



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	HCS12X
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, I ² C, IrDA, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 12x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xet256maa

The state of the ROMCTL signal is latched into the ROMON bit in the MMCCTL1 register on the rising edge of $\overline{\text{RESET}}$. The state of the EROMCTL signal is latched into the EROMON bit in the MMCCTL1 register on the rising edge of $\overline{\text{RESET}}$.

Table 1-12. Chip Modes and Data Sources

Chip Modes	MODC	MODB	MODA	ROMCTL	EROMCTL	Data Source ⁽¹⁾
Normal single chip	1	0	0	X	X	Internal
Special single chip	0	0	0			
Emulation single chip	0	0	1	X	0	Emulation memory
				X	1	Internal Flash
Normal expanded	1	0	1	0	X	External application
				1	X	Internal Flash
Emulation expanded	0	1	1	0	X	External application
				1	0	Emulation memory
				1	1	Internal Flash
Special test	0	1	0	0	X	External application
				1	X	Internal Flash

1. Internal means resources inside the MCU are read/written.

Internal Flash means Flash resources inside the MCU are read/written.

Emulation memory means resources inside the emulator are read/written (PRU registers, Flash replacement, RAM, EEPROM, and register space are always considered internal).

External application means resources residing outside the MCU are read/written.

1.4.1.1 Normal Expanded Mode

Ports K, A, and B are configured as a 23-bit address bus, ports C and D are configured as a 16-bit data bus, and port E provides bus control and status signals. This mode allows 16-bit external memory and peripheral devices to be interfaced to the system. The fastest external bus rate is divide by 2 from the internal bus rate.

1.4.1.2 Normal Single-Chip Mode

There is no external bus in this mode. The processor program is executed from internal memory. Ports A, B,C,D, K, and most pins of port E are available as general-purpose I/O.

1.4.1.3 Special Single-Chip Mode

This mode is used for debugging single-chip operation, boot-strapping, or security related operations. The background debug module BDM is active in this mode. The CPU executes a monitor program located in an on-chip ROM. BDM firmware waits for additional serial commands through the BKGD pin. There is no external bus after reset in this mode.

1.4.1.4 Emulation of Expanded Mode

Developers use this mode for emulation systems in which the users target application is normal expanded mode. Code is executed from external memory or from internal memory depending on the state of ROMON and EROMON bit. In this mode the internal operation is visible on external bus interface.



2.3.26 Port T Polarity Select Register (PPST)

Address 0x0245

Access: User read/write⁽¹⁾

	7	6	5	4	3	2	1	0
R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-24. Port T Polarity Select Register (PPST)

1. Read: Anytime.
Write: Anytime.

Table 2-25. PPST Register Field Descriptions

Field	Description
7-0 PPST	Port T pull device select —Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.27 PIM Reserved Register

Address 0x0246

Access: User read⁽¹⁾

	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 2-25. PIM Reserved Register

1. Read: Always reads 0x00
Write: Unimplemented

2.3.28 PIM Reserved Register

Address 0x0247

Access: User read⁽¹⁾

	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 2-26. PIM Reserved Register

1. Read: Always reads 0x00
Write: Unimplemented

Table 2-33. PTM Register Field Descriptions (continued)

Field	Description
4 PTM	<p>Port M general purpose input/output data—Data Register Port M pin 4 is associated with the RXCAN signal of CAN2 and the routed CAN4 and CAN0, as well as with MOSI signals of SPI0. The CAN2 function takes precedence over the routed CAN0, routed CAN4, the routed SPI0 and the general purpose I/O function if the CAN2 module is enabled. The routed CAN0 function takes precedence over the routed CAN4, the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed CAN4 function takes precedence over the routed SPI0 and general purpose I/O function if the routed CAN4 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.</p>
3 PTM	<p>Port M general purpose input/output data—Data Register Port M pin 5 is associated with the TXCAN signal of CAN1 and the routed CAN0, as well as with $\overline{SS0}$ signals of SPI0. The CAN1 function takes precedence over the routed CAN0, the routed SPI0 and the general purpose I/O function if the CAN1 module is enabled. The routed CAN0 function takes precedence over the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.</p>
2 PTM	<p>Port M general purpose input/output data—Data Register Port M pin 4 is associated with the RXCAN signal of CAN1 and the routed CAN0, as well as with MISO signals of SPI0. The CAN1 function takes precedence over the routed CAN0, the routed SPI0 and the general purpose I/O function if the CAN1 module is enabled. The routed CAN0 function takes precedence over the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.</p>
1-0 PTM	<p>Port M general purpose input/output data—Data Register Port M pins 1 and 0 are associated with TXCAN and RXCAN signals of CAN0, respectively. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.</p>

2.3.38 Port M Input Register (PTIM)

Address 0x0251

Access: User read⁽¹⁾

	7	6	5	4	3	2	1	0
R	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
W								
Reset	u	u	u	u	u	u	u	u


 = Unimplemented or Reserved
u = Unaffected by reset

Figure 2-36. Port M Input Register (PTIM)

Table 7-7. Firmware Commands

Command ⁽¹⁾	Opcode (hex)	Data	Description
READ_NEXT ⁽²⁾	62	16-bit data out	Increment X index register by 2 ($X = X + 2$), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X index register by 2 ($X = X + 2$), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL ⁽³⁾	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

1. If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.
2. When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.
3. System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the "UNTIL" condition (BDM active again) is reached (see [Section 7.4.7, "Serial Interface Hardware Handshake Protocol"](#) last Note).

7.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

ADD

Add without Carry

ADD

Operation

$RS1 + RS2 \Rightarrow RD$
 $RD + IMM16 \Rightarrow RD$ (translates to ADDL RD, #IMM16[7:0]; ADDH RD, #IMM16[15:8])
Performs a 16 bit addition and stores the result in the destination register RD.

NOTE

When using immediate addressing mode (ADD RD, #IMM16), the V-flag and the C-Flag of the first instruction (ADDL RD, #IMM16[7:0]) are not considered by the second instruction (ADDH RD, #IMM16[15:8]).
 \Rightarrow Don't rely on the V-Flag if $RD + IMM16[7:0] \geq 2^{15}$.
 \Rightarrow Don't rely on the C-Flag if $RD + IMM16[7:0] \geq 2^{16}$.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

- N: Set if bit 15 of the result is set; cleared otherwise.
Z: Set if the result is \$0000; cleared otherwise.
V: Set if a two's complement overflow resulted from the operation; cleared otherwise.
 $RS1[15] \& RS2[15] \& \overline{RD[15]}_{new} \mid \overline{RS1[15]} \& RS2[15] \& RD[15]_{new}$
Refer to ADDH instruction for #IMM16 operations.
C: Set if there is a carry from bit 15 of the result; cleared otherwise.
 $RS1[15] \& RS2[15] \mid RS1[15] \& \overline{RD[15]}_{new} \mid RS2[15] \& \overline{RD[15]}_{new}$
Refer to ADDH instruction for #IMM16 operations.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
ADD RD, RS1, RS2	TRI	0	0	0	1	1	RD	RS1	RS2	1	0	P
ADD RD, #IMM16	IMM8	1	1	1	0	0	RD	IMM16[7:0]				P
	IMM8	1	1	1	0	1	RD	IMM16[15:8]				P

CMPL

Compare Immediate 8 bit Constant (Low Byte)

CMPL

Operation

RS.L – IMM8 ⇒ NONE, only condition code flags get updated

Subtracts the 8 bit constant IMM8 contained in the instruction code from the low byte of the source register RS.L using binary subtraction and updates the condition code register accordingly.

Remark: There is no equivalent operation using triadic addressing. Comparing the values of two registers can be performed by using the subtract instruction with R0 as destination register.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

- N: Set if bit 7 of the result is set; cleared otherwise.
- Z: Set if the 8 bit result is \$00; cleared otherwise.
- V: Set if a two's complement overflow resulted from the 8 bit operation; cleared otherwise.
 $RS[7] \& IMM8[7] \& result[7] \mid \overline{RS[7]} \& \overline{IMM8[7]} \& \overline{result[7]}$
- C: Set if there is a carry from the Bit 7 to Bit 8 of the result; cleared otherwise.
 $\overline{RS[7]} \& IMM8[7] \mid RS[7] \& result[7] \mid IMM8[7] \& result[7]$

Code and CPU Cycles

Source Form	Address Mode	Machine Code						Cycles
CMPL RS, #IMM8	IMM8	1	1	0	1	0	RS IMM8	P

13.4.2.2 General-Purpose Digital Port Operation

The input channel pins can be multiplexed between analog and digital data. As analog inputs, they are multiplexed and sampled as analog channels to the A/D converter. The analog/digital multiplex operation is performed in the input pads. The input pad is always connected to the analog input channels of the ADC12B16C. The input pad signal is buffered to the digital port registers. This buffer can be turned on or off with the ATDDIEN register. This is important so that the buffer does not draw excess current when analog potentials are presented at its input.

13.5 Resets

At reset the ADC12B16C is in a power down state. The reset state of each individual bit is listed within the Register Description section (see [Section 13.3.2, “Register Descriptions”](#)) which details the registers and their bit-field.

13.6 Interrupts

The interrupts requested by the ADC12B16C are listed in [Table 13-24](#). Refer to MCU specification for related vector address and priority.

Table 13-24. ATD Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Sequence Complete Interrupt	1 bit	ASCIE in ATDCTL2
Compare Interrupt	1 bit	ACMPIE in ATDCTL2

See [Section 13.3.2, “Register Descriptions”](#) for further details.

Table 15-7. IIC Divider and Hold Values (Sheet 2 of 6)

IBC[7:0] (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
06	34	10	13	18
07	40	10	16	21
08	28	7	10	15
09	32	7	12	17
0A	36	9	14	19
0B	40	9	16	21
0C	44	11	18	23
0D	48	11	20	25
0E	56	13	24	29
0F	68	13	30	35
10	48	9	18	25
11	56	9	22	29
12	64	13	26	33
13	72	13	30	37
14	80	17	34	41
15	88	17	38	45
16	104	21	46	53
17	128	21	58	65
18	80	9	38	41
19	96	9	46	49
1A	112	17	54	57
1B	128	17	62	65
1C	144	25	70	73
1D	160	25	78	81
1E	192	33	94	97
1F	240	33	118	121
20	160	17	78	81
21	192	17	94	97
22	224	33	110	113
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000F CANTXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
	W								
0x0010–0x0013 CANIDAR0–3	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
	W								
0x0014–0x0017 CANIDMRx	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
	W								
0x0018–0x001B CANIDAR4–7	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
	W								
0x001C–0x001F CANIDMR4–7	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
	W								
0x0020–0x002F CANRXFG	R	See Section 16.3.3, “Programmer’s Model of Message Storage”							
	W								
0x0030–0x003F CANTXFG	R	See Section 16.3.3, “Programmer’s Model of Message Storage”							
	W								

= Unimplemented or Reserved

Figure 16-3. MSCAN Register Summary (continued)

16.3.2 Register Descriptions

This section describes in detail all the registers and register bits in the MSCAN module. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order. All bits of all registers in this module are completely synchronous to internal clocks during a register read.

16.3.2.1 MSCAN Control Register 0 (CANCTL0)

The CANCTL0 register provides various control bits of the MSCAN module as described below.

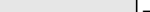
Module Base + 0x0000				Access: User read/write ⁽¹⁾				
	7	6	5	4	3	2	1	0
R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
W								
Reset:	0	0	0	0	0	0	0	1
			= Unimplemented					

Figure 16-4. MSCAN Control Register 0 (CANCTL0)



0x02F4

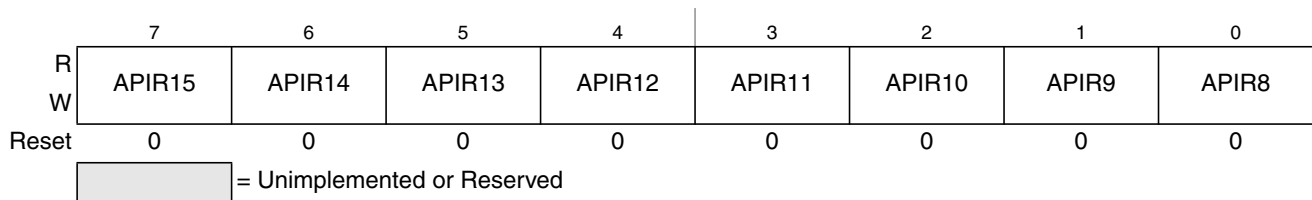


Figure 23-6. Autonomous Periodical Interrupt Rate High Register (VREGAPIRH)

0x02F5

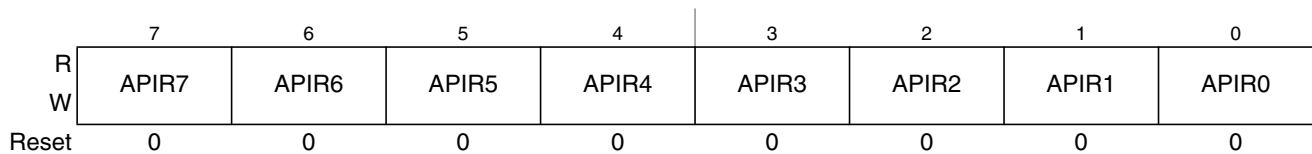


Figure 23-7. Autonomous Periodical Interrupt Rate Low Register (VREGAPIRL)

Table 23-8. VREGAPIRH / VREGAPIRL Field Descriptions

Field	Description
15-0 APIR[15:0]	Autonomous Periodical Interrupt Rate Bits — These bits define the timeout period of the API. See Table 23-9 for details of the effect of the autonomous periodical interrupt rate bits. Writable only if APIFE = 0 of VREGAPICL register.

Table 23-9. Selectable Autonomous Periodical Interrupt Periods

APICLK	APIR[15:0]	Selected Period
0	0000	0.2 ms ⁽¹⁾
0	0001	0.4 ms ¹
0	0002	0.6 ms ¹
0	0003	0.8 ms ¹
0	0004	1.0 ms ¹
0	0005	1.2 ms ¹
0
0	FFFD	13106.8 ms ¹
0	FFFE	13107.0 ms ¹
0	FFFF	13107.2 ms ¹
1	0000	2 * bus clock period
1	0001	4 * bus clock period
1	0002	6 * bus clock period
1	0003	8 * bus clock period
1	0004	10 * bus clock period
1	0005	12 * bus clock period
1
1	FFFD	131068 * bus clock period
1	FFFE	131070 * bus clock period

Table 25-17. FSTAT Field Descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag — The CCIF flag indicates that a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation. 0 Flash command in progress 1 Flash command has completed
5 ACCERR	Flash Access Error Flag — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 25.4.1.2) or issuing an illegal Flash command or when errors are encountered while initializing the EEE buffer ram during the reset sequence. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR. 0 No access error detected 1 Access error detected
4 FPVIOL	Flash Protection Violation Flag —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation detected 1 Protection violation detected
3 MGBUSY	Memory Controller Busy Flag — The MGBUSY flag reflects the active state of the Memory Controller. 0 Memory Controller is idle 1 Memory Controller is busy executing a Flash command (CCIF = 0) or is handling internal EEE operations
2 RSVD	Reserved Bit — This bit is reserved and always reads 0.
1–0 MGSTAT[1:0]	Memory Controller Command Completion Status Flag — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. See Section 25.4.2 , “Flash Command Description,” and Section 25.6 , “Initialization” for details.

25.3.2.8 Flash Error Status Register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

Offset Module Base + 0x0007

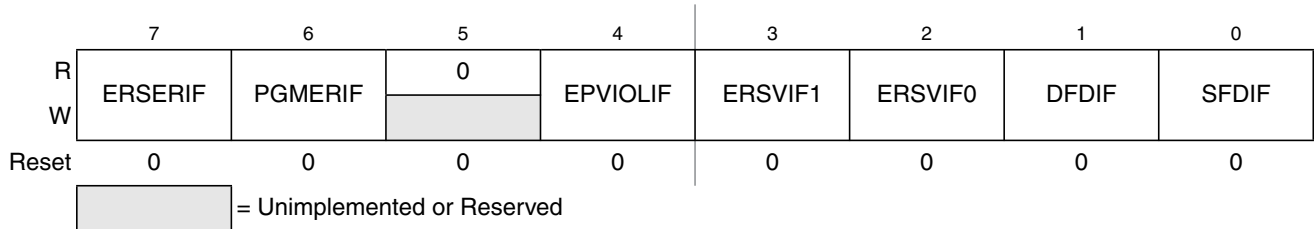


Figure 25-12. Flash Error Status Register (FERSTAT)

All flags in the FERSTAT register are readable and only writable to clear the flag.

Table 26-15. FCNFG Field Descriptions (continued)

Field	Description
1 FDFD	Force Double Bit Fault Detect — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. The FECCR registers will not be updated during the Flash array read operation with FDFD set unless an actual double bit fault is detected. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 26.3.2.7) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 26.3.2.6)
0 FSFD	Force Single Bit Fault Detect — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. The FECCR registers will not be updated during the Flash array read operation with FSFD set unless an actual single bit fault is detected. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 26.3.2.7) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 26.3.2.6)

26.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

Offset Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	ERSERIE	PGMERIE	0	EPVIOIE	ERSVIE1	ERSVIE0	DFDIE	SFDIE
W								
Reset	0	0	0	0	0	0	0	0

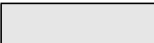
 = Unimplemented or Reserved

Figure 26-10. Flash Error Configuration Register (FERCNFG)

All assigned bits in the FERCNFG register are readable and writable.

Table 26-16. FERCNFG Field Descriptions

Field	Description
7 ERSERIE	EEE Erase Error Interrupt Enable — The ERSERIE bit controls interrupt generation when a failure is detected during an EEE erase operation. 0 ERSERIF interrupt disabled 1 An interrupt will be requested whenever the ERSERIF flag is set (see Section 26.3.2.8)
6 PGMERIE	EEE Program Error Interrupt Enable — The PGMERIE bit controls interrupt generation when a failure is detected during an EEE program operation. 0 PGMERIF interrupt disabled 1 An interrupt will be requested whenever the PGMERIF flag is set (see Section 26.3.2.8)
4 EPVIOIE	EEE Protection Violation Interrupt Enable — The EPVIOIE bit controls interrupt generation when a protection violation is detected during a write to the buffer RAM EEE partition. 0 EPVIOIF interrupt disabled 1 An interrupt will be requested whenever the EPVIOIF flag is set (see Section 26.3.2.8)

Offset Module Base + 0x0008

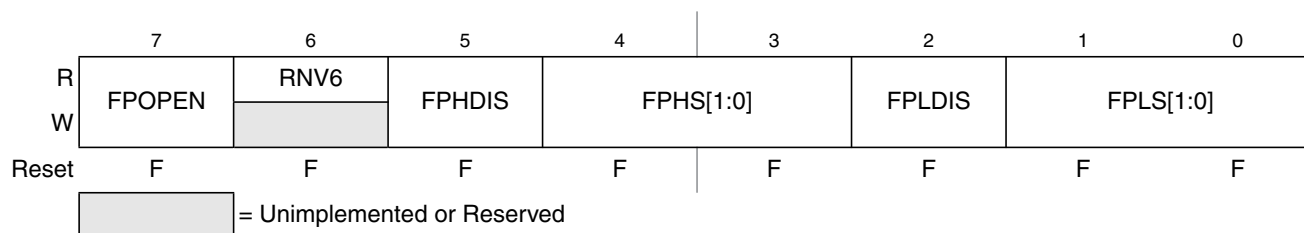


Figure 26-13. Flash Protection Register (FPROT)

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 26-23](#)).

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 0x7F_FF0C located in P-Flash memory (see [Table 26-3](#)) 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.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 26-19. FPROT Field Descriptions

Field	Description
7 FPOPEN	Flash Protection Operation Enable — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 26-20 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	Reserved Nonvolatile Bit — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	Flash Protection Higher Address Range Disable — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x7F_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	Flash Protection Higher Address Size — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 26-21 . The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	Flash Protection Lower Address Range Disable — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x7F_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	Flash Protection Lower Address Size — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 26-22 . The FPLS bits can only be written to while the FPLDIS bit is set.

- Program a duplicate DFPART to the EEE nonvolatile information register at global address 0x12_0002 (see [Table 26-7](#))
- Program ERPART to the EEE nonvolatile information register at global address 0x12_0004 (see [Table 26-7](#))
- Program a duplicate ERPART to the EEE nonvolatile information register at global address 0x12_0006 (see [Table 26-7](#))

The D-Flash user partition will start at global address 0x10_0000. The buffer RAM EEE partition will end at global address 0x13_FFFF. After the Partition D-Flash operation has completed, the CCIF flag will set.

Running the Partition D-Flash command a second time will result in the ACCERR bit within the FSTAT register being set. The data value written corresponds to the number of 256 byte sectors allocated for either direct D-Flash access (DFPART) or buffer RAM EEE access (ERPART).

Table 26-78. Partition D-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 26-30)
		Set if partitions have already been defined
		Set if an invalid DFPART or ERPART selection is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read
FERSTAT	EPVIOLIF	None

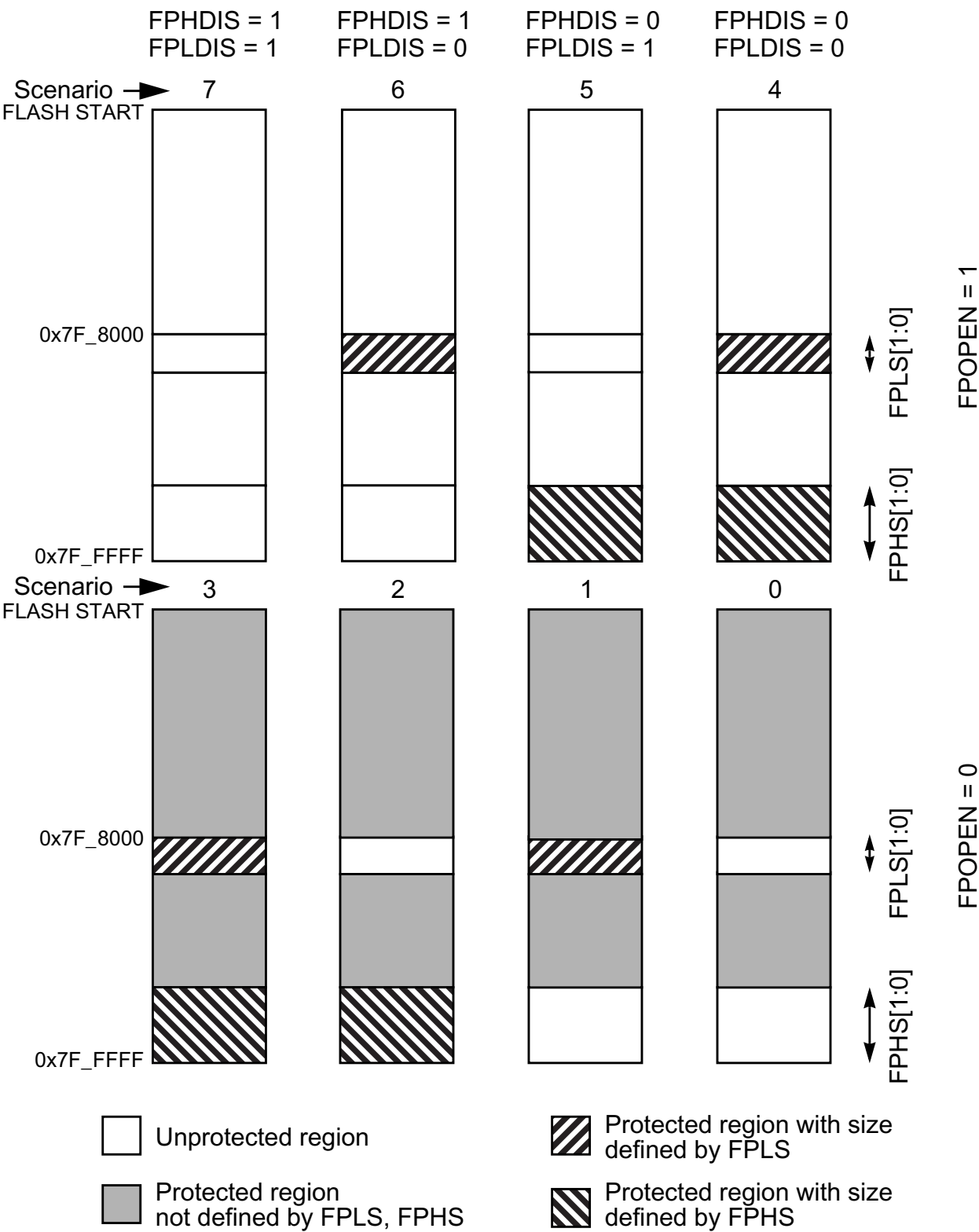


Figure 27-14. P-Flash Protection Scenarios

Table 27-76. EEPROM Emulation Query Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 27-30)
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

27.4.2.22 Partition D-Flash Command

The Partition D-Flash command allows the user to allocate sectors within the D-Flash block for applications and a partition within the buffer RAM for EEPROM access. The D-Flash block consists of 128 sectors with 256 bytes per sector. The Erase All Blocks command must be run prior to launching the Partition D-Flash command.

Table 27-77. Partition D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x20	Not required
001	Number of 256 byte sectors for the D-Flash user partition (DFPART)	
010	Number of 256 byte sectors for buffer RAM EEE partition (ERPART)	

Upon clearing CCIF to launch the Partition D-Flash command, the following actions are taken to define a partition within the D-Flash block for direct access (DFPART) and a partition within the buffer RAM for EEE use (ERPART):

- Validate the DFPART and ERPART values provided:
 - DFPART ≤ 128 (maximum number of 256 byte sectors in D-Flash block)
 - ERPART ≤ 16 (maximum number of 256 byte sectors in buffer RAM)
 - If ERPART > 0 , $128 - \text{DFPART} \geq 12$ (minimum number of 256 byte sectors in the D-Flash block required to support EEE)
 - If ERPART > 0 , $((128 - \text{DFPART}) / \text{ERPART}) \geq 8$ (minimum ratio of D-Flash EEE space to buffer RAM EEE space to support EEE)
- Erase verify the D-Flash block and the EEE nonvolatile information register
- Program DFPART to the EEE nonvolatile information register at global address 0x12_0000 (see [Table 27-7](#))

Table 29-38. Erase Verify P-Flash Section Command Error Handling

Register	Error Bit	Error Condition
FERSTAT	EPVIOLIF	None

29.4.2.4 Read Once Command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash block 0. The Read Once field is programmed using the Program Once command described in [Section 29.4.2.7](#). The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 29-39. Read Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x04	Not Required
001	Read Once phrase index (0x0000 - 0x0007)	
010	Read Once word 0 value	
011	Read Once word 1 value	
100	Read Once word 2 value	
101	Read Once word 3 value	

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag will set after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block 0 will return invalid data.

Table 29-40. Read Once Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 29-30)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read
FERSTAT	EPVIOLIF	None

29.4.2.5 Load Data Field Command

The Load Data Field command is executed to provide FCCOB parameters for multiple P-Flash blocks for a future simultaneous program operation in the P-Flash memory space.

CAUTION

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

Table 29-43. Program P-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x06	Global address [22:16] to identify P-Flash block
001	Global address [15:0] of phrase location to be programmed ⁽¹⁾	
010	Word 0 program value	
011	Word 1 program value	
100	Word 2 program value	
101	Word 3 program value	

1. Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller will program the data words to the supplied global address and will then proceed to verify the data words read back as expected. The CCIF flag will set after the Program P-Flash operation has completed.

Table 29-44. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 29-30)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if a Load Data Field command sequence is currently active and the selected block has previously been selected in the same command sequence
		Set if a Load Data Field command sequence is currently active and global address [17:0] does not match that previously supplied in the same command sequence
	FPVIOL	Set if the global address [22:0] points to a protected area
FERSTAT	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
	EPVIOLIF	None

29.4.2.7 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash block 0. The Program Once reserved field can be read using the Read Once command as described in [Section 29.4.2.4](#). The Program Once command must only be issued once since the nonvolatile information register in P-Flash block 0 cannot be erased. The Program

$$t = 350 \cdot \frac{1}{f_{\text{NVMBUS}}}$$

A.3.1.15 Full Partition D-Flash (FCMD=0x0F)

The maximum time for partitioning the D-flash (ERPART=16, DFPART=0) is given by :

$$t_{\text{part}} \approx 21800 \cdot \frac{1}{f_{\text{NVMOP}}} + 400000 \cdot \frac{1}{f_{\text{NVMBUS}}} + t_{\text{mass}}$$

A.3.1.16 Erase Verify D-Flash Section (FCMD=0x10)

Erase Verify D-Flash for a given number of words N_W is given by .

$$t_{\text{check}} \approx (840 + N_W) \cdot \frac{1}{f_{\text{NVMBUS}}}$$

A.3.1.17 D-Flash Programming (FCMD=0x11)

D-Flash programming time is dependent on the number of words being programmed and their location with respect to a row boundary, because programming across a row boundary requires extra steps. The D-Flash programming time is specified for different cases (1,2,3,4 words and 4 words across a row boundary) at a 50MHz bus frequency. The typical programming time can be calculated using the following equation, whereby N_W denotes the number of words; BC=0 if no boundary is crossed and BC=1 if a boundary is crossed.

$$t_{\text{dpgm}} = \left((15 + (54 \cdot N_W) + (16 \cdot \text{BC})) \cdot \frac{1}{f_{\text{NVMOP}}} \right) + \left((460 + (640 \cdot N_W) + (500 \cdot \text{BC})) \cdot \frac{1}{f_{\text{NVMBUS}}} \right)$$

The maximum programming time can be calculated using the following equation

$$t_{\text{dpgm}} = \left((15 + (56 \cdot N_W) + (16 \cdot \text{BC})) \cdot \frac{1}{f_{\text{NVMOP}}} \right) + \left((460 + (840 \cdot N_W) + (500 \cdot \text{BC})) \cdot \frac{1}{f_{\text{NVMBUS}}} \right)$$

A.3.1.18 Erase D-Flash Sector (FCMD=0x12)

Typical D-Flash sector erase times are those expected on a new device, where no margin verify fails occur. They can be calculated using the following equation.

$$t_{\text{eradf}} \approx 5025 \cdot \frac{1}{f_{\text{NVMOP}}} + 700 \cdot \frac{1}{f_{\text{NVMBUS}}}$$

Maximum D-Flash sector erase times can be calculated using the following equation.

$$t_{\text{eradf}} \approx 20100 \cdot \frac{1}{f_{\text{NVMOP}}} + 3300 \cdot \frac{1}{f_{\text{NVMBUS}}}$$

The D-Flash sector erase time on a new device is ~5ms and can extend to 20ms as the flash is cycled.



0x00340–0x0367 – Periodic Interrupt Timer (PIT) Map (Sheet 3 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0360	PITLD6 (hi)	R W	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
0x0361	PITLD6 (lo)	R W	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
0x0362	PITCNT6 (hi)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
0x0363	PITCNT6 (lo)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0
0x0364	PITLD7 (hi)	R W	PLD15	PLD14	PLD13	PLD12	PLD11	PLD10	PLD9	PLD8
0x0365	PITLD7 (lo)	R W	PLD7	PLD6	PLD5	PLD4	PLD3	PLD2	PLD1	PLD0
0x0366	PITCNT7 (hi)	R W	PCNT15	PCNT14	PCNT13	PCNT12	PCNT11	PCNT10	PCNT9	PCNT8
0x0367	PITCNT7 (lo)	R W	PCNT7	PCNT6	PCNT5	PCNT4	PCNT3	PCNT2	PCNT1	PCNT0