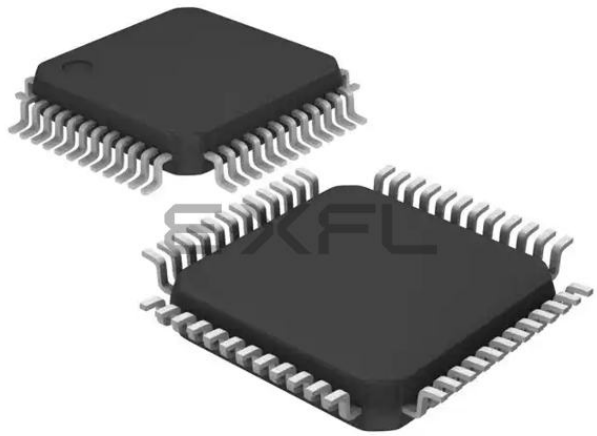


Welcome to E-XFL.COM

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	HCS12
Core Size	16-Bit
Speed	16MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	31
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K 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	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12q12j2vfae1r

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com/

A full list of family members and options is included in the appendices.

The following revision history table summarizes changes contained in this document.

This document contains information for all constituent modules, with the exception of the S12 CPU. For S12 CPU information please refer to the CPU S12 Reference Manual.

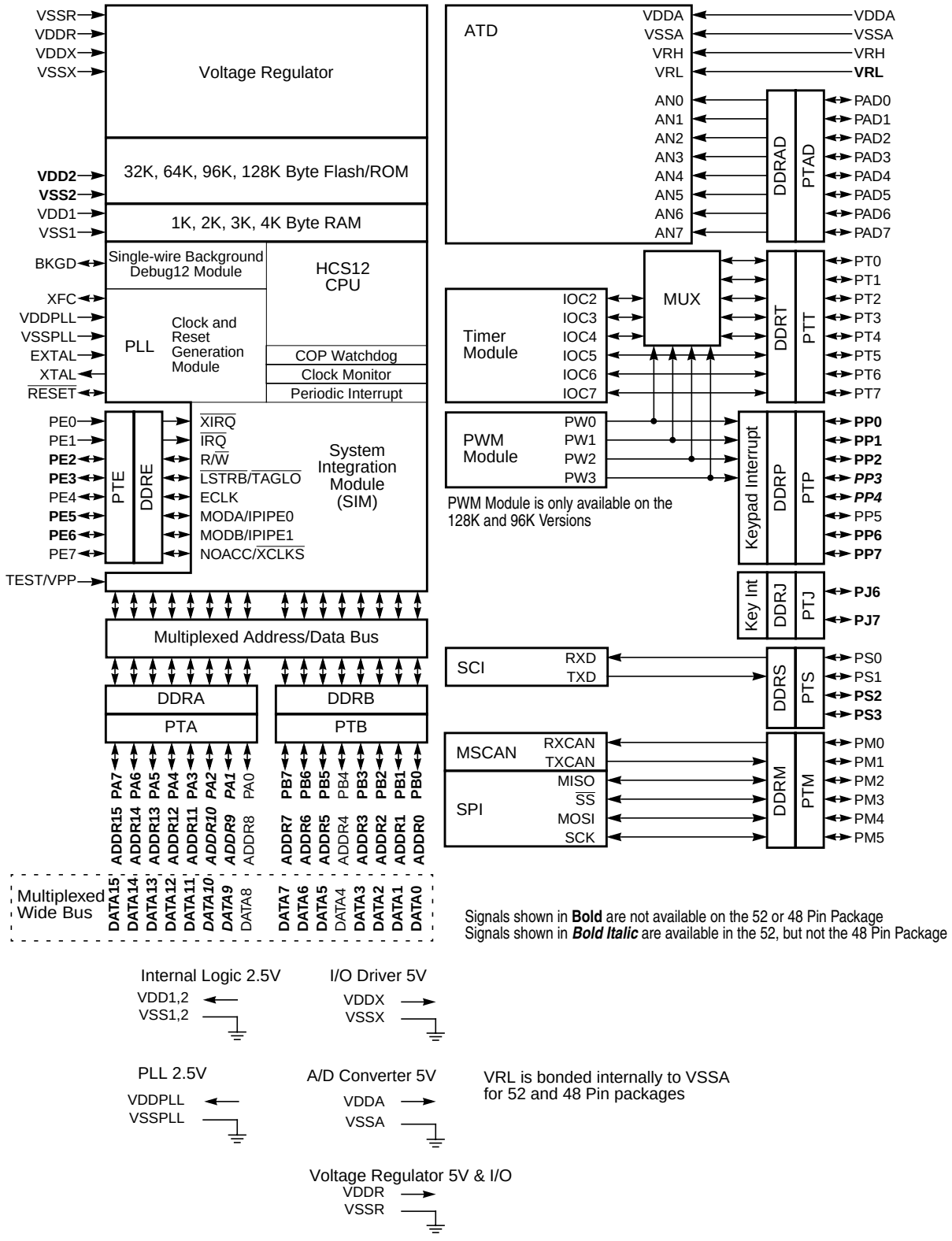
Date	Revision Level	Description
January 2006	01.05	Updated LVI levels in electrical parameter section Fixed incorrect reference to TSCR2
May 2006	01.06	Added 0M66G PART ID number Corrected missing overbars on pin names Added units to MSCAN timing table
Dec 2006	01.07	Corrected CRGFLG contents in register summary Corrected unintended symbol font
May 2007	01.08	Added emulation package info. Corrected TIM and PWM channel count in PIM section Updated ATD section Corrected typos and inconsistent register listing format
Dec 2007	01.09	Added new PartID
May 2010	01.10	Updated TIM section (see TIM rev. history)

Chapter 3
Module Mapping Control (MMCV4) Block Description

Chapter 4
Multiplexed External Bus Interface (MEBIV3)

1.1.3 Block Diagram

Figure 1-1. MC9S12Q128-Family Block Diagram



1.5.3.3 Wait

This mode is entered by executing the CPU WAI instruction. In this mode the CPU will not execute instructions. The internal CPU signals (address and data bus) will be fully static. All peripherals stay active. For further power consumption reduction the peripherals can individually turn off their local clocks.

1.5.3.4 Run

Although this is not a low-power mode, unused peripheral modules should not be enabled in order to save power.

1.6 Resets and Interrupts

Consult the Exception Processing section of the CPU12 Reference Manual for information.

1.6.1 Vectors

Table 1-9 lists interrupt sources and vectors in default order of priority.

Table 1-9. Interrupt Vector Locations

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
0xFFFE, 0xFFFF	External reset, power on reset, or low voltage reset (see CRG flags register to determine reset source)	None	None	—
0xFFFC, 0xFFFD	Clock monitor fail reset	None	COPCTL (CME, FCME)	—
0xFFFA, 0xFFFB	COP failure reset	None	COP rate select	—
0xFFF8, 0xFFF9	Unimplemented instruction trap	None	None	—
0xFFF6, 0xFFF7	SWI	None	None	—
0xFFF4, 0xFFF5	XIRQ	X-Bit	None	—
0xFFF2, 0xFFF3	IRQ	I bit	INTCR (IRQEN)	0x00F2
0xFFF0, 0xFFF1	Real time Interrupt	I bit	CRGINT (RTIE)	0x00F0
$FFEE, $FFEF	Reserved			
$FFEC, $FFED	Reserved			
0xFFEA, 0xFFEB	Standard timer channel 2	I bit	TIE (C2I)	0x00EA
0xFFE8, 0xFFE9	Standard timer channel 3	I bit	TIE (C3I)	0x00E8
0xFFE6, 0xFFE7	Standard timer channel 4	I bit	TIE (C4I)	0x00E6
0xFFE4, 0xFFE5	Standard timer channel 5	I bit	TIE (C5I)	0x00E4
0xFFE2, 0xFFE3	Standard timer channel 6	I bit	TIE (C6I)	0x00E2
0xFFE0, 0xFFE1	Standard timer channel 7	I bit	TIE (C7I)	0x00E0
0xFFDE, 0xFFDF	Standard timer overflow	I bit	TMSK2 (TOI)	0x00DE
0xFFDC, 0xFFDD	Pulse accumulator A overflow	I bit	PACTL (PAOVI)	0x00DC

Table 1-11. Device Specific Flash PAGE Mapping

Device	PAGE	PAGE Visible with PPAGE Contents
MC9S12Q32	3E	$00,$02,$04,$06,$08,$0A,$0C,$0E,$10,$12.....$2C,$2E,$30,$32,$34,$36,$38,$3A,$3C,$3E
	3F	$01,$03,$05,$07,$09,$0B,$0D,$0F,$11,$13.....$2D,$2F,$31,$33,$35,$37,$39,$3B,$3D,$3F
MC9S12Q64	3C	$04,$0C,$14,$1C,$24,$2C,$34,$3C
	3D	$05,$0D,$15,$1D,$25,$2D,$35,$3D
	3E	$06,$0E,$16,$1E,$26,$2E,$36,$3E
	3F	$07,$0F,$17,$1F,$27,$2F,$37,$3F
MC9S12Q96	3A	$02,$0A,$12,$1A,$22,$2A,$32,$3A
	3B	$03,$0B,$13,$1B,$23,$2B,$33,$3B
	3C	$04,$0C,$14,$1C,$24,$2C,$34,$3C
	3D	$05,$0D,$15,$1D,$25,$2D,$35,$3D
	3E	$06,$0E,$16,$1E,$26,$2E,$36,$3E
	3F	$07,$0F,$17,$1F,$27,$2F,$37,$3F
MC9S12Q128	38	$00,$08,$10,$18,$20,$28,$30,$38
	39	$01,$09,$11,$19,$21,$29,$31,$39
	3A	$02,$0A,$12,$1A,$22,$2A,$32,$3A
	3B	$03,$0B,$13,$1B,$23,$2B,$33,$3B
	3C	$04,$0C,$14,$1C,$24,$2C,$34,$3C
	3D	$05,$0D,$15,$1D,$25,$2D,$35,$3D
	3E	$06,$0E,$16,$1E,$26,$2E,$36,$3E
	3F	$07,$0F,$17,$1F,$27,$2F,$37,$3F

1.7.2 BDM Alternate Clock

The BDM section reference to alternate clock is equivalent to the oscillator clock.

1.7.3 Extended Address Range Emulation Implications

In order to emulate the devices, external addressing of a 128K memory map is required. This is provided in a 112 LQFP package version which includes the 3 necessary extra external address bus signals via PortK[2:0]. This package version is for emulation only and not provided as a general production package.

The reset state of DDRK is 0x0000, configuring the pins as inputs.

The reset state of PUPKE in the PUCR register is “1” enabling the internal Port K pullups.

In this reset state the pull-ups provide a defined state and prevent a floating input, thereby preventing unnecessary current flow at the input stage.

To prevent unnecessary current flow in production package options, the states of DDRK and PUPKE should not be changed by software.

2.3.2.3.6 Port M Polarity Select Register (PPSM)

Module Base + 0x0015

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

= Unimplemented or Reserved

Figure 2-22. Port M Polarity Select Register (PPSM)

Read: Anytime.

Write: Anytime.

Table 2-20. PPSM Field Descriptions

Field	Description
5–0 PPSM[5:0]	**Polarity Select Port M** — This register selects whether a pull-down or a pull-up device is connected to the pin. 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 input or as wired-or output. 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 input.

2.3.2.3.7 Port M Wired-OR Mode Register (WOMM)

Module Base + 0x0016

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

= Unimplemented or Reserved

Figure 2-23. Port M Wired-OR Mode Register (WOMM)

Read: Anytime.

Write: Anytime.

Table 2-21. WOMM Field Descriptions

Field	Description
5–0 WOMM[5:0]	**Wired-OR Mode Port M** — 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. This bit has no influence on pins used as inputs. 0 Output buffers operate as push-pull outputs. 1 Output buffers operate as open-drain outputs.

2.3.2.4.7 Port P Interrupt Enable Register (PIEP)

Module Base + 0x001E

	7	6	5	4	3	2	1	0
R W	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
Reset	0	0	0	0	0	0	0	0

Figure 2-30. Port P Interrupt Enable Register (PIEP)

Read: Anytime.

Write: Anytime.

Table 2-26. PIEP Field Descriptions

Field	Description
7–0 PIEP[7:0]	**Pull Select Port P** — This register disables or enables on a per pin basis the edge sensitive external interrupt associated with port P. 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

2.3.2.4.8 Port P Interrupt Flag Register (PIFP)

Module Base + 0x001F

	7	6	5	4	3	2	1	0
R W	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
Reset	0	0	0	0	0	0	0	0

Figure 2-31. Port P Interrupt Flag Register (PIFP)

Read: Anytime.

Write: Anytime.

Table 2-27. PIFP Field Descriptions

Field	Description
7–0 PIFP[7:0]	**Interrupt Flags Port P** — 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 PPSP register. To clear this flag, write a “1” to the corresponding bit in the PIFP register. Writing a “0” has no effect. 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.

3.3.2.9 Program Page Index Register (PPAGE)

Module Base + 0x0030
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
W								
Reset[1]	—	—	—	—	—	—	—	—

1. The reset state of this register is controlled at chip integration. Please refer to the device overview section to determine the actual reset state of this register.

= Unimplemented or Reserved

Figure 3-11. Program Page Index Register (PPAGE)

Read: Anytime

Write: Determined at chip integration. Generally it's: "write anytime in all modes;" on some devices it will be: "write only in special modes." Check specific device documentation to determine which applies.

Reset: Defined at chip integration as either 0x00 (paired with write in any mode) or 0x3C (paired with write only in special modes), see device overview chapter.

The HCS12 core architecture limits the physical address space available to 64K bytes. The program page index register allows for integrating up to 1M byte of FLASH or ROM into the system by using the six page index bits to page 16K byte blocks into the program page window located from 0x8000 to 0xBFFF as defined in Table 3-14. CALL and RTC instructions have special access to read and write this register without using the address bus.

NOTE

Normal writes to this register take one cycle to go into effect. Writes to this register using the special access of the CALL and RTC instructions will be complete before the end of the associated instruction.

Table 3-13. MEMSIZ0 Field Descriptions

Field	Description
5:0 PIX[5:0]	**Program Page Index Bits 5:0** — These page index bits are used to select which of the 64 FLASH or ROM array pages is to be accessed in the program page window as shown in Table 3-14.

Table 4-6. PEAR Field Descriptions

Field	Description
7 NOACCE	**CPU No Access Output Enable** Normal: write once Emulation: write never Special: write anytime 1 The associated pin (port E, bit 7) is general-purpose I/O. 0 The associated pin (port E, bit 7) is output and indicates whether the cycle is a CPU free cycle. This bit has no effect in single-chip or special peripheral modes.
5 PIPOE	**Pipe Status Signal Output Enable** Normal: write once Emulation: write never Special: write anytime. 0 The associated pins (port E, bits 6:5) are general-purpose I/O. 1 The associated pins (port E, bits 6:5) are outputs and indicate the state of the instruction queue This bit has no effect in single-chip or special peripheral modes.
4 NECLK	**No External E Clock** Normal and special: write anytime Emulation: write never 0 The associated pin (port E, bit 4) is the external E clock pin. External E clock is free-running if ESTR = 0 1 The associated pin (port E, bit 4) is a general-purpose I/O pin. External E clock is available as an output in all modes.
3 LSTRE	**Low Strobe ($\overline{\text{LSTRB}}$) Enable** Normal: write once Emulation: write never Special: write anytime. 0 The associated pin (port E, bit 3) is a general-purpose I/O pin. 1 The associated pin (port E, bit 3) is configured as the $\overline{\text{LSTRB}}$ bus control output. If BDM tagging is enabled, $\overline{\text{TAGLO}}$ is multiplexed in on the rising edge of ECLK and $\overline{\text{LSTRB}}$ is driven out on the falling edge of ECLK. This bit has no effect in single-chip, peripheral, or normal expanded narrow modes. **Note:** $\overline{\text{LSTRB}}$ is used during external writes. After reset in normal expanded mode, $\overline{\text{LSTRB}}$ is disabled to provide an extra I/O pin. If $\overline{\text{LSTRB}}$ is needed, it should be enabled before any external writes. External reads do not normally need $\overline{\text{LSTRB}}$ because all 16 data bits can be driven even if the system only needs 8 bits of data.
2 RDWE	**Read/Write Enable** Normal: write once Emulation: write never Special: write anytime 0 The associated pin (port E, bit 2) is a general-purpose I/O pin. 1 The associated pin (port E, bit 2) is configured as the R/$\overline{\text{W}}$ pin This bit has no effect in single-chip or special peripheral modes. **Note:** R/$\overline{\text{W}}$ is used for external writes. After reset in normal expanded mode, R/$\overline{\text{W}}$ is disabled to provide an extra I/O pin. If R/$\overline{\text{W}}$ is needed it should be enabled before any external writes.

6.4.9 SYNC — Request Timed Reference Pulse

The SYNC command is unlike other BDM commands because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency (the lowest serial communication frequency is determined by the crystal oscillator or the clock chosen by CLKSW.)
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic 1.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next falling edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

6.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. As soon as this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

10.3.2.6 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

	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 10-9. MSCAN Receiver Interrupt Enable Register (CANRIER)

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.

Read: Anytime
Write: Anytime when not in initialization mode

Table 10-10. CANRIER Register Field Descriptions

Field	Description
7 WUPIE[(1)]	**Wake-Up Interrupt Enable** 0 No interrupt request is generated from this event. 1 A wake-up event causes a Wake-Up interrupt request.
6 CSCIE	**CAN Status Change Interrupt Enable** 0 No interrupt request is generated from this event. 1 A CAN Status Change event causes an error interrupt request.
5:4 RSTATE[1:0]	**Receiver Status Change Enable** — These RSTAT enable bits control the sensitivity level in which receiver state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level the RSTAT flags continue to indicate the actual receiver state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by receiver state changes. 01 Generate CSCIF interrupt only if the receiver enters or leaves "bus-off" state. Discard other receiver state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the receiver enters or leaves "RxErr" or "bus-off"[(2)] state. Discard other receiver state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
3:2 TSTATE[1:0]	**Transmitter Status Change Enable** — These TSTAT enable bits control the sensitivity level in which transmitter state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level, the TSTAT flags continue to indicate the actual transmitter state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by transmitter state changes. 01 Generate CSCIF interrupt only if the transmitter enters or leaves "bus-off" state. Discard other transmitter state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the transmitter enters or leaves "TxErr" or "bus-off" state. Discard other transmitter state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.

13.3.2.1 SCI Baud Rate Registers (SCIBDH and SCHBDL)

Module Base + 0x_0000

	7	6	5	4	3	2	1	0
R	0	0	0	SBR12	SBR11	SBR10	SBR9	SBR8
W								
Reset	0	0	0	0	0	0	0	0

Module Base + 0x_0001

	7	6	5	4	3	2	1	0
R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
W								
Reset	0	0	0	0	0	1	0	0

= Unimplemented or Reserved

Figure 13-3. SCI Baud Rate Registers (SCIBDH and SCIBDL)

The SCI Baud Rate Register is used by the counter to determine the baud rate of the SCI. The formula for calculating the baud rate is:

SCI baud rate = SCI module clock / (16 x BR)

where:

BR is the content of the SCI baud rate registers, bits SBR12 through SBR0. The baud rate registers can contain a value from 1 to 8191.

Read: Anytime. If only SCIBDH is written to, a read will not return the correct data until SCIBDL is written to as well, following a write to SCIBDH.

Write: Anytime

Table 13-1. SCIBDH AND SCIBDL Field Descriptions

Field	Description
4–0 7–0 SBR[12:0]	**SCI Baud Rate Bits** — The baud rate for the SCI is determined by these 13 bits. **Note:** The baud rate generator is disabled until the TE bit or the RE bit is set for the first time after reset. The baud rate generator is disabled when BR = 0. Writing to SCIBDH has no effect without writing to SCIBDL, since writing to SCIBDH puts the data in a temporary location until SCIBDL is written to.

Table 13-3. Loop Functions

LOOPS	RSRC	Function
0	x	Normal operation
1	0	Loop mode with Rx input internally connected to Tx output
1	1	Single-wire mode with Rx input connected to TXD

13.3.2.3 SCI Control Register 2 (SCICR2)

Module Base + 0x_0003

	7	6	5	4	3	2	1	0
R W	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

Figure 13-5. SCI Control Register 2 (SCICR2)

Read: Anytime

Write: Anytime

Table 13-4. SCICR2 Field Descriptions

Field	Description
7 TIE	**Transmitter Interrupt Enable Bit** — TIE enables the transmit data register empty flag, TDRE, to generate interrupt requests. 0 TDRE interrupt requests disabled 1 TDRE interrupt requests enabled
6 TCIE	**Transmission Complete Interrupt Enable Bit** — TCIE enables the transmission complete flag, TC, to generate interrupt requests. 0 TC interrupt requests disabled 1 TC interrupt requests enabled
5 RIE	**Receiver Full Interrupt Enable Bit** — RIE enables the receive data register full flag, RDRF, or the overrun flag, OR, to generate interrupt requests. 0 RDRF and OR interrupt requests disabled 1 RDRF and OR interrupt requests enabled
4 ILIE	**Idle Line Interrupt Enable Bit** — ILIE enables the idle line flag, IDLE, to generate interrupt requests. 0 IDLE interrupt requests disabled 1 IDLE interrupt requests enabled
3 TE	**Transmitter Enable Bit** — TE enables the SCI transmitter and configures the TXD pin as being controlled by the SCI. The TE bit can be used to queue an idle preamble. 0 Transmitter disabled 1 Transmitter enabled
2 RE	**Receiver Enable Bit** — RE enables the SCI receiver. 0 Receiver disabled 1 Receiver enabled

Table 17-4. FSEC Field Descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of the backdoor key access to the Flash module as shown in Table 17-5.
5–2 NV[5:2]	**Nonvolatile Flag Bits** — The NV[5:2] bits are available to the user as nonvolatile flags.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 17-6. If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

Table 17-5. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01(1)	DISABLED
10	ENABLED
11	DISABLED

1. Preferred KEYEN state to disable Backdoor Key Access.

Table 17-6. Flash Security States

SEC[1:0]	Status of Security
00	Secured
01(1)	Secured
10	Unsecured
11	Secured

1. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 17.4.3, “Flash Module Security”.

17.3.2.3 RESERVED1

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 17-6. RESERVED1

All bits read 0 and are not writable.

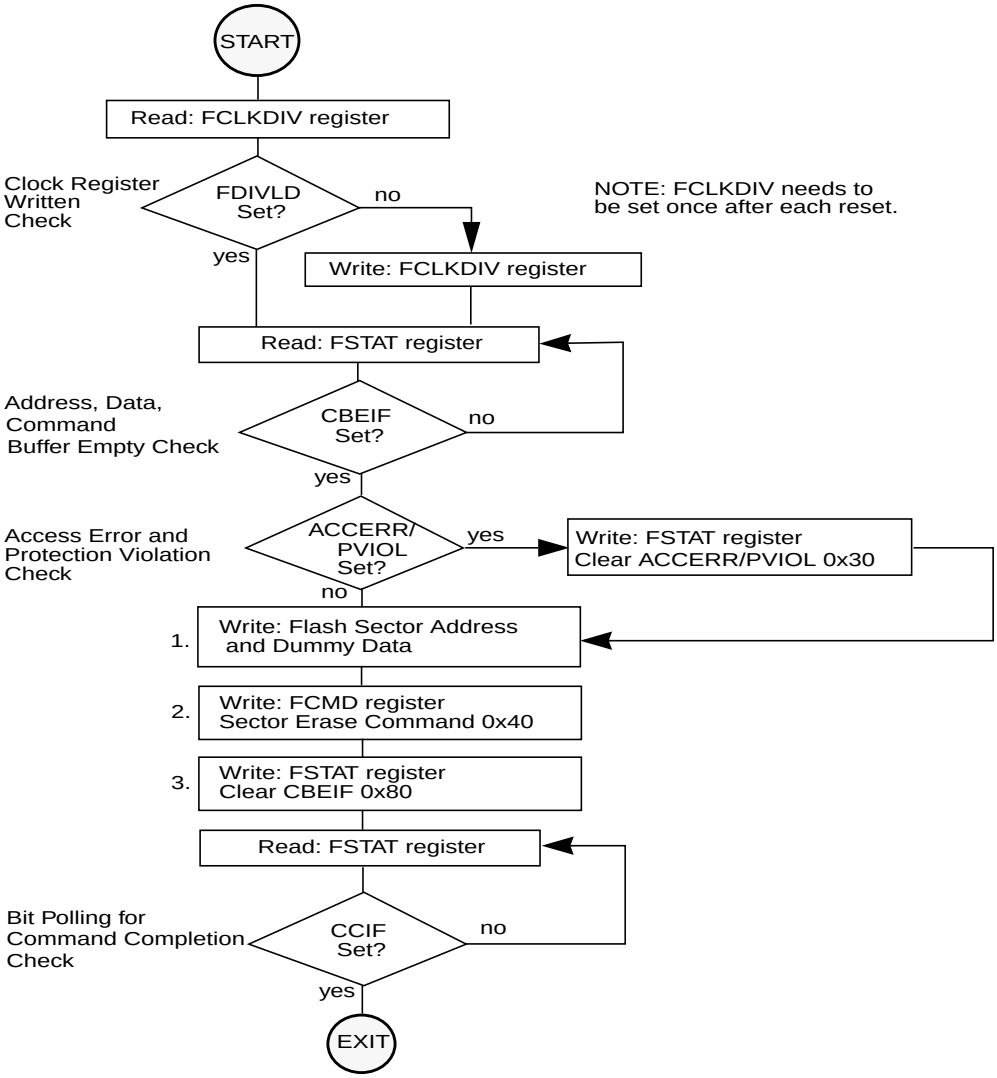


Figure 17-24. Example Sector Erase Command Flow

18.4.1.4 Illegal Flash Operations

18.4.1.4.1 Access Error

The ACCERR flag in the FSTAT register will be set during the command write sequence if any of the following illegal Flash operations are performed causing the command write sequence to immediately abort:

1. Writing to the Flash address space before initializing the FCLKDIV register
2. Writing a misaligned word or a byte to the valid Flash address space
3. Writing to the Flash address space while CBEIF is not set
4. Writing a second word to the Flash address space before executing a program or erase command on the previously written word
5. Writing to any Flash register other than FCMD after writing a word to the Flash address space
6. Writing a second command to the FCMD register before executing the previously written command
7. Writing an invalid command to the FCMD register
8. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the FCMD register
9. The part enters stop mode and a program or erase command is in progress. The command is aborted and any pending command is killed
10. When security is enabled, a command other than mass erase originating from a non-secure memory or from the background debug mode is written to the FCMD register
11. A 0 is written to the CBEIF bit in the FSTAT register to abort a command write sequence.

The ACCERR flag will not be set if any Flash register is read during the command write sequence. If the Flash array is read during execution of an algorithm (CCIF=0), the Flash module will return invalid data and the ACCERR flag will not be set. If an ACCERR flag is set in the FSTAT register, the Flash command controller is locked. It is not possible to launch another command until the ACCERR flag is cleared.

18.4.1.4.2 Protection Violation

The PVIOL flag in the FSTAT register will be set during the command write sequence after the word write to the Flash address space if any of the following illegal Flash operations are performed, causing the command write sequence to immediately abort:

1. Writing a Flash address to program in a protected area of the Flash array (see Section 18.3.2.5).
2. Writing a Flash address to erase in a protected area of the Flash array.
3. Writing the mass erase command to the FCMD register while any protection is enabled.

If the PVIOL flag is set, the Flash command controller is locked. It is not possible to launch another command until the PVIOL flag is cleared.

19.4.1.2 Command Write Sequence

The Flash command controller is used to supervise the command write sequence to execute program, erase, and erase verify algorithms.

Before starting a command write sequence, the ACCERR and PVIOL flags in the FSTAT register must be clear and the CBEIF flag should be tested to determine the state of the address, data, and command buffers. If the CBEIF flag is set, indicating the buffers are empty, a new command write sequence can be started. If the CBEIF flag is clear, indicating the buffers are not available, a new command write sequence will overwrite the contents of the address, data, and command buffers.

A command write sequence consists of three steps which must be strictly adhered to with writes to the Flash module not permitted between the steps. However, Flash register and array reads are allowed during a command write sequence. The basic command write sequence is as follows:

1. Write to a valid address in the Flash array memory.
2. Write a valid command to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the command.

The address written in step 1 will be stored in the FADDR registers and the data will be stored in the FDATA registers. When the CBEIF flag is cleared in step 3, the CCIF flag is cleared by the Flash command controller indicating that the command was successfully launched. For all command write sequences, the CBEIF flag will set after the CCIF flag is cleared indicating that the address, data, and command buffers are ready for a new command write sequence to begin. A buffered command will wait for the active operation to be completed before being launched. Once a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag in the FSTAT register. The CCIF flag will set upon completion of all active and buffered commands.

19.4.1.3.3 Sector Erase Command

The sector erase operation will erase all addresses in a 1024 byte sector of the Flash array using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 19-24. The sector erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while MCU address bits [9:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

If a Flash sector to be erased is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. Once 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 new command write sequence has been buffered.

Chapter 20
128 Kbyte Flash Module (S12FTS128K1V1)

20.1 Introduction

The FTS128K1 module implements a 128 Kbyte Flash (nonvolatile) memory. The Flash memory contains one array of 128 Kbytes organized as 1024 rows of 128 bytes with an erase sector size of eight rows (1024 bytes). The Flash array may be read as either bytes, aligned words, or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

The Flash array is ideal for program and data storage for single-supply applications allowing for field reprogramming without requiring external voltage sources for program or erase. Program and erase functions are controlled by a command driven interface. The Flash module supports both mass erase and sector erase. An erased bit reads 1 and a programmed bit reads 0. The high voltage required to program and erase is generated internally. It is not possible to read from a Flash array while it is being erased or programmed.

CAUTION

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

20.1.1 Glossary

Command Write Sequence — A three-step MCU instruction sequence to program, erase, or erase verify the Flash array memory.

20.1.2 Features

- 128 Kbytes of Flash memory comprised of one 128 Kbyte array divided into 128 sectors of 1024 bytes
- Automated program and erase algorithm
- Interrupts on Flash command completion and command buffer empty
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for Flash program and erase operations
- Security feature to prevent unauthorized access to the Flash array memory

A.4.3 Phase Locked Loop

The oscillator provides the reference clock for the PLL. The PLL´s Voltage Controlled Oscillator (VCO) is also the system clock source in self clock mode.

A.4.3.1 XFC Component Selection

This section describes the selection of the XFC components to achieve a good filter characteristics.

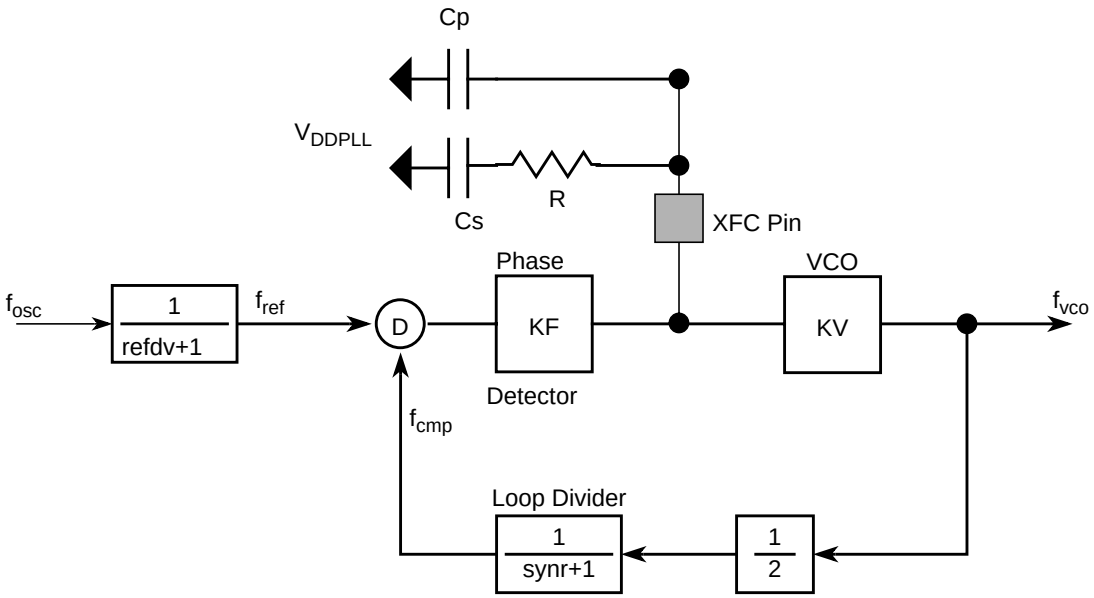


Figure A-2. Basic PLL Functional Diagram

The following procedure can be used to calculate the resistance and capacitance values using typical values for K_1, f_1 and i_{ch} from Table A-17.

The grey boxes show the calculation for f_{VCO} = 50MHz and f_{ref} = 1MHz. E.g., these frequencies are used for f_{OSC} = 4MHz and a 16MHz bus clock.

The VCO Gain at the desired VCO frequency is approximated by:

$$K_V = K_1 \cdot e^{\frac{(f_1 - f_{vco})}{K_1 \cdot 1V}} = -100 \cdot e^{\frac{(60-50)}{-100}} = -90.48\text{MHz/V}$$

The phase detector relationship is given by:

$$K_\Phi = -|i_{ch}| \cdot K_V = 316.7\text{Hz}/\Omega$$

i_{ch} is the current in tracking mode.