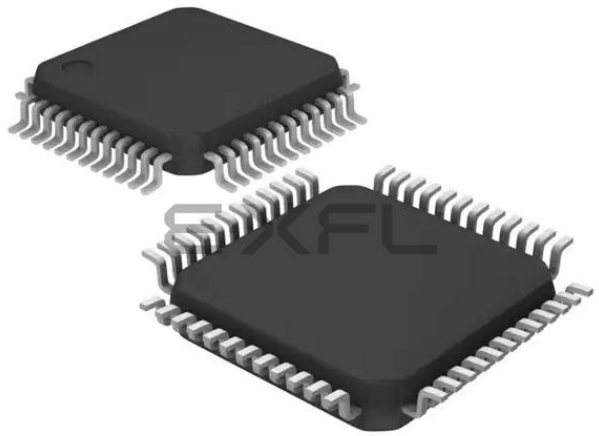


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	40
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	3K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g96f0clfr

The MC9S12G-Family is optimized for lower program memory sizes down to 16k. In order to simplify customer use it features an EEPROM with a small 4 bytes erase sector size.

The MC9S12G-Family deliver all the advantages and efficiencies of a 16-bit MCU while retaining the low cost, power consumption, EMC, and code-size efficiency advantages currently enjoyed by users of NXP's existing 8-bit and 16-bit MCU families. Like the MC9S12XS-Family, the MC9S12G-Family run 16-bit wide accesses without wait states for all peripherals and memories. The MC9S12G-Family is available in 100-pin LQFP, 64-pin LQFP, 48-pin LQFP/QFN, 32-pin LQFP and 20-pin TSSOP package options and aims to maximize the amount of functionality especially for the lower pin count packages. In addition to the I/O ports available in each module, further I/O ports are available with interrupt capability allowing wake-up from stop or wait modes.

1.2 Features

This section describes the key features of the MC9S12G-Family.

1.2.1 MC9S12G-Family Comparison

Table 1-1 provides a summary of different members of the MC9S12G-Family and their features. This information is intended to provide an understanding of the range of functionality offered by this microcontroller family.

Table 1-1. MC9S12G-Family Overview[1]

Feature	S12GN16	S12GNA16	S12GN32	S12GNA32	S12GN48	S12G48	S12GA48	S12G64	S12GA64	S12G96	S12GA96	S12G128	S12GA128	S12G192	S12GA192	S12G240	S12GA240
CPU	CPU12V1																
Flash memory [kBytes]	16	16	32	32	48	48	48	64	64	96	96	128	128	192	192	240	240
EEPROM [kBytes]	0.5	0.5	1	1	1.5	1.5	1.5	2	2	3	3	4	4	4	4	4	4
RAM [kBytes]	1	1	2	2	4	4	4	4	4	8	8	8	8	11	11	11	11
MSCAN	—	—	—	—	—	1	1	1	1	1	1	1	1	1	1	1	1
SCI	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	3	3
SPI	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	3	3
16-Bit Timer channels	6	6	6	6	6	6	6	6	6	8	8	8	8	8	8	8	8
8-Bit PWM channels	6	6	6	6	6	6	6	6	6	8	8	8	8	8	8	8	8
10-Bit ADC channels	8	—	8	—	12	12	—	12	—	12	—	12	—	16	—	16	—
12-Bit ADC channels	—	8	—	8	—	—	12	—	12	—	12	—	12	—	16	—	16
Temperature Sensor	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Yes	—	Yes
RVA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	YES	—	YES
8-Bit DAC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	2	—	2

Table 2-4. Signals and Priorities

Port	Pin	Signal	Signals per Device and Package (signal priority on pin from top to bottom)																				I/O	Description
			GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	G64 / GA64 / G48 / GA48	GN48	GA240 / GA192	G240 / G192	G128 / GA128 / G96 / GA96	G64 / GA64 / G48 / GA48	GN48	GN32 / GNA32	GN16 / GNA16	G64 / G48	GN48	GN32	GN16	GN32	GN16		
			100			64					48							32				20			
P	PP7-PP6	PWM7-PWM6	?	?	?	?	?	?																O	PWM channel
		[PTP7:PTP6]/ KWP7-KWP6	?	?	?	?	?	?	?	?														I/O	GPIO with interrupt
	PP5-PP4	PWM5-PWM4	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?							O	PWM channel
		[PTP5:PTP4]/ KWP5-KWP4	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?							I/O	GPIO with interrupt
	PP3-PP2	PWM3-PWM2	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			O	PWM channel
		ETRIG3- ETRIG2	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I	ADC external trigger
		[PTP3:PTP2]/ KWP3-KWP2	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	GPIO with interrupt
	PP1	PWM1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			O	PWM channel
		ECLKX2				?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			O	Free-running clock (ECLK x 2)
		ETRIG1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I	ADC external trigger
		[PTP1]/ KWP1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	GPIO with interrupt
	PP0	PWM0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			O	PWM channel
		API_EXTCLK				?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			O	API Clock
		ETRIG0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I	ADC external trigger
		[PTP0]/ KWP0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?			I/O	GPIO with interrupt

Legend	
?	Signal available on pin
?	Routing option on pin
?	Routing reset location
	Not available on pin

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
         LDX      #SUB_1
MARK1    JMP      0,X                    ; IRQ interrupt occurs during execution of this
MARK2    NOP                             ;

SUB_1    BRN      *                      ; JMP Destination address TRACE BUFFER ENTRY 1
                                         ; RTI Destination address TRACE BUFFER ENTRY 3
         NOP                             ;
ADDR1    DBNE     A,PART5                ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR  LDAB     #$F0                   ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
         STAB     VAR_C1
         RTI                             ;
```

The execution flow taking into account the IRQ is as follows

```
         LDX      #SUB_1
MARK1    JMP      0,X                    ;
IRQ_ISR  LDAB     #$F0                   ;
         STAB     VAR_C1
         RTI                             ;
SUB_1    BRN      *
         NOP                             ;
ADDR1    DBNE     A,PART5                ;
```

8.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user’s code that the DBG module is designed to help find.

8.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where the code is in error. This mode also features information bit storage to the trace buffer, for each address byte

8.5.6 Scenario 5

Trigger if following event A, event C precedes event B. i.e. the expected execution flow is A->B->C.

Figure 8-34. Scenario 5

Scenario 5 is possible with the S12SDBGV1 SCR encoding

8.5.7 Scenario 6

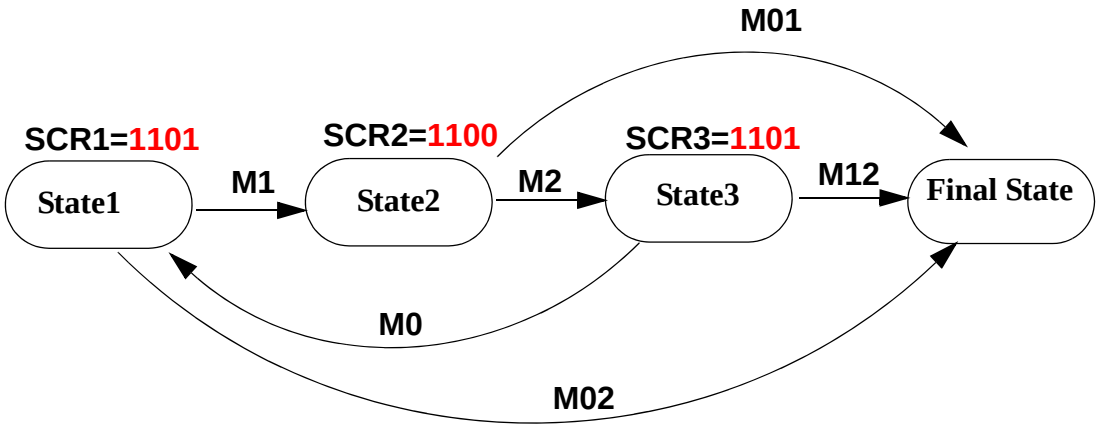
Trigger if event A occurs twice in succession before any of 2 other events (BC) occurs. This scenario is not possible using the S12SDBGV1 SCR encoding. S12SDBGV2 includes additions shown in red. The change in SCR1 encoding also has the advantage that a State1->State3 transition using M0 is now possible. This is advantageous because range and data bus comparisons use channel0 only.

Figure 8-35. Scenario 6

8.5.8 Scenario 7

Trigger when a series of 3 events is executed out of order. Specifying the event order as M1,M2,M0 to run in loops (120120120). Any deviation from that order should trigger. This scenario is not possible using the S12SDBGV1 SCR encoding because OR possibilities are very limited in the channel encoding. By adding OR forks as shown in red this scenario is possible.

Figure 8-36. Scenario 7



Chapter 9
Security (S12XS9SECV2)

Table 9-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
02.00	27 Aug 2004		reviewed and updated for S12XD architecture
02.01	21 Feb 2007		added S12XE, S12XF and S12XS architectures
02.02	19 Apr 2007		corrected statement about Backdoor key access via BDM on XE, XF, XS

9.1 Introduction

This specification describes the function of the security mechanism in the MC9S12G-Family (9SEC).

NOTE

No security feature is absolutely secure. However, NXP's strategy is to make reading or copying the FLASH and/or EEPROM difficult for unauthorized users.

9.1.1 Features

The user must be reminded that part of the security must lie with the application code. An extreme example would be application code that dumps the contents of the internal memory. This would defeat the purpose of security. At the same time, the user may also wish to put a backdoor in the application program. An example of this is the user downloads a security key through the SCI, which allows access to a programming routine that updates parameters stored in another section of the Flash memory.

The security features of the MC9S12G-Family (in secure mode) are:

- Protect the content of non-volatile memories (Flash, EEPROM)
- Execution of NVM commands is restricted
- Disable access to internal memory via background debug module (BDM)

9.1.2 Modes of Operation

Table 9-2 gives an overview over availability of security relevant features in unsecure and secure modes.

Table 9-2. Feature Availability in Unsecure and Secure Modes on S12XS

	Unsecure Mode						Secure Mode					
	NS	SS	NX	ES	EX	ST	NS	SS	NX	ES	EX	ST
Flash Array Access	?	?					?	?				

10.4 Functional Description

10.4.1 Phase Locked Loop with Internal Filter (PLL)

The PLL is used to generate a high speed PLLCLK based on a low frequency REFCLK.

The REFCLK is by default the IRCCLK which is trimmed to f_{IRC1M_TRIM}=1MHz.

If using the oscillator (OSCE=1) REFCLK will be based on OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency REFCLK using the REFDIV[3:0] bits. Based on the SYNDIV[5:0] bits the PLL generates the VCOCLK by multiplying the reference clock by a 2, 4, 6,... 126, 128. Based on the POSTDIV[4:0] bits the VCOCLK can be divided in a range of 1,2, 3, 4, 5, 6,... to 32 to generate the PLLCLK.

If oscillator is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

If oscillator is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

$$f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$$

If PLL is locked (LOCK=1) $$f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$$

If PLL is not locked (LOCK=0) $$f_{PLL} = \frac{f_{VCO}}{4}$$

If PLL is selected (PLLSEL=1) $$f_{bus} = \frac{f_{PLL}}{2}$$

NOTE

Although it is possible to set the dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.

Several examples of PLL divider settings are shown in Table 10-25. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF}.

Table 10-25. Examples of PLL Divider Settings

f_{osc}	REFDIV[3:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{bus}
off	$00	1MHz	00	$18	50MHz	01	$03	12.5MHz	6.25MHz

This buffer can be turned on or off with the ATDDIEN register for each ATD input pin.
This is important so that the buffer does not draw excess current when an ATD input pin is selected as analog input to the ADC12B8C.

12.5 Resets

At reset the ADC12B8C is in a power down state. The reset state of each individual bit is listed within the Register Description section (see Section 12.3.2, “Register Descriptions”) which details the registers and their bit-field.

12.6 Interrupts

The interrupts requested by the ADC12B8C are listed in Table 12-24. Refer to MCU specification for related vector address and priority.

Table 12-24. ATD Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Sequence Complete Interrupt	I bit	ASCIE in ATDCTL2
Compare Interrupt	I bit	ACMPIE in ATDCTL2

See Section 12.3.2, “Register Descriptions” for further details.

Table 22-11. TSCR2 Field Descriptions

Field	Description
7 TOI	Timer Overflow Interrupt Enable 0 Interrupt inhibited. 1 Hardware interrupt requested when TOF flag set.
2:0 PR[2:0]	**Timer Prescaler Select** — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 22-12.

Table 22-12. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	Bus Clock / 1
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

NOTE

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

22.3.2.10 Main Timer Interrupt Flag 1 (TFLG1)

Module Base + 0x000E

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	C5F	C4F	C3F	C2F	C1F	C0F
Reset	0	0	0	0	0	0	0	0

Figure 22-16. Main Timer Interrupt Flag 1 (TFLG1)

Read: Anytime

Write: Used in the clearing mechanism (set bits cause corresponding bits to be cleared). Writing a zero will not affect current status of the bit.

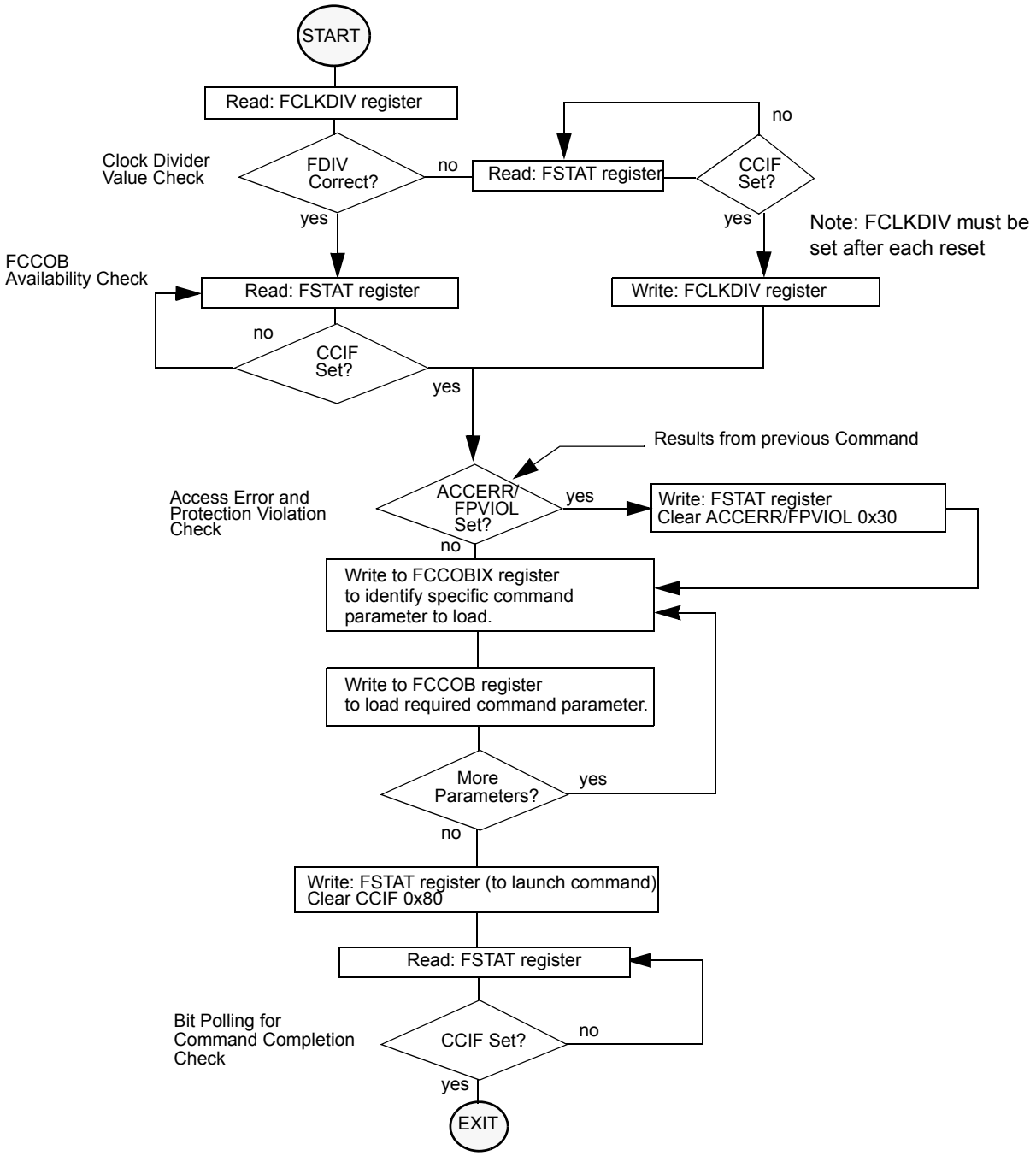


Figure 24-25. Generic Flash Command Write Sequence Flowchart

Table 24-47. Erase P-Flash Sector Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 24-25)
		Set if an invalid global address [17:16] is supplied see Table 24-3)[1]
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

1 As defined by the memory map for FTMRG32K1.

24.4.6.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 24-48. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 24-49. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 24-25)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation[1]
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation[1]

1 As found in the memory map for FTMRG32K1.

Table 24-57. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 24-25)
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied (See Table 24-32)[1]
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

1 As defined by the memory map for FTMRG32K1.

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

24.4.6.14 Erase Verify EEPROM Section Command

The Erase Verify EEPROM Section command will verify that a section of code in the EEPROM is erased. The Erase Verify EEPROM Section command defines the starting point of the data to be verified and the number of words.

Table 24-58. Erase Verify EEPROM Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify EEPROM Section command, the Memory Controller will verify the selected section of EEPROM memory is erased. The CCIF flag will set after the Erase Verify EEPROM Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

25.4.4.3 Valid Flash Module Commands

Table 25-27 present the valid Flash commands, as enabled by the combination of the functional MCU mode (Normal SingleChip NS, Special Singlechip SS) with the MCU security state (Unsecured, Secured).

Special Singlechip mode is selected by input mmc_ss_mode_ts2 asserted. MCU Secured state is selected by input mmc_secure input asserted.

Table 25-27. Flash Commands by Mode and Security State

FCMD	Command	Unsecured		Secured	
		NS[1]	SS[2]	NS[3]	SS[4]
0x01	Erase Verify All Blocks	*	*	*	*
0x02	Erase Verify Block	*	*	*	*
0x03	Erase Verify P-Flash Section	*	*	*	
0x04	Read Once	*	*	*	
0x06	Program P-Flash	*	*	*	
0x07	Program Once	*	*	*	
0x08	Erase All Blocks		*		*
0x09	Erase Flash Block	*	*	*	
0x0A	Erase P-Flash Sector	*	*	*	
0x0B	Unsecure Flash		*		*
0x0C	Verify Backdoor Access Key	*		*	
0x0D	Set User Margin Level	*	*	*	
0x0E	Set Field Margin Level		*		
0x10	Erase Verify EEPROM Section	*	*	*	
0x11	Program EEPROM	*	*	*	
0x12	Erase EEPROM Sector	*	*	*	

1 Unsecured Normal Single Chip mode
2 Unsecured Special Single Chip mode.
3 Secured Normal Single Chip mode.
4 Secured Special Single Chip mode.

25.4.4.4 P-Flash Commands

Table 25-28 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

Table 25-28. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and EEPROM) blocks are erased.

Table 26-16. FERSTAT Field Descriptions

Field	Description
1 DFDIF	**Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF.[2] 0 No double bit fault detected 1 Double bit fault detected or a Flash array read operation returning invalid data was attempted while command running
0 SFDIF	**Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.[1] The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or a Flash array read operation returning invalid data was attempted while command running

1 The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault but never both). A simultaneous access collision (Flash array read operation returning invalid data attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

2 There is a one cycle delay in storing the ECC DFDIF and SFDIF fault flags in this register. At least one NOP is required after a flash memory read before checking FERSTAT for the occurrence of ECC errors.

26.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Offset Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	FPOPEN	RNV6	FPHDIS	FPHS[1:0]		FPLDIS	FPLS[1:0]	
W								
Reset	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]

= Unimplemented or Reserved

Figure 26-13. Flash Protection Register (FPROT)

1 Loaded from IFR Flash configuration field, during reset sequence.

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 26.3.2.9.1, “P-Flash Protection Restrictions,” and Table 26-21).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 26-4) as indicated by reset condition ‘F’ in Figure 26-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Table 26-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

26.3.2.12 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000C

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-18. Flash Reserved1 Register (FRSV1)

All bits in the FRSV1 register read 0 and are not writable.

26.3.2.13 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-19. Flash Reserved2 Register (FRSV2)

All bits in the FRSV2 register read 0 and are not writable.

26.3.2.14 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

P-Flash memory (see Table 28-4) as indicated by reset condition F in Table 28-23. To change the EEPROM protection that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 28-22. EEPROT Field Descriptions

Field	Description
7 DPOPEN	EEPROM Protection Control 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
6–0 DPS[6:0]	**EEPROM Protection Size** — The DPS[6:0] bits determine the size of the protected area in the EEPROM memory, this size increase in step of 32 bytes, as shown in Table 28-23 .

Table 28-23. EEPROM Protection Address Range

DPS[6:0]	Global Address Range	Protected Size
0000000	0x0_0400 – 0x0_041F	32 bytes
0000001	0x0_0400 – 0x0_043F	64 bytes
0000010	0x0_0400 – 0x0_045F	96 bytes
0000011	0x0_0400 – 0x0_047F	128 bytes
0000100	0x0_0400 – 0x0_049F	160 bytes
0000101	0x0_0400 – 0x0_04BF	192 bytes
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
1011111 - to - 1111111	0x0_0400 – 0x0_0FFF	3,072 bytes

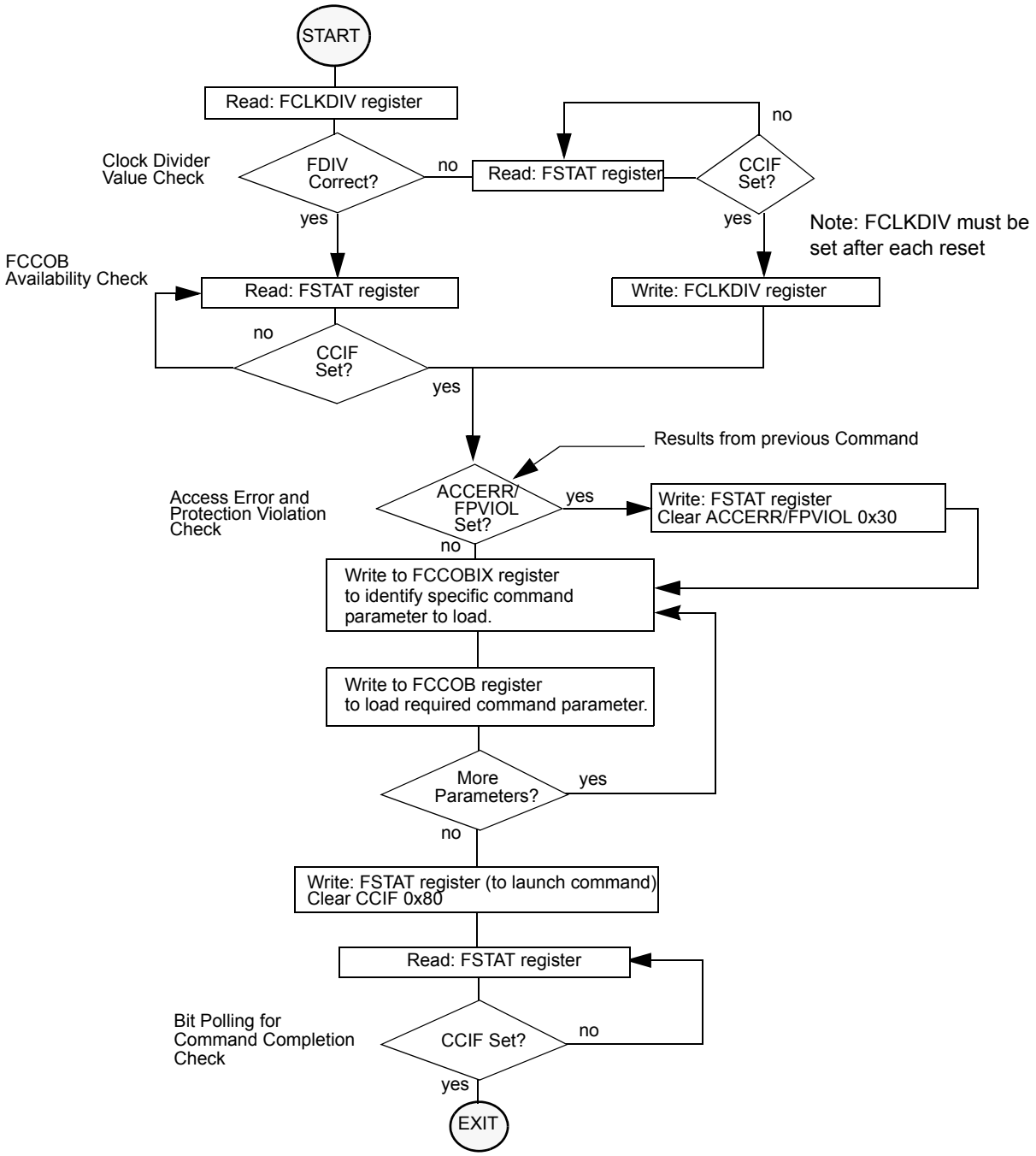


Figure 28-26. Generic Flash Command Write Sequence Flowchart

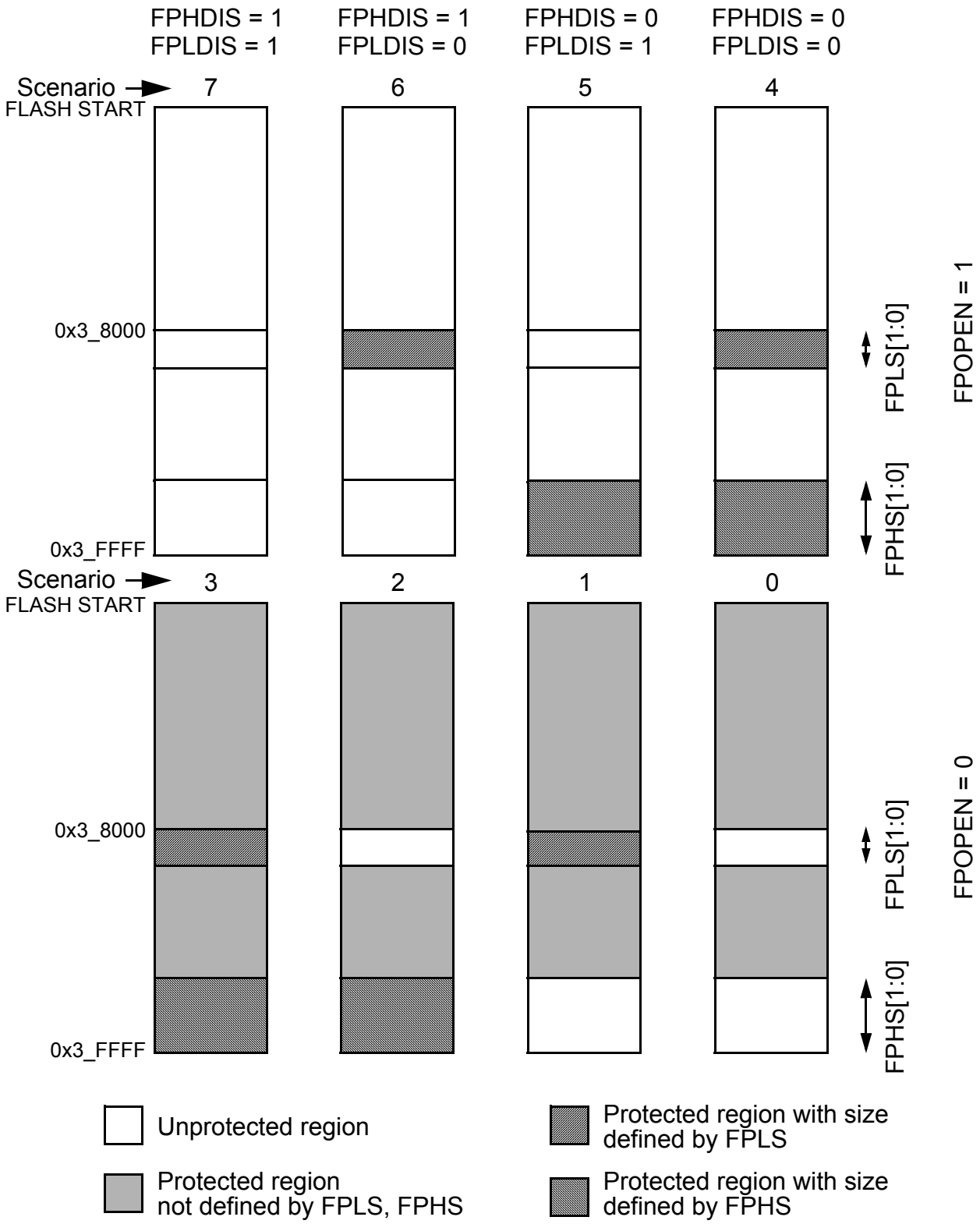


Figure 31-14. P-Flash Protection Scenarios

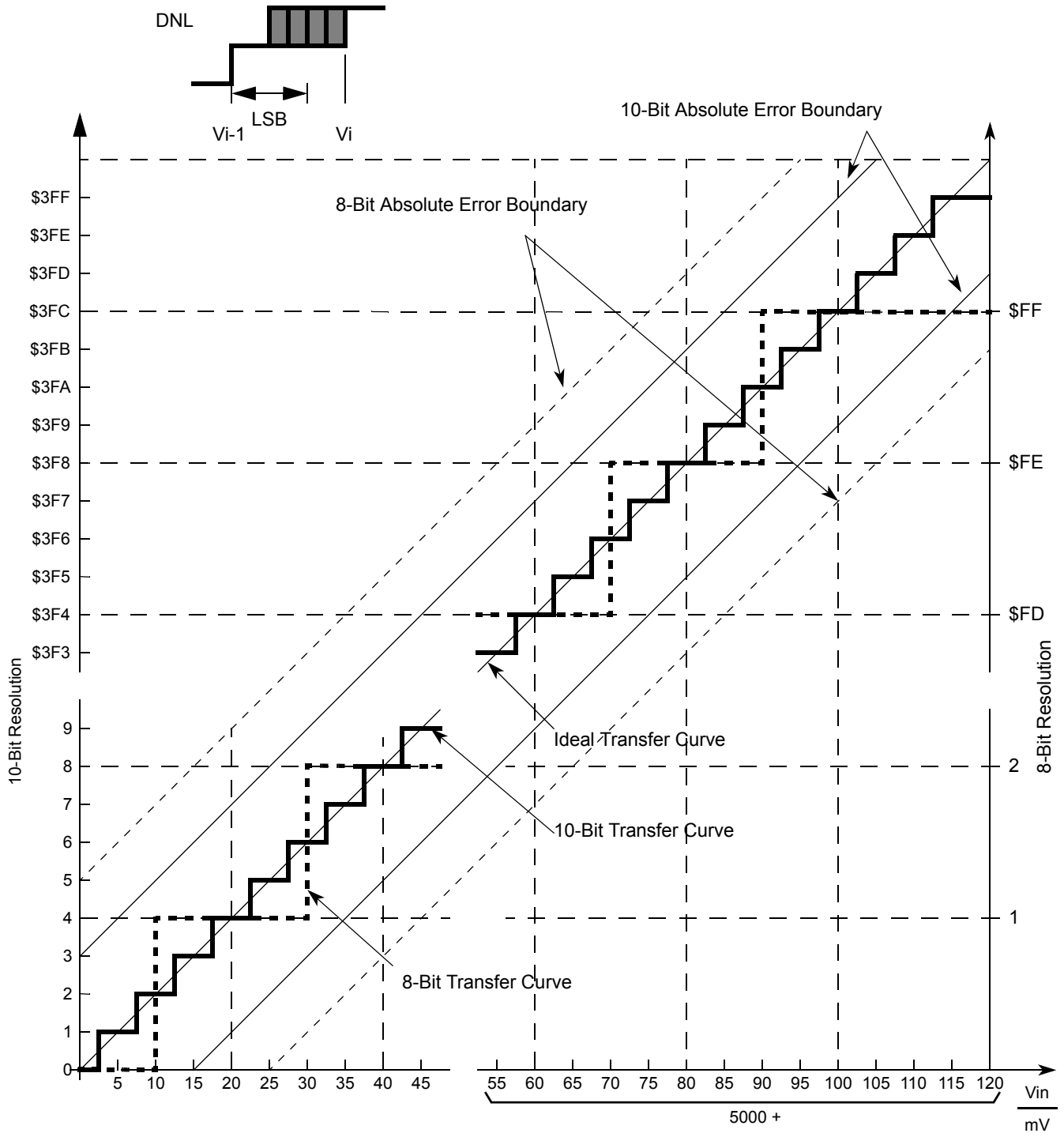


Figure A-1. ADC Accuracy Definitions

NOTE

Figure A-1 shows only definitions, for specification values refer to Table A-21 and Table A-26.

In Table A-52 the timing characteristics for slave mode are listed.

Table A-52. SPI Slave Mode Timing Characteristics

Conditions are 4.5 V < V_{DD35} < 5.5 V junction temperature from –40°C to T_{Jmax}.							
Num	**C**	**Characteristic**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	D	SCK Frequency	f_{sck}	DC	—	1/4	f_{bus}
1	D	SCK Period	t_{sck}	4	—	∞	t_{bus}
2	D	Enable Lead Time	t_L	4	—	—	t_{bus}
3	D	Enable Trail Time	t_T	4	—	—	t_{bus}
4	D	Clock (SCK) High or Low Time	t_{wsck}	4	—	—	t_{bus}
5	D	Data Setup Time (Inputs)	t_{su}	8	—	—	ns
6	D	Data Hold Time (Inputs)	t_{hi}	8	—	—	ns
7	D	Slave Access Time (time to data active)	t_a	—	—	20	ns
8	D	Slave MISO Disable Time	t_{dis}	—	—	22	ns
9	D	Data Valid after SCK Edge	t_{vsck}	—	—	$28 + 0.5 \cdot t_{bus}$ [1]	ns
10	D	Data Valid after $\overline{SS}$ fall	t_{vss}	—	—	$28 + 0.5 \cdot t_{bus}$ [1]	ns
11	D	Data Hold Time (Outputs)	t_{ho}	20	—	—	ns
12	D	Rise and Fall Time Inputs	t_{rfi}	—	—	9	ns
13	D	Rise and Fall Time Outputs	t_{rfo}	—	—	9	ns

[1] $0.5t_{bus}$ added due to internal synchronization delay

A.16 ADC Conversion Result Reference

The reference voltage V_{DDF} is measured under the conditions shown in Table A-53. The value stored in the IFR is the average of eight consecutive conversions at T_j=150 °C and eight consecutive conversions at T_j=-40 °C.

Table A-53. Measurement Conditions

Description	Symbol	Value	Unit
Regulator supply voltage	V_{DDR}	5	V
I/O supply voltage	V_{DDX}	5	V
Analog supply voltage	V_{DDA}	5	V
ADC reference voltage	V_{RH}	5	V
ADC clock	f_{ADCCLK}	2	MHz
ADC sample time	t_{SMP}	4	ADC clock cycles
Bus frequency	f_{bus}	24	MHz
Junction temperature	T_j	150 and -40	°C
Code execution		from RAM	