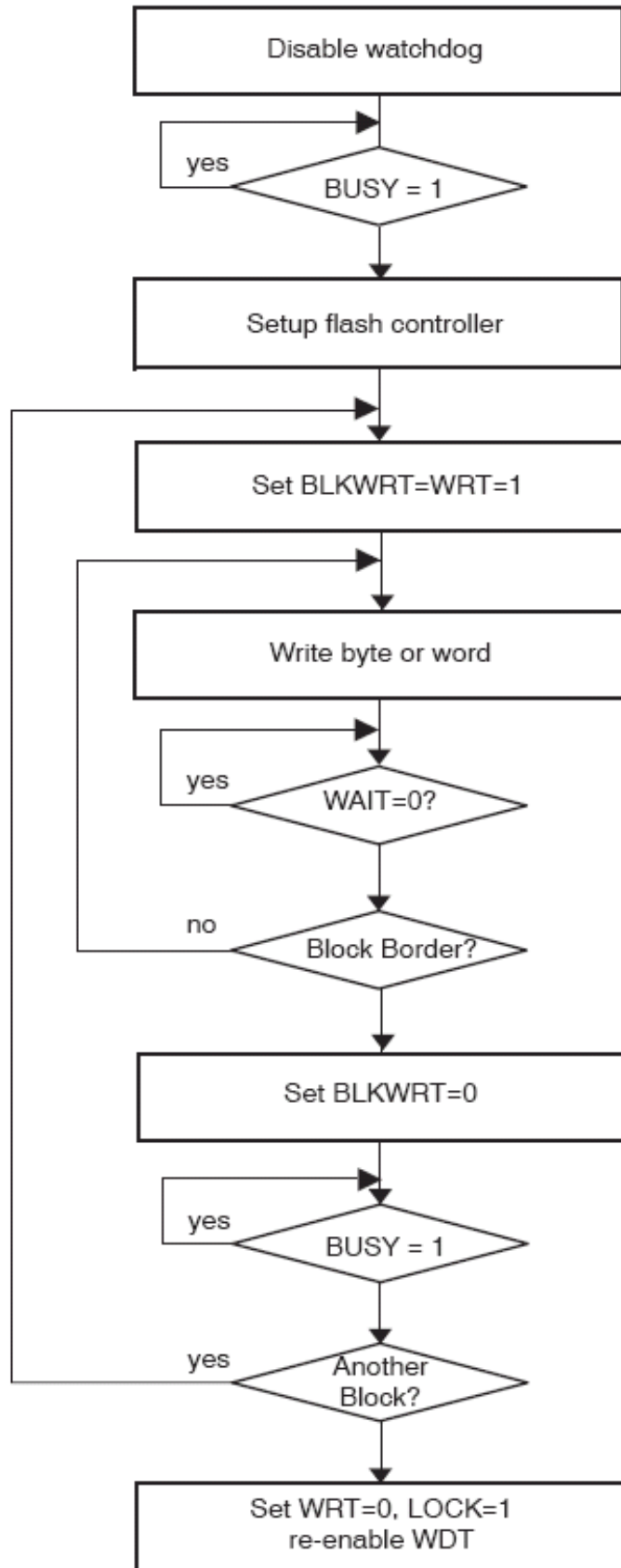


图5-11 Flash存储器块写入流程

```

; 写入一个其实地址为0F000h的块.
; 必须由RAM存储器启动执行, 假设Flash存储器已经擦除
; 514 kHz < SMCLK < 952 kHz
; 假设ACCVIE = NMIIE = OFIE = 0
MOV    #32, R5                ; 作为写入计数器
MOV    #0F000h, R6           ; 写操作数指针
MOV    #WDTPW+WDTHOLD, &WDTCTL ; 停止看门狗WDT
L1     BIT    #BUSY, &FCTL3    ; 测试BUSY位
JNZ    L1                    ; BUSY=1, 即忙, 则循环执行L1
MOV    #FWKEY+FSSEL1+FN0, &FCTL2 ; SMCLK/2
MOV    #FWKEY, &FCTL3        ; 清除LOCK位
MOV    #FWKEY+BLKWRT+WRT, &FCTL1 ; 使能块写入功能
L2     MOV    Write_Value,0(R6) ; 写入地址
L3     BIT    #WAIT, &FCTL3    ; 测试WAIT位
JZ     L3                    ; 当WAIT=0时, 循环执行L3
INCD   R6                   ; 指向下一个字
DEC    R5                    ; 计数器减1
JNZ    L2                    ; 写入是否完成? 没有则循环执行L2
MOV    #FWKEY, &FCTL1        ; 清除WRT和BLKWRT位
L4     BIT    #BUSY, &FCTL3    ; 测试BUSY位
JNZ    L4                    ; BUSY=1, 即忙, 则循环执行L4
MOV    #FWKEY+LOCK, &FCTL3    ; LOCK置位
.....                        ; 需要的话, 重新使能看门狗WDT

```

5.3.4 写入/擦除期间的Flash存储器访问

在由RAM存储器启动的任意写入/除操或BUSY=1时, CPU不能读/写Flash。否则, 将产生一个非法进入, ACCVIFG (非法访问中断标志) 置位, 写入结果将不确定。当然, 如果在WRT=0时对Flash进行写入, ACCVIFG (非法访问中断标志) 同样产生置位。Flash存储器不受影响。

在由Flash存储器启动的任意写入/除操或BUSY=1时, Flash控制器将在取出下一个指令时给CPU返回操作代码03FFFh, 相当于指令JMP PC, 这就会使CPU循环执行, 直到Flash写入操作完成。在操作完成且BUSY=0时, Flash控制器允许CPU取出正确的操作代码并重新运行程序。

在BUSY=1时, 对Flash的访问条件如表5-3所示。

表5-3 Flash的访问条件 (BUSY=1)

Flash的操作	Flash访问	WAIT位取值	结果
擦除或单字节/单字 写入	读	0	ACCVIFG=0, 读出结果为03FFFh
	写	0	ACCVIFG=1, 写操作被忽略
	取指令	0	ACCVIFG=0, CPU读取03FFFh, 相当于指令JMP PC
块写入	任意操作	0	ACCVIFG=1, LOCK=1
	读	1	ACCVIFG=0, 读出结果为03FFFh
	写	1	ACCVIFG=0, Flash执行写操作
	取指令	1	ACCVIFG=1, LOCK=1

当EEI=0, EEIEX=0时, Flash操作期间中断功能自动无效; 在MSP430x20xx系列没有EEI和

EEIEX位的芯片也同样适用。Flash操作结束后，中断功能自动重新使能。在操作期间发生的中断将相应标志位置位，在重新使能后产生中断请求。当EEIEX = 1, GIE = 1时，中断将立即停止Flash的操作，并置位FAIL位。当EEI=1, GIE=1, EEIEX=0时，每32个 f_{FTG} 周期段擦除将产生一个中断，在中断服务之后，段擦除将继续执行32个 f_{FTG} 周期，直至擦除结束。在中断服务程序执行时，BUSY保持置位，但是CPU可访问Flash存储器而不发生非法进入。不支持嵌套中断。

在擦除周期之前，看门狗（工作于看门狗模式）要停止工作。复位将停止擦除操作，使擦除结果不确定。在擦除周期结束之后，要重新使能看门狗。

5.3.5 停止写入/擦除周期

置位EMEX(紧急退出位)位，任意的写入/擦除操作均可以在正常结束前被停止。设置EMEX位能立即停止当前操作和Flash控制器。所有Flash操作停止，恢复到默认状态——读模式，FCTL1寄存器将全部复位。预计操作的结果将不可靠。

5.3.6 Flash存储控制寄存器设置和访问

FCTLx是一个16位，有安全键值保护，可读/写的寄存器。必须用字指令访问，写操作还需在高字节加入安全键值0A5h。键值出错将使KEYV置位并产生PUC（上电清除信号）系统复位。对FCTLx的读出需在高字节加入安全键值096h。

在擦除或单字节/单字写入操作期间对FCTL1进行写操作是非法的，将使ACCVIFG置位。在块写入模式下，当WAIT=1时，允许写FCTL1；但如果WAIT=0将置位ACCVIFG。

当BUSY=1时，对FCTL2的写入是非法的。

对FCTLx的读操作要在BUSY=1是进行，否则将产生非法操作。

5.3.7 Flash存储控制寄存器中断

Flash控制器有两个中断源：KEYV和ACCVIFG。发生非法操作时ACCVIFG位置位，在写入或擦除操作之后重新置位ACCVIFG位，此时ACCVIFG位的置位将产生一个中断请求。ACCVIFG是NMI（非屏蔽中断），所以不受GIE是否置位的影响。需进行软件检测ACCVIFG，确认是否有非法操作发生。ACCVIFG得由软件复位。

当写入错误键值时，KEYV位（安全键值错位）置位，同时立即产生PUC（上电清除信号）使系统复位。

5.3.8 Flash存储器的编程

对MSP430 Flash型单片机有三种编程方式的选择。所有方式都支持在线编程。

- 通过JTAG接口编程
- 通过引导加载程序编程
- 通过自定义方式编程

通过JTAG接口编程

MSP430单片机可以通过JTAG接口进行编程。JTAG接口需要四根信号线（在20或28引脚芯片上需要5根信号线），地线，VCC和RST/NMI。

JTAG接口由熔丝进行保护，熔丝的激活是不可逆的，一旦激活就不能再访问片内系统。详细情况请参看www.msp430.com上的使用报告中的《Programming a Flash-Based MSP430 Using the JTAG Interface》一文。

通过引导加载程序（Bootstrap Loader, BSL）编程

MSP430 Flash型单片机均包含一个引导装入程序，BSL使用户能通过UART串行接口对Flash存储器和RAM存储器进行编程。通过BSL访问Flash存储器由256位，用户自定义的口令进行保护。详细情况请参看www.ti.com/msp430上的使用报告中的《Features of the MSP430 Bootstrap Loader》一文。

通过自定义方式编程

MSP430 CPU对Flash存储器的在线和外部用户自定义写入方式如图5-12所示，用户可以选择通过UART, SPI等方式进行编程。用户自行开发的软件也可以接收数据或对Flash存储器编程，从而能方便的进行编程，擦除，更新Flash存储器的应用。

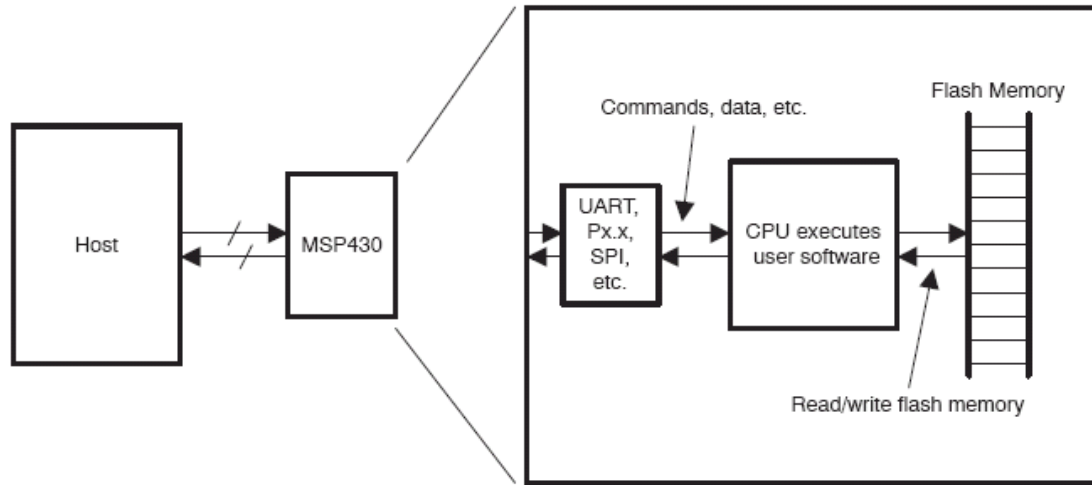


图5-12 用户自定义编程方式

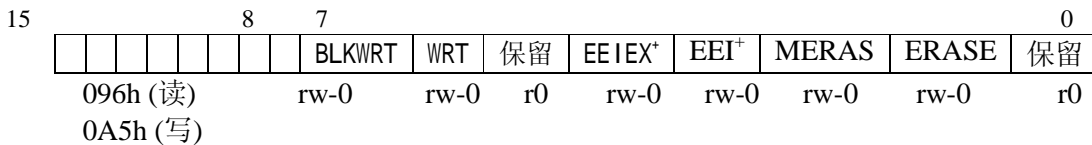
5.4 Flash 存储器控制寄存器

Flash存储器控制寄存器如所示。

表5-4 Flash存储器控制寄存器一览表

寄存器	简称	寄存器类型	地址	初始值(PUC后)
Flash memory control register 1	FCTL1	可读/写	0128h	09600h
Flash memory control register 2	FCTL2	可读/写	012Ah	09642h
Flash memory control register 3	FCTL3	可读/写	012Ch	09658h
Interrupt Enable 1	IE1	可读/写	0000h	Reset

Flash存储器控制寄存器FCTL1



注：在MSP430x20xx系列中无

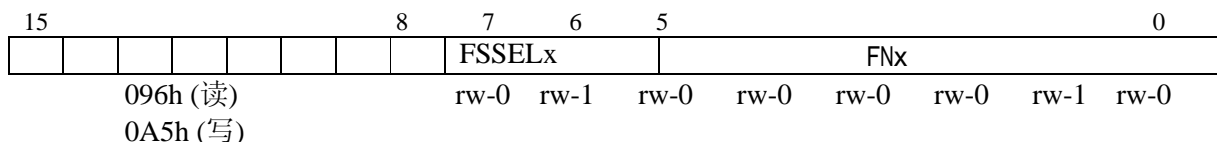
b15-8: **FRKEY/FWKEY, FCTLx安全键值**
读操作为096h，写操作为0A5h，写入错误将产生PUC。

b7: **BLKWRT, 块写入模式选择位**
在写模式下WRT也必须置位，EMEX置位时，WRT将自动复位。
0: 块写入模式关闭
1: 块写入模式开启

- b6/b0 **WRT**，写模式选择位
此位是用于选择写模式的。EMEX置位时，WRT将自动复位。
0: 写入模式关闭
1: 写入模式开启
- b5 **Reserved**，保留位
值为0。
- b4 **EEIEX**，紧急中断退出使能
通过设置此位，能在GIE = 1时，产生紧急中断，从而跳出Flash的操作。EMEX置位后，EEIEX自动复位。
0: 无作用
1: 中断使能
- b3 **EEL**，擦除中断使能
设置此位则允许段擦除产生中断请求，中断服务结束之后擦除周期继续进行。EMEX置位后，EEL自动复位。
0: 无作用
1: 使能段擦除中断
- b2 **MERAS**，集中擦除
- b1 **ERASE**，擦除
这两位同时用于选择擦除模式。EMEX置位后，MERAS和ERASE将自动复位。

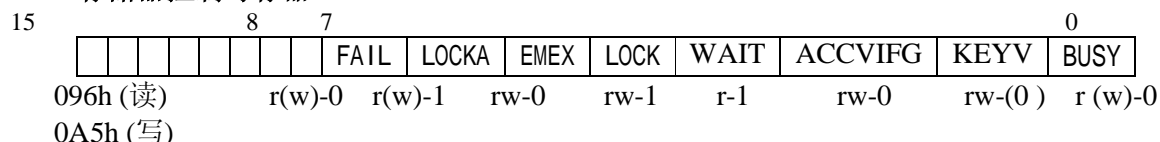
MERAS	ERASE	擦除周期
0	0	无擦除
0	1	只进行单段的擦除
1	0	擦除所有的主存储器段
1	1	LOCKA=0时，擦除所有主存储器和信息存储器段 LOCKA=1时，擦除所有主存储器

Flash存储器控制寄存器FCTL2



- b15-8: **FWKEYx**，FCTLx安全键值
读操作为096h，写操作为0A5h，写入错误将产生PUC。
- b7-6: **FSSELx**，时钟源选择位
00 ACLK
01 MCLK
10/11 SMCLK
- B5-0: **FNx**，定义时钟源信号的分频因子
为FNx值加1。例如：FNx=00h，分频因子为1；当FNx=03Fh，分频因子为64

Flash存储器控制寄存器FCTL3



- b15-8: **FWKEYx, FCTLx安全键值**
读操作为096h, 写操作为0A5h, 写入错误将产生PUC。
- b7: **FAIL, 操作失败**
在时序发生器输入时钟失效或者Flash操作停止时 (EEIEX=1) 该位置位, 需用软件复位。
0 操作成功
1 操作失败
- b6: **LOCKA, Segment A 封锁键**
输入1值改变状态, 输入0值无影响。
0 Segment A未封锁, 在集中擦除操作时, 所有信息存储器均被擦除
1 Segment A封锁, 在集中擦除操作时, 所有信息存储器均被保护起来
- b5: **EMEX, 紧急退出键**
0 无作用
1 产生紧急退出
- b4: **LOCK, 封锁键**
在写入和擦除操作时, 不对Flash存储器进行封锁。LOCK位可以在单字节/单字写入或擦除操作的任意时刻置位, 且不影响操作的正常进行。在块写入模式中, 如果BLKWRT=WAIT=1时LOCK位置位, BLKWRT和WAIT位将复位, 但操作仍正常进行。
0 未封锁
1 封锁
- b3: **WAIT, 等待键**
用于指示Flash存储器可以进行写入操作。
0 Flash存储器未做好准备进行下一次单字节/单字的写入操作
1 Flash存储器已做好准备进行下一次单字节/单字的写入操作
- b2: **ACCVIFG, 非法访问中断标志**
必须用软件清除。
0 无待处理的中断
1 有待处理的中断
- b1: **KEYV, 安全键值错**
该位用于指示写入FCTLx的安全键值错误, 同时产生PUC。KEYV必须用软件复位。
0 FCTLx的安全键值正确
1 FCTLx的安全键值错误
- b0: **BUSY**
该位用于指示Flash时序发生器的状态
0 不忙
1 忙

中断使能寄存器IE1

7	6	5	4	3	2	1	0
		ACCVIE					
rw-0							

- b7-6,b4-0 这些位用于其他模块, 请参看具体的芯片数据手册
- b5 **Flash存储器非法访问中断使能位**
该位用于使能ACCVIFG。建议使用在设置和清除该位时用指令: **BIS.B** 或 **BIC.B**,

优于使用MOV.B或CLR.B。



MSP430F22x4 开发板

专业提供 MSP430 单片机开发工具