ST10F163

ST Embedded Algorithm Kernel (STEAK) for Flash Programming and Erasing

APPLICATION NOTE

The ST10F163 has 128Kbytes of on-chip Flash memory. This application note describes how to use the ST Embedded Algorithm Kernel (STEAK™) to program, erase and read the status of the ST10F163 Flash memory.

For Flash programming or erasing, GPRs are loaded with the address and data to be programed, or sector to be erased. STEAK uses embedded routines, which check the validity of the programmed parameters, decode and then execute the programming or erase command. During operation, the STEAK routines carry out checks and retries to verify proper cell programming or erasing. When an error occurs, STEAK returns an error-code which identifies the cause of the error.

This application note describes the three stages of running the STEAK - loading the parameters, initiating the Unlock Sequence and reading the return values. Tables of STEAK parameters, code and error code definition, and return values are given in *Section 1*.

A more detailed description of single word, double word and block programming routines is given in *Section 2*. The causes of error, and possible solutions are provided for each error code. Examples of programming code and the PRESTO programming algorithm are given.

Similarly, for Flash erase, a detailed description including analysis of error codes, a code example and the PRESTO erase algorithm is in *Section 3*.

The STEAK Read Status Command is used to read the STEAK revision and is described in *Section 4.1*.

Table of Contents

1 Running the STEAK - 4

2 Flash Programming - 7

2.1 Programming routines - 7

2.1.1 Single word programming -7

2.1.2 Double word programming -8

2.1.3 Block programming -8

2.1.4 PRESTO programming algorithm - - - - - - - - - - - - - - - - - - - 10

2.2 Error-codes for STEAK programming - - - - - - - - - - - - - - 11

2.2.1 Programming error-code 01: - 11

2.2.2 Programming error-code 02: - 12

2.2.3 Programming error-code 03: - 12

2.2.4 Programming error-code 04: - 12

2.2.5 Programming error-code 05: - 13

2.2.6 Programming error-code 06: - 13

2.2.7 Programming error-code 09: - 13

2.2.8 Programming error-code 0A: - 13

2.2.9 Programming error-code FF: - 13

2.3 Code examples - 14

2.3.1 Programming a single word using STEAK - - - - - - - - - - - - - - - 14

2.3.2 Programming multiple words using STEAK - - - - - - - - - - - - - - 16

3 Flash Erasing - 17

3.1 Erase routine - 17

3.1.1 PRESTO erase algorithm - 18

3.2 Error-codes for Flash erasing - - - - - - - - - - - - - - - - - - 19

3.2.1 Erase error-code 01: - 19

3.2.2 Erase error-code 02: - 19

3.2.3 Erase error-code 03: - 19

3.2.4 Erase error-code 05: - 19

3.2.5 Erase error-code 06: - 19

3.2.6 Erase error-code 07: - 19

3.2.7 Erase error-code 08: - 20

3.2.8 Erase error-code FF: - 20

3.3 Code example - 21

4 STEAK Read Status - 22
4.1 Reading the STEAK revision - - - - - - - - - - - - - - - - - - - 22
4.2 Error-codes for STEAK read status - - - - - - - - - - - - - - - - 22
4.2.1 Read status error-code 01: - 22
4.2.2 Read status error-code 02: - 22
4.2.3 Read status error-code 05: - 22
4.2.4 Read status error-code 06: - 22
4.2.5 Read status error-code FF: - 23
4.3 Code example - 23

1 Running the STEAK

There are three stages to running the STEAK:

1 Load the registers R0 to R4 with the STEAK command, the address and data to be programed, or sector to be erased.

Table 1 gives the STEAK parameters for each type of Flash programming/erasing operation. Table 2 defines the codes used in Table 1.

The Flash programing routine for each of the three Flash programming types (single word programming - double word programming and block programming) is described in “Flash Programming” on page 7.

2 Initiate the Unlock Sequence.

The STEAK routine is stored in the Testflash. In normal operation, the Testflash is not mapped in the Flash address area and is, therefore, “non-visible” to the user. Because of this, an Unlock Sequence is required to activate the STEAK routine.

The Unlock Sequence is composed of two consecutive writes to an even address in the Flash active address space - the first write has direct addressing mode (MOV mem, Rwn) - the second write has indirect addressing mode (MOV [Rwm], Rwn). Rwn can be any unused word-GPR (R6 to R15) loaded with a value resulting in the same even address as “mem”.

3 Read the return values in R0.

When the embedded programming/erasing algorithm returns to trigger point, return values are given in R0. Table 3 gives the error-code definitions, Table 4 gives the return values in each register for each type of Flash programming/erasing command.

Note *The Flash Embedded Presto Algorithms require at least 50 words on the Internal System Stack. The program verifies that there is enough free space on the System Stack, before performing a programming or erasing operation.The MDH, MDL and MDC register content are modified.*

Code examples for programming and erasing the STEAK are given in *Section 2.3* and *Section 3.1* respectively.

COMMAND	R0	R1	R2	R3	R4
Single word programming	55Ash	AddOff	W	nu	2TCL
Double word programming	DD4sh	AddOff	DWL	DWH	2TCL
Block programming	AA5sh	BegAddOff	EndAddOff	SourceAddr	2TCL
Sector Erasing	EEEEh	5555h	Bnk	Bnk	2TCL
Read Status	7777h	nu	nu	nu	2TCL

Table 1 Steak parameters

Abbreviation	Definition
s	Segment of the target flash memory cell
AddOff	Segment Offset of the target flash memory cell which must be even an value (word-aligned address).
W	Data (word) to be written in flash.
DWL,DWH	Data (double word, DHL = low word, DWH = high word to be written in Flash,
BegAddOff	Segment Offset of the FIRST target flash memory word to be written in a multiple programming command. This value must be even (word-aligned address)
EndAddOff	Segment Offset of the LAST Target Flash Memory word to be written in a Multiple programming command. • Must be even value (word-aligned address). • The value D = (EndAddOff - BegAddOff) must be: $0 \leq D < 16384$ (i.e. up to one page (16 kBytes) can be written in the flash with one multi-word programming command).
SourceAdd	Start address for the source data (block) to be programmed. This address uses implicitly the data paging mechanism of the CPU. SourceAdd value must respect the rules: • SourceAdd + (EndAddOff - BegAddOff) < 16384. • Page 0 and 1 can NOT be used for source data if SYSCON bit ROMS1 = '1' **Note**: source data can be located in flash (In pages 0, 1, 6, 7, 8, 9, 10 or 11 if bit ROMS1 = '0', or in pages 4, 5, 6, 7, 8, 9, 10 or 11 if bit ROMS1 = '1'.
Bnk	Number of the Bank to be erased. Note that for security, R2 and R3 must hold the same value.
2TCL	CPU clock period in nseconds (e.g. R4 = 40d means CPU frequency is 25MHz).

Table 2 Programming code definition

ERROR CODE	MEANING
00h	Operation was successful
01h	ROMEN bit inside SYSCON is not set
02h	Vpp voltage not present
03h	Programming operation failed
04h	Address value (R1) incorrect: not in Flash address area or odd
05h	CPU period out of range (must be between 10 ns to 1000 ns)
06h	Not enough free space on system stack for proper operation
07h	Incorrect bank number (R2,R3) specified
08h	Erase operation failed
09h	Bad source address for multi-word programming command
0Ah	Bad number of words to be copied in multi-word programming command: one destination will be out of flash, or one source operand will be out of the source page
FFh	Unknown or bad command

Table 3 Error code definition

Programming command	R0	R1	R2	R3	R4-R15
Single or double word programming	Error code	Unchanged	Data in Flash for location Segment+Segment Offset (R0.[3:0] with R1)	Data in Flash for location Segment+Segment Offset +2 (R0[3:0] with R1+2)	Unchanged
Block programming	Error code	The last segment offset address of the last written word in flash (failing flash address if R0 is not equal to zero)	Undefined		Unchanged
Erasing	Error code	Undefined			Unchanged
After status read	Error code	Flash embedded rev 0100h for STEAK rev1 MSByte = major release LSByte = minor revision	Circuit identifiers: R2 = #0787h R3 = #0101h for this device		Unchanged

Table 4 Return values for each programming command

2 Flash Programming

There are 3 types of Flash programming routine:

- Single word programming -16 bits per STEAK call.
- Double word programming - 32 bits per STEAK call.
- Block programming (up to one page) -16Kbytes per STEAK call.

When performing a programming command, the Embedded Algorithm Kernel automatically times the program pulse widths (taking in account the CPU period provided as a parameter by the user) and verifies proper cell programming.

Each of the Flash programming routines uses the same PRESTO programming algorithm to program and verify the Flash. The PRESTO programming algorithm is described in Figure 1 on page 10.

2.1 Programming routines

2.1.1 Single word programming

This routine is run when the Single Word Programming Command value (55Ash) is loaded inside R0 register before executing the Unlock Sequence.

The CPU calls the Embedded Kernel routine and performs the following actions:

- Return to user's appli. with R0 = 01 if the Flash is NOT enabled (bit ROMEN of SYSCON not set).
- Return to user's appli. with R0 = 05 if CPUPER parameter (value of R4 register) is > 500ns (F_{CPU} < 2 Mhz) or < 30 ns (F_{CPU} > 33 Mhz).
- Return to user's appli. with R0 = 06 if there are NOT enough free words on System Stack for proper STEAK execution, i.e. if (SP - STKOV < 100 bytes).
- Save internally used registers on System Stack.
- wait (~ 10 µs) for internal stabilization of Vpp voltage.
- Return to user's appli. if Vpp voltage is internally sensed < ~11 volts.
- Return to user's appli. with R0 = 04 if the address formed by R0 (segment address in the 4 least significant bits) and R1 (segment offset address) is NOT a word address (i.e. even value), or is NOT inside the address area of the flash (taking in account the current mapping of the Flash, i.e. setting of bit ROMS1 in SYSCON).

- Call of 'PRESTO programming' algorithm.
- Return to user's appli. with error-code of PRESTO algorithm returned in register R0.

2.1.2 Double word programming

This routine is run when the Double Word Programming Command value (DD4sh) value is loaded inside R0 register before executing the Unlock Sequence.

The CPU calls the Embedded Kernel routine and performs the following actions:

- Return to user's appli. with R0 = 01 if the Flash is NOT enabled (bit ROMEN of SYSCON not set).
- Return to user's appli. with R0 = 05 if CPUPER parameter (value of R4 register) is >500ns (F_{CPU} < 2 Mhz) or < 30 ns (F_{CPU} > 33 Mhz).
- Return to user's appli. with R0 = 06 if there is NOT enough free words on System Stack for proper STEAK execution, i.e. if (SP - STKOV < 100 bytes).
- Save internally used registers on System Stack.
- wait (~ 10 µs) for internal stabilization of Vpp voltage.
- Return to user's appli. if Vpp voltage is internally sensed < ~11 volts.
- Return to user's appli. with R0 = 04 if address formed by R0 (segment address in the 4 least significant bits) and R1 (segment offset address) is NOT a double-word address (i.e. multiple of 4 value), or is NOT inside the address area of the flash (taking in account the current mapping of the Flash, i.e setting of bit ROMS1 in SYSCON).
- Call 'PRESTO programming' algorithm.
- Return to user's appli. with error-code of PRESTO algorithm returned in register R0.

2.1.3 Block programming

This routine is run when the Block Programming Command value (AA5sh) value is loaded inside R0 register before executing the Unlock Sequence.

The CPU calls the Embedded Kernel routine and performs the following actions:

- Return to user's appli. with R0 = 01 if the Flash is NOT enabled (bit ROMEN of SYSCON not set).
- Return to user's appli. with R0 = 05 if CPUPER parameter (value of R4 register) is >500ns (F_{CPU} < 2 Mhz) or < 30 ns (F_{CPU} > 33 Mhz).

- Return to user's appli. with R0 = 06 if there are NOT enough free words on System Stack for proper STEAK execution, i.e. if (SP - STKOV < 100 bytes).
- Save internally used registers on System Stack.
- Wait (~ 10 µs) for internal stabilization of Vpp voltage.
- Return to user's appli. if Vpp voltage is internally sensed < ~11 volts.
- Return to user's appli. with R0 = 04 if the first flash address to be programmed is formed by R0 (segment address in the 4 least significant bits), and R1 (segment offset address) is NOT a word address (i.e. even value), or is NOT inside the address area of the flash (taking in account the current mapping of the Flash, i.e setting of bit ROMS1 in SYSCON).
- Return to user's appli. with R0 = 04 if last Flash address to be programmed is formed by R0 (segment address in the 4 least significant bits), and R2 (segment offset address) is NOT a word address (i.e. even value), or is NOT inside the address area of the flash (taking in account the current mapping of the Flash, i.e setting of bit ROMS1 in SYSCON).
- Return to user's appli. with R0 = 04 if the last Flash address to be programmed (in R2 register) is less than first address (in R1 register).
- Return to user's appli. with R0 = 0Ah if the number of words to be programmed exceeds a page size (8192 words, i.e. 16 Kbytes).
- Return to user's appli. with R0 = 09h if the source page offset (in R3 register) + 2* number of words to be programmed, exceeds the source page boundary.
- Return to user's appli. with R0 = 09h if the source address is inside page 0 and 1, while the flash is mapped in segment 1.
- For the number of words to be programmed, call of 'PRESTO programming' algorithm with data read at source address (in R3), destination is address defined by R0/R1 register then increment source address (R3) by 2, and destination address by 2 (R0/R1).
- If PRESTO programming algorithm returns an error, return to user's appli. with error-code of PRESTO algorithm returned in register R0, else program next word.

2.1.4 PRESTO programming algorithm

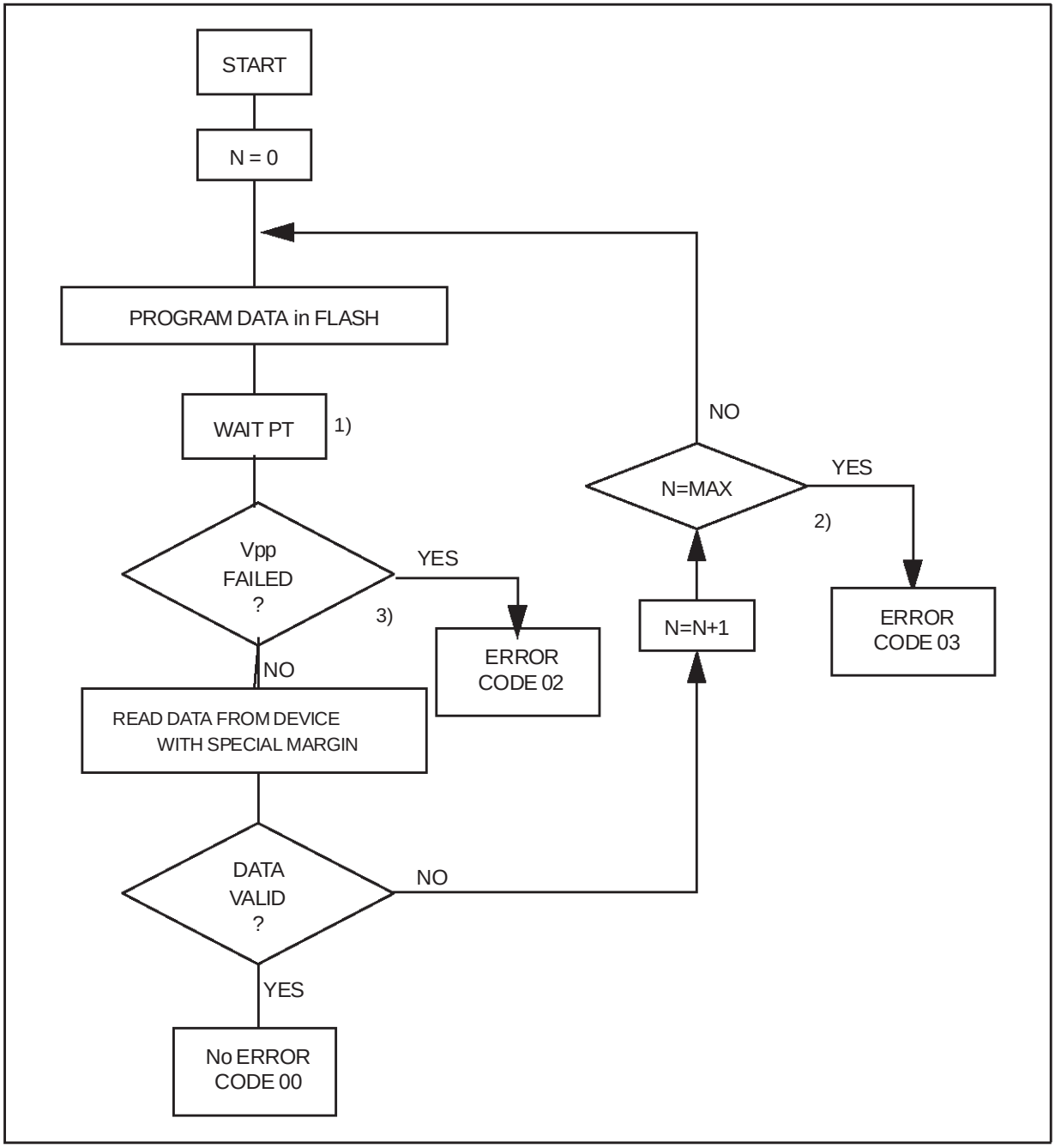

Figure 1 Presto Programming Algorithm

1) PT = 256 x CPU period [ns]
2) MAX = 80 µs / PT = 313 [ns] / CPU period [ns]
3) Vpp failed if Vpp drops below 11 V during the programming pulse.

2.2 Error-codes for STEAK programming

The STEAK programming command routines perform many checks and retries. When an error occurs, the STEAK returns an error-code in R0 which can be used to identify the cause of the problem. This section gives a correction guide for each error-code:

2.2.1 Programming error-code 01: (ROMEN bit inside SYSCON is not set)

- Cause: The Flash memory is not enabled, i.e. bit ROMEN in the SYSCON register is not set.
- Solution: Set bit ROMEN in the SYSCON register and then recall the STEAK programming command.

  When the Flash is enabled, pay attention to the application mapping versus the Flash mapping.

  Reload the DPPx values to take account of the new CPU memory mapping (even if the DPPx registers are loaded with values that will not be inside the Flash address area).

  Perform a jump segmented to the next code instruction (needed to also reload IP register correctly).

  The following code gives an example of this solution:

```
; Flash is being enabled, ie. bit ROMEN will be set in SYSCON.
BSET    ROMEN           ; Enable the Flash (default mapping is in
                        ; segment 0, so this code must neither be run in
                        ; address range 00'0000h - 00'7FFFh, nor in
                        ; 01'8000h - 02'FFFFh
MOV     DPP0, #00h      ; Reload DPPx registers (here with their default
MOV     DPP1, #01h      ; values), requested for the CPU to take into
                        ; account the new memory mapping
MOV     DPP2, #02h
MOV     DPP3, #03h
JMPS    NextInstruction ; Performs a segmented jump to next code
                        ; intruction. Needed for reloading CS and IP to
                        ; take into account the new memory mapping
Next Intruction:        ; Destination of the segmented jump.
(next instructions.......)
```

2.2.2 Programming error-code 02: Vpp failed

- Cause: The internal Flash Vpp voltage is not stabilized before calling the PRESTO programming algorithm, or the Vpp voltage drops below 11 V during a programming pulse when the PRESTO programming algorithm is running. This failure can be caused by a Vpp supply glitch.
- Solution: Recall the STEAK programming command if you suspect a glitch on the Vpp supply. See “Programming a single word using STEAK” on page 14 for a code example.

2.2.3 Programming error-code 03: Programming failed

- Cause: The PRESTO programming algorithm did not successfully read the correct data from the Flash after a maximum number of retries. This occurs when:
  - At least one bit of the DATA to be programmed is at ‘1’ while the correspondent bit in the Flash matrix is already at ‘0’ (i.e. a bit of the Flash is already programmed, but the user wants to program a ‘virgin’ bit).
  - At least one bit of the DATA to be programmed at ‘0’ can not be actually programmed in the flash matrix (i.e. the flash is dead).
- Solution: Read the Flash content for the failing address(es), and compare with the data (word or double word) to be programmed:
  - If a bit of data at ‘1’ is already at ‘0’ in the Flash, then the Flash must be erased (at least, the current bank).
  - If a bit of data at ‘0’ is read at ‘1’ in the Flash, then the Flash is out of order (the part must be replaced!).

Note *Recalling the STEAK programming command after this kind of error can give correct programming which will then be impossible to erase.*

2.2.4 Programming error-code 04: Address value incorrect

- Cause: The destination address, formed by the 4 lower bits of the R0 register (segment number) and R1 (16-bit offset address in segment) does not correspond to an address in the Flash.
- Solution: Check the current memory mapping of the Flash (setting of bit ROMS1 in the SYSCON register). This is especially important when programming the address range 00’0000h to 00’7FFFEh or 01’0000h to 01’7FFEh.

2.2.5 Programming error-code 05: CPU period out of range: must be between 30-500 ns

- Cause: The CPU period given in R4 is out of valid range. This parameter is used by the STEAK to time some waiting loops inside the STEAK routines.
- Solution: Provide a valid value in R4 before calling STEAK command.

Note *It is very important to use the correct value for the CPU period - otherwise, incorrect time-outs are generated and programming may not be successful.*
It is NOT possible to program the Flash if f_{cpu}<2MHz or >33MHz.

2.2.6 Programming error-code 06: Not enough free space on system stack

- Cause: the STEAK programming routine needs at least 50 words on System Stack. The STEAK routine performs the following check:"if (SP - STKOV) < 100 then return error 06".
- Solution: Change the initialization setting of SYSCON (bitfield STKSZ) to provide a bigger stack size, or flush the System Stack to external memory before calling the STEAK programming command, and then restore it after STEAK return.

2.2.7 Programming error-code 09: Bad source address for multi-word programming

- Cause: The source data address for a multi-word programming command (value in R3 register) is an odd address,

  or, this address plus the number of bytes to be programmed, exceeds a page boundary,

  or, this source address is inside page 0 or 1, while the flash is mapped in segment 1.
- Solution: Ensure that the following condition is true *before* calling STEAK multi-word programming command:"(R3 AND #03FFFh) + R2 -R1 < 04000h".

2.2.8 Programming error-code 0A: Bad number of words for multi-word programming

- Cause: The number of bytes to be programmed in a multi-word programming command exceed a page size (i.e. R2 - R1 ≥ #04000h).
- Solution: Ensure that the following condition is true before calling STEAK multi-word programming command:"(R3 AND #03FFFh) + R2 - R1 < 04000h".

2.2.9 Programming error-code FF: Bad STEAK command

- Cause: R0 content is not a valid command number.
- Solution: Ensure that R0 = #055Ash or #0DD4sh or #0AA5sh *before* calling the STEAK erase routine.

2.3 Code examples

2.3.1 Programming a single word using STEAK

The following code programs a single word, then checks and reacts to error-code returned by STEAK:

```
; code hereafter assumes that flash is mapped in segment 1
; ie. bit ROMS1 = ‘1’ in SYSCON register
; Flash must also be enabled, ie. bit ROMEN = ‘1’ in SYSCON.
PrgSgl:
MOV     R0, #055A0h          ; 55Ash : Single word programming command
OR      R0, #01h             ; Selects segment 1 in flash memory
MOV     R1, #00224h          ; Address to be programmed is 01’0224h
MOV     R2, #03456h          ; Data to be programmed at 01’0224h
MOV     R4, #050d            ; 50ns is 20 MHz CPU clock frequency
MOV     R7, #08000h          ; R7 used for Flash trigger sequence
#define FCR 08000h
; Flash Unlock Sequence: consists in two consecutive writes, with the
direct addressing mode and then the indirect addressing mode. FCR must
represent an even address in the active address space of the Flash
memory, and Rwn can be any unused word GPR (R6 to R15)loaded with a
value resulting in the same even address than FCR
EXTS    #1, #2               ; flash can be mapped in segment 0 or 1
MOV     FCR, R7              ; first part
MOV     [R7], R7             ; second part
NOP                          ; WARNING: place 2 NOP operations after
NOP                          ; the Unlock sequence to avoid all
                             possible
                             ; pipeline conflict in STEAK programs
CMP     R0, #0               ; Check error-code returned by STEAK
JMP     cc_EQ, PrgDone       ; Programming OK!

CMP     R0, #2               ; Vpp failed, we can retry programming
JMP     cc_NE, CkR03
SUB     RetryCnt, #1         ; command, with a max number of retry
                             ; loaded in RetryCnt variable.
JMP     cc_NZ, PrgSgl        ; If retry counter > 0, recall STEAK
JMP     cc_UC, ErrorVpp      ; else say that Vpp if failling.

ChkR03:
```

```
; code hereafter assumes that flash is mapped in segment 1
CMP     R0, #3              ; error-code = 03?
JMP     cc_NE, ChkR04       ; No: next R0 check
                            ; Yes: Programming failed: we must check
                            ; the data versus flash content.
EXTS    #1, #1              ; read flash content using EXTended
MOV     R0, 0224h           ; instruction.
AND     R0, #03456h         ; Flash content is ANDed with data to be
                            ; programmed in flash.
CMP     R0, #03456h         ; If (Flash cont. AND Data) is NOT equal
                            ; to Data, then at least one bit of the
                            ; flash is at '0' while it is at '1' in the
JMP     cc_NE, EraseBnk     ; Data: Flash bank must be erased prior
JMP     cc_UC, DeadFlash    ; else: the flash failed to be be
                            ; programmed (i.e. at least one bit in
                            ; Flash was not able to be programmed to
                            ; '0'): the flash part is dead!

ChkR04:
CMP     R0, #4              ; Error-code = 04?
JMP     cc_NE, ChkR05       ; No: next R0 check
JMP     cc_UC, ErrorAd      ; Yes: say that address was not correct

ChkR05:
CMP     R0, #5              ; Error-code = 05?
JMP     cc_NE, ChkR06       ; No: next R0 check
JMP     cc_UC, ErrorCPUPER  ; Yes: say that CPU period was not
                            ; correct

ChkR06:
CMP     R0, #6              ; Error Code = 06?
JMP     cc_NE, BadError     ; No: say that STEAK report incorrect
                            ; error number (could not happen!)
JMP     cc_UC, ErrorSysStk  ; Yes: say that not enough free words on
                            ; System Stack.
```

2.3.2 Programming multiple words using STEAK

The following code programs a block of data. The area of Flash to be programmed is address 01'9000h to 01'9FFEh (inclusive). The source data (to be copied into flash) is located in external RAM from address 03'1000h (to 03'1FFEh, implicitly):

```
; code hereafter assumes that flash is mapped in segment 1
; i.e. bit ROMS1 = '1' in SYSCON register
; Flash must also be enabled, i.e. bit ROMEN = '1' in SYSCON.
MOV    R0, #0AA50h  ; AA5sh: Multi word programming command
OR     R0, #01h     ; Selects segment 1 in flash memory
MOV    R1, #09000h  ; First Flash Segment Offset Address*
MOV    R2, #09FFEh  ; Last Flash Segment Offset Address
MOV    R3, #09000h  ; Source data address: use DPP2 as
                    ; data page pointer
SCXT   DPP2,#0Ch    ; Source is in page 12 (0Ch): save previous
                    ; DPP2 value and load it with source page
                    ; number
MOV    R4, #050d    ; 50ns is 20 MHz CPU clock frequency
MOV    R7, #08000h  ; R7 used for Flash trigger sequence
#define FCR 08000h
EXTS   #1, #2       ; flash can be mapped in segment 0 or 1
MOV    FCR, R7      ; first part
MOV    [R7], R7     ; second part
NOP                 ; WARNING: place 2 NOP operations after
NOP                 ; the Unlock sequence to avoid all possible
                    ; pipeline conflict in STEAK programs
POP    DPP2         ; restore DPP2
; Check R0 return value
(see previous example)
```

3 Flash Erasing

3.1 Erase routine

The STEAK Erasing command erases one sector (or bank) at a time, automatically, in two steps:

1 It programs the whole bank with 0000h values for each word of the bank, using the same PRESTO programming algorithm as for STEAK programming commands,

2 It then erases the bank by using the PRESTO Erase Algorithm.

When performing an erasing command, the Embedded Algorithm Kernel automatically preprograms the bank to be erased if it is not already programmed. During erase, the Embedded Algorithm Kernel automatically times the erase pulse widths (taking in account the CPU period provided as a parameter by the user) and verifies proper cell erasing.

3.1.1 PRESTO erase algorithm

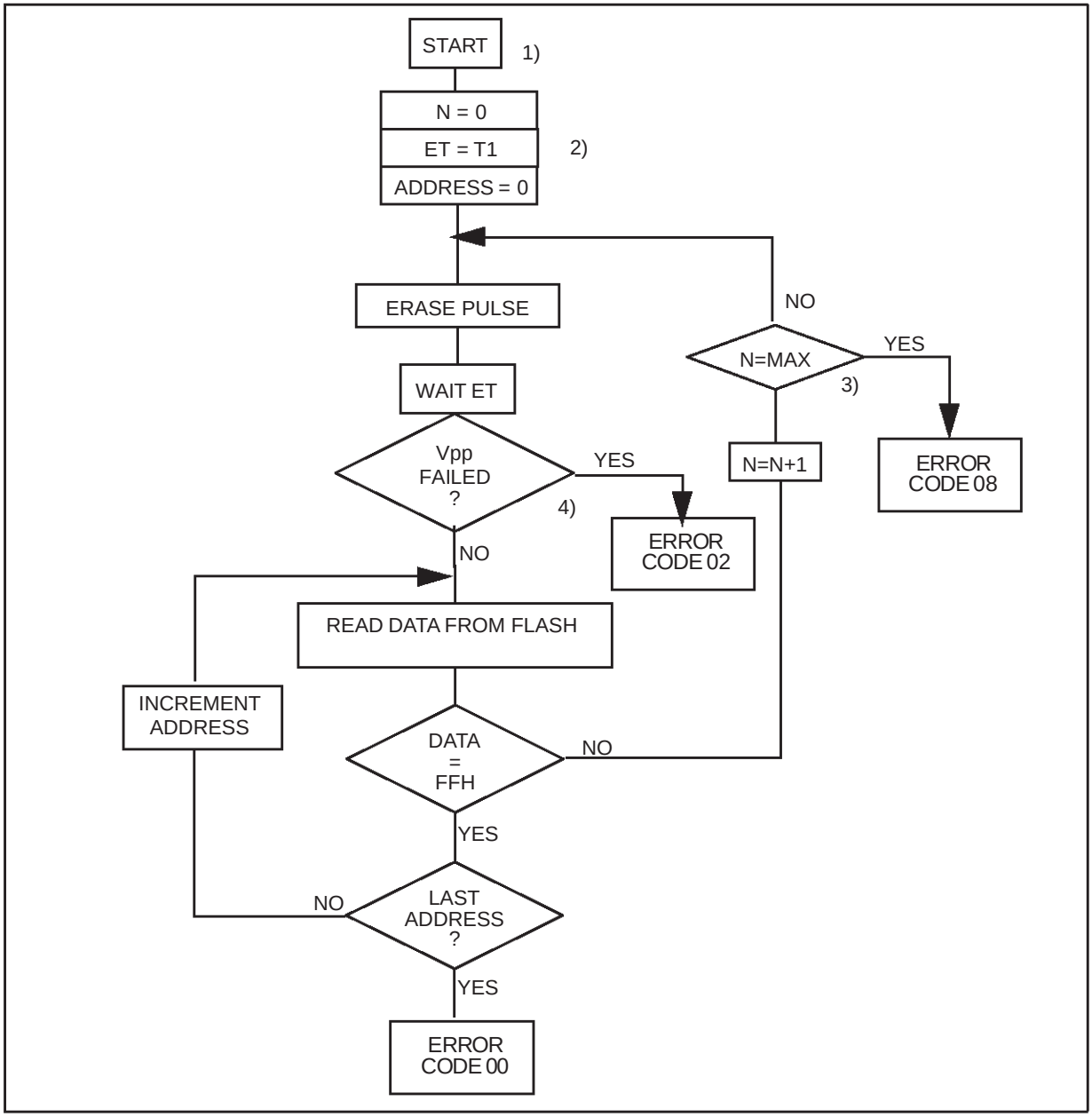

Figure 2 Presto Erase Algorithm

1) All words in the bank are programmed to 0000H before starting PRESTO erase algorithm.
2) ET = 32.768 ∗ CPU period
3) MAX = 60 s / ET = 1.831.055 [ns] / CPU period [ns]
4) Vpp failed if Vpp drops below 11 V during the programming pulse.

3.2 Error-codes for Flash erasing

The STEAK erasing command routines perform many checks and retries. When an error occurs, the STEAK returns an error-code in R0 which can be used to identify the cause of the problem. This section gives a correction guide for each error-code:

3.2.1 Erase error-code 01: ROMEN bit inside SYSCON is not set

- Cause: The Flash memory is not enabled, i.e. bit ROMEN in SYSCON register is not set.
- Solution: Set bit ROMEN in the SYSCON register and then recall the STEAK programming command. See“Programming error-code 01:” on page 11 for more details.

3.2.2 Erase error-code 02: Vpp failed

- Cause: The internal Flash Vpp voltage is not stabilized before calling the PRESTO algorithm, or the Vpp voltage drops below 11 V during a programming/erasing pulse when the PRESTO programming/erasing algorithm is running. This failure can be caused by a Vpp supply glitch.
- Solution: Recall the STEAK programming/erasing command if you suspect a glitch on the Vpp supply.

3.2.3 Erase error-code 03: Programming failed

- Cause: the PRESTO programming algorithm did not successfully program the whole bank with zeros, after the maximum number of retries. This means the Flash is dead.
- Solution: Change the device.

3.2.4 Erase error-code 05: CPU period out of range, must be 30 - 500 ns

- Cause: The CPU period given in R4 is out of valid range. This parameter is used by the STEAK to time some waiting loops inside the STEAK routines.
- Solution: Provide a valid value in R4 before calling STEAK command.

3.2.5 Erase error-code 06: Not enough free space on system stack

- Cause: the STEAK programming routine needs at least 50 words on System Stack. The STEAK routine performs the following check:”if (SP - STKOV) < 100 then return error 06”.
- Solution: Change the initialization setting of SYSCON (bitfield STKSZ) to provide a bigger stack size, or flush the System Stack to external memory before calling the STEAK programming command, and then restore it after STEAK return.

3.2.6 Erase error-code 07: Incorrect Bank number specified

- Cause: The bank number specified in the R2 and R3 registers is not equal to 0,1,2 or 3.
- Solution: Ensure that R2=R3=0,1,2 or 3 before calling STEAK erasing routine.

3.2.7 Erase error-code 08: Erasing failed

- Cause: The PRESTO erasing algorithm did not successfully erase the whole bank after maximum number of retries. Therefore, the flash is dead.
- Solution: Change the device.

3.2.8 Erase error-code FF: Bad STEAK command

- Cause: The content of R2 register does not equal the content of R3 register,

  or, the content of R1 register does not equal #0555h,

  or, the R0 content is not a valid command number.
- Solution: Ensure that R2=R3, R1 = #0555h and R0 = #0EEEEh *before* calling the STEAK erasing routine.

3.3 Code example

The following code is provided as an example to erase a bank of Flash. The Flash content can be undefined before calling the STEAK Erasing Command:

```
; code hereafter assumes that flash is mapped in segment 1
; i.e. bit ROMS1 = ‘1’ in SYSCON register
; Flash must also be enabled, i.e. bit ROMEN = ‘1’ in SYSCON.
EraseBnk:
MOV         R0, #0EEEEh      ; EEEEh: Bank Erasing command
MOV         R1, #05555h      ; Security data
MOV         R2, #01h         ; Bank 1 to be erased
MOV         R3, #01h         ; Security: R3 must be equal to R2
MOV         R4, #050d        ; 50ns is 20 MHz CPU clock frequency
MOV         R7, #08000h      ; R7 used for Flash trigger sequence
#define FCR 08000h
EXTS        #1, #2           ; flash can be mapped in segment 0 or 1
MOV         FCR, R7          ; first part
MOV         [R7], R7         ; second part
NOP                          ; WARNING: place 2 NOP operations after
NOP                          ; the Unlock sequence to avoid all poss
                             ; pipeline conflict in STEAK programs
; Check R0 return value
(see previous example)
CMP         R0, #0           ; Check error-code returned by STEAK
JMP         cc_EQ, EraseDone ; Erasing OK!

CMP         R0, #2           ; Vpp failed, we can retry programming
JMP         cc_NE, EraseError
SUB         RetryCnt, #1     ; command, with a max number of retry
                             ; loaded in RetryCnt variable.
JMP         cc_NZ, EraseBnk  ; If retry counter > 0, recall STEAK
JMP         cc_UC, ErrorVpp  ; else say that Vpp if failling.

EraseDone:
(next instructions)
```

4 STEAK Read Status

The STEAK Read Status command checks:

- If Vpp voltage is present and > 11 V.
- The current revision of STEAK.
- The device internal code identifier.

4.1 Reading the STEAK revision

The R1 register content is modified by the STEAK Read Status command, and the value returned is the STEAK revision code.The code is made up of the major revision (on the 8 MSB of R1), and minor revision (on the 8 LSB of R1). For examples:

- R1 = 0100h: STEAK major revision is 1, minor revision 0: the STEAK revision is rev. 1.0 (qualified revision).
- R1 = 0005h: STEAK major revision is 0, minor revision 5: the STEAK revision is rev. 0.5 (preliminary revision).

4.2 Error-codes for STEAK read status

The STEAK Read Status Command routine performs checks and returns an error-code that describes detected problems. The error-codes are described in this section:

4.2.1 Read status error-code 01: ROMEN bit inside SYSCON is not set

- Cause: The Flash memory is not enabled, i.e. bit ROMEN in the SYSCON register is not set.

4.2.2 Read status error-code 02: Vpp failed

- Cause: The Flash internal Vpp voltage is not stabilized.

4.2.3 Read status error-code 05: CPU period out of range must be between 30-500ns

- Cause: The CPU period given in R4 is out of valid range. This parameter is used by the STEAK to time some waiting loops inside the STEAK routines.

4.2.4 Read status error-code 06: Not enough free space on system stack

- Cause: the STEAK erasing routine needs at least 50 words on System Stack. The STEAK routine performs the following check:”if (SP - STKOV) < 100 then return error 06”.

- Solution: Change the initialization setting of SYSCON (bitfield STKSZ) to provide a bigger stack size, or flush the System Stack to external memory before calling the STEAK programming command, and then restore it after STEAK return.

4.2.5 Read status error-code FF: Bad STEAK command

- Cause: The R0 register content does not equal the code of any STEAK command.

4.3 Code example

The following code gives a Read Status programming example:

```
; code hereafter assumes that flash is mapped in segment 0
; i.e. bit ROMS1 = ‘0’ in SYSCON register
; Flash must also be enabled, i.e. bit ROMEN = ‘1’ in SYSCON.
ReadStatus:
MOV    R0, #07777h  ; 7777h: Read Status command
MOV    R4, #050d    ; 50ns is 20 MHz CPU clock frequency
MOV    R7, #00000h  ; R7 used for Flash trigger sequence
#define FCR 00000h
EXTS   #0, #2       ; flash is mapped in segment 0
MOV    FCR, R7      ; first part
MOV    [R7], R7     ; second part
NOP                 ; WARNING: place 2 NOP operations after
NOP                 ; the Unlock sequence to avoid all possible
                    ; pipeline conflict in STEAK programs
; R0 contains error-code if any or zero, R1 contains STEAK rev code.
```

littlediode
COMPONENTS FOR ALL

Littlediode supplies new, hard to find or obsolete electronic components and semiconductors all over the world.

With over two million different components listed you are sure to find the part you need.

Feel free to visit us today at our online store:

LittleDiode.com

Looking forward to providing you with the best possible service.