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	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12kg256vpve

5. If the KEYACC bit does not remain set while the four 16-bit words are written.
6. If any two of the four 16-bit words are written on successive MCU clock cycles.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0xFF00–0xFF07 in the Flash configuration field.

The security as defined in the Flash security byte (0xFF0F) is not changed by using the backdoor key access sequence to unsecure. The backdoor keys stored in addresses 0xFF00–0xFF07 are unaffected by the backdoor key access sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0xFF0F). The backdoor key access sequence has no effect on the program and erase protections defined in the Flash protection register.

It is not possible to unsecure the MCU in special single-chip mode by using the backdoor key access sequence via the background debug mode (BDM).

2.6.2 Unsecuring the Flash Module in Special Single-Chip Mode using BDM

The MCU can be unsecured in special single-chip mode by erasing the Flash module by the following method:

- Reset the MCU into special single-chip mode, delay while the erase test is performed by the BDM secure ROM, send BDM commands to disable protection in the Flash module, and execute a mass erase command write sequence to erase the Flash memory.

After the CCIF flag sets to indicate that the mass operation has completed, reset the MCU into special single-chip mode. The BDM secure ROM will verify that the Flash memory is erased and will assert the UNSEC bit in the BDM status register. This BDM action will cause the MCU to override the Flash security state and the MCU will be unsecured. All BDM commands will be enabled and the Flash security byte may be programmed to the unsecure state by the following method:

- Send BDM commands to execute a word program sequence to program the Flash security byte to the unsecured state and reset the MCU.

2.7 Resets

2.7.1 Flash Reset Sequence

On each reset, the Flash module executes a reset sequence to hold CPU activity while loading the following registers from the Flash memory according to Table 2-2:

- FPROT — Flash Protection Register (see Section 2.3.2.5).

  If a double bit fault is detected during the read of the protection field as part of the reset sequence, the FPOPEN bit in the FPROT register will be clear and remaining bits will be set leaving the Flash block fully protected from program and erase.
- FCTL — Flash Control Register (see Section 2.3.2.9).

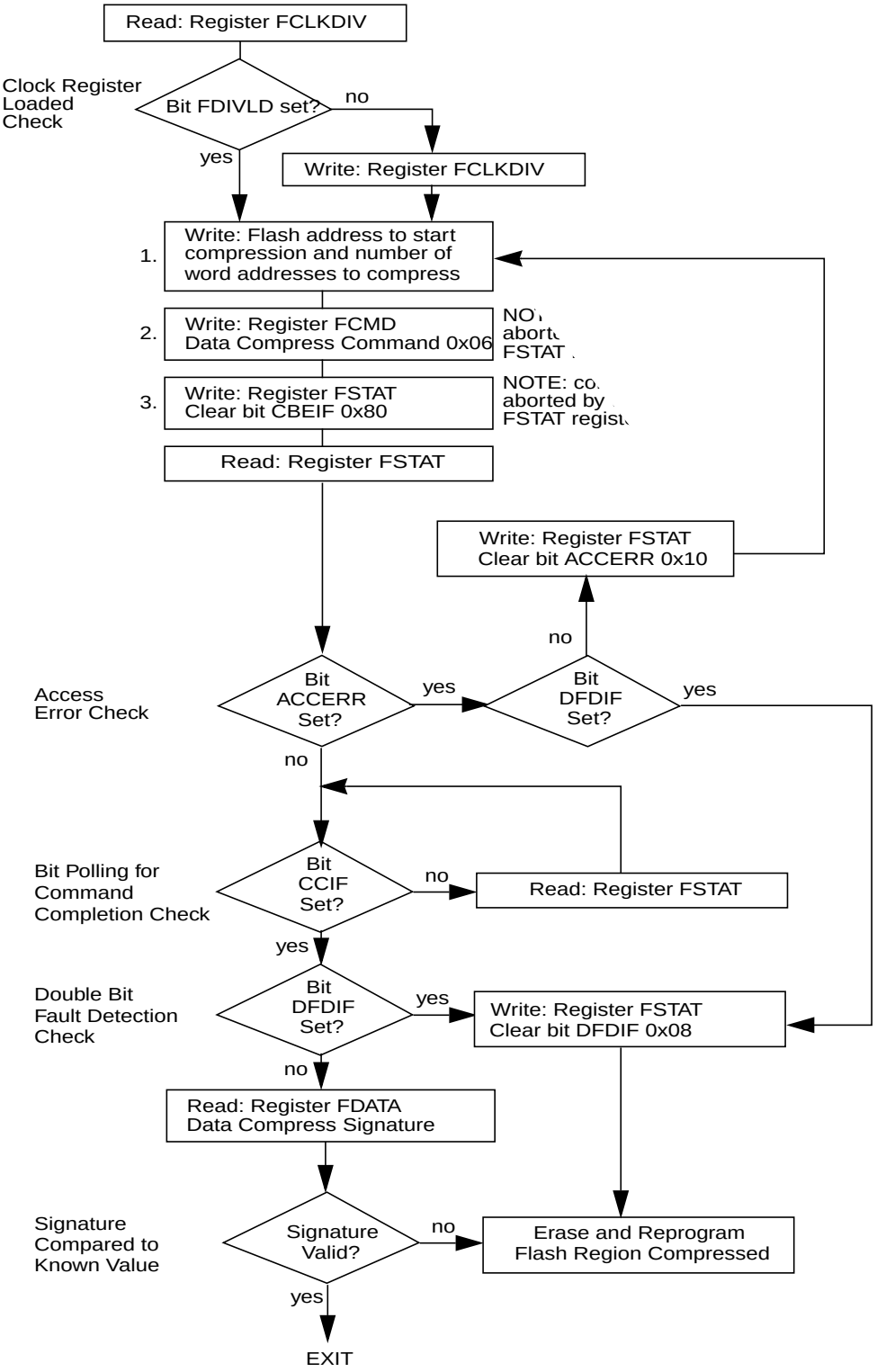

Figure 3-25. Example Data Compress Command Flow

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	EPOPEN	NV6	NV5	NV4	EPDIS	EP2	EP1	EP0
W								
Reset	F	F	F	F	F	F	F	F

= Unimplemented or Reserved

Figure 4-8. EEPROM Protection Register (EPROT)

The EPROT register is loaded from EEPROM array address 0x0FFD during reset, as indicated by the F in Figure 4-8.

All bits in the EPROT register are readable. Bits NV[6:4] are not writable. The EPOPEN and EPDIS bits in the EPROT register can only be written to the protected state (i.e., 0). The EP[2:0] bits can be written anytime until bit EPDIS is cleared. If the EPOPEN bit is cleared, then the state of the EPDIS and EP[2:0] bits is irrelevant.

To change the EEPROM protection that will be loaded on reset, the upper sector of EEPROM must first be unprotected, then the EEPROM protect byte located at address 0x0FFD must be written to.

A protected EEPROM sector is disabled by the EPDIS bit while the size of the protected sector is defined by the EP bits in the EPROT register.

Trying to alter any of the protected areas will result in a protect violation error and PVIOL flag will be set in the ESTAT register. A mass erase of a whole EEPROM block is only possible when protection is fully disabled by setting the EPOPEN and EPDIS bits. An attempt to mass erase an EEPROM block while protection is enabled will set the PVIOL flag in the ESTAT register.

Table 4-5. EPROT Field Descriptions

Field	Description
7 EPOPEN	**Opens EEPROM for Program or Erase** 0 The whole EEPROM array is protected. In this case, the EPDIS and EP bits within the protection register are ignored. 1 The EEPROM sectors not protected are enabled for program or erase.
6:4 NV[6:4]	**Nonvolatile Flag Bits** — These three bits are available to the user as nonvolatile flags.
3 EPDIS	**EEPROM Protection Address Range Disable** — The EPDIS bit determines whether there is a protected area in the space of the EEPROM address map. 0 Protection enabled 1 Protection disabled
2:0 EP[2:0]	**EEPROM Protection Address Size** — The EP[2:0] bits determine the size of the protected sector. Refer to Table 4-6.

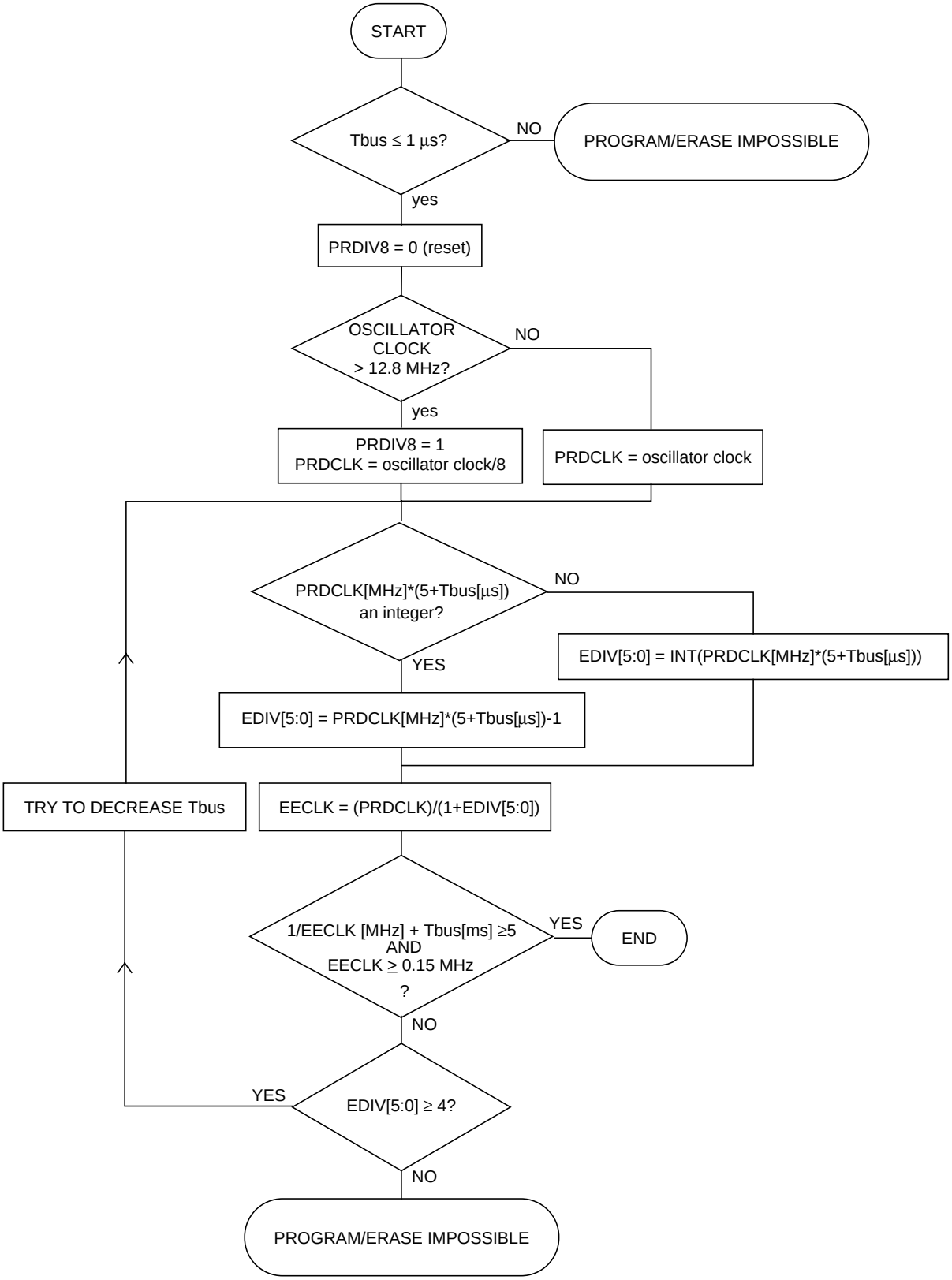

Figure 4-17. PRDIV8 and EDIV Bits Determination Procedure

Table 5-1. Pin Functions and Priorities (Sheet 4 of 4)

Port	Pin Name	Pin Function and Priority	Description	Pull Mode after Reset	Pin Function after Reset
Port H	PH7	SS2	Serial Peripheral Interface 2 slave select output in master mode, input in slave mode or master mode.	Hi-Z	GPIO
		GPIO/KWH7	General-purpose I/O with interrupt		
	PH6	SCK2	Serial Peripheral Interface 2 serial clock pin		
		GPIO/KWH6	General-purpose I/O with interrupt		
	PH5	MOSI2	Serial Peripheral Interface 2 master out/slave in pin		
		GPIO/KWH5	General-purpose I/O with interrupt		
	PH4	MISO2	Serial Peripheral Interface 2 master in/slave out pin		
		GPIO/KWH4	General-purpose I/O with interrupt		
	PH3	SS1	Serial Peripheral Interface 1 slave select output in master mode, input in slave mode or master mode.		
		GPIO/KWH3	General-purpose I/O with interrupt		
	PH2	SCK1	Serial Peripheral Interface 1 serial clock pin		
		GPIO/KWH2	General-purpose I/O with interrupt		
	PH1	MOSI1	Serial Peripheral Interface 1 master out/slave in pin		
		GPIO/KWH1	General-purpose I/O with interrupt		
	PH0	MISO1	Serial Peripheral Interface 1 master in/slave out pin		
		GPIO/KWH0	General-purpose I/O with interrupt		
Port J	PJ7	TXCAN4	MSCAN4 transmit pin	Pull-up	GPIO
		SCL	Inter Integrated Circuit serial clock line		
		GPIO/KWJ7	General-purpose I/O with interrupt		
	PJ6	RXCAN4	MSCAN4 receive pin		
		SDA	Inter Integrated Circuit serial data line		
		GPIO/KWJ6	General-purpose I/O with interrupt		
	PJ1	GPIO/KWJ1	General-purpose I/O with interrupt		
	PJ0	GPIO/KWJ0	General-purpose I/O with interrupt		

[1] If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.

5.3.3.5 Port M Pull Device Enable Register (PERM)

Module Base + 0x0014

	7	6	5	4	3	2	1	0
R W	PERM7	PERM6	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
Reset	0	0	0	0	0	0	0	0

Figure 5-19. Port M Pull Device Enable Register (PERM)

Read: Anytime. Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or wired-OR output. This bit has no effect if the port is used as push-pull output. Out of reset no pull device is enabled.

Table 5-14. PERM Field Descriptions

Field	Description
7–0 PERM[7:0]	**Pull Device Enable Port M** 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

5.3.3.6 Port M Polarity Select Register (PPSM)

Module Base + 0x0015

	7	6	5	4	3	2	1	0
R W	PPSM7	PPSM6	PPSM5	PPSM4	PPSM3	PPSM2	PPSM1	PPSM0
Reset	0	0	0	0	0	0	0	0

Figure 5-20. Port M Polarity Select Register (PPSM)

Read: Anytime. Write: Anytime.

This register selects whether a pull-down or a pull-up device is connected to the pin. If CAN is active a pull-up device can be activated on the receiver inputs, but not a pull-down.

Table 5-15. PPSM Field Descriptions

Field	Description
7–0 PPSM[7:0]	**Pull Select Port M** 0 A pull-up device is connected to the associated port M pin, if enabled by the associated bit in register PERM and if the port is used as general purpose, RXCAN input. 1 A pull-down device is connected to the associated port M pin, if enabled by the associated bit in register PERM and if the port is used as a general purpose but not as RXCAN.

5.3.5.7 Port H Interrupt Enable Register (PIEH)

Module Base + 0x0026

	7	6	5	4	3	2	1	0
R W	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
Reset	0	0	0	0	0	0	0	0

Figure 5-37. Port H Interrupt Enable Register (PIEH)

Read: Anytime. Write: Anytime.

This register disables or enables on a per pin basis the edge sensitive external interrupt associated with port H.

Table 5-34. PIEH Field Descriptions

Field	Description
7–0 PIEH[7:0]	**Interrupt Enable Port H** 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

5.3.5.8 Port H Interrupt Flag Register (PIFH)

Module Base + 0x0027

	7	6	5	4	3	2	1	0
R W	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
Reset	0	0	0	0	0	0	0	0

Figure 5-38. Port H Interrupt Flag Register (PIFH)

Read: Anytime. Write: Anytime.

Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSH register. To clear this flag, write “1” to the corresponding bit in the PIFH register. Writing a “0” has no effect.

Table 5-35. PIFH Field Descriptions

Field	Description
7–0 PIFH[7:0]	**Interrupt Flags Port H** 0 No active edge pending. Writing a “0” has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). Writing a “1” clears the associated flag.

5.3.6.3 Port J Data Direction Register (DDRJ)

Module Base + 0x002A

	7	6	5	4	3	2	1	0
R	DDRJ7	DDRJ6	0	0	0	0	DDRJ1	DDRJ0
W								
Reset	0	0	—	—	—	—	0	0

= Unimplemented or Reserved

Figure 5-41. Port J Data Direction Register (DDRJ)

Read: Anytime. Write: Anytime.

This register configures each port J pin as either input or output.

If enable, CAN4 forces the I/O state to be an output on PJ7 (TXCAN4) and an input on pin PJ6 (RXCAN4). If CAN4 is disabled, the IIC takes control of the I/O if enabled. In these cases the data direction bits will not change.

The DDRJ bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled.

Table 5-36. Field Descriptions

Field	Description
7, 6, 1, 0 DDRJ[7:6] DDRJ[1:0]	**Data Direction Port J** 0 Associated pin is configured as input. 1 Associated pin is configured as output.

5.3.6.4 Port J Reduced Drive Register (RDRJ)

Module Base + 0x002B

	7	6	5	4	3	2	1	0
R	RDRJ7	RDRJ6	0	0	0	0	RDRJ1	RDRJ0
W								
Reset	0	0	—	—	—	—	0	0

= Unimplemented or Reserved

Figure 5-42. Port J Reduced Drive Register (RDRJ)

Read: Anytime. Write: Anytime.

This register configures the drive strength of each port J output pin as either full or reduced. If the port is used as input this bit is ignored.

Table 5-37. RDRJ Field Descriptions

Field	Description
7, 6, 1, 0 RDRJ[7:6] RDRJ[1:0]	**Reduced Drive Port J** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

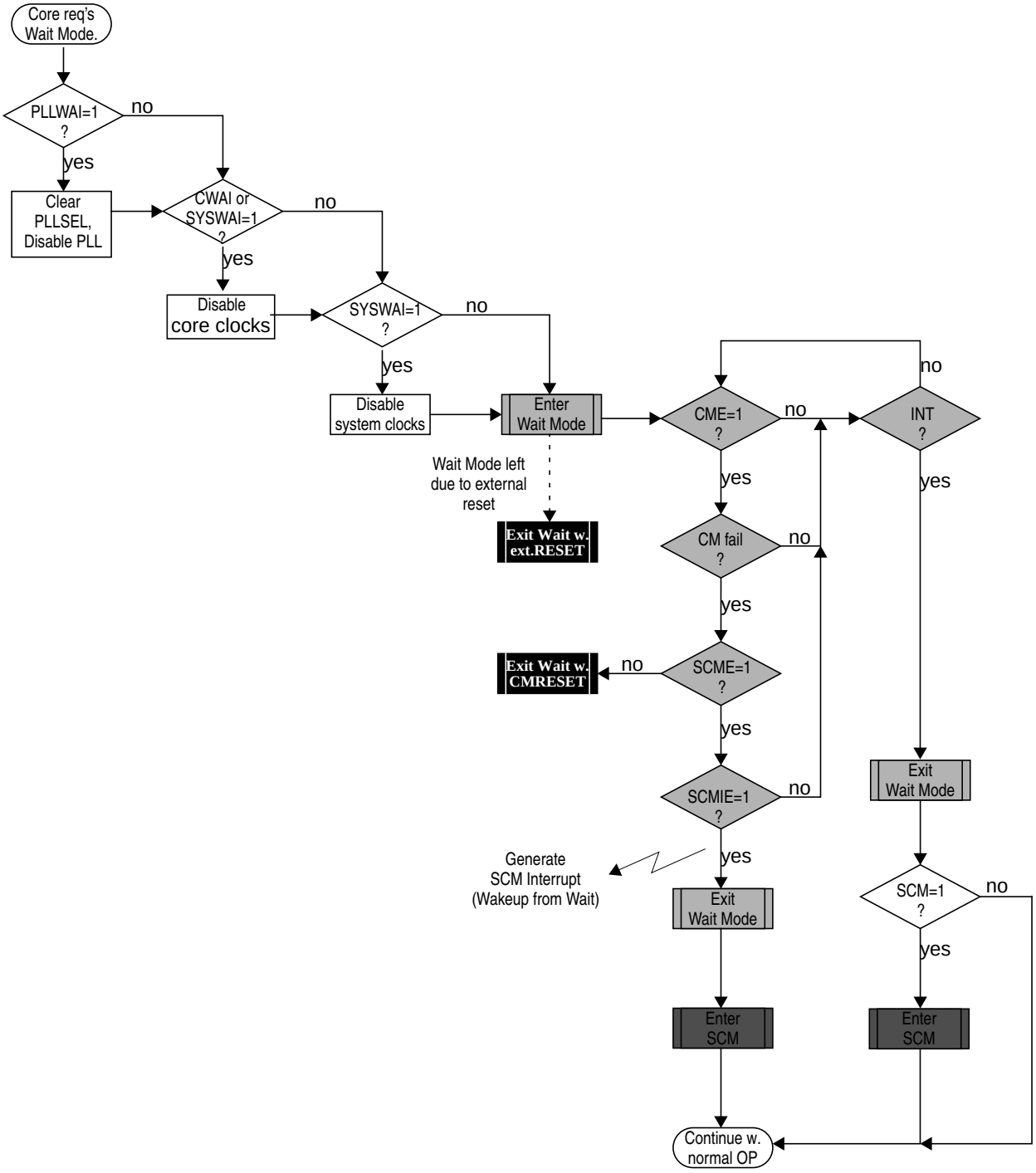

Figure 6-23. Wait Mode Entry/Exit Sequence

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	CANE	CLKSRC	LOOPB	LISTEN		WUPM	SLPAK	INITAK
W								
Reset:	0	0	0	1	0	0	0	1

= Unimplemented

Figure 10-5. MSCAN Control Register 1 (CANCTL1)

Read: Anytime
Write: Anytime when INITRQ = 1 and INITAK = 1, except CANE which is write once in normal and anytime in special system operation modes when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1).

Table 10-2. CANCTL1 Register Field Descriptions

Field	Description
7 CANE	**MSCAN Enable** 0 MSCAN module is disabled 1 MSCAN module is enabled
6 CLKSRC	**MSCAN Clock Source** — This bit defines the clock source for the MSCAN module (only for systems with a clock generation module; Section 10.4.3.2, "Clock System," and Section Figure 10-42., "MSCAN Clocking Scheme,"). 0 MSCAN clock source is the oscillator clock 1 MSCAN clock source is the bus clock
5 LOOPB	**Loopback Self Test Mode** — When this bit is set, the MSCAN performs an internal loopback which can be used for self test operation. The bit stream output of the transmitter is fed back to the receiver internally. The RXCAN input pin is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 0 Loopback self test disabled 1 Loopback self test enabled
4 LISTEN	**Listen Only Mode** — This bit configures the MSCAN as a CAN bus monitor. When LISTEN is set, all valid CAN messages with matching ID are received, but no acknowledgement or error frames are sent out (see Section 10.4.4.4, "Listen-Only Mode"). In addition, the error counters are frozen. Listen only mode supports applications which require "hot plugging" or throughput analysis. The MSCAN is unable to transmit any messages when listen only mode is active. 0 Normal operation 1 Listen only mode activated
2 WUPM	**Wake-Up Mode** — If WUPE in CANCTL0 is enabled, this bit defines whether the integrated low-pass filter is applied to protect the MSCAN from spurious wake-up (see Section 10.4.5.4, "MSCAN Sleep Mode"). 0 MSCAN wakes up on any dominant level on the CAN bus 1 MSCAN wakes up only in case of a dominant pulse on the CAN bus that has a length of T_{wup}

10.3.3.3 Data Length Register (DLR)

This register keeps the data length field of the CAN frame.

Module Base + 0x00XB

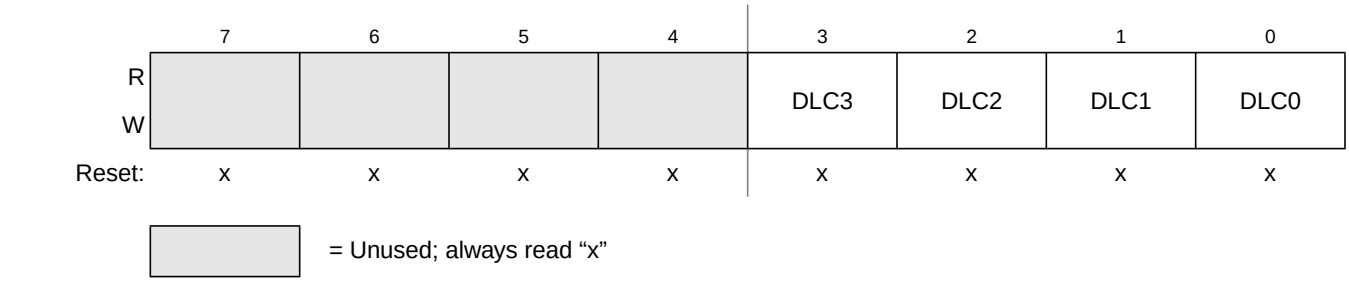

Figure 10-34. Data Length Register (DLR) — Extended Identifier Mapping

Table 10-31. DLR Register Field Descriptions

Field	Description
3:0 DLC[3:0]	**Data Length Code Bits** — The data length code contains the number of bytes (data byte count) of the respective message. During the transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted data bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 10-32 shows the effect of setting the DLC bits.

Table 10-32. Data Length Codes

Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8

10.3.3.4 Transmit Buffer Priority Register (TBPR)

This register defines the local priority of the associated message buffer. The local priority is used for the internal prioritization process of the MSCAN and is defined to be highest for the smallest binary number. The MSCAN implements the following internal prioritization mechanisms:

- All transmission buffers with a cleared TXEx flag participate in the prioritization immediately before the SOF (start of frame) is sent.
- The transmission buffer with the lowest local priority field wins the prioritization.

11.3.2.6 SCI Data Registers (SCIDRH and SCIDRL)

Module Base + 0x_0006

	7	6	5	4	3	2	1	0
R	R8	T8	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Module Base + 0x_0007

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 11-8. SCI Data Registers (SCIDRH and SCIDRL)

Read: Anytime; reading accesses SCI receive data register

Write: Anytime; writing accesses SCI transmit data register; writing to R8 has no effect

Table 11-7. SCIDRH AND SCIDRL Field Descriptions

Field	Description
7 R8	**Received Bit 8** — R8 is the ninth data bit received when the SCI is configured for 9-bit data format (M = 1).
6 T8	**Transmit Bit 8** — T8 is the ninth data bit transmitted when the SCI is configured for 9-bit data format (M = 1).
7–0 R[7:0] T[7:0]	**Received Bits** — Received bits seven through zero for 9-bit or 8-bit data formats **Transmit Bits** — Transmit bits seven through zero for 9-bit or 8-bit formats

NOTE

If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten.The same value is transmitted until T8 is rewritten

In 8-bit data format, only SCI data register low (SCIDRL) needs to be accessed.

When transmitting in 9-bit data format and using 8-bit write instructions, write first to SCI data register high (SCIDRH), then SCIDRL.

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

13.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

13.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

13.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 13.4.1, “PWM Clock Select” for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 13-19. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 13-19 and described in Section 13.4.2.5, “Left Aligned Outputs” and Section 13.4.2.6, “Center Aligned Outputs”.

Table 14-5. OC7M Field Descriptions

Field	Description
7:0 OC7M[7:0]	**Output Compare 7 Mask** — Setting the OC7Mx (x ranges from 0 to 6) will set the corresponding port to be an output port when the corresponding TIOSx (x ranges from 0 to 6) bit is set to be an output compare. **Note:** A successful channel 7 output compare overrides any channel 6:0 compares. For each OC7M bit that is set, the output compare action reflects the corresponding OC7D bit.

14.3.2.4 Output Compare 7 Data Register (OC7D)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R W	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
Reset	0	0	0	0	0	0	0	0

Figure 14-9. Output Compare 7 Data Register (OC7D)

Read: Anytime

Write: Anytime

Table 14-6. OC7D Field Descriptions

Field	Description
7:0 OC7D[7:0]	**Output Compare 7 Data** — A channel 7 output compare can cause bits in the output compare 7 data register to transfer to the timer port data register depending on the output compare 7 mask register.

14.3.2.5 Timer Count Register (TCNT)

Module Base + 0x0004

	15	14	13	12	11	10	9	9
R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
Reset	0	0	0	0	0	0	0	0

Figure 14-10. Timer Count Register High (TCNTH)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Reset	0	0	0	0	0	0	0	0

Figure 14-11. Timer Count Register Low (TCNTL)

16.3.2.2 BDM CCR Holding Register (BDMCCR)

0xFF06

	7	6	5	4	3	2	1	0
R W	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
Reset	0	0	0	0	0	0	0	0

Figure 16-4. BDM CCR Holding Register (BDMCCR)

Read: All modes

Write: All modes

NOTE

When BDM is made active, the CPU stores the value of the CCR register in the BDMCCR register. However, out of special single-chip reset, the BDMCCR is set to 0xD8 and not 0xD0 which is the reset value of the CCR register.

When entering background debug mode, the BDM CCR holding register is used to save the contents of the condition code register of the user's program. It is also used for temporary storage in the standard BDM firmware mode. The BDM CCR holding register can be written to modify the CCR value.

16.3.2.3 BDM Internal Register Position Register (BDMINR)

0xFF07

	7	6	5	4	3	2	1	0
R	0	REG14	REG13	REG12	REG11	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-5. BDM Internal Register Position (BDMINR)

Read: All modes

Write: Never

Table 16-4. BDMINR Field Descriptions

Field	Description
6:3 REG[14:11]	**Internal Register Map Position** — These four bits show the state of the upper five bits of the base address for the system's relocatable register block. BDMINR is a shadow of the INITRG register which maps the register block to any 2K byte space within the first 32K bytes of the 64K byte address space.

— Data associated with event B trigger modes
— Detail report mode stores address and data for all cycles except program (P) and free (f) cycles
— Current instruction address when in profiling mode
— BGND is not considered a change-of-flow (cof) by the debugger

17.1.2 Modes of Operation

There are two main modes of operation: breakpoint mode and debug mode. Each one is mutually exclusive of the other and selected via a software programmable control bit.

In the breakpoint mode there are two sub-modes of operation:

- Dual address mode, where a match on either of two addresses will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).
- Full breakpoint mode, where a match on address and data will cause the system to enter background debug mode (BDM) or initiate a software interrupt (SWI).

In debug mode, there are several sub-modes of operation.

- Trigger modes

  There are many ways to create a logical trigger. The trigger can be used to capture bus information either starting from the trigger or ending at the trigger. Types of triggers (A and B are registers):

  — A only
  — A or B
  — A then B
  — Event only B (data capture)
  — A then event only B (data capture)
  — A and B, full mode
  — A and not B, full mode
  — Inside range
  — Outside range

- Capture modes

  There are several capture modes. These determine which bus information is saved and which is ignored.

  — Normal: save change-of-flow program fetches
  — Loop1: save change-of-flow program fetches, ignoring duplicates
  — Detail: save all bus operations except program and free cycles
  — Profile: poll target from external device

17.1.3 Block Diagram

Figure 17-1 is a block diagram of this module in breakpoint mode. Figure 17-2 is a block diagram of this module in debug mode.

17.3.2.3 Debug Trace Buffer Register (DBGTB)

Module Base + 0x0022
Starting address location affected by INITRG register setting.

	15	14	13	12	11	10	9	8
R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
Reset	u	u	u	u	u	u	u	u

= Unimplemented or Reserved

Figure 17-6. Debug Trace Buffer Register High (DBGTBH)

Module Base + 0x0023
Starting address location affected by INITRG register setting.

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

= Unimplemented or Reserved

Figure 17-7. Debug Trace Buffer Register Low (DBGTBL)

Table 17-7. DBGTB Field Descriptions

Field	Description
15:0	**Trace Buffer Data Bits** — The trace buffer data bits contain the data of the trace buffer. This register can be read only as a word read. Any byte reads or misaligned access of these registers will return 0 and will not cause the trace buffer pointer to increment to the next trace buffer address. The same is true for word reads while the debugger is armed. In addition, this register may appear to contain incorrect data if it is not read with the same capture mode bit settings as when the trace buffer data was recorded (See Section 17.4.2.9, “Reading Data from Trace Buffer”). Because reads will reflect the contents of the trace buffer RAM, the reset state is undefined.

19.3.2.16 Port K Data Direction Register (DDRK)

Module Base + 0x0033
Starting address location affected by INITRG register setting.

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

Figure 19-20. Port K Data Direction Register (DDRK)

Read: Anytime

Write: Anytime

This register determines the primary direction for each port K pin configured as general-purpose I/O. This register is not in the map in peripheral or expanded modes while the EMK control bit in MODE register is set. Therefore, these accesses will be echoed externally.

Table 19-14. EBICTL Field Descriptions

Field	Description
7:0 DDRK	**Data Direction Port K Bits** 0 Associated pin is a high-impedance input 1 Associated pin is an output **Note:** It is unwise to write PORTK and DDRK as a word access. If you are changing port K pins from inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTK before enabling as outputs. **Note:** To ensure that you read the correct value from the PORTK pins, always wait at least one cycle after writing to the DDRK register before reading from the PORTK register.

19.4 Functional Description

19.4.1 Detecting Access Type from External Signals

The external signals $\overline{\text{LSTRB}}$, R/$\overline{\text{W}}$, and AB0 indicate the type of bus access that is taking place. Accesses to the internal RAM module are the only type of access that would produce $\overline{\text{LSTRB}}$ = AB0 = 1, because the internal RAM is specifically designed to allow misaligned 16-bit accesses in a single cycle. In these cases the data for the address that was accessed is on the low half of the data bus and the data for address + 1 is on the high half of the data bus. This is summarized in Table 19-15.

Table 19-15. Access Type vs. Bus Control Pins

$\overline{\text{LSTRB}}$	AB0	R/$\overline{\text{W}}$	Type of Access
1	0	1	8-bit read of an even address
0	1	1	8-bit read of an odd address
1	0	0	8-bit write of an even address
0	1	0	8-bit write of an odd address

Table 20-20. 48K Byte Physical FLASH/ROM Allocated

Address Space	Page Window Access	ROMHM	ECS	XAB19:14
0x0000–0x3FFF	N/A	N/A	1	0x3D
0x4000–0x7FFF	N/A	0	0	0x3E
	N/A	1	1	
0x8000–0xBFFF	External	N/A	1	PIX[5:0]
	Internal	N/A	0	
0xC000–0xFFFF	N/A	N/A	0	0x3F

Table 20-21. 64K Byte Physical FLASH/ROM Allocated

Address Space	Page Window Access	ROMHM	ECS	XAB19:14
0x0000–0x3FFF	N/A	0	0	0x3D
	N/A	1	1	
0x4000–0x7FFF	N/A	0	0	0x3E
	N/A	1	1	
0x8000–0xBFFF	External	N/A	1	PIX[5:0]
	Internal	N/A	0	
0xC000–0xFFFF	N/A	N/A	0	0x3F