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	Not For New Designs
Core Processor	HCS12X
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	42
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xf512mlh

Table 4-7. EEE Nonvolatile Information Register Fields

Global Address (EEEIFRON)	Size (Bytes)	Description
0x12_0000 – 0x12_0001	2	D-Flash User Partition (DFPART) Refer to Section 4.4.2.15, “Full Partition D-Flash Command”
0x12_0002 – 0x12_0003	2	D-Flash User Partition (duplicate(1))
0x12_0004 – 0x12_0005	2	Buffer RAM EEE Partition (ERPART) Refer to Section 4.4.2.15, “Full Partition D-Flash Command”
0x12_0006 – 0x12_0007	2	Buffer RAM EEE Partition (duplicate1)
0x12_0008 – 0x12_007F	120	Reserved

1. Duplicate value used if primary value generates a double bit fault when read during the reset sequence.

4.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between Flash module base + 0x0000 and 0x0013. A summary of the Flash module registers is given in Figure 4-4 with detailed descriptions in the following subsections.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIV6	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0002 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0003 FECCRIX	R	0	0	0	0	0	ECCRIX2	ECCRIX1	ECCRIX0
	W								
0x0004 FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
	W								
0x0005 FERCNFG	R	ERSERIE	PGMERIE	0	EPVIOLIE	ERSVIE1	ERSVIE0	DFDIE	SFDIE
	W								

Figure 4-4. FTM384K2 Register Summary

Table 4-20. P-Flash Protection Function

FPOPEN	FPHDIS	FPLDIS	Function[1]
1	1	1	No P-Flash Protection
1	1	0	Protected Low Range
1	0	1	Protected High Range
1	0	0	Protected High and Low Ranges
0	1	1	Full P-Flash Memory Protected
0	1	0	Unprotected Low Range
0	0	1	Unprotected High Range
0	0	0	Unprotected High and Low Ranges

1. For range sizes, refer to Table 4-21 and Table 4-22.

Table 4-21. P-Flash Protection Higher Address Range

FPHS[1:0]	Global Address Range	Protected Size
00	0x7F_F800–0x7F_FFFF	2 Kbytes
01	0x7F_F000–0x7F_FFFF	4 Kbytes
10	0x7F_E000–0x7F_FFFF	8 Kbytes
11	0x7F_C000–0x7F_FFFF	16 Kbytes

Table 4-22. P-Flash Protection Lower Address Range

FPLS[1:0]	Global Address Range	Protected Size
00	0x7F_8000–0x7F_83FF	1 Kbyte
01	0x7F_8000–0x7F_87FF	2 Kbytes
10	0x7F_8000–0x7F_8FFF	4 Kbytes
11	0x7F_8000–0x7F_9FFF	8 Kbytes

All possible P-Flash protection scenarios are shown in Figure 4-14. Although the protection scheme is loaded from the Flash memory at global address 0x7F_FF0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

Table 8-18. FERSTAT Field Descriptions

Field	Description
7 ERSERIF	**EEE Erase Error Interrupt Flag** — The setting of the ERSERIF flag occurs due to an error in a Flash erase command that resulted in the erase operation not being successful during EEE operations. The ERSERIF flag is cleared by writing a 1 to ERSERIF. Writing a 0 to the ERSERIF flag has no effect on ERSERIF. While ERSERIF is set, it is possible to write to the buffer RAM EEE partition but the data written will not be transferred to the D-Flash EEE partition. 0 Erase command successfully completed on the D-Flash EEE partition 1 Erase command failed on the D-Flash EEE partition
6 PGMERIF	**EEE Program Error Interrupt Flag** — The setting of the PGMERIF flag occurs due to an error in a Flash program command that resulted in the program operation not being successful during EEE operations. The PGMERIF flag is cleared by writing a 1 to PGMERIF. Writing a 0 to the PGMERIF flag has no effect on PGMERIF. While PGMERIF is set, it is possible to write to the buffer RAM EEE partition but the data written will not be transferred to the D-Flash EEE partition. 0 Program command successfully completed on the D-Flash EEE partition 1 Program command failed on the D-Flash EEE partition
4 EPVIOLIF	**EEE Protection Violation Interrupt Flag** —The setting of the EPVIOLIF flag indicates an attempt was made to write to a protected area of the buffer RAM EEE partition. The EPVIOLIF flag is cleared by writing a 1 to EPVIOLIF. Writing a 0 to the EPVIOLIF flag has no effect on EPVIOLIF. While EPVIOLIF is set, it is possible to write to the buffer RAM EEE partition as long as the address written to is not in a protected area. 0 No EEE protection violation 1 EEE protection violation detected
3 ERSVIF1	**EEE Error Interrupt 1 Flag** —The setting of the ERSVIF1 flag indicates that the memory controller was unable to change the state of a D-Flash EEE sector. The ERSVIF1 flag is cleared by writing a 1 to ERSVIF1. Writing a 0 to the ERSVIF1 flag has no effect on ERSVIF1. While ERSVIF1 is set, it is possible to write to the buffer RAM EEE partition but the data written will not be transferred to the D-Flash EEE partition. 0 No EEE sector state change error detected 1 EEE sector state change error detected
2 ERSVIF0	**EEE Error Interrupt 0 Flag** —The setting of the ERSVIF0 flag indicates that the memory controller was unable to format a D-Flash EEE sector for EEE use. The ERSVIF0 flag is cleared by writing a 1 to ERSVIF0. Writing a 0 to the ERSVIF0 flag has no effect on ERSVIF0. While ERSVIF0 is set, it is possible to write to the buffer RAM EEE partition but the data written will not be transferred to the D-Flash EEE partition. 0 No EEE sector format error detected 1 EEE sector format error detected
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 was attempted on a Flash block that was under a Flash command operation. The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF. 0 No double bit fault detected 1 Double bit fault detected or an invalid Flash array read operation attempted
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 was attempted on a Flash block that was under a Flash command operation. 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 an invalid Flash array read operation attempted

8.3.2.9 P-Flash Protection Register (FPROT)

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

containing the EEE protection byte during the reset sequence, the EPOPEN bit will be cleared and remaining bits in the EPROT register will be set to leave the buffer RAM EEE partition fully protected.

Trying to write data to any protected area in the buffer RAM EEE partition will result in a protection violation error and the EPVIOLIF flag will be set in the FERSTAT register. Trying to write data to any protected area in the buffer RAM partitioned for user access will not be prevented and the EPVIOLIF flag in the FERSTAT register will not set.

Table 9-24. EPROT Field Descriptions

Field	Description
7 EPOPEN	**Enables writes to the Buffer RAM partitioned for EEE** 0 The entire buffer RAM EEE partition is protected from writes 1 Unprotected buffer RAM EEE partition areas are enabled for writes
6–4 RNV[6:4]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements
3 EPDIS	**Buffer RAM Protection Address Range Disable** — The EPDIS bit determines whether there is a protected area in a specific region of the buffer RAM EEE partition. 0 Protection enabled 1 Protection disabled
2–0 EPS[2:0]	**Buffer RAM Protection Size** — The EPS[2:0] bits determine the size of the protected area in the buffer RAM EEE partition as shown inTable 9-21. The EPS bits can only be written to while the EPDIS bit is set.

Table 9-25. Buffer RAM EEE Partition Protection Address Range

EPS[2:0]	Global Address Range	Protected Size
000	0x13_FFC0 – 0x13_FFFF	64 bytes
001	0x13_FF80 – 0x13_FFFF	128 bytes
010	0x13_FF40 – 0x13_FFFF	192 bytes
011	0x13_FF00 – 0x13_FFFF	256 bytes
100	0x13_FEC0 – 0x13_FFFF	320 bytes
101	0x13_FE80 – 0x13_FFFF	384 bytes
110	0x13_FE40 – 0x13_FFFF	448 bytes
111	0x13_FE00 – 0x13_FFFF	512 bytes

9.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Table 9-36. Erase Verify Block 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 an invalid global address [22:16] 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

9.4.2.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases. The section to be verified cannot cross a 256 Kbyte boundary in the P-Flash memory space.

Table 9-37. Erase Verify P-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x03	Global address [22:16] of a P-Flash block
001	Global address [15:0] of the first phrase to be verified	
010	Number of phrases to be verified	

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed.

Table 9-38. Erase Verify P-Flash Section 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 9-30)
		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 the requested section crosses a 256 Kbyte boundary
	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

Table 13-28. PSR0 Field Descriptions (Sheet 2 of 2)

Field	Description
7–4 STARTUP STATE	**Startup State** — protocol related variable: *vPOC!StartupState.* This field indicates the current sub-state of the startup procedure. 0000 reserved 0001 reserved 0010 *POC:coldstart collision resolution* 0011 *POC:coldstart listen* 0100 *POC:integration consistency check* 0101 *POC:integrationi listen* 0110 reserved 0111 *POC:initialize schedule* 1000 reserved 1001 reserved 1010 *POC:coldstart consistency check* 1011 reserved 1100 reserved 1101 *POC:integration coldstart check* 1110 *POC:coldstart gap* 1111 *POC:coldstart join*
2–0 WAKEUP STATUS	**Wakeup Status** — protocol related variable: *vPOC!WakeupStatus*. This field provides the outcome of the execution of the wakeup mechanism. 000 UNDEFINED 001 RECEIVED_HEADER 010 RECEIVED_WUP 011 COLLISION_HEADER 100 COLLISION_WUP 101 COLLISION_UNKNOWN 110 TRANSMITTED 111 reserved

13.5.2.21 Protocol Status Register 1 (PSR1)

Module Base + 0x002A Additional Reset: CSAA, CSP, CPN: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSAA	CSP	0	REMCSAT					CPN	HHR	FRZ	APTAC				
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-21. Protocol Status Register 1 (PSR1)

Write: Normal Mode

13.6 Functional Description

This section provides a detailed description of the functionality implemented in the FlexRay block.

13.6.1 Message Buffer Concept

The FlexRay block uses a data structure called *message buffer* to store frame data, configuration, control, and status data. Each message buffer consists of two parts, the *message buffer control data* and the *physical message buffer.* The message buffer control data are located in dedicated registers. The structure of the message buffer control data depends on the message buffer type and is described in Section 13.6.3, “Message Buffer Types”. The physical message buffer is located in the FRM and is described in Section 13.6.2, “Physical Message Buffer”.

13.6.2 Physical Message Buffer

All FlexRay messages and related frame and slot status information of received frames and of frames to be transmitted to the FlexRay bus are stored in data structures called *physical message buffers*. The physical message buffers are located in the FRM.The structure of a physical message buffer is depicted in Figure 13-100.

A physical message buffer consists of two fields, the *message buffer header field* and the *message buffer data field*. The message buffer header field contains the *frame header*, the *data field offset*, and the *slot status*.The message buffer data field contains the *frame data*.

The connection between the two fields is established by the *data field offset*.

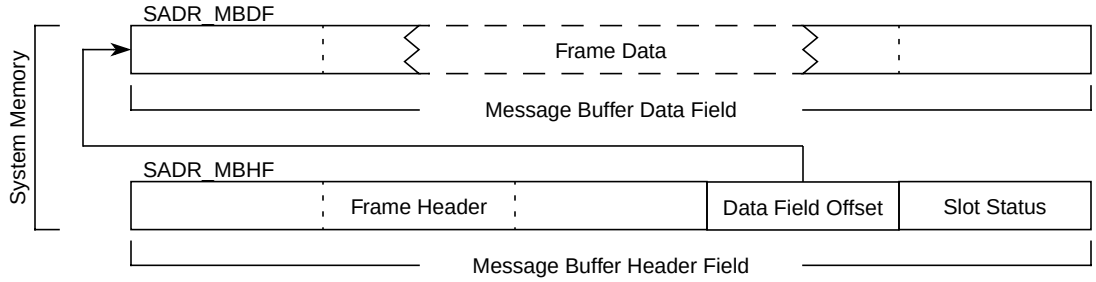

Figure 13-100. Physical Message Buffer Structure

13.6.2.1 Message Buffer Header Field

The message buffer header field is a contiguous region in the FRM and occupies ten bytes. It contains the frame header, the data field offset, and the slot status. Its structure is shown in Figure 13-100. The physical start address *SADR_MBHF* of the message buffer header field must be 16-bit aligned.

13.6.2.1.1 Frame Header

The frame header occupies the first six bytes in the message buffer header field. It contains all FlexRay frame header related information according to the *FlexRay Communications System Protocol*

Table 14-9. XGVBR Field Descriptions

Field	Description
15–1 XBVBR[15:1]	**Vector Base Address** — The XGVBR register holds the start address of the vector block in the XGATE memory map.

14.3.1.8 XGATE Channel Interrupt Flag Vector (XGIF)

The XGATE Channel Interrupt Flag Vector (Figure 14-10) provides access to the interrupt flags of all channels. Each flag may be cleared by writing a "1" to its bit location. Refer to Section 14.5.2, “Outgoing Interrupt Requests” for further information.

Module Base +0x0008

	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
R	0	0	0	0	0	0	0	XGIF_78	XGF_77	XGIF_76	XGIF_75	XGIF_74	XGIF_73	XGIF_72	XGIF_71	XGIF_70
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
R	XGIF_6F	XGIF_6E	XGIF_6D	XGIF_6C	XGIF_6B	XGIF_6A	XGIF_69	XGIF_68	XGF_67	XGIF_66	XGIF_65	XGIF_64	XGIF_63	XGIF_62	XGIF_61	XGIF_60
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
R	XGIF_5F	XGIF_5E	XGIF_5D	XGIF_5C	XGIF_5B	XGIF_5A	XGIF_59	XGIF_58	XGF_57	XGIF_56	XGIF_55	XGIF_54	XGIF_53	XGIF_52	XGIF_51	XGIF_50
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
R	XGIF_4F	XGIF_4E	XGIF_4D	XGIF_4C	XGIF_4B	XGIF_4A	XGIF_49	XGIF_48	XGF _47	XGIF_46	XGIF_45	XGIF_44	XGIF_43	XGIF_42	XGIF_41	XGIF_40
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-10. XGATE Channel Interrupt Flag Vector (XGIF)

BGE Branch if Greater than or Equal to Zero BGE

Operation

If N ^ V = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Branch instruction to compare signed numbers.

Branch if RS1 ≥ RS2:

```
SUB     R0,RS1,RS2
BGE     REL9
```

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BGE REL9	REL9	0	0	1	1	0	1	0	REL9	PP/P

STW Store Word to Memory STW

Operation

RS ⇒ M[RB, #OFFS5]
RS ⇒ M[RB, RI]
RS ⇒ M[RB, RI]; RI+2 ⇒ RI;
RI–2 ⇒ RI; RS ⇒ M[RB, RI][1]

Stores the content of register RS to memory.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
STW RS, (RB, #OFFS5)	IDO5	0	1	0	1	1	RS	RB	OFFS5			PW
STW RS, (RB, RI)	IDR	0	1	1	1	1	RS	RB	RI	0	0	PW
STW RS, (RB, RI+)	IDR+	0	1	1	1	1	RS	RB	RI	0	1	PW
STW RS, (RB, -RI)	-IDR	0	1	1	1	1	RS	RB	RI	1	0	PW

1. If the same general purpose register is used as index (RI) and source register (RS), the unmodified content of the source register is written to the memory: RS ⇒ M[RB, RS–2]; RS–2 ⇒ RS

Figure 15-14 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode. Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Since this is not a probable situation, the protocol does not prevent this conflict from happening.

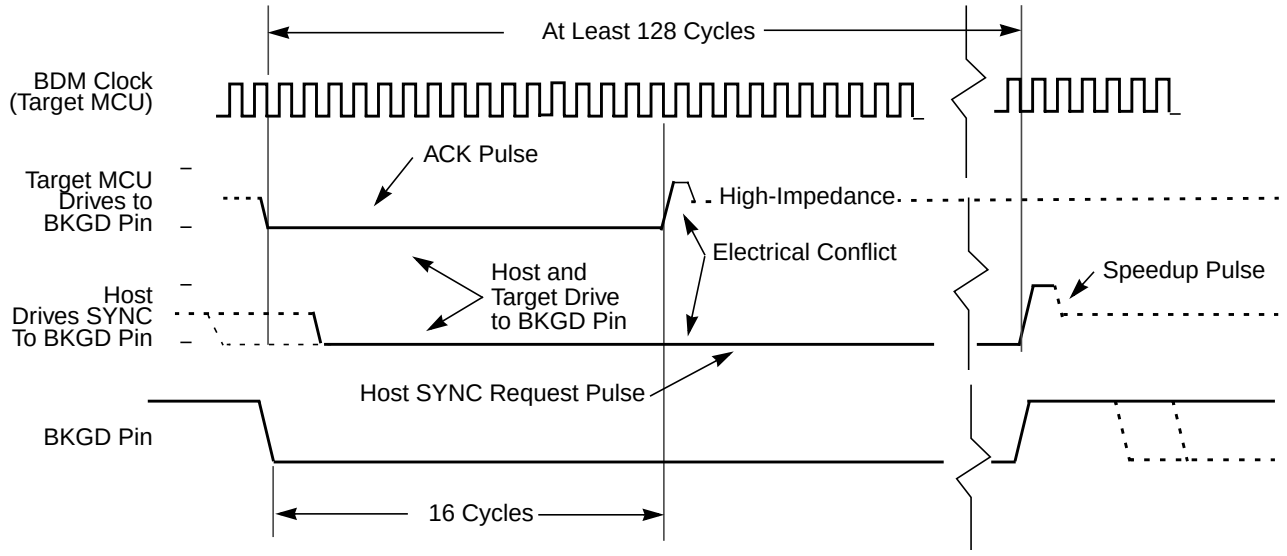

Figure 15-14. ACK Pulse and SYNC Request Conflict

NOTE

This information is being provided so that the MCU integrator will be aware that such a conflict could eventually occur.

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices which are not able to execute the hardware handshake protocol. It also allows for new POD devices, that support the hardware handshake protocol, to freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target will issue the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands will ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands will ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin and when the data bus cycle is complete. See Section 15.4.3, “BDM Hardware Commands” and Section 15.4.4, “Standard BDM Firmware Commands” for more information on the BDM commands.

20.3.2.21 PMF Frequency Control A Register (PMFFQCA)

Address: $0021

	7	6	5	4	3	2	1	0
R W	LDFQA				HALFA	PRSCA		PWMRFA
Reset	0	0	0	0	0	0	0	0

Figure 20-27. PMF Frequency Control A Register (PMFFQCA)

Read and write anytime.

Table 20-26. PMFFQCA Field Descriptions

Field	Description
7–4 LDFQA	**Load Frequency A** — This field selects the PWM load frequency according to Table 20-27. See Section 20.4.7.2, “Load Frequency” for more details. **Note:** The LDFQA field takes effect when the current load cycle is complete, regardless of the state of the load okay bit, LDOKA. Reading the LDFQA field reads the buffered value and not necessarily the value currently in effect.
3 HALFA	**Half Cycle Reload A** — This bit enables half-cycle reloads in center-aligned PWM mode. This bit has no effect on edge-aligned PWMs. 0 Half-cycle reloads disabled 1 Half-cycle reloads enabled
2–1 PRSCA	**Prescaler A** — This buffered field selects the PWM clock frequency illustrated in Table 20-28. **Note:** Reading the PRSCA field reads the buffered value and not necessarily the value currently in effect. The PRSCA field takes effect at the beginning of the next PWM cycle and only when the load okay bit, LDOKA, is set.
0 PWMRFA	**PWM Reload Flag A** — This flag is set at the beginning of every reload cycle regardless of the state of the LDOKA bit. Clear PWMRFA by reading PMFFQCA with PWMRFA set and then writing a logic one to the PWMRFA bit. If another reload occurs before the clearing sequence is complete, writing logic one to PWMRFA has no effect. 0 No new reload cycle since last PWMRFA clearing 1 New reload cycle since last PWMRFA clearing **Note:** Clearing PWMRFA satisfies pending PWMRFA CPU interrupt requests.

Table 20-27. PWM Reload Frequency A

LDFQA	PWM Reload Frequency	LDFQ[3:0]	PWM Reload Frequency
0000	Every PWM opportunity	1000	Every 9 PWM opportunities
0001	Every 2 PWM opportunities	1001	Every 10 PWM opportunities
0010	Every 3 PWM opportunities	1010	Every 11 PWM opportunities
0011	Every 4 PWM opportunities	1011	Every 12 PWM opportunities
0100	Every 5 PWM opportunities	1100	Every 13 PWM opportunities
0101	Every 6 PWM opportunities	1101	Every 14 PWM opportunities
0110	Every 7 PWM opportunities	1110	Every 15 PWM opportunities
0111	Every 8 PWM opportunities	1111	Every 16 PWM opportunities

Table 21-9. Time Segment 2 Values

TSEG22	TSEG21	TSEG20	Time Segment 2
0	0	0	1 Tq clock cycle(1)
0	0	1	2 Tq clock cycles
:	:	:	:
1	1	0	7 Tq clock cycles
1	1	1	8 Tq clock cycles

1. This setting is not valid. Please refer to Table 21-36 for valid settings.

Table 21-10. Time Segment 1 Values

TSEG13	TSEG12	TSEG11	TSEG10	Time segment 1
0	0	0	0	1 Tq clock cycle(1)
0	0	0	1	2 Tq clock cycles1
0	0	1	0	3 Tq clock cycles1
0	0	1	1	4 Tq clock cycles
:	:	:	:	:
1	1	1	0	15 Tq clock cycles
1	1	1	1	16 Tq clock cycles

1. This setting is not valid. Please refer to Table 21-36 for valid settings.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (Tq) clock cycles per bit (as shown in Table 21-9 and Table 21-10).

Eqn. 21-1

$$\text{Bit Time} = \frac{(\text{Prescaler value})}{f_{CANCLK}} \bullet (1 + \text{TimeSegment1} + \text{TimeSegment2})$$

21.3.2.5 MSCAN Receiver Flag Register (CANRFLG)

A flag can be cleared only by software (writing a 1 to the corresponding bit position) when the condition which caused the setting is no longer valid. Every flag has an associated interrupt enable bit in the CANRIER register.

Module Base + 0x0004 — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 21-8. MSCAN Receiver Flag Register (CANRFLG)

1. Read: Anytime
   Write: Anytime when not in initialization mode, except RSTAT[1:0] and TSTAT[1:0] flags which are read-only; write of 1 clears flag; write of 0 is ignored

NOTE

The CANRFLG register is held in the reset state[1] when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Table 21-11. CANRFLG Register Field Descriptions

Field	Description
7 WUPIF	**Wake-Up Interrupt Flag** — If the MSCAN detects CAN bus activity while in sleep mode (see Section 21.4.5.5, "MSCAN Sleep Mode,") and WUPE = 1 in CANTCTL0 (see Section 21.3.2.1, "MSCAN Control Register 0 (CANCTL0)"), the module will set WUPIF. If not masked, a wake-up interrupt is pending while this flag is set. 0 No wake-up activity observed while in sleep mode 1 MSCAN detected activity on the CAN bus and requested wake-up
6 CSCIF	**CAN Status Change Interrupt Flag** — This flag is set when the MSCAN changes its current CAN bus status due to the actual value of the transmit error counter (TEC) and the receive error counter (REC). An additional 4-bit (RSTAT[1:0], TSTAT[1:0]) status register, which is split into separate sections for TEC/REC, informs the system on the actual CAN bus status (see Section 21.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)"). If not masked, an error interrupt is pending while this flag is set. CSCIF provides a blocking interrupt. That guarantees that the receiver/transmitter status bits (RSTAT/TSTAT) are only updated when no CAN status change interrupt is pending. If the TECs/RECs change their current value after the CSCIF is asserted, which would cause an additional state change in the RSTAT/TSTAT bits, these bits keep their status until the current CSCIF interrupt is cleared again. 0 No change in CAN bus status occurred since last interrupt 1 MSCAN changed current CAN bus status
5-4 RSTAT[1:0]	**Receiver Status Bits** — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate receiver related CAN bus status of the MSCAN. The coding for the bits RSTAT1, RSTAT0 is: 00 RxOK: 0 ≤ receive error counter ≤ 96 01 RxWRN: 96 < receive error counter ≤ 127 10 RxERR: 127 < receive error counter 11 Bus-off(1): transmit error counter > 255
3-2 TSTAT[1:0]	**Transmitter Status Bits** — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate transmitter related CAN bus status of the MSCAN. The coding for the bits TSTAT1, TSTAT0 is: 00 TxOK: 0 ≤ transmit error counter ≤ 96 01 TxWRN: 96 < transmit error counter ≤ 127 10 TxERR: 127 < transmit error counter ≤ 255 11 Bus-Off: transmit error counter > 255

1. The RSTAT[1:0], TSTAT[1:0] bits are not affected by initialization mode.

Table 21-11. CANRFLG Register Field Descriptions (continued)

Field	Description
1 OVRIF	**Overrun Interrupt Flag** — This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set. 0 No data overrun condition 1 A data overrun detected
0 RXF(2)	**Receive Buffer Full Flag** — RXF is set by the MSCAN when a new message is shifted in the receiver FIFO. This flag indicates whether the shifted buffer is loaded with a correctly received message (matching identifier, matching cyclic redundancy code (CRC) and no other errors detected). After the CPU has read that message from the RxFG buffer in the receiver FIFO, the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the shifting of the next FIFO entry into the foreground buffer (RxFG). If not masked, a receive interrupt is pending while this flag is set. 0 No new message available within the RxFG 1 The receiver FIFO is not empty. A new message is available in the RxFG

1. Redundant information for the most critical CAN bus status which is "bus-off". This only occurs if the Tx error counter exceeds a number of 255 errors. Bus-off affects the receiver state. As soon as the transmitter leaves its bus-off state the receiver state skips to RxOK too. Refer also to TSTAT[1:0] coding in this register.
2. To ensure data integrity, do not read the receive buffer registers while the RXF flag is cleared. For MCUs with dual CPUs, reading the receive buffer registers while the RXF flag is cleared may result in a CPU fault condition.

21.3.2.6 MSCAN Receiver Interrupt Enable Register (CANRIER)

This register contains the interrupt enable bits for the interrupt flags described in the CANRFLG register.

Module Base + 0x0005 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
Reset:	0	0	0	0	0	0	0	0

Figure 21-9. MSCAN Receiver Interrupt Enable Register (CANRIER)

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANRIER register is held in the reset state when the initialization mode is active (INITRQ=1 and INITAK=1). This register is writable when not in initialization mode (INITRQ=0 and INITAK=0).

The RSTATE[1:0], TSTATE[1:0] bits are not affected by initialization mode.

```
                MOVB    #$01,PITINTE            ; enable interrupt channel 0
                MOVB    #$80,PITCFLMT           ; enable PIT
                CLI                             ; clear Interrupt disable Mask bit

;******************** Main Program ************************************************************

MAIN:           BRA *                           ; loop until interrupt

;******************** Channel 0 Interrupt Routine *********************************************

CH0_ISR:        LDAA    PITTF                   ; 8 bit read of PIT time out flags
                MOVB    #$01,PITTF              ; clear PIT channel 0 time out flag
                RTI                             ; return to MAIN
```

26.3.2.11 Timer System Control Register 2 (TSCR2)

Module Base + 0x000D

	7	6	5	4	3	2	1	0
R	TOI	0	0	0	TCRE	PR2	PR1	PR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-16. Timer System Control Register 2 (TSCR2)

Read or write: Anytime

All bits reset to zero.

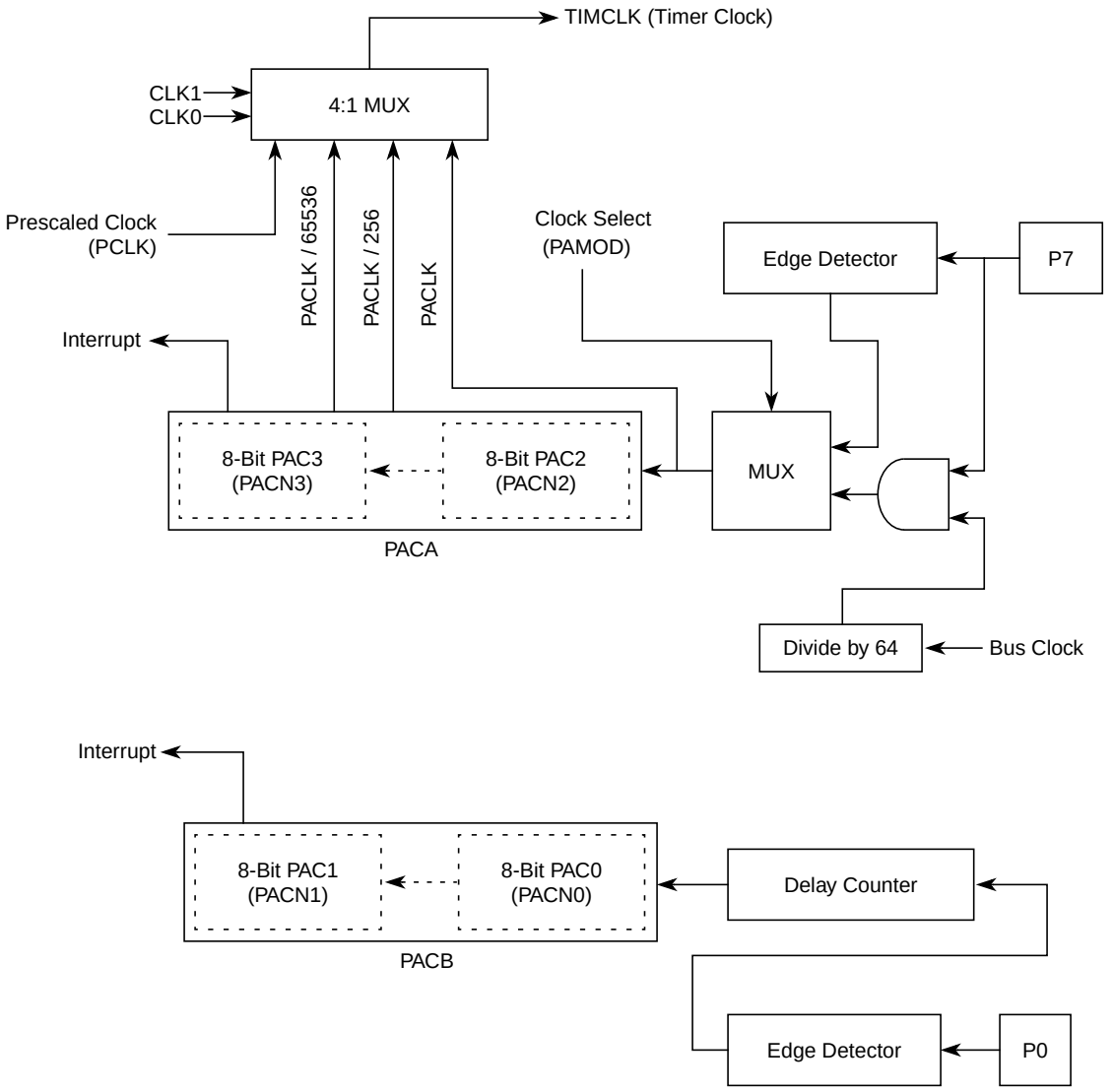

Figure 26-72. 16-Bit Pulse Accumulators Block Diagram

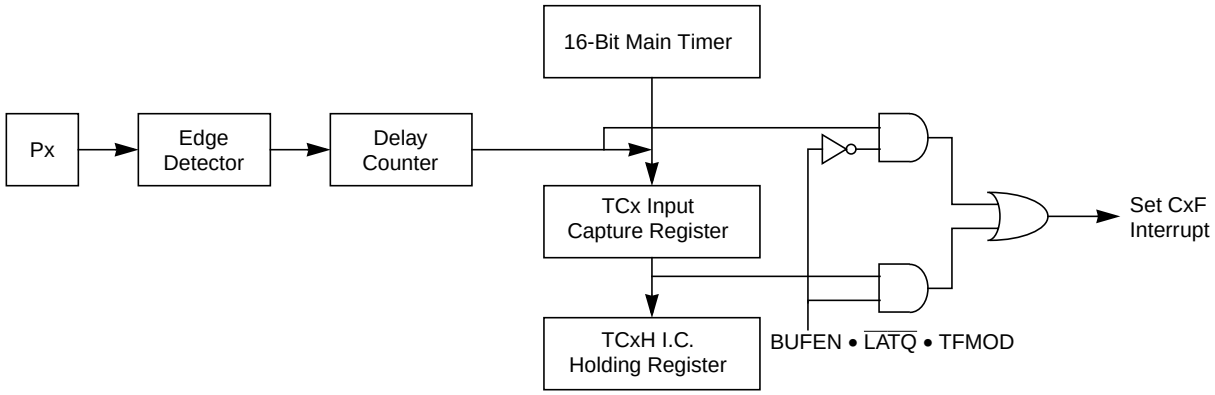

Figure 26-73. Block Diagram for Port 7 with Output Compare/Pulse Accumulator A

A.1.4 Current Injection

Power supply must maintain regulation within operating V_{DD35} or V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DD35}$) is greater than I_{DD35}, the injection current may flow out of V_{DD35} and could result in external power supply going out of regulation. Ensure external V_{DD35} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power; e.g., if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

0x0200–0x023F PMF

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0217	PMFVAL3	R	PMFVAL3							
		W								
0x0218	PMFVAL4	R	PMFVAL4							
		W								
0x0219	PMFVAL4	R	PMFVAL4							
		W								
0x021A	PMFVAL5	R	PMFVAL5							
		W								
0x021B	PMFVAL5	R	PMFVAL5							
		W								
0x021C - 0x021F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0220	PMFENCA	R	PWMENA	0	0	0	0	0	LDOKA	PWMRIEA
		W								
0x0221	PMFFQCA	R	LDFQA				HALFA	PRSCA		PWMRFA
		W								
0x0222	PMFCNTA	R	0	PMFCNTA						
		W								
0x0223	PMFCNTA	R	PMFCNTA							
		W								
0x0224	PMFMODA	R	0	PMFMODA						
		W								
0x0225	PMFMODA	R	PMFMODA							
		W								
0x0226	PMFDTMA	R	0	0	0	0	PMFDTMA			
		W								
0x0227	PMFDTMA	R	PMFDTMA							
		W								
0x0228	PMFENCB	R	PWMENB	0	0	0	0	0	LDOKB	PWMRIEB
		W								
0x0229	PMFFQCB	R	LDFQB				HALFB	PRSCB		PWMRFB
		W								
0x022A	PMFCNTB	R	0	PMFCNTB						
		W								
0x022B	PMFENCA	R	PMFCNTB							
		W								
0x022C	PMFMODB	R	0	PMFMODB						
		W								
0x022D	PMFMODB	R	PMFMODB							
		W								
0x022E	PMFDTMB	R	0	0	0	0	PMFDTMB			
		W								
0x022F	PMFDTMB	R	PMFDTMB							
		W								
0x0230	PMFENCC	R	PWMENC	0	0	0	0	0	LDOKC	PWMRIEC
		W								