MSP430 Flash Memories

Aleksandar Milenkovic

Electrical and Computer Engineering

The University of Alabama in Huntsville

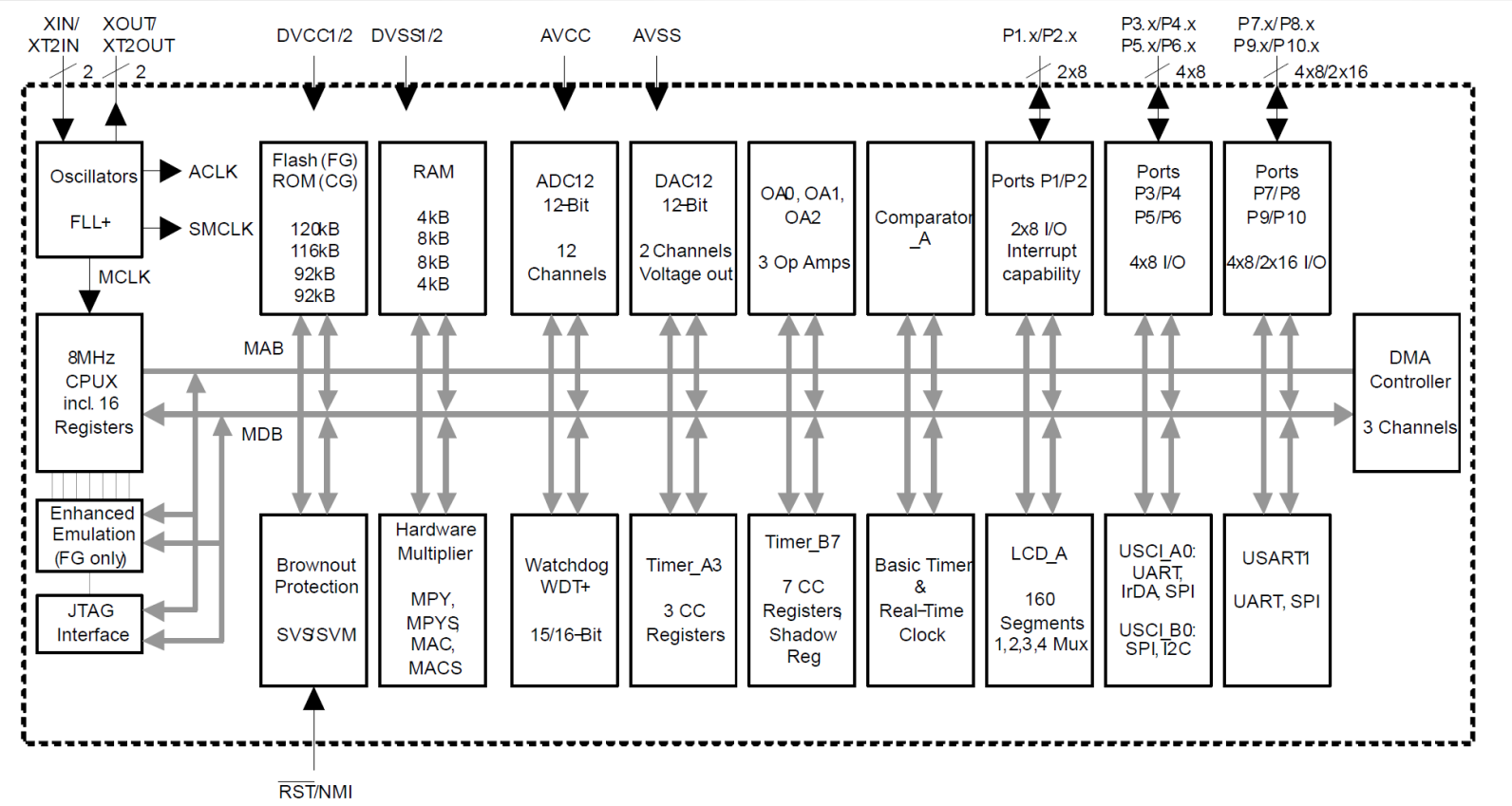
Web: http://www.ece.uah.edu/~milenka

Email: milenka@uah.edu

Outline

- Motivation
- Background
  - MSP430 Flash Memory
  - Cornell Work

MSP430FG4618: Functional Block Diagram

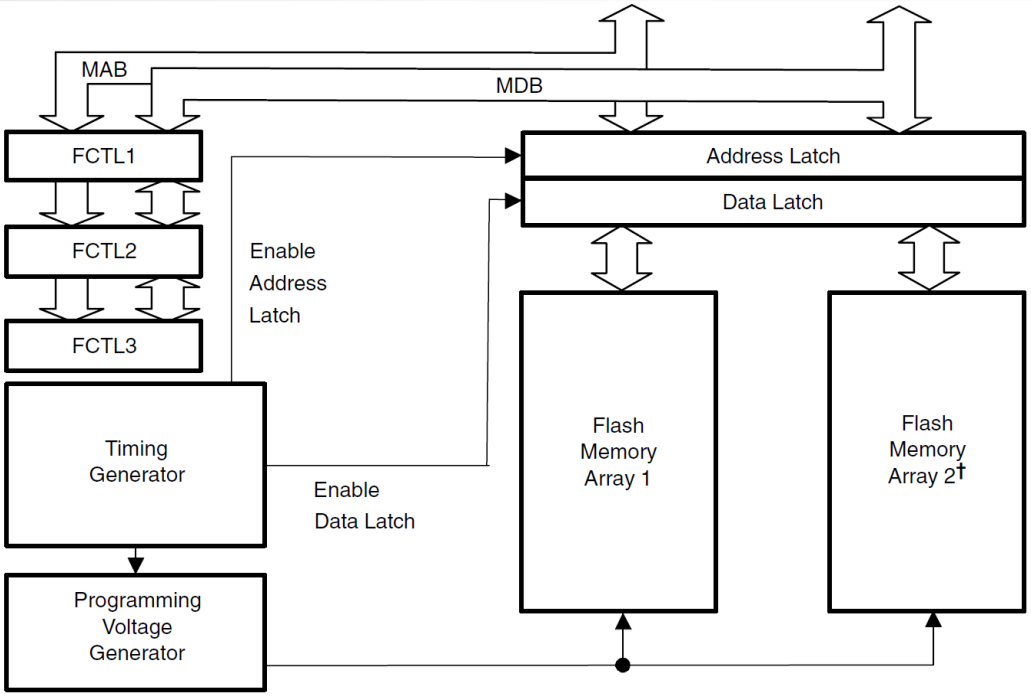


MSP430 Flash Memory Introduction

- Default view of Flash memory is Read Only
  - Contains code and data constants
- In-system programmable Flash memory
  - Read/write into the Flash memory from the running program
- But writing into the Flash requires a special interface
- MSP430 flash memory features
  - Bit-, byte-, and word-addressable and programmable
  - Internal programming voltage generation
  - Segment erase and mass erase

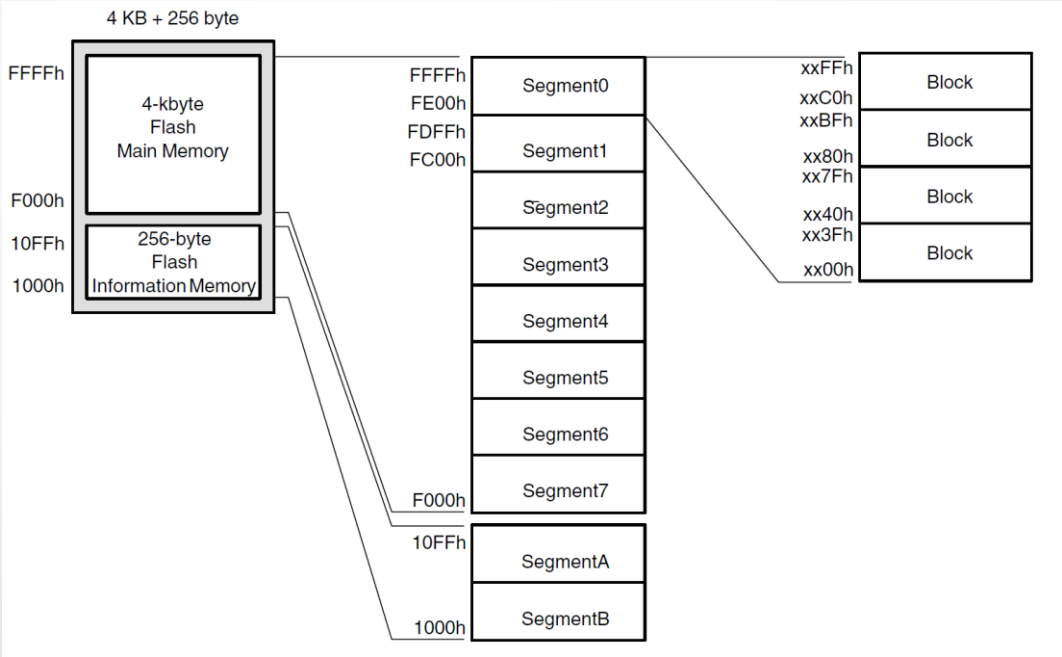
MSP430 Flash Memory Block Diagram

- Flash memory arrays: Information, Main
- Control registers
- Timing and Voltage generators



Flash Memory Segmentation

- Two partitions:
  Main memory + Information Memory
- Segment:
  smallest unit that
  can be erased
- Information Mem:
  2 128-byte segments
- Main Mem:
  n 512-byte segments



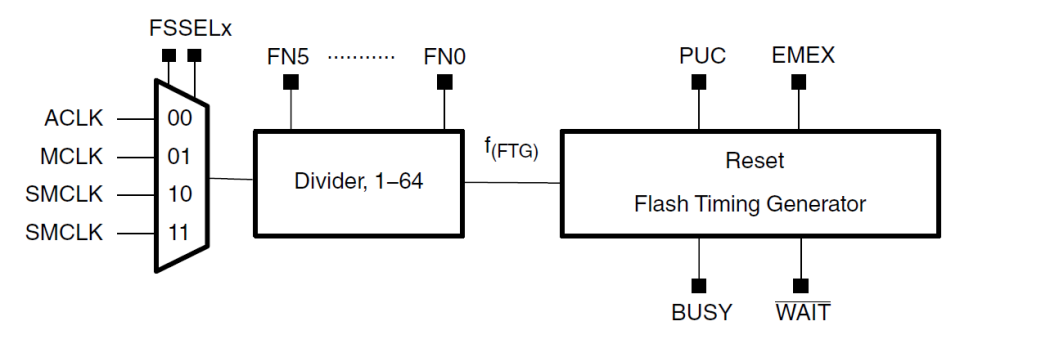
- Segments are divided into blocks (64-byte)

Flash Memory Operation

- Default is read mode: cannot be written to or erased, voltage generator is off, memory operates like ROM
- CPU can program its own flash memory (ISP – In System Programmable)
  - Byte/word write
  - Block write
  - Segment erase
  - Mass erase (all main memory segments)
  - All erase (all segments)
- Reading or writing to flash memory while it is programmed is prohibited
- If CPU execution is required, the executed code must be in RAM

Flash Timing Generator

- Write and erase are controlled by the flash timing generator
- The $f_{(FTG)}$ must be in range from ~257 KHz to ~ 476 KHz
- Can be sourced by ACLK, SMCLK, or MCLK; FN bits specify divisor
- If $f_{(FTG)}$ deviates from the specification, the result of the write or erase my be unpredictable (? Can we use this)



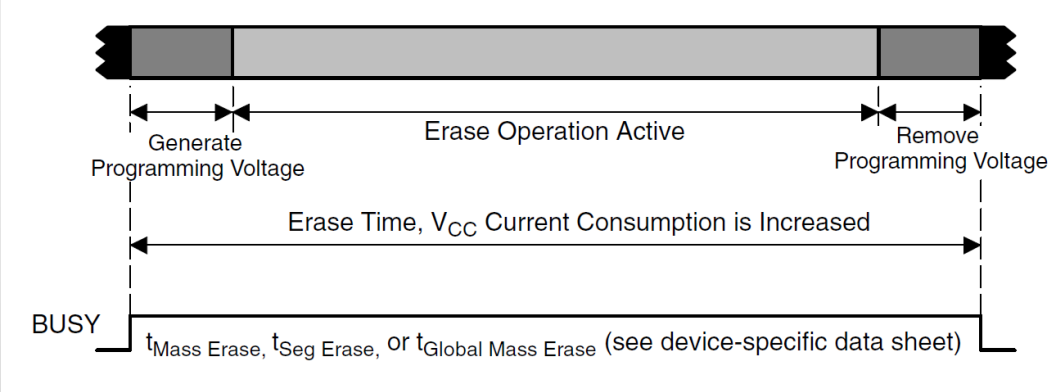
Erasing Flash Memory

- Erased level of a flash memory bit is 1
- Each bit can be programmed from 1 to 0, but to reprogram from 0 to 1 require an erase cycle
- Erase is trigged by a dummy write into the address range
- Erase modes on MSP430FG461x

GMERAS	MERAS	ERASE	Erase Mode
X	0	1	Segment erase
0	1	0	Mass erase (all main memory segments of selected memory array)
0	1	1	Erase all flash memory (main and information segments of selected memory array)
1	1	0	Global mass erase (all main memory segments of both memory arrays)
1	1	1	Erase main memory and information segments of both memory arrays

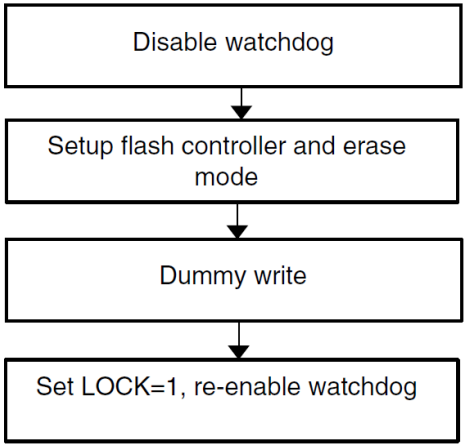
Erase Cycle Timing

- BUSY bit is set immediately after the dummy write and remains set during the erase cycle
- BUSY, GMERAS, MERAS, and ERASE are automatically cleared when the cycle completes



Initiating Erase From Flash

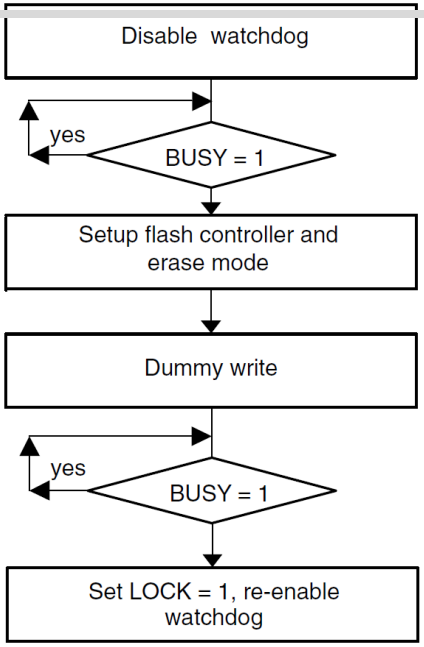
- Flash segment erase is initiated from within flash memory, all timing is controlled by the flash controller, CPU is held while the erase cycle completes




```
; Segment Erase from flash. 514 kHz < SMCLK < 952 kHz
; Assumes ACCVIE = NMIIE = OFIE = 0.
   MOV    #WDTPW+WDTHOLD,&WDTCTL    ; Disable WDT
   MOV    #FWKEY+FSSEL1+FN0,&FCTL2  ; SMCLK/2
   MOV    #FWKEY,&FCTL3             ; Clear LOCK
   MOV    #FWKEY+ERASE,&FCTL1       ; Enable segment erase
   CLR    &0FC10h                   ; Dummy write, erase S1
   MOV    #FWKEY+LOCK,&FCTL3        ; Done, set LOCK
   ...                              ; Re-enable WDT?
```

Initiating Erase From RAM

- CPU can continue to execute from RAM
- Must poll BUSY bit to determine the end of the erase cycle before any flash memory access
- If access occurs while BUSY=1, an access violation bit is set (ACCVIFG), and the erase cycle is unpredictable



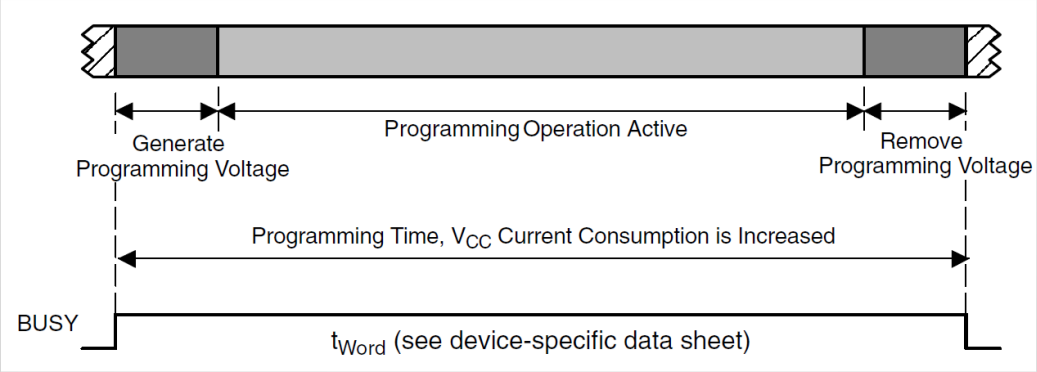

```
; Segment Erase from RAM. 514 kHz < SMCLK < 952 kHz
; Assumes ACCVIE = NMIIE = OFIE = 0.
    MOV    #WDTPW+WDTHOLD,&WDTCTL    ; Disable WDT
L1  BIT    #BUSY,&FCTL3              ; Test BUSY
    JNZ    L1                        ; Loop while busy
    MOV    #FWKEY+FSSEL1+FN0,&FCTL2  ; SMCLK/2
    MOV    #FWKEY,&FCTL3             ; Clear LOCK
    MOV    #FWKEY+ERASE,&FCTL1       ; Enable erase
    CLR    &0FC10h                   ; Dummy write, erase S1
L2  BIT    #BUSY,&FCTL3              ; Test BUSY
    JNZ    L2                        ; Loop while busy
    MOV    #FWKEY+LOCK,&FCTL3        ; Done, set LOCK
    ...                              ; Re-enable WDT?
```

Writing Flash Memory

- Write modes: WRT (byte/word) write, BLKWRT (block write)
- Both use individual write instructions, but using block write is approximately twice as fast as byte/word write, because voltage generator remains on for the complete block write
- A flash word must not be written more than twice between erasures (otherwise damage may occur)

Byte/Word Write

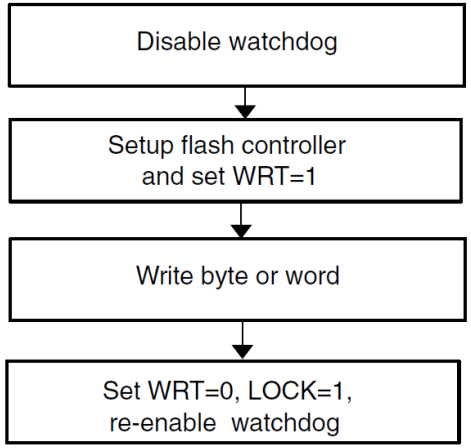
- Can be initiated from within flash memory or from RAM
  - flash: CPU is held while the write completes
  - RAM: CPU continues execution, but must wait for BUSY to go down before accessing flash memory
- Timing cycle



Cumulative Programming Time

- In byte/word mode, the internally generated programming voltage is applied to the complete 64-byte block each time a byte or word is written for (t_{WORD} – 3) f_{FTG} cycles
- With each byte or word write, the amount of time the block is subjected to the programming voltage accumulates. The cumulative programming time, t_{CPT}, must not be exceeded for any block.
- If the cumulative programming time is met, the block must be erased before performing any further writes to any address within the block

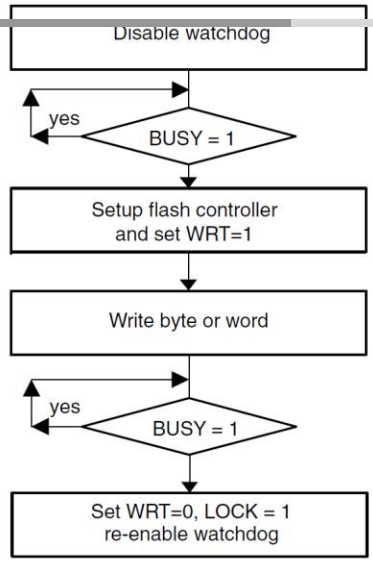
Initiating Byte/Word Write From Flash




```
; Byte/word write from flash. 514 kHz < SMCLK < 952 kHz
; Assumes 0FF1Eh is already erased
; Assumes ACCVIE = NMIIE = OFIE = 0.
   MOV    #WDTPW+WDTHOLD,&WDTCTL    ; Disable WDT
   MOV    #FWKEY+FSSEL1+FN0,&FCTL2  ; SMCLK/2
   MOV    #FWKEY,&FCTL3             ; Clear LOCK
   MOV    #FWKEY+WRT,&FCTL1         ; Enable write
   MOV    #0123h,&0FF1Eh            ; 0123h    -> 0FF1Eh
   MOV    #FWKEY,&FCTL1             ; Done. Clear WRT
   MOV    #FWKEY+LOCK,&FCTL3        ; Set LOCK
   ...                              ; Re-enable WDT?
```

Initiating a Byte/Word Write From RAM

- CPU can continue to execute from RAM
- Must poll BUSY bit to determine the end of the write cycle before any flash memory access

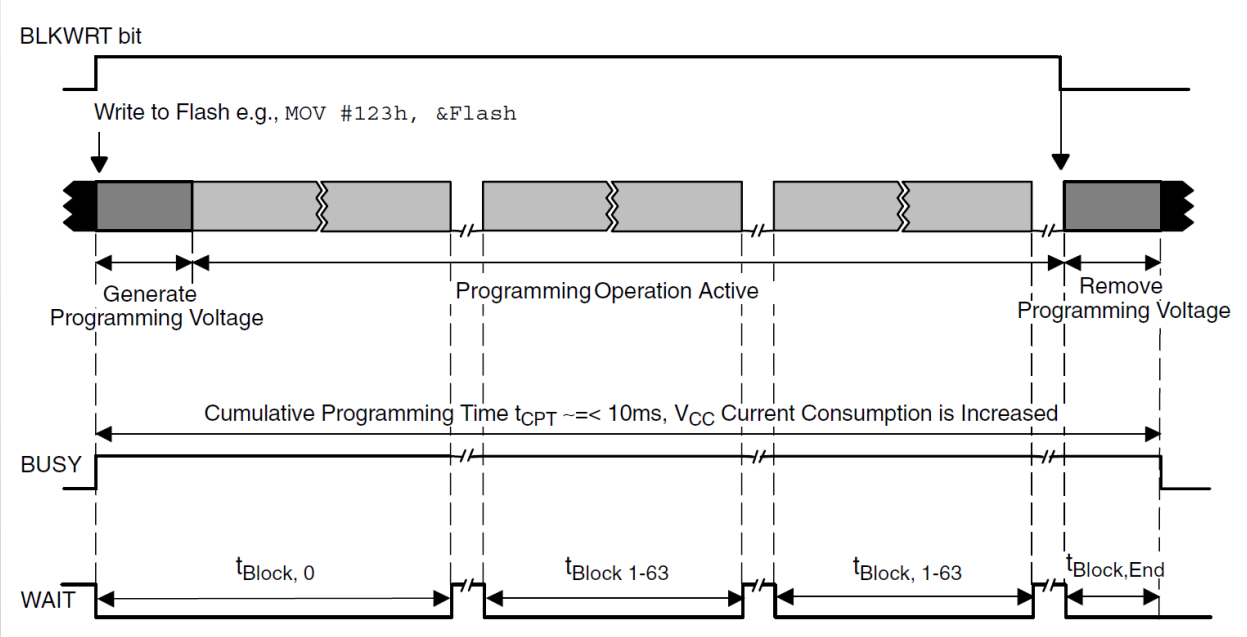



```
; Byte/word write from RAM. 514 kHz < SMCLK < 952 kHz
; Assumes 0FF1Eh is already erased
; Assumes ACCVIE = NMIIE = OFIE = 0.
    MOV   #WDTPW+WDTHOLD,&WDTCTL    ; Disable WDT
L1  BIT   #BUSY,&FCTL3              ; Test BUSY
    JNZ   L1                        ; Loop while busy
    MOV   #FWKEY+FSSEL1+FN0,&FCTL2  ; SMCLK/2
    MOV   #FWKEY,&FCTL3             ; Clear LOCK
    MOV   #FWKEY+WRT,&FCTL1         ; Enable write
    MOV   #0123h,&0FF1Eh            ; 0123h -> 0FF1Eh
L2  BIT   #BUSY,&FCTL3              ; Test BUSY
    JNZ   L2                        ; Loop while busy
    MOV   #FWKEY,&FCTL1             ; Clear WRT
    MOV   #FWKEY+LOCK,&FCTL3        ; Set LOCK
    ...                             ; Re-enable WDT?
```

Block Write

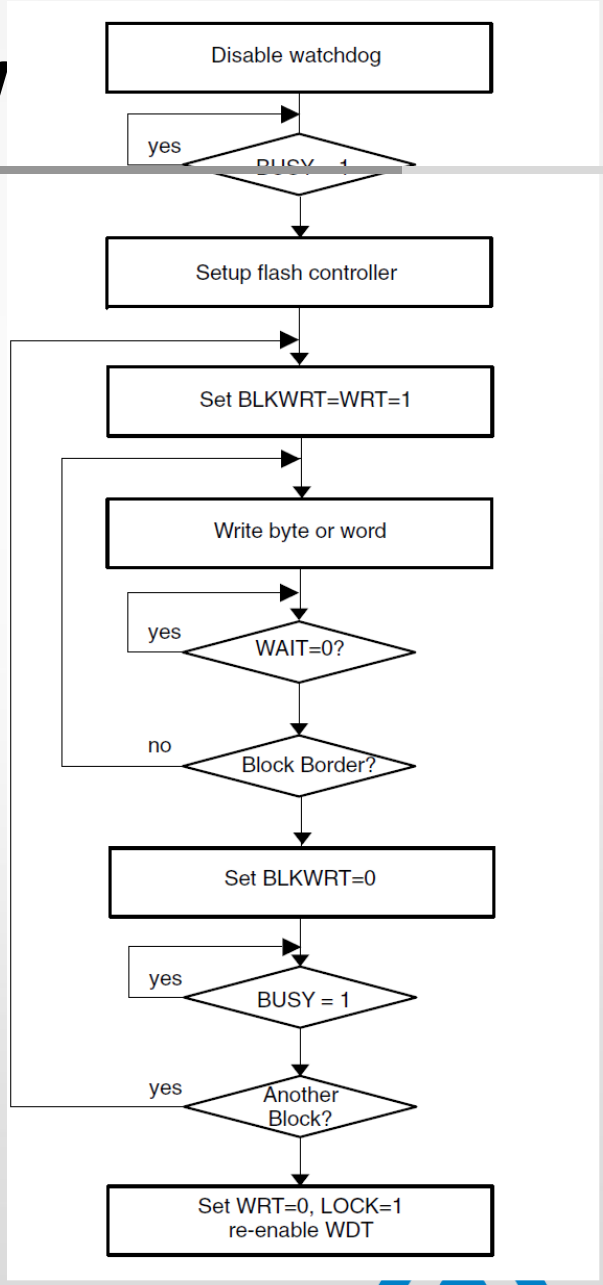
- Accelerates the flash write process when many sequential bytes or words need to be programmed
- The flash programming voltage remains on for the duration n writing the 64-byte block
- The cumulative programming time must not be exceeded
- Must be initiated from RAM
- BUSY bit remains set throughout duration of block write
- WAIT must be checked between writing each byte/word in the block
- BLKWRT must be cleared after the current block is complete

Block Write Cycle Timing



Block Write Flow

```
; Write one block starting at 0F000h.
; Must be executed from RAM, Assumes Flash is already erased.
; 514 kHz < SMCLK < 952 kHz
; Assumes ACCVIE = NMIIE = OFIE = 0.
    MOV    #32,R5                    ; Use as write counter
    MOV    #0F000h,R6                ; Write pointer
    MOV    #WDTPW+WDTHOLD,&WDTCTL    ; Disable WDT
L1  BIT    #BUSY,&FCTL3              ; Test BUSY
    JNZ    L1                        ; Loop while busy
    MOV    #FWKEY+FSSEL1+FN0,&FCTL2  ; SMCLK/2
    MOV    #FWKEY,&FCTL3             ; Clear LOCK
    MOV    #FWKEY+BLKWRT+WRT,&FCTL1  ; Enable block write
L2  MOV    Write_Value,0(R6)         ; Write location
L3  BIT    #WAIT,&FCTL3              ; Test WAIT
    JZ     L3                        ; Loop while WAIT=0
    INCD   R6                        ; Point to next word
    DEC    R5                        ; Decrement write counter
    JNZ    L2                        ; End of block?
    MOV    #FWKEY,&FCTL1             ; Clear WRT,BLKWRT
L4  BIT    #BUSY,&FCTL3              ; Test BUSY
    JNZ    L4                        ; Loop while busy
    MOV    #FWKEY+LOCK,&FCTL3        ; Set LOCK
    ...                              ; Re-enable WDT if needed
```



Stopping a Write or Erase Cycle

- Any write or erase cycle can be stopped before its normal completion by setting the emergency exit bit EMEX
- Setting EMEX bit stops the active operation immediately and stops the flash controller
- All flash operations cease, the flash returns to read mode, all bits in the FCTL1 register are reset, and the result of operation is unpredictable

Flash Memory Specs

electrical characteristics over recommended operating free-air temperature (unless otherwise noted)

flash memory (MSP430FG461x devices only)

PARAMETER		TEST CONDITIONS	V_{CC}	MIN	TYP	MAX	UNIT
$V_{CC(PGM/ERASE)}$	Program and Erase supply voltage			2.7		3.6	V
f_{FTG}	Flash Timing Generator frequency			257		476	kHz
I_{PGM}	Supply current from DV_{CC} during program		2.7 V/ 3.6 V		3	5	mA
I_{ERASE}	Supply current from DV_{CC} during erase	See Note 3	2.7 V/ 3.6 V		3	7	mA
$I_{GMERASE}$	Supply current from DV_{CC} during global mass erase	See Note 4	2.7 V/ 3.6 V		6	14	mA
t_{CPT}	Cumulative program time	See Note 1	2.7 V/ 3.6 V			10	ms
$t_{CMErase}$	Cumulative mass erase time		2.7 V/ 3.6 V	20			ms
	Program/Erase endurance			10^4	10^5		cycles
$t_{Retention}$	Data retention duration	T_J = 25°C		100			years
t_{Word}	Word or byte program time	See Note 2			30		t_{FTG}
$t_{Block, 0}$	Block program time for 1st byte or word				25		
$t_{Block, 1-63}$	Block program time for each additional byte or word				18		
$t_{Block, End}$	Block program end-sequence wait time				6		
$t_{Mass Erase}$	Mass erase time				10593		
$t_{Global Mass Erase}$	Global mass erase time				10593		
$t_{Seg Erase}$	Segment erase time				4819		

NOTES:
1. The cumulative program time must not be exceeded during a block-write operation. This parameter is only relevant if the block write feature is used.
2. These values are hardwired into the Flash Controller's state machine ($t_{FTG} = 1/f_{FTG}$).
3. Lower 64-KB or upper 64-KB Flash memory erased.
4. All Flash memory erased.