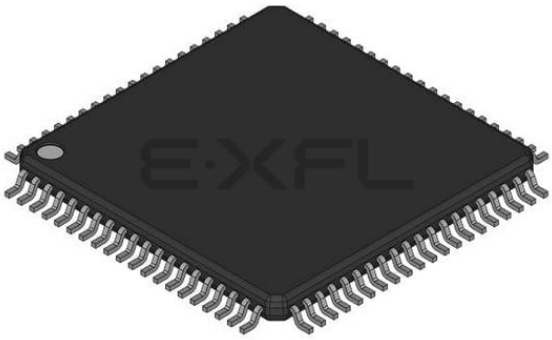

Welcome to E-XFL.COM

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	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12gc16mfue

Chapter 17
16 Kbyte Flash Module (S12FTS16KV1)

Chapter 18
32 Kbyte Flash Module (S12FTS32KV1)

Chapter 19
64 Kbyte Flash Module (S12FTS64KV4)

0x00C8–0x00CF SCI (Asynchronous Serial Interface) (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00CD	SCISR2	Read:	0	0	0	0	0	BRK13	TXDIR	RAF
		Write:								
0x00CE	SCIDRH	Read:	R8	T8	0	0	0	0	0	0
		Write:								
0x00CF	SCIDRL	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0

0x00D0–0x00D7 Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00D0–0x00D7	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

0x00D8–0x00DF SPI (Serial Peripheral Interface)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00D8	SPICR1	Read: Write:	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
0x00D9	SPICR2	Read:	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		Write:								
0x00DA	SPIBR	Read:	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		Write:								
0x00DB	SPISR	Read:	SPIF	0	SPTEF	MODF	0	0	0	0
		Write:								
0x00DC	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x00DD	SPIDR	Read: Write:	Bit7	6	5	4	3	2	1	Bit0
0x00DE	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								
0x00DF	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

2.4 Functional Description

Each pin can act as general purpose I/O. In addition the pin can act as an output from a peripheral module or an input to a peripheral module.

A set of configuration registers is common to all ports. All registers can be written at any time, however a specific configuration might not become active.

Example: Selecting a pull-up resistor. This resistor does not become active while the port is used as a push-pull output.

2.4.1 Registers

2.4.1.1 I/O Register

This register holds the value driven out to the pin if the port is used as a general purpose I/O. Writing to this register has only an effect on the pin if the port is used as general purpose output. When reading this address, the value of the pins are returned if the data direction register bits are set to 0.

If the data direction register bits are set to 1, the contents of the I/O register is returned. This is independent of any other configuration (Figure 2-46).

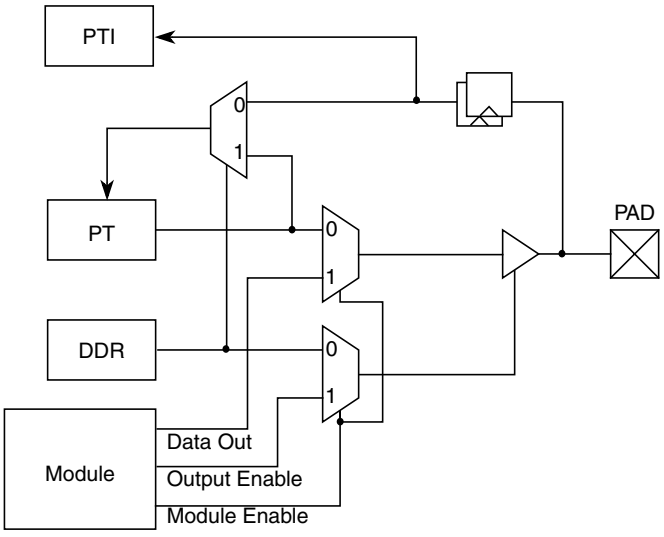

Figure 2-46. Illustration of I/O Pin Functionality

2.4.1.2 Input Register

This is a read-only register and always returns the value of the pin (Figure 2-46).

2.4.1.3 Data Direction Register

This register defines whether the pin is used as an input or an output. If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-46).

vector spaces, expansion windows, and on-chip memory are mapped so that their address ranges do not overlap. The MMC will make only one select signal active at any given time. This activation is based upon the priority outlined in Table 3-15. If two or more blocks share the same address space, only the select signal for the block with the highest priority will become active. An example of this is if the registers and the RAM are mapped to the same space, the registers will have priority over the RAM and the portion of RAM mapped in this shared space will not be accessible. The expansion windows have the lowest priority. This means that registers, vectors, and on-chip memory are always visible to a program regardless of the values in the page select registers.

Table 3-15. Select Signal Priority

Priority	Address Space
Highest	BDM (internal to core) firmware or register space
...	Internal register space
...	RAM memory block
...	EEPROM memory block
...	On-chip FLASH or ROM
Lowest	Remaining external space

In expanded modes, all address space not used by internal resources is by default external memory space. The data registers and data direction registers for ports A and B are removed from the on-chip memory map and become external accesses. If the EME bit in the MODE register (see MEBI block description chapter) is set, the data and data direction registers for port E are also removed from the on-chip memory map and become external accesses.

In special peripheral mode, the first 16 registers associated with bus expansion are removed from the on-chip memory map (PORTA, PORTB, DDRA, DDRB, PORTE, DDRE, PEAR, MODE, PUCR, RDRIV, and the EBI reserved registers).

In emulation modes, if the EMK bit in the MODE register (see MEBI block description chapter) is set, the data and data direction registers for port K are removed from the on-chip memory map and become external accesses.

3.4.2.2 Emulation Chip Select Signal

When the EMK bit in the MODE register (see MEBI block description chapter) is set, port K bit 7 is used as an active-low emulation chip select signal, $\overline{\text{ECS}}$. This signal is active when the system is in emulation mode, the EMK bit is set and the FLASH or ROM space is being addressed subject to the conditions outlined in Section 3.4.3.2, “Extended Address (XAB19:14) and ECS Signal Functionality.” When the EMK bit is clear, this pin is used for general purpose I/O.

3.4.2.3 External Chip Select Signal

When the EMK bit in the MODE register (see MEBI block description chapter) is set, port K bit 6 is used as an active-low external chip select signal, $\overline{\text{XCS}}$. This signal is active only when the $\overline{\text{ECS}}$ signal described above is not active and when the system is addressing the external address space. Accesses to

4.4.3.2 Special Operating Modes

There are two special operating modes that correspond to normal operating modes. These operating modes are commonly used in factory testing and system development.

4.4.3.2.1 Special Single-Chip Mode

When the MCU is reset in this mode, the background debug mode is enabled and active. The MCU does not fetch the reset vector and execute application code as it would in other modes. Instead the active background mode is in control of CPU execution and BDM firmware is waiting for additional serial commands through the BKGD pin. When a serial command instructs the MCU to return to normal execution, the system will be configured as described below unless the reset states of internal control registers have been changed through background commands after the MCU was reset.

There is no external expansion bus after reset in this mode. Ports A and B are initially simple bidirectional I/O pins that are configured as high-impedance inputs with internal pull resistors disabled; however, writing to the mode select bits in the MODE register (which is allowed in special modes) can change this after reset. All of the Port E pins (except PE4/ECLK) are initially configured as general purpose high-impedance inputs with internal pull resistors enabled. PE4/ECLK is configured as the E clock output in this mode.

The pins associated with Port E bits 6, 5, 3, and 2 cannot be configured for their alternate functions IPIPE1, IPIPE0, $\overline{\text{LSTRB}}$, and R/$\overline{\text{W}}$ while the MCU is in single chip modes. In single chip modes, the associated control bits PIPOE, LSTRE and RDWE are reset to zero. Writing the opposite value into these bits in single chip mode does not change the operation of the associated Port E pins.

Port E, bit 4 can be configured for a free-running E clock output by clearing NECLK=0. Typically the only use for an E clock output while the MCU is in single chip modes would be to get a constant speed clock for use in the external application system.

4.4.3.2.2 Special Test Mode

In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E provides bus control and status signals. In special test mode, the write protection of many control bits is lifted so that they can be thoroughly tested without needing to go through reset.

4.4.3.3 Test Operating Mode

There is a test operating mode in which an external master, such as an I.C. tester, can control the on-chip peripherals.

4.4.3.3.1 Peripheral Mode

This mode is intended for factory testing of the MCU. In this mode, the CPU is inactive and an external (tester) bus master drives address, data and bus control signals in through Ports A, B and E. In effect, the whole MCU acts as if it was a peripheral under control of an external CPU. This allows faster testing of on-chip memory and peripherals than previous testing methods. Since the mode control register is not accessible in peripheral mode, the only way to change to another mode is to reset the MCU into a different

Chapter 10
Freescale's Scalable Controller Area Network (S12MSCANV2)

10.1 Introduction

Freescale's scalable controller area network (S12MSCANV2) definition is based on the MSCAN12 definition, which is the specific implementation of the MSCAN concept targeted for the M68HC12 microcontroller family.

The module is a communication controller implementing the CAN 2.0A/B protocol as defined in the Bosch specification dated September 1991. For users to fully understand the MSCAN specification, it is recommended that the Bosch specification be read first to familiarize the reader with the terms and concepts contained within this document.

Though not exclusively intended for automotive applications, CAN protocol is designed to meet the specific requirements of a vehicle serial data bus: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness, and required bandwidth.

MSCAN uses an advanced buffer arrangement resulting in predictable real-time behavior and simplified application software.

10.1.1 Glossary

ACK: Acknowledge of CAN message

CAN: Controller Area Network

CRC: Cyclic Redundancy Code

EOF: End of Frame

FIFO: First-In-First-Out Memory

IFS: Inter-Frame Sequence

SOF: Start of Frame

CPU bus: CPU related read/write data bus

CAN bus: CAN protocol related serial bus

oscillator clock: Direct clock from external oscillator

bus clock: CPU bus realated clock

CAN clock: CAN protocol related clock

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID2	ID1	ID0	RTR	IDE (=0)			
Reset:	x	x	x	x	x	x	x	x

= Unused; always read 'x'

Figure 10-30. Identifier Register 1 — Standard Mapping

Table 10-29. IDR1 Register Field Descriptions

Field	Description
7:5 ID[2:0]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 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. See also ID bits in Table 10-28.
4 RTR	**Remote Transmission Request** — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame
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)

Module Base + 0x00X2

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

= Unused; always read 'x'

Figure 10-31. Identifier Register 2 — Standard Mapping

— a) the 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages or

— b) the 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages. Figure 10-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDA3, CANIDMR0–3CANIDMR) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.

- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier. Figure 10-41 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

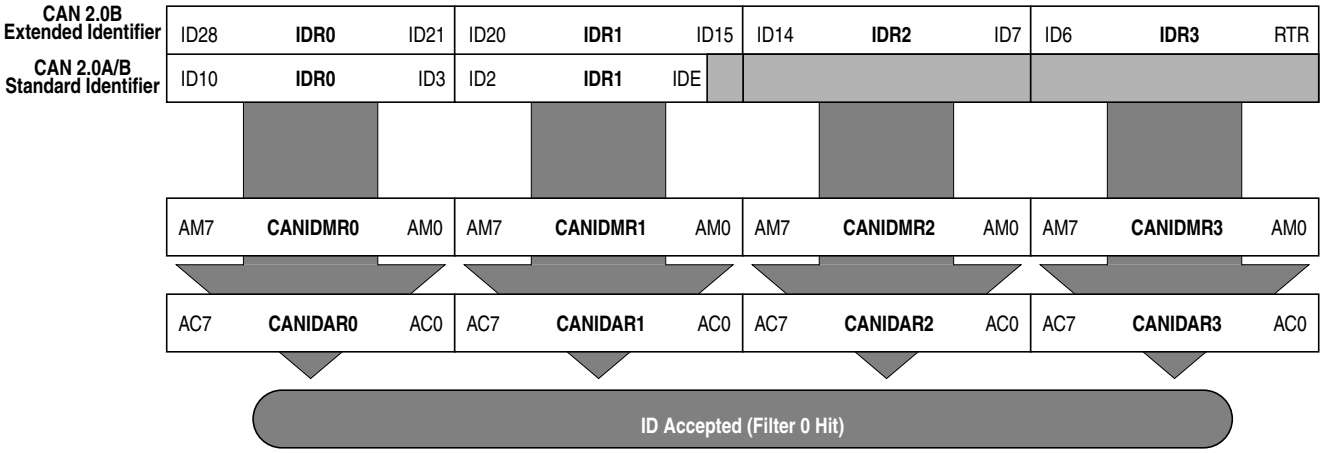

Figure 10-39. 32-bit Maskable Identifier Acceptance Filter

12.2.4 PWM2 — Pulse Width Modulator Channel 2 Pin

This pin serves as waveform output of PWM channel 2.

12.2.5 PWM1 — Pulse Width Modulator Channel 1 Pin

This pin serves as waveform output of PWM channel 1.

12.2.6 PWM0 — Pulse Width Modulator Channel 0 Pin

This pin serves as waveform output of PWM channel 0.

12.3 Memory Map and Register Definition

This subsection describes in detail all the registers and register bits in the PWM8B6CV1 module.

The special-purpose registers and register bit functions that would not normally be made available to device end users, such as factory test control registers and reserved registers are clearly identified by means of shading the appropriate portions of address maps and register diagrams. Notes explaining the reasons for restricting access to the registers and functions are also explained in the individual register descriptions.

12.3.1 Module Memory Map

The following paragraphs describe the content of the registers in the PWM8B6CV1 module. The base address of the PWM8B6CV1 module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset. Table 12-1 shows the registers associated with the PWM and their relative offset from the base address. The register detail description follows the order in which they appear in the register map.

Reserved bits within a register will always read as 0 and the write will be unimplemented. Unimplemented functions are indicated by shading the bit.

Table 12-1 shows the memory map for the PWM8B6CV1 module.

NOTE

Register address = base address + address offset, where the base address is defined at the MCU level and the address offset is defined at the module level.

Table 12-9. PWMCTL Field Descriptions

Field	Description
6 CON45	**Concatenate Channels 4 and 5** 0 Channels 4 and 5 are separate 8-bit PWMs. 1 Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high-order byte and channel 5 becomes the low-order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.
5 CON23	**Concatenate Channels 2 and 3** 0 Channels 2 and 3 are separate 8-bit PWMs. 1 Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high-order byte and channel 3 becomes the low-order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.
4 CON01	**Concatenate Channels 0 and 1** 0 Channels 0 and 1 are separate 8-bit PWMs. 1 Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high-order byte and channel 1 becomes the low-order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.
3 PSWAI	**PWM Stops in Wait Mode** — Enabling this bit allows for lower power consumption in wait mode by disabling the input clock to the prescaler. 0 Allow the clock to the prescaler to continue while in wait mode. 1 Stop the input clock to the prescaler whenever the MCU is in wait mode.
2 PFRZ	**PWM Counters Stop in Freeze Mode** — In freeze mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that after normal program flow is continued, the counters are re-enabled to simulate real-time operations. Because the registers remain accessible in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode. 0 Allow PWM to continue while in freeze mode. 1 Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

13.3.2.2 SCI Control Register 1 (SCICR1)

Module Base + 0x_0002

	7	6	5	4	3	2	1	0
R W	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

Figure 13-4. SCI Control Register 1 (SCICR1)

Read: Anytime

Write: Anytime

Table 13-2. SCICR1 Field Descriptions

Field	Description
7 LOOPS	**Loop Select Bit** — LOOPS enables loop operation. In loop operation, the RXD pin is disconnected from the SCI and the transmitter output is internally connected to the receiver input. Both the transmitter and the receiver must be enabled to use the loop function.See Table 13-3. 0 Normal operation enabled 1 Loop operation enabled **Note:** The receiver input is determined by the RSRC bit.
6 SCISWAI	**SCI Stop in Wait Mode Bit** — SCISWAI disables the SCI in wait mode. 0 SCI enabled in wait mode 1 SCI disabled in wait mode
5 RSRC	**Receiver Source Bit** — When LOOPS = 1, the RSRC bit determines the source for the receiver shift register input. 0 Receiver input internally connected to transmitter output 1 Receiver input connected externally to transmitter
4 M	**Data Format Mode Bit** — MODE determines whether data characters are eight or nine bits long. 0 One start bit, eight data bits, one stop bit 1 One start bit, nine data bits, one stop bit
3 WAKE	**Wakeup Condition Bit** — WAKE determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received data character or an idle condition on the RXD. 0 Idle line wakeup 1 Address mark wakeup
2 ILT	**Idle Line Type Bit** — ILT determines when the receiver starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. 0 Idle character bit count begins after start bit 1 Idle character bit count begins after stop bit
1 PE	**Parity Enable Bit** — PE enables the parity function. When enabled, the parity function inserts a parity bit in the most significant bit position. 0 Parity function disabled 1 Parity function enabled
0 PT	**Parity Type Bit** — PT determines whether the SCI generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. 0 Even parity 1 Odd parity

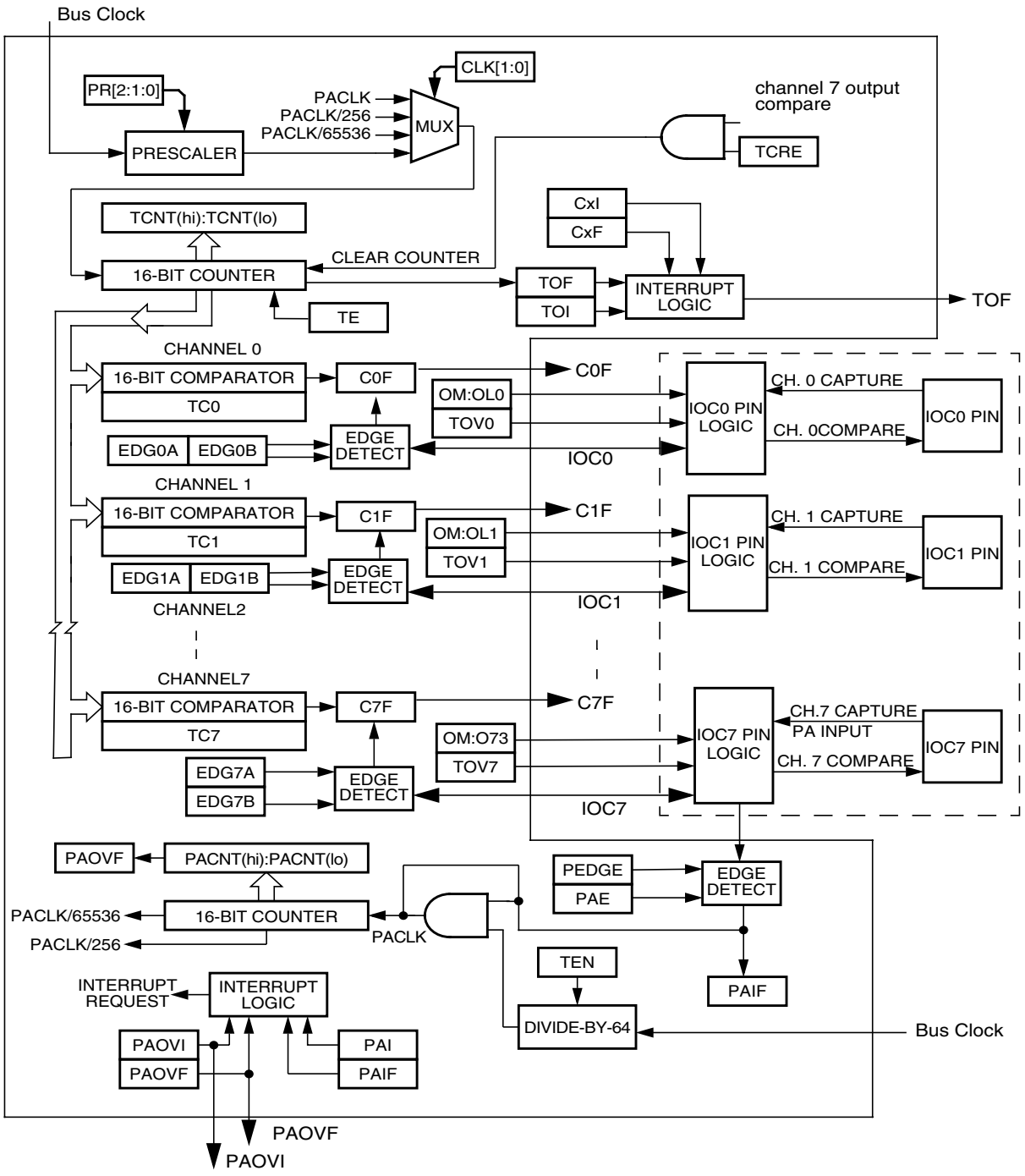

Figure 15-28. Detailed Timer Block Diagram

15.4.1 Prescaler

The prescaler divides the bus clock by 1,2,4,8,16,32,64 or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in timer system control register 2 (TSCR2).

17.3.2 Register Descriptions

The Flash module contains a set of 16 control and status registers located between module base + 0x0000 and 0x000F. A summary of the Flash module registers is given in Figure 17-3. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 FCLKDIV	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
	W								
0x0002 RESERVED1 (1)	R	0	0	0	0	0	0	0	0
	W								
0x0003 FCNFG	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
	W								
0x0004 FPROT	R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	NV2	NV1	NV0
	W								
0x0005 FSTAT	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
	W								
0x0006 FCMD	R	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
	W								
0x0007 RESERVED2[1]	R	0	0	0	0	0	0	0	0
	W								
0x0008 FADDRHI[1]	R	0	0	0	FABHI				
	W								
0x0009 FADDRLO[1]	R	FABLO							
	W								
0x000A FDATAHI[1]	R	FDHI							
	W								
0x000B FDATALO[1]	R	FDLO							
	W								
0x000C RESERVED3[1]	R	0	0	0	0	0	0	0	0
	W								
0x000D RESERVED4[1]	R	0	0	0	0	0	0	0	0
	W								
0x000E RESERVED5[1]	R	0	0	0	0	0	0	0	0
	W								
0x000F RESERVED6[1]	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 17-3. Flash Register Summary

1. Intended for factory test purposes only.

- Tbus as the period of the bus clock
- INT(x) as taking the integer part of x (e.g., INT(4.323) = 4),

then FCLKDIV register bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 19-24.

For example, if the oscillator clock frequency is 950 kHz and the bus clock is 10 MHz, FCLKDIV bits FDIV[5:0] should be set to 4 (000100) and bit PRDIV8 set to 0. The resulting FCLK is then 190 kHz. As a result, the Flash algorithm timings are increased over optimum target by:

$$(200 - 190) / 200 \times 100 \; = \; 5\%$$

Command execution time will increase proportionally with the period of FCLK.

CAUTION

Because of the impact of clock synchronization on the accuracy of the functional timings, programming or erasing the Flash array cannot be performed if the bus clock runs at less than 1 MHz. Programming or erasing the Flash array with an input clock < 150 kHz should be avoided. Setting FCLKDIV to a value such that FCLK < 150 kHz can destroy the Flash array due to overstress. Setting FCLKDIV to a value such that (1/FCLK + Tbus) < 5µs can result in incomplete programming or erasure of the Flash array cells.

If the FCLKDIV register is written, the bit FDIVLD is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written to, the Flash command loaded during a command write sequence will not execute and the ACCERR flag in the FSTAT register will set.

19.4.1.3.2 Program Command

The program operation will program a previously erased word in the Flash array using an embedded algorithm.

An example flow to execute the program operation is shown in Figure 19-26. The program command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the program command. The data written will be programmed to the Flash array address written.
2. Write the program command, 0x20, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the program command.

If a word to be programmed is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the program command will not launch. Once the program command has successfully launched, the CCIF flag in the FSTAT register will set after the program operation has completed unless a new command write sequence has been buffered. By executing a new program command write sequence on sequential words after the CBEIF flag in the FSTAT register has been set, up to 55% faster programming time per word can be effectively achieved than by waiting for the CCIF flag to set after each program operation.

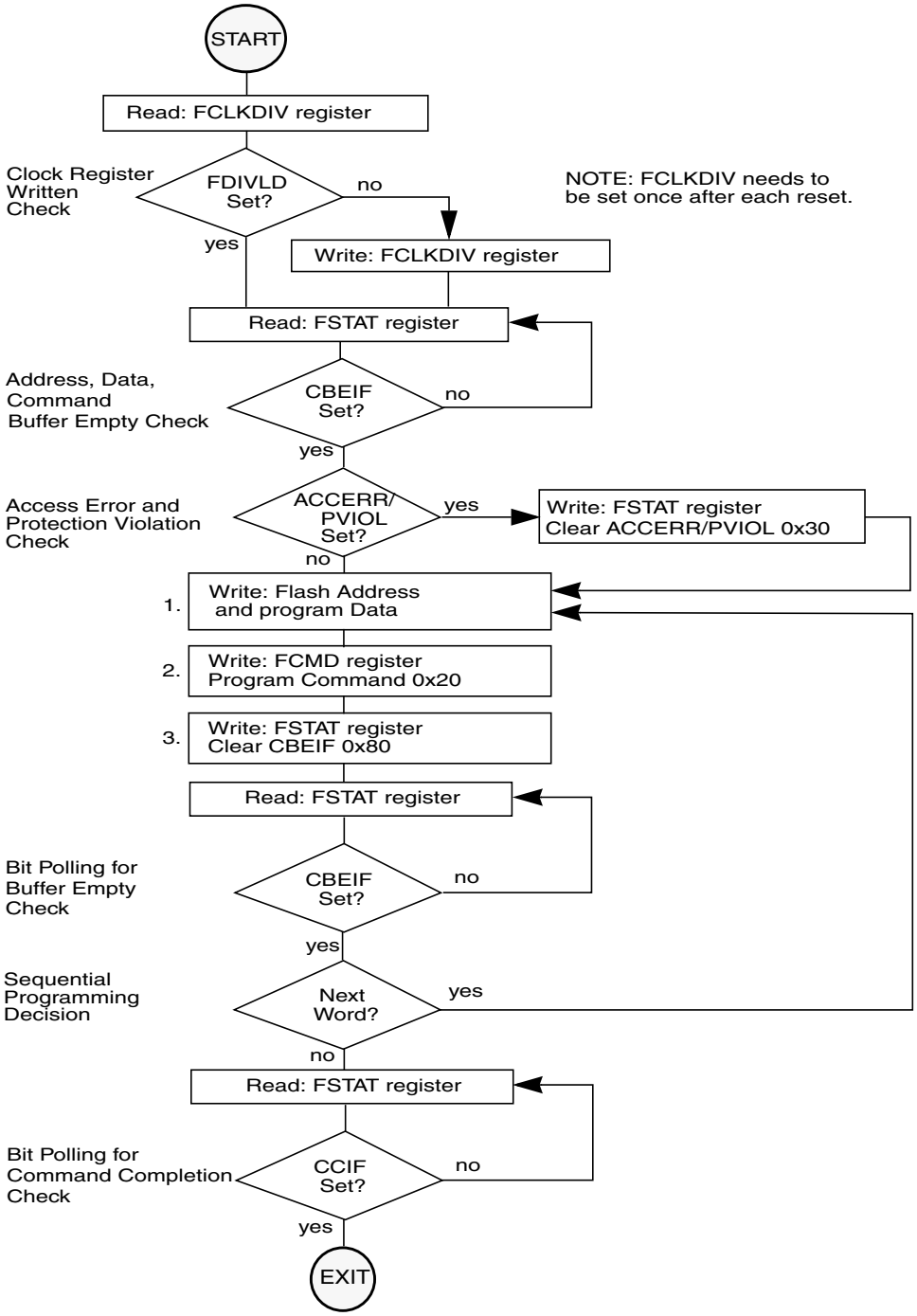

Figure 19-26. Example Program Command Flow

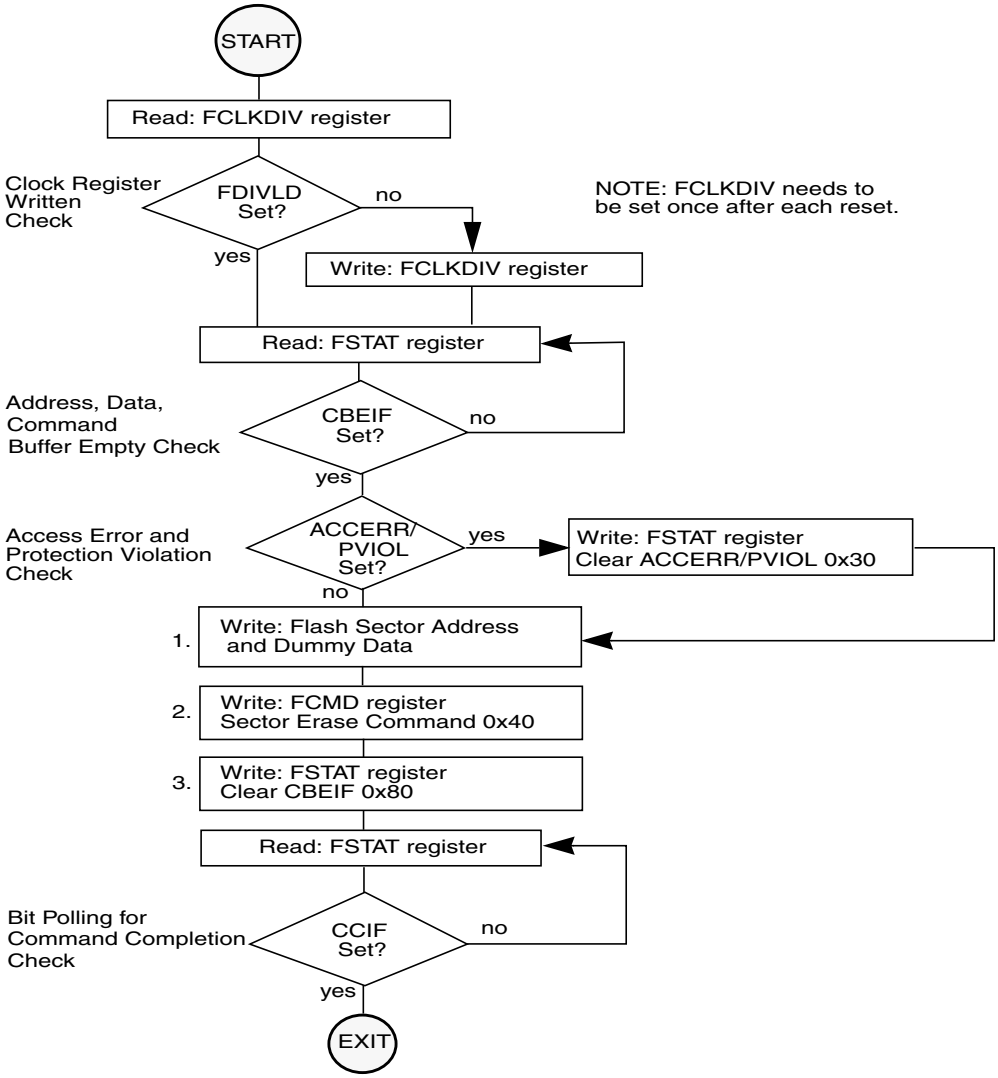

Figure 20-26. Example Sector Erase Command Flow

21.3.2 Register Descriptions

The Flash module contains a set of 16 control and status registers located between module base + 0x0000 and 0x000F. A summary of the Flash module registers is given in Figure 21-3. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 FCLKDIV	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
	W								
0x0002 RESERVED1 (1)	R	0	0	0	0	0	0	0	0
	W								
0x0003 FCNFG	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
	W								
0x0004 FPROT	R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0005 FSTAT	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
	W								
0x0006 FCMD	R	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
	W								
0x0007 RESERVED2[1]	R	0	0	0	0	0	0	0	0
	W								
0x0008 FADDRHI[1]	R	FABHI							
	W								
0x0009 FADDRLO[1]	R	FABLO							
	W								
0x000A FDATAHI[1]	R	FDHI							
	W								
0x000B FDATALO[1]	R	FDLO							
	W								
0x000C RESERVED3[1]	R	0	0	0	0	0	0	0	0
	W								
0x000D RESERVED4[1]	R	0	0	0	0	0	0	0	0
	W								
0x000E RESERVED5[1]	R	0	0	0	0	0	0	0	0
	W								
0x000F RESERVED6[1]	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 21-3. Flash Register Summary

1. Intended for factory test purposes only.

Table A-7. 3.3V I/O Characteristics

Conditions are V_{DDX}=3.3V +/-10%, Temperature from –40˚C to +140˚C, unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Input High Voltage	V_{IH}	0.65*V_{DD5}	—	—	V
	T	Input High Voltage	V_{IH}	—	—	V_{DD5} + 0.3	V
2	P	Input Low Voltage	V_{IL}	—	—	0.35*V_{DD5}	V
	T	Input Low Voltage	V_{IL}	V_{SS5} – 0.3	—	—	V
3	C	Input Hysteresis	V_{HYS}	—	250	—	mV
4	P	Input Leakage Current (pins in high ohmic input mode)(1) V_{in} = V_{DD5} or V_{SS5}	I_{in}	–1	—	1	µA
5	C	Output High Voltage (pins in output mode) Partial Drive I_{OH} = –0.75mA	V_{OH}	V_{DD5} – 0.4	—	—	V
6	P	Output High Voltage (pins in output mode) Full Drive I_{OH} = –4mA	V_{OH}	V_{DD5} – 0.4	—	—	V
7	C	Output Low Voltage (pins in output mode) Partial Drive I_{OL} = +0.9mA	V_{OL}	—	—	0.4	V
8	P	Output Low Voltage (pins in output mode) Full Drive I_{OL} = +4.75mA	V_{OL}	—	—	0.4	V
9	P	Internal Pull Up Device Current, tested at V_{IL} Max.	I_{PUL}	—	—	–60	µA
10	C	Internal Pull Up Device Current, tested at V_{IH} Min.	I_{PUH}	-6	—	—	µA
11	P	Internal Pull Down Device Current, tested at V_{IH} Min.	I_{PDH}	—	—	60	µA
12	C	Internal Pull Down Device Current, tested at V_{IL} Max.	I_{PDL}	6	—	—	µA
11	D	Input Capacitance	C_{in}	—	7	—	πΦ
12	T	Injection current(2) Single Pin limit Total Device Limit. Sum of all injected currents	I_{ICS} I_{ICP}	–2.5 –25	—	2.5 25	µA
13	P	Port P, J Interrupt Input Pulse filtered(3)	t_{PIGN}	—	—	3	µs
14	P	Port P, J Interrupt Input Pulse passed3	t_{PVAL}	10	—	—	µs

1. Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8 C to 12 C in the temperature range from 50 C to 125 C
2. Refer to Section A.1.4, “Current Injection”, for more details
3. Parameter only applies in STOP or Pseudo STOP mode.

A.1.10 Supply Currents

This section describes the current consumption characteristics of the device as well as the conditions for the measurements.

A.1.10.1 Measurement Conditions

All measurements are without output loads. Unless otherwise noted the currents are measured in single chip mode, internal voltage regulator enabled and at 25MHz bus frequency using a 4MHz oscillator.