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	59
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/mcs12kg256vfue

0x001C–0x001D MMC Map 3 of 4 (HCS12 Module Mapping Control, Device Guide)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x001C	MEMSIZ0	R	reg_sw0	0	eep_sw1	eep_sw0	0	ram_sw2	ram_sw1	ram_sw0
		W								
0x001D	MEMSIZ1	R	rom_sw1	rom_sw0	0	0	0	0	pag_sw1	pag_sw0
		W								

0x001E–0x001E MEBI Map 2 of 3 (HCS12 Multiplexed External Bus Interface)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x001E	INTCR	R	IRQE	IRQEN	0	0	0	0	0	0
		W								

0x001F–0x001F INT Map 2 of 2 (HCS12 Interrupt)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x001F	HPRIO	R	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
		W								

0x0020–0x002F DBG Map 1 of 1 (HCS12 Debug)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0020	DBGC1	R	DBGEN	ARM	TRGSEL	BEGIN	DBGBRK	0	CAPMOD	
	—	W								
0x0021	DBGSC	R	AF	BF	CF	0	TRG			
	—	W								
0x0022	DBGTBH	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	—	W								
0x0023	DBGTBL	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	—	W								
0x0024	DBGCNT	R	TBF	0	CNT					
	—	W								
0x0025	DBGCCX	R	PAGSEL		EXTCMP					
	—	W								
0x0026	DBGCCH	R	Bit 15	14	13	12	11	10	9	Bit 8
	—	W								
0x0027	DBGCCL	R	Bit 7	6	5	4	3	2	1	Bit 0
	—	W								
0x0028	DBGC2	R	BKABEN	FULL	BDM	TAGAB	BKCEN	TAGC	RWCEN	RWC
	BKPCT0	W								
0x0029	DBGC3	R	BKAMBH	BKAMBL	BKBMBH	BKBMBL	RWAEN	RWA	RWBEN	RWB
	BKPCT1	W								
0x002A	DBGCAX	R	PAGSEL		EXTCMP					
	BKP0X	W								
0x002B	DBGCAH	R	Bit 15	14	13	12	11	10	9	Bit 8
	BKP0H	W								

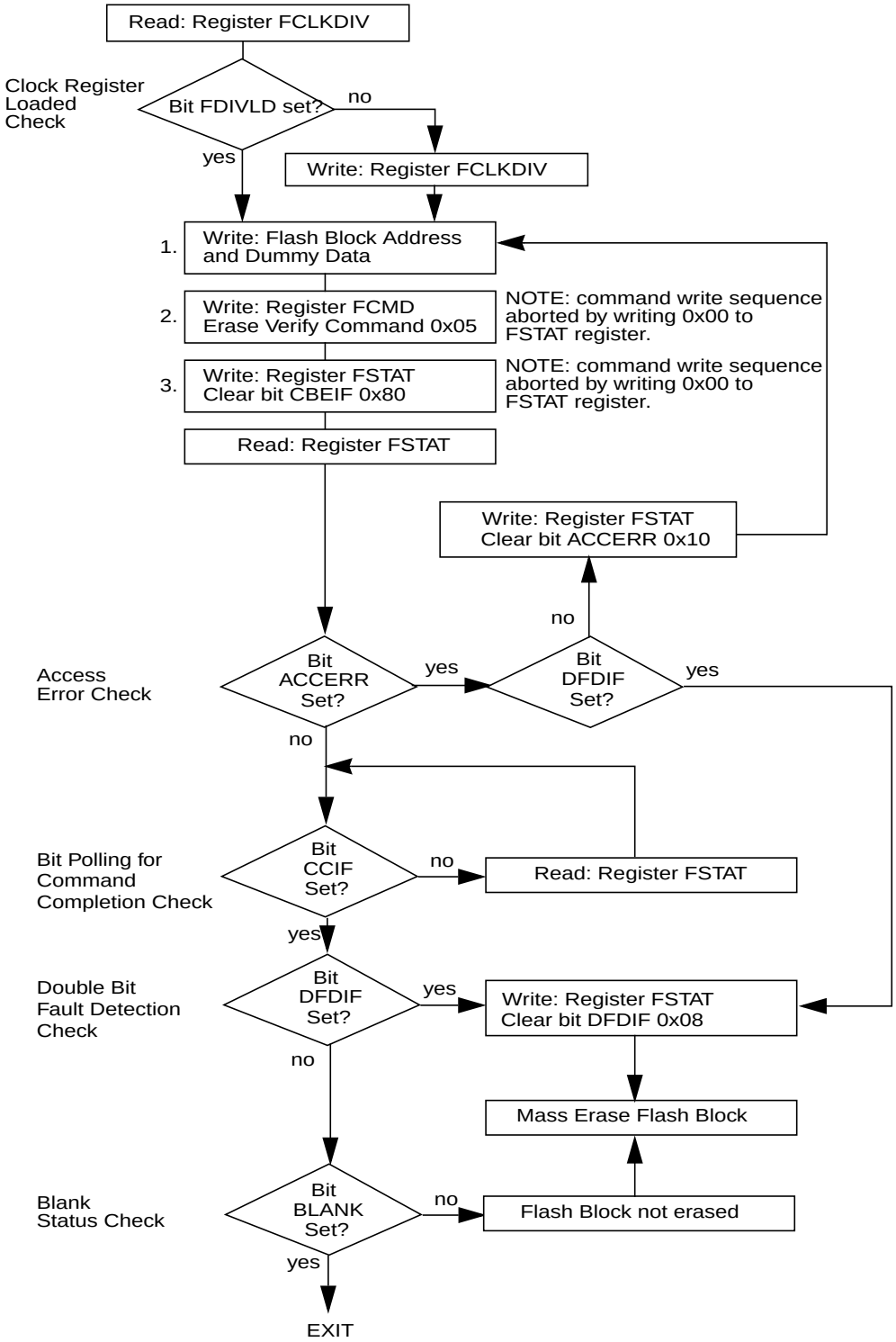


Figure 2-24. Example Erase Verify Command Flow

2.4.1.3.4 Sector Erase Command

The sector erase command is used to erase the addressed sector in the Flash memory using an embedded algorithm. If the Flash sector to be erased is in a protected area of the Flash block, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. After the sector erase command has successfully launched, the CCIF flag in the FSTAT register will set after the sector erase operation has completed unless a second command has been buffered.

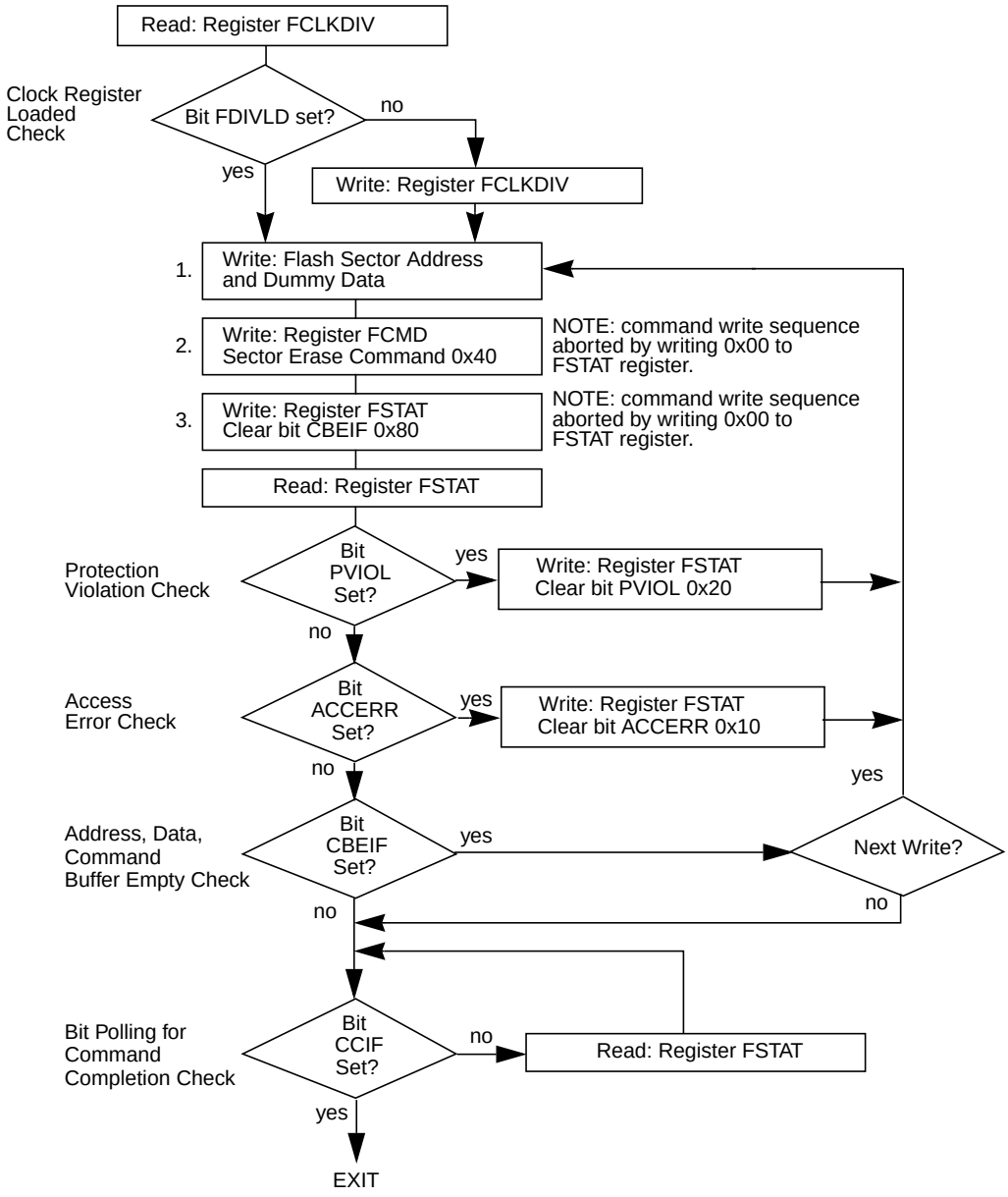


Figure 2-28. Example Sector Erase Command Flow

3.1.1 Glossary

Banked Register — A memory-mapped register operating on one Flash block which shares the same register address as the equivalent registers for the other Flash blocks. The active register bank is selected by the BKSEL bit in the FCNFG register.

Command Write Sequence — A three-step MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

Common Register — A memory-mapped register which operates on all Flash blocks.

3.1.2 Features

- 256 Kbytes of Flash memory comprised of two 128 Kbyte blocks with each block divided into 128 sectors of 1024 bytes with every word (two bytes) accompanied by 6 ECC parity bits
- Single bit fault correction per word during read operations
- Automated program and erase algorithm with generation of ECC parity bits
- Interrupts on Flash command completion, command buffer empty and double bit fault detection
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Sector erase abort feature for critical interrupt response
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for all Flash operations including program and erase
- Security feature to prevent unauthorized access to the Flash memory
- Code integrity check using built-in data compression

3.1.3 Modes of Operation

Program, erase, erase verify, and data compress operations (please refer to Section 3.4.1 for details).

3.1.4 Block Diagram

A block diagram of the Flash module is shown in Figure 3-1.

3.4.1.3.5 Mass Erase Command

The mass erase command is used to erase a Flash memory block using an embedded algorithm. If the Flash block to be erased contains any protected area, the PVIOL flag in the FSTAT register will set and the mass erase command will not launch. After the mass erase command has successfully launched, the CCIF flag in the FSTAT register will set after the mass erase operation has completed unless a second command has been buffered.

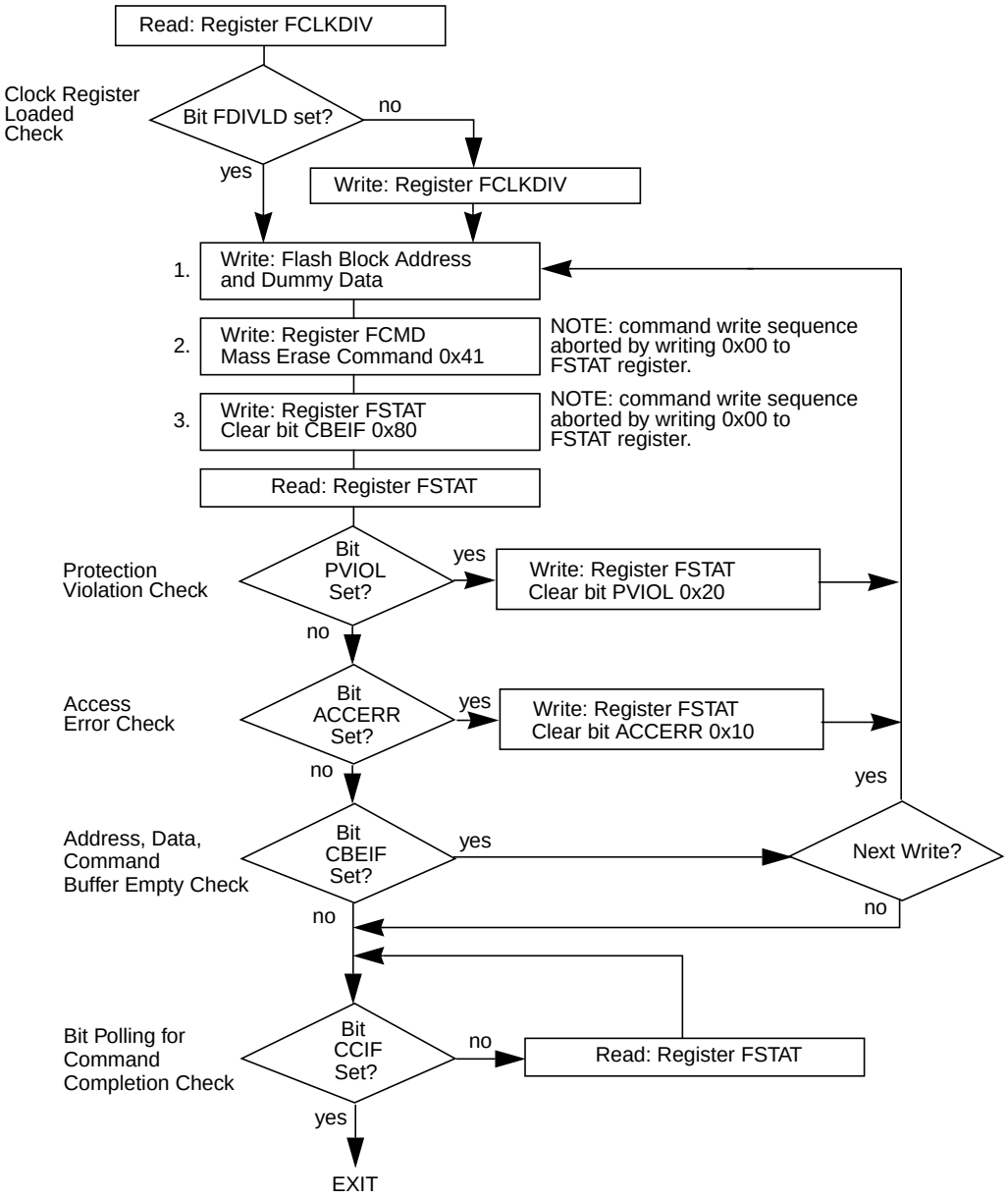


Figure 3-29. Example Mass Erase Command Flow

3.4.1.4 Illegal Flash Operations

The ACCERR flag will be set during the command write sequence if any of the following illegal steps are performed, causing the command write sequence to immediately abort:

1. Writing to a Flash address before initializing the FCLKDIV register.
2. Writing to a Flash address in the range 0x8000–0xBFFF when the PPAGE register does not select a 16 Kbyte page in the Flash block selected by the BKSEL bit in the FCNFG register.
3. Writing to a Flash address in the range 0x4000–0x7FFF or 0xC000–0xFFFF with the BKSEL bit in the FCNFG register not selecting Flash block 0.
4. Writing a byte or misaligned word to a valid Flash address.
5. Starting a command write sequence while a data compress operation is active.
6. Starting a command write sequence while a sector erase abort operation is active.
7. Writing a second word to a Flash address in the same command write sequence.
8. Writing to any Flash register other than FCMD after writing a word to a Flash address.
9. Writing a second command to the FCMD register in the same command write sequence.
10. Writing an invalid command to the FCMD register.
11. When security is enabled, writing a command other than mass erase to the FCMD register when the write originates from a non-secure memory location or from the Background Debug Mode.
12. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the FCMD register.
13. Writing a 0 to the CBEIF flag in the FSTAT register to abort a command write sequence.

The ACCERR flag will not be set if any Flash register is read during a valid command write sequence.

The ACCERR flag will also be set if any of the following events occur:

1. Launching the sector erase abort command while a sector erase operation is active which results in the early termination of the sector erase operation (see Section 3.4.1.3.6, “Sector Erase Abort Command”)
2. A double bit fault is detected in any of the following Flash operations:
   a) Array read
   b) Erase Verify
   c) Data Compress
   d) Reset Sequence Array Read (Configuration Field)
3. The MCU enters stop mode and a program or erase operation is in progress. The operation is aborted immediately and any pending command is purged (see Section 3.5.2, “Stop Mode”).

If the Flash memory is read during execution of an algorithm (i.e., CCIF flag in the FSTAT register is low), the read operation will return invalid data and the ACCERR flag will not be set.

If the ACCERR flag is set in any of the banked FSTAT registers, the user must clear the ACCERR flag in all of the banked FSTAT registers before starting another command write sequence (see Section 3.3.2.7, “Flash Status Register (FSTAT)”).

5.3.3.7 Port M Wired-OR Mode Register (WOMM)

Module Base + 0x0016

	7	6	5	4	3	2	1	0
R W	WOMM7	WOMM6	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
Reset	0	0	0	0	0	0	0	0

Figure 5-21. Port M Wired-OR Mode Register (WOMM)

Read: Anytime. Write: Anytime.

This register configures the output pins as wired-OR. If enabled the output is driven active low only (open-drain). A logic level of "1" is not driven. It applies also to the CAN outputs and allows a multipoint connection of several serial modules. This bit has no influence on pins used as inputs.

Table 5-16. WOMM Field Descriptions

Field	Description
7–0 WOMM[7:0]	**Wired-OR Mode Port M** 0 Output buffers operate as push-pull outputs. 1 Output buffers operate as open-drain outputs.

5.3.3.8 Module Routing Register (MODRR)

Module Base + 0x0017

	7	6	5	4	3	2	1	0
R	0	MODRR6	MODRR5	MODRR4	MODRR3	MODRR2	MODRR1	MODRR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-22. Module Routing Register (MODRR)

Read: Anytime. Write: Anytime.

This register configures the re-routing of CAN0, CAN4, SPI0, SPI1 and SPI2 on defined port pins.

Table 5-17. MODRR Field Descriptions

Field	Description
6 MODRR6	**SPI2 Routing Bit** — See Table 5-22.
5 MODRR5	**SPI1 Routing Bit** — See Table 5-21.
4 MODRR4	**SPI0 Routing Bit** — See Table 5-20.

Table 5-23. Implemented Modules on Derivatives

Number of Modules	MSCAN Modules			SPI Modules		
	CAN0	CAN1	CAN4	SPI0	SPI1	SPI2
3	X	X	X	X	X	X
2	X	—	X	X	X	—
1	X	—	—	X	—	—

If the SPI0 module is routed on PM[5:4] and used in bidirectional master mode with disabled $\overline{SS}$ output, PM[3:2] are free to be used with CAN or GPIO.

6.1.2 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the CRG.

- Run mode

  All functional parts of the CRG are running during normal run mode. If RTI or COP functionality is required the individual bits of the associated rate select registers (COPCTL, RTICTL) have to be set to a nonzero value.

- Wait mode

  This mode allows to disable the system and core clocks depending on the configuration of the individual bits in the CLKSEL register.

- Stop mode

  Depending on the setting of the PSTP bit, stop mode can be differentiated between full stop mode (PSTP = 0) and pseudo-stop mode (PSTP = 1).

  — Full stop mode

  The oscillator is disabled and thus all system and core clocks are stopped. The COP and the RTI remain frozen.

  — Pseudo-stop mode

  The oscillator continues to run and most of the system and core clocks are stopped. If the respective enable bits are set the COP and RTI will continue to run, else they remain frozen.

- Self-clock mode

  Self-clock mode will be entered if the clock monitor enable bit (CME) and the self-clock mode enable bit (SCME) are both asserted and the clock monitor in the oscillator block detects a loss of clock. As soon as self-clock mode is entered the CRGV4 starts to perform a clock quality check. Self-clock mode remains active until the clock quality check indicates that the required quality of the incoming clock signal is met (frequency and amplitude). Self-clock mode should be used for safety purposes only. It provides reduced functionality to the MCU in case a loss of clock is causing severe system conditions.

6.1.3 Block Diagram

Figure 6-1 shows a block diagram of the CRGV4.

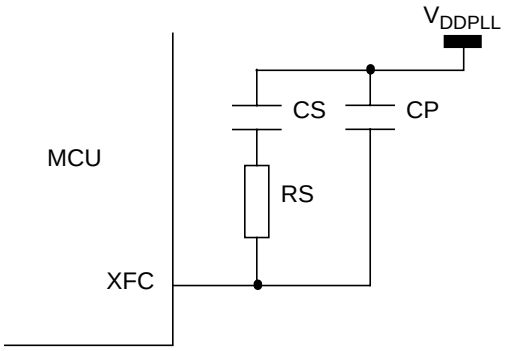


Figure 6-2. PLL Loop Filter Connections

6.2.3 $\overline{\text{RESET}}$ — Reset Pin

$\overline{\text{RESET}}$ is an active low bidirectional reset pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that an system reset (internal to MCU) has been triggered.

6.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the CRGV4.

6.3.1 Module Memory Map

Table 6-1 gives an overview on all CRGV4 registers.

Table 6-1. CRGV4 Memory Map

Address Offset	Use	Access
0x0000	CRG Synthesizer Register (SYNR)	R/W
0x0001	CRG Reference Divider Register (REFDV)	R/W
0x0002	CRG Test Flags Register (CTFLG)[1]	R/W
0x0003	CRG Flags Register (CRGFLG)	R/W
0x0004	CRG Interrupt Enable Register (CRGINT)	R/W
0x0005	CRG Clock Select Register (CLKSEL)	R/W
0x0006	CRG PLL Control Register (PLLCTL)	R/W
0x0007	CRG RTI Control Register (RTICTL)	R/W
0x0008	CRG COP Control Register (COPCTL)	R/W
0x0009	CRG Force and Bypass Test Register (FORBYP)[2]	R/W
0x000A	CRG Test Control Register (CTCTL)[3]	R/W
0x000B	CRG COP Arm/Timer Reset (ARMCOP)	R/W

1 CTFLG is intended for factory test purposes only.

2 FORBYP is intended for factory test purposes only.

3 CTCTL is intended for factory test purposes only.

The PLL filter can be manually or automatically configured into one of two possible operating modes:

- Acquisition mode

  In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start-up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the TRACK status bit is cleared in the CRGFLG register.
- Tracking mode

  In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct and the TRACK bit is set in the CRGFLG register.

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the PLL clock (PLLCLK) is safe to use as the source for the system and core clocks. If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and then check the LOCK bit. If CPU interrupts are disabled, software can poll the LOCK bit continuously (during PLL start-up, usually) or at periodic intervals. In either case, only when the LOCK bit is set, is the PLLCLK clock safe to use as the source for the system and core clocks. If the PLL is selected as the source for the system and core clocks and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application.

The following conditions apply when the PLL is in automatic bandwidth control mode (AUTO = 1):

- The TRACK bit is a read-only indicator of the mode of the filter.
- The TRACK bit is set when the VCO frequency is within a certain tolerance, Δ_{trk}, and is clear when the VCO frequency is out of a certain tolerance, Δ_{unt}.
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance, Δ_{Lock}, and is cleared when the VCO frequency is out of a certain tolerance, Δ_{unl}.
- CPU interrupts can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

The PLL can also operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below the maximum system frequency (f_{sys}) and require fast start-up. The following conditions apply when in manual mode:

- ACQ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the ACQ bit should be asserted to configure the filter in acquisition mode.
- After turning on the PLL by setting the PLLON bit software must wait a given time (t_{acq}) before entering tracking mode (ACQ = 0).
- After entering tracking mode software must wait a given time (t_{al}) before selecting the PLLCLK as the source for system and core clocks (PLLSEL = 1).

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001D ATDDR6L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x001E ATDD47H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x001F ATDD47L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								

= Unimplemented or Reserved

Figure 8-2. ATD Register Summary (Sheet 5 of 5)

8.3.2.1 ATD Control Register 0 (ATDCTL0)

Writes to this register will abort current conversion sequence but will not start a new sequence.

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	WRAP2	WRAP1	WRAP0
W								
Reset	0	0	0	0	0	1	1	1

= Unimplemented or Reserved

Figure 8-3. ATD Control Register 0 (ATDCTL0)

Read: Anytime

Write: Anytime

Table 8-1. ATDCTL0 Field Descriptions

Field	Description
2–0 WRAP[2:0]	**Wrap Around Channel Select Bits** — These bits determine the channel for wrap around when doing multi-channel conversions. The coding is summarized in Table 8-2.

Table 8-2. Multi-Channel Wrap Around Coding

WRAP2	WRAP1	WRAP0	Multiple Channel Conversions (MULT = 1) Wrap Around to AN0 after Converting
0	0	0	Reserved
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5

10.3.3 Programmer's Model of Message Storage

The following section details the organization of the receive and transmit message buffers and the associated control registers.

To simplify the programmer interface, the receive and transmit message buffers have the same outline. Each message buffer allocates 16 bytes in the memory map containing a 13 byte data structure.

An additional transmit buffer priority register (TBPR) is defined for the transmit buffers. Within the last two bytes of this memory map, the MSCAN stores a special 16-bit time stamp, which is sampled from an internal timer after successful transmission or reception of a message. This feature is only available for transmit and receiver buffers, if the TIME bit is set (see Section 10.3.2.1, "MSCAN Control Register 0 (CANCTL0)").

The time stamp register is written by the MSCAN. The CPU can only read these registers.

Table 10-23. Message Buffer Organization

Offset Address	Register	Access
0x00X0	Identifier Register 0	
0x00X1	Identifier Register 1	
0x00X2	Identifier Register 2	
0x00X3	Identifier Register 3	
0x00X4	Data Segment Register 0	
0x00X5	Data Segment Register 1	
0x00X6	Data Segment Register 2	
0x00X7	Data Segment Register 3	
0x00X8	Data Segment Register 4	
0x00X9	Data Segment Register 5	
0x00XA	Data Segment Register 6	
0x00XB	Data Segment Register 7	
0x00XC	Data Length Register	
0x00XD	Transmit Buffer Priority Register[1]	
0x00XE	Time Stamp Register (High Byte)[2]	
0x00XF	Time Stamp Register (Low Byte)[3]	

1 Not applicable for receive buffers
2 Read-only for CPU
3 Read-only for CPU

Figure 10-23 shows the common 13-byte data structure of receive and transmit buffers for extended identifiers. The mapping of standard identifiers into the IDR registers is shown in Figure 10-24.

All bits of the receive and transmit buffers are 'x' out of reset because of RAM-based implementation[1]. All reserved or unused bits of the receive and transmit buffers always read 'x'.

1. Exception: The transmit priority registers are 0 out of reset.

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
Reset:	x	x	x	x	x	x	x	x

Figure 10-26. Identifier Register 1 (IDR1) — Extended Identifier Mapping

Table 10-25. IDR1 Register Field Descriptions — Extended

Field	Description
7:5 ID[20:18]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
4 SRR	**Substitute Remote Request** — This fixed recessive bit is used only in extended format. It must be set to 1 by the user for transmission buffers and is stored as received on the CAN bus for receive buffers.
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)
2:0 ID[17:15]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
Reset:	x	x	x	x	x	x	x	x

Figure 10-27. Identifier Register 2 (IDR2) — Extended Identifier Mapping

Table 10-26. IDR2 Register Field Descriptions — Extended

Field	Description
7:0 ID[14:7]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

The transmitting device can address messages to selected receivers by including addressing information in the initial frame or frames of each message.

The WAKE bit in SCI control register 1 (SCICR1) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wakeup or address mark wakeup.

11.4.4.6.1 Idle Input Line Wakeup (WAKE = 0)

In this wakeup method, an idle condition on the **Rx Input** signal clears the RWU bit and wakes up the SCI. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another idle character appears on the **Rx Input** signal.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

The idle character that wakes a receiver does not set the receiver idle bit, IDLE, or the receive data register full flag, RDRF.

The idle line type bit, ILT, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit. ILT is in SCI control register 1 (SCICR1).

11.4.4.6.2 Address Mark Wakeup (WAKE = 1)

In this wakeup method, a logic 1 in the most significant bit (msb) position of a frame clears the RWU bit and wakes up the SCI. The logic 1 in the msb position marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow.Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another address frame appears on the **Rx Input** signal.

The logic 1 msb of an address frame clears the receiver's RWU bit before the stop bit is received and sets the RDRF flag.

Address mark wakeup allows messages to contain idle characters but requires that the msb be reserved for use in address frames.{sci_wake}

NOTE

With the WAKE bit clear, setting the RWU bit after the **Rx Input** signal has been idle can cause the receiver to wake up immediately.

12.4.3 Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device, slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

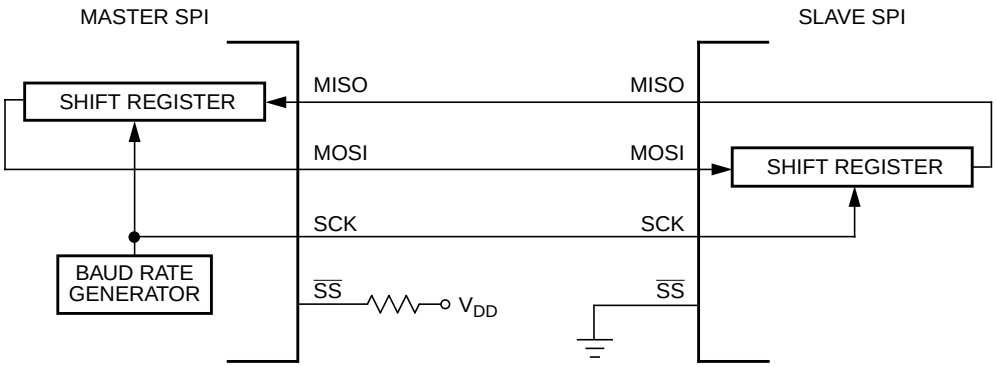


Figure 12-8. Master/Slave Transfer Block Diagram

12.4.3.1 Clock Phase and Polarity Controls

Using two bits in the SPI Control Register1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

12.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

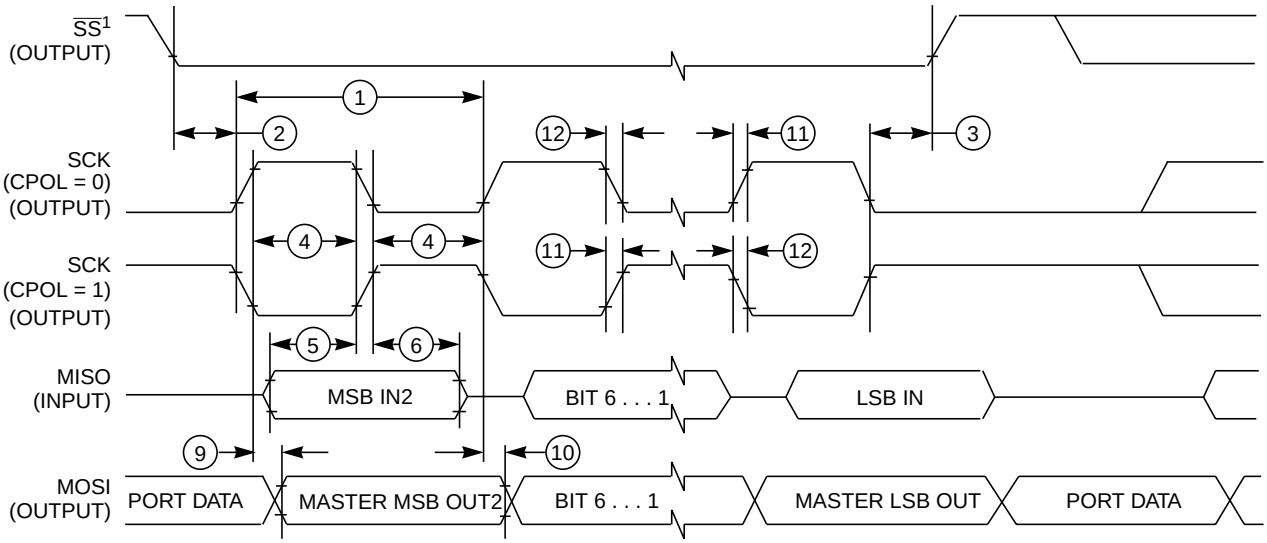
13.6 Interrupts

The PWM module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM7 channel changes while PWM7ENA = 1 or when PWMENA is being asserted while the level at PWM7 is active.

In stop mode or wait mode (with the PSWAI bit set), the emergency shutdown feature will drive the PWM outputs to their shutdown output levels but the PWMIF flag will not be set.

A description of the registers involved and affected due to this interrupt is explained in Section 13.3.2.15, “PWM Shutdown Register (PWMSDN)”.

The PWM block only generates the interrupt and does not service it. The interrupt signal name is PWM interrupt signal.



1. If configured as output
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure A-6. SPI Master Timing (CPHA =1)

Table A-22. SPI Master Mode Timing Characteristics

Conditions are shown in Table A-4 unless otherwise noted, CLOAD = 200pF on all outputs							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Operating Frequency	f_{op}	DC	—	1/4	f_{bus}
1	P	SCK Period	t_{sck}	4	—	2048	t_{bus}
2	D	Enable Lead Time	t_{lead}	1/2	—	—	t_{sck}
3	D	Enable Lag Time	t_{lag}	1/2	—		t_{sck}
4	D	Clock (SCK) High or Low Time	t_{wsck}	t_{bus} – 30	—	1024 t_{bus}	ns
5	D	Data Setup Time (Inputs)	t_{su}	25	—	—	ns
6	D	Data Hold Time (Inputs)	t_{hi}	0	—	—	ns
9	D	Data Valid (after SCK Edge)	t_{v}	—	—	25	ns
10	D	Data Hold Time (Outputs)	t_{ho}	0	—	—	ns
11	D	Rise Time Inputs and Outputs	t_{r}	—	—	25	ns
12	D	Fall Time Inputs and Outputs	t_{f}	—	—	25	ns

Table A-24. Expanded Bus Timing Characteristics

Conditions are shown in Table A-4 unless otherwise noted, C_{LOAD} = 50pF							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Frequency of operation (E-clock)	f_o	0	—	25.0	MHz
2	P	Cycle time	t_{cyc}	40	—	—	ns
3	D	Pulse width, E low	PW_{EL}	17	—	—	ns
4	D	Pulse width, E high[1]	PW_{EH}	17	—	—	ns
5	D	Address delay time	t_{AD}	—	—	8	ns
6	D	Address valid time to E rise (PW_{EL}–t_{AD})	t_{AV}	11	—	—	ns
7	D	Muxed address hold time	t_{MAH}	2	—	—	ns
8	D	Address hold to data valid	t_{AHDS}	7	—	—	ns
9	D	Data hold to address	t_{DHA}	2	—	—	ns
10	D	Read data setup time	t_{DSR}	13	—	—	ns
11	D	Read data hold time	t_{DHR}	0	—	—	ns
12	D	Write data delay time	t_{DDW}	—	—	7	ns
13	D	Write data hold time	t_{DHW}	2	—	—	ns
14	D	Write data setup time[1] (PW_{EH}–t_{DDW})	t_{DSW}	10	—	—	ns
15	D	Address access time[1] (t_{cyc}–t_{AD}–t_{DSR})	t_{ACCA}	19	—	—	ns
16	D	E high access time[1] (PW_{EH}–t_{DSR})	t_{ACCE}	4	—	—	ns
17	D	Non-multiplexed address delay time	t_{NAD}	—	—	7	ns
18	D	Non-muxed address valid to E rise (PW_{EL}–t_{NAD})	t_{NAV}	10	—	—	ns
19	D	Non-multiplexed address hold time	t_{NAH}	2	—	—	ns
20	D	Chip select delay time	t_{CSD}	—	—	16	ns
21	D	Chip select access time[1] (t_{cyc}–t_{CSD}–t_{DSR})	t_{ACCS}	11	—	—	ns
22	D	Chip select hold time	t_{CSH}	2	—	—	ns
23	D	Chip select negated time	t_{CSN}	8	—	—	ns
24	D	Read/write delay time	t_{RWD}	—	—	7	ns
25	D	Read/write valid time to E rise (PW_{EL}–t_{RWD})	t_{RWV}	10	—	—	ns
26	D	Read/write hold time	t_{RWH}	2	—	—	ns
27	D	Low strobe delay time	t_{LSD}	—	—	7	ns
28	D	Low strobe valid time to E rise (PW_{EL}–t_{LSD})	t_{LSV}	10	—	—	ns
29	D	Low strobe hold time	t_{LSH}	2	—	—	ns
30	D	NOACC strobe delay time	t_{NOD}	—	—	7	ns
31	D	NOACC valid time to E rise (PW_{EL}–t_{LSD})	t_{NOV}	10	—	—	ns
32	D	NOACC hold time	t_{NOH}	2	—	—	ns
33	D	PIPO0 delay time	t_{P0D}	2	—	7	ns
34	D	PIPO0 valid time to E rise (PW_{EL}–t_{P0D})	t_{P0V}	10	—		ns
35	D	PIPO1 delay time[1] (PW_{EH}-t_{P1V})	t_{P1D}	2	—	7	ns
36	D	PIPO1 valid time to E fall	t_{P1V}	10	—	—	ns

1 Affected by clock stretch: add N x t_{cyc} where N=0,1,2 or 3, depending on the number of clock stretches.