



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, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	96KB (96K 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	52-QFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12c96mpbe

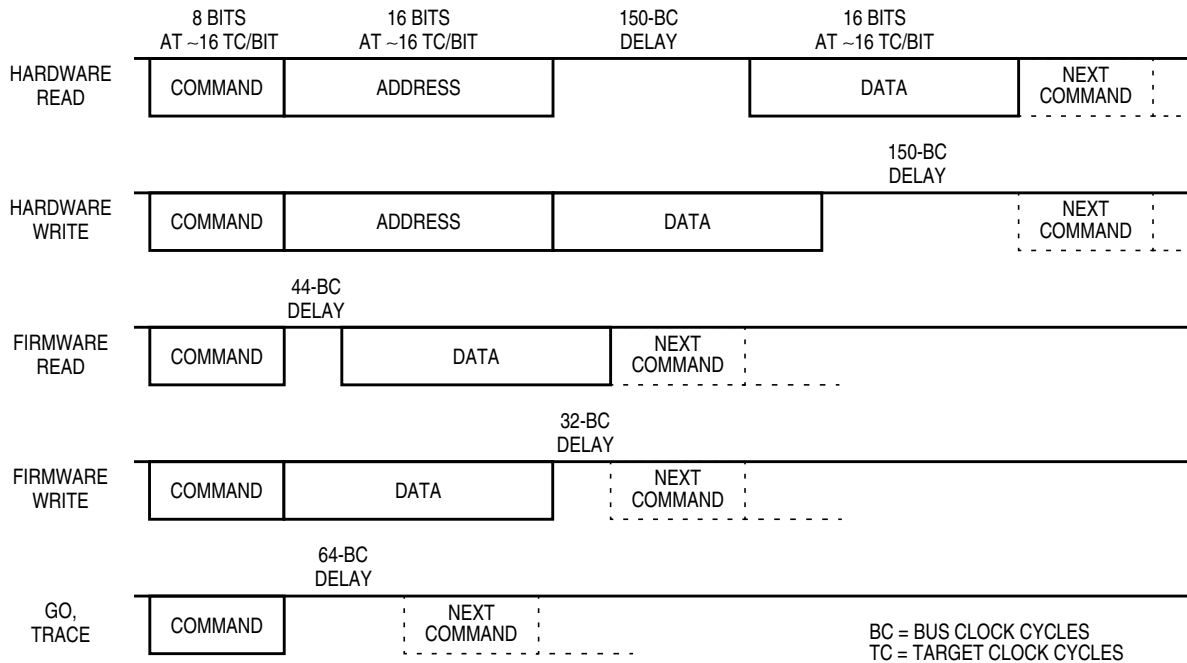


Figure 6-6. BDM Command Structure

6.4.6 BDM Serial Interface

The BDM communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDM.

The BDM serial interface is timed using the clock selected by the CLKS_W bit in the status register see [Section 6.3.2.1, “BDM Status Register \(BDMSTS\).”](#) This clock will be referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and that drivers connected to BKGD do not typically drive the high level. Because R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in [Figure 6-7](#) and that of target-to-host in [Figure 6-8](#) and [Figure 6-9](#). All four cases begin when the host drives the BKGD pin low to generate a falling edge. Because the host and target are operating from separate clocks, it can take the target system up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Upon return to standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

6.4.11 Instruction Tagging

The instruction queue and cycle-by-cycle CPU activity are reconstructible in real time or from trace history that is captured by a logic analyzer. However, the reconstructed queue cannot be used to stop the CPU at a specific instruction. This is because execution already has begun by the time an operation is visible outside the system. A separate instruction tagging mechanism is provided for this purpose.

The tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue, the CPU enters active BDM rather than executing the instruction.

NOTE

Tagging is disabled when BDM becomes active and BDM serial commands are not processed while tagging is active.

Executing the BDM TAGGO command configures two system pins for tagging. The $\overline{\text{TAGLO}}$ signal shares a pin with the $\overline{\text{LSTRB}}$ signal, and the $\overline{\text{TAGHI}}$ signal shares a pin with the BKGD signal.

Table 6-7 shows the functions of the two tagging pins. The pins operate independently, that is the state of one pin does not affect the function of the other. The presence of logic level 0 on either pin at the fall of the external clock (ECLK) performs the indicated function. High tagging is allowed in all modes. Low tagging is allowed only when low strobe is enabled (LSTRB is allowed only in wide expanded modes and emulation expanded narrow mode).

Table 6-7. Tag Pin Function

$\overline{\text{TAGHI}}$	$\overline{\text{TAGLO}}$	Tag
1	1	No tag
1	0	Low byte
0	1	High byte
0	0	Both bytes

6.4.12 Serial Communication Time-Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

7.3.2.4 Debug Count Register (DBGCNT)

Module Base + 0x0024

Starting address location affected by INITRG register setting.

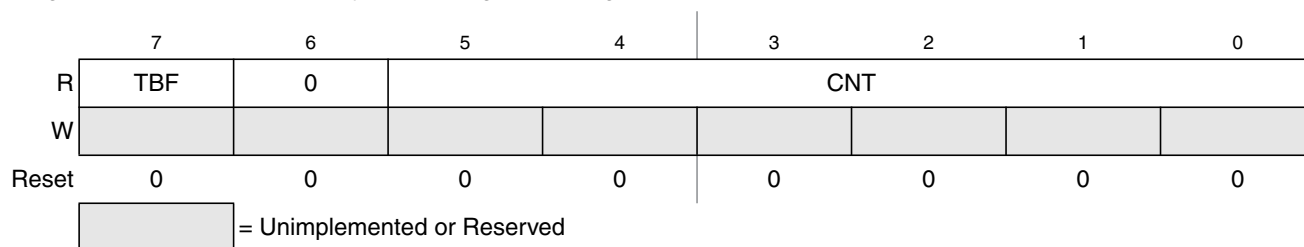


Figure 7-8. Debug Count Register (DBGCNT)

Table 7-8. DBG CNT Field Descriptions

Field	Description
7 TBF	Trace Buffer Full — The TBF bit indicates that the trace buffer has stored 64 or more words of data since it was last armed. If this bit is set, then all 64 words will be valid data, regardless of the value in CNT[5:0]. The TBF bit is cleared when ARM in DBG C1 is written to a 1.
5:0 CNT	Count Value — The CNT bits indicate the number of valid data words stored in the trace buffer. Table 7-9 shows the correlation between the CNT bits and the number of valid data words in the trace buffer. When the CNT rolls over to 0, the TBF bit will be set and incrementing of CNT will continue if DBG is in end-trigger mode. The DBG CNT register is cleared when ARM in DBG C1 is written to a 1.

Table 7-9. CNT Decoding Table

TBF	CNT	Description
0	000000	No data valid
0	000001	1 word valid
0	000010	2 words valid

	111110	62 words valid
0	111111	63 words valid
1	000000	64 words valid; if BEGIN = 1, the ARM bit will be cleared. A breakpoint will be generated if DBGBRK = 1
1	000001	64 words valid, oldest data has been overwritten by most recent data
	..	
	..	
	111111	

7.4.2.6 Capture Modes

The DBG in DBG mode can operate in four capture modes. These modes are described in the following subsections.

7.4.2.6.1 Normal Mode

In normal mode, the DBG module uses comparator A and B as triggering devices. Change-of-flow information or data will be stored depending on TRG in DBGSC.

7.4.2.6.2 Loop1 Mode

The intent of loop1 mode is to prevent the trace buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the trace buffer, the DBG module writes this value into the C comparator and the C comparator is placed in ignore address mode. This will prevent duplicate address entries in the trace buffer resulting from repeated bit-conditional branches. Comparator C will be cleared when the ARM bit is set in loop1 mode to prevent the previous contents of the register from interfering with loop1 mode operation. Breakpoints based on comparator C are disabled.

Loop1 mode only inhibits duplicate source address entries that would typically be stored in most tight looping constructs. It will not inhibit repeated entries of destination addresses or vector addresses, because repeated entries of these would most likely indicate a bug in the user's code that the DBG module is designed to help find.

NOTE

In certain very tight loops, the source address will have already been fetched again before the C comparator is updated. This results in the source address being stored twice before further duplicate entries are suppressed. This condition occurs with branch-on-bit instructions when the branch is fetched by the first P-cycle of the branch or with loop-construct instructions in which the branch is fetched with the first or second P cycle. See examples below:

```

LOOP  INCX                ; 1-byte instruction fetched by 1st P-cycle of BRCLR
      BRCLR CMPTMP,#$0c,LOOP ; the BRCLR instruction also will be fetched by 1st P-cycle of BRCLR

LOOP2 BRN   *              ; 2-byte instruction fetched by 1st P-cycle of DBNE
      NOP                ; 1-byte instruction fetched by 2nd P-cycle of DBNE
      DBNE  A,LOOP2       ; this instruction also fetched by 2nd P-cycle of DBNE
    
```

NOTE

Loop1 mode does not support paged memory, and inhibits duplicate entries in the trace buffer based solely on the CPU address. There is a remote possibility of an erroneous address match if program flow alternates between paged and unpaged memory space.

Table 10-10. CANRIER Register Field Descriptions (continued)

Field	Description
1 OVRIE	Overrun Interrupt Enable 0 No interrupt request is generated from this event. 1 An overrun event causes an error interrupt request.
0 RXFIE	Receiver Full Interrupt Enable 0 No interrupt request is generated from this event. 1 A receive buffer full (successful message reception) event causes a receiver interrupt request.

1. WUPIE and WUPE (see Section 10.3.2.1, "MSCAN Control Register 0 (CANCTL0)") must both be enabled if the recovery mechanism from stop or wait is required.
2. Bus-off state is defined by the CAN standard (see Bosch CAN 2.0A/B protocol specification: for only transmitters. Because the only possible state change for the transmitter from bus-off to TxOK also forces the receiver to skip its current state to RxOK, the coding of the RXSTAT[1:0] flags define an additional bus-off state for the receiver (see Section 10.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)").

10.3.2.7 MSCAN Transmitter Flag Register (CANTFLG)

The transmit buffer empty flags each have an associated interrupt enable bit in the CANTIER register.

Module Base + 0x0006

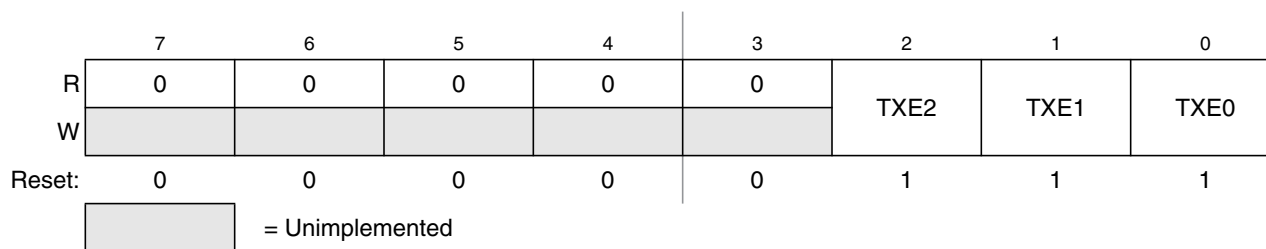


Figure 10-10. MSCAN Transmitter Flag Register (CANTFLG)

NOTE

The CANTFLG 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).

Read: Anytime

Write: Anytime for TXEx flags when not in initialization mode; write of 1 clears flag, write of 0 is ignored

12.3.2.12 PWM Channel Counter Registers (PWMCNTx)

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register – 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0.

Any value written to the counter causes the counter to reset to 0x0000, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see Section 12.4.2.5, “Left Aligned Outputs,” and Section 12.4.2.6, “Center Aligned Outputs,” for more details). When the channel is disabled (PWME_x = 0), the PWMCNT_x register does not count. When a channel becomes enabled (PWME_x = 1), the associated PWM counter starts at the count in the PWMCNT_x register. For more detailed information on the operation of the counters, reference Section 12.4.2.4, “PWM Timer Counters.”

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low- or high-order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

NOTE

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

Module Base + 0x000C

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

Figure 12-15. PWM Channel Counter Registers (PWMCNT0)

Module Base + 0x000D

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

Figure 12-16. PWM Channel Counter Registers (PWMCNT1)

Note: in Table 15-11, the IOS7 and IOSx should be set to 1

IOSx is the register TIOS bit x,

OC7Mx is the register OC7M bit x,

TCx is timer Input Capture/Output Compare register,

IOCx is channel x,

OMx/OLx is the register TCTL1/TCTL2,

OC7Dx is the register OC7D bit x.

$IOCx = OC7Dx + OMx/OLx$, means that both OC7 event and OCx event will change channel x value.

15.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOS_x, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TC_x.

The minimum pulse width for the input capture input is greater than two bus clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 must be set to one) while clearing CxF (writing one to CxF).

15.4.3 Output Compare

Setting the I/O select bit, IOS_x, configures channel x as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OM_x and OL_x, select set, clear, toggle on output compare. Clearing both OM_x and OL_x disconnects the pin from the output logic.

Setting a force output compare bit, FOC_x, causes an output compare on channel x. A forced output compare does not set the channel flag.

A successful output compare on channel 7 overrides output compares on all other output compare channels. The output compare 7 mask register masks the bits in the output compare 7 data register. The timer counter reset enable bit, TCRE, enables channel 7 output compares to reset the timer counter. A channel 7 output compare can reset the timer counter even if the IOC7 pin is being used as the pulse accumulator input.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

When TCRE is set and TC7 is not equal to 0, then TCNT will cycle from 0 to TC7. When TCNT reaches TC7 value, it will last only one bus cycle then reset to 0.

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

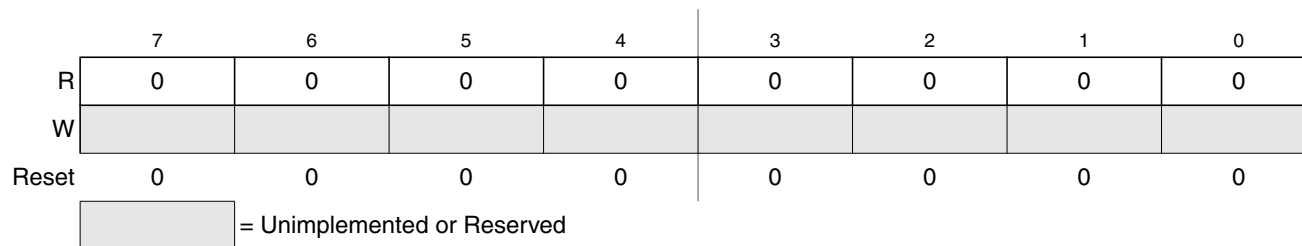


Figure 17-6. RESERVED1

All bits read 0 and are not writable.

18.3 Memory Map and Registers

This section describes the [FTS32K](#) memory map and registers.

18.3.1 Module Memory Map

The [FTS32K](#) memory map is shown in [Figure 18-2](#). The HCS12 architecture places the Flash array addresses between [0x4000](#) and [0xFFFF](#), which corresponds to three 16 Kbyte pages. The content of the HCS12 Core PPAGE register is used to map the logical [middle](#) page ranging from address [0x8000](#) to [0xBFFF](#) to any physical 16K byte page in the Flash array memory.¹ The FPROT register (see [Section 18.3.2.5](#)) can be set to globally protect the entire Flash array. Three separate areas, one starting from the Flash array starting address (called [lower](#)) towards higher addresses, one growing downward from the Flash array end address (called [higher](#)), and the remaining addresses, can be activated for protection. The Flash array addresses covered by these protectable regions are shown in [Figure 18-2](#). The higher address area is mainly targeted to hold the boot loader code since it covers the vector space. The lower address area can be used for EEPROM emulation in an MCU without an EEPROM module since it can be left unprotected while the remaining addresses are protected from program or erase. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field described in [Table 18-1](#).

Table 18-1. Flash Configuration Field

Flash Address	Size (bytes)	Description
0xFF00–0xFF07	8	Backdoor Key to unlock security
0xFF08–0xFF0C	5	Reserved
0xFF0D	1	Flash Protection byte Refer to Section 18.3.2.5 , “Flash Protection Register (FPROT)”
0xFF0E	1	Reserved
0xFF0F	1	Flash Security/Options byte Refer to Section 18.3.2.2 , “Flash Security Register (FSEC)”

1. By placing [0x3E/0x3F](#) in the HCS12 Core PPAGE register, the [bottom/top fixed](#) 16 Kbyte pages can be seen twice in the MCU memory map.

Module Base + 0x000F

	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 18-20. RESERVED6

All bits read 0 and are not writable.

18.4 Functional Description

18.4.1 Flash Command Operations

Write operations are used for the program, erase, and erase verify algorithms described in this section. The program and erase algorithms are controlled by a state machine whose timebase FCLK is derived from the oscillator clock via a programmable divider. The FCMD register as well as the associated FADDR and FDATA registers operate as a buffer and a register (2-stage FIFO) so that a new command along with the necessary data and address can be stored to the buffer while the previous command is still in progress. This pipelined operation allows a time optimization when programming more than one word on a specific row, as the high voltage generation can be kept active in between two programming commands. The pipelined operation also allows a simplification of command launching. Buffer empty as well as command completion are signalled by flags in the FSTAT register with corresponding interrupts generated, if enabled.

The next sections describe:

- How to write the FCLKDIV register
- Command write sequence used to program, erase or erase verify the Flash array
- Valid Flash commands
- Errors resulting from illegal Flash operations

18.4.1.1 Writing the FCLKDIV Register

Prior to issuing any Flash command after a reset, it is first necessary to write the FCLKDIV register to divide the oscillator clock down to within the 150-kHz to 200-kHz range. Since the program and erase timings are also a function of the bus clock, the FCLKDIV determination must take this information into account.

If we define:

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

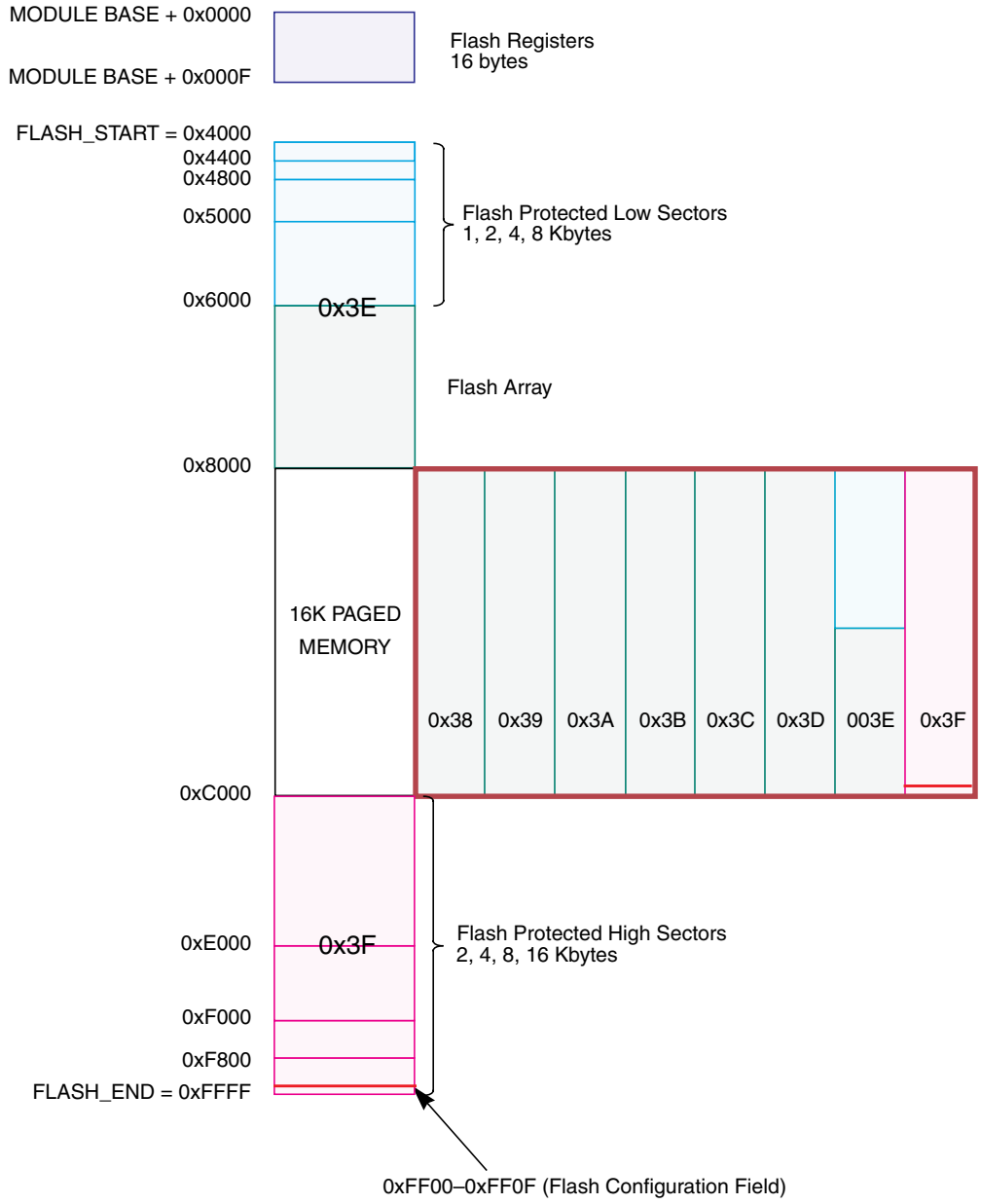
18.4.1.3.4 Mass Erase Command

The mass erase operation will erase all addresses in a Flash array using an embedded algorithm.

An example flow to execute the mass erase operation is shown in [Figure 18-25](#). The mass erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the mass erase command. The address and data written will be ignored.
2. Write the mass erase command, 0x41, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the mass erase command.

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



Note: 0x38–0x3F correspond to the PPAGE register content

Figure 20-3. Flash Memory Map

20.3.2.4 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash interrupts and gates the security backdoor key writes.

Module Base + 0x0003

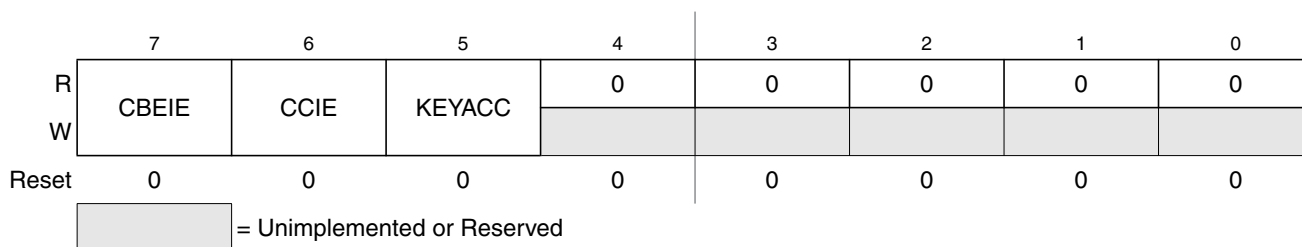


Figure 20-9. Flash Configuration Register (FCNFG)

CBEIE, CCIE, and KEYACC are readable and writable while remaining bits read 0 and are not writable. KEYACC is only writable if the KEYEN bit in the FSEC register is set to the enabled state (see [Section 20.3.2.2](#)).

Table 20-8. FCNFG Field Descriptions

Field	Description
7 CBEIE	Command Buffer Empty Interrupt Enable — The CBEIE bit enables the interrupts in case of an empty command buffer in the Flash module. 0 Command Buffer Empty interrupts disabled 1 An interrupt will be requested whenever the CBEIF flag is set (see Section 20.3.2.6)
6 CCIE	Command Complete Interrupt Enable — The CCIE bit enables the interrupts in case of all commands being completed in the Flash module. 0 Command Complete interrupts disabled 1 An interrupt will be requested whenever the CCIF flag is set (see Section 20.3.2.6)
5 KEYACC	Enable Security Key Writing. 0 Flash writes are interpreted as the start of a command write sequence 1 Writes to the Flash array are interpreted as a backdoor key while reads of the Flash array return invalid data

20.3.2.5 Flash Protection Register (FPROT)

The FPROT register defines which Flash sectors are protected against program or erase.

Module Base + 0x0004

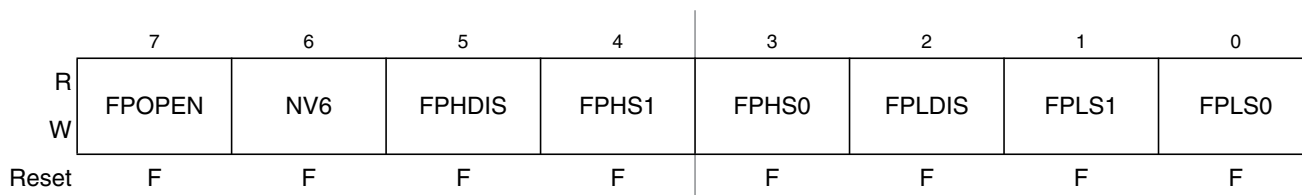


Figure 20-10. Flash Protection Register (FPROT)

The FPROT register is readable in normal and special modes. FPOPEN can only be written from a 1 to a 0. [FPLS\[1:0\]](#) can be written anytime until [FPLDIS](#) is cleared. [FPHS\[1:0\]](#) can be written anytime until

Table 21-9. Flash Protection Function

FPOpen	FPHDis	FPHS[1]	FPHS[0]	FPLDis	FPLS[1]	FPLS[0]	Function ⁽¹⁾
1	1	x	x	1	x	x	No protection
1	1	x	x	0	x	x	Protect low range
1	0	x	x	1	x	x	Protect high range
1	0	x	x	0	x	x	Protect high and low ranges
0	1	x	x	1	x	x	Full Flash array protected
0	0	x	x	1	x	x	Unprotected high range
0	1	x	x	0	x	x	Unprotected low range
0	0	x	x	0	x	x	Unprotected high and low ranges

1. For range sizes refer to [Table 21-10](#) and [Table 21-11](#) or .

Table 21-10. Flash Protection Higher Address Range

FPHS[1:0]	Address Range	Range Size
00	0xF800–0xFFFF	2 Kbytes
01	0xF000–0xFFFF	4 Kbytes
10	0xE000–0xFFFF	8 Kbytes
11	0xC000–0xFFFF	16 Kbytes

Table 21-11. Flash Protection Lower Address Range

FPLS[1:0]	Address Range	Range Size
00	0x4000–0x43FF	1 Kbyte
01	0x4000–0x47FF	2 Kbytes
10	0x4000–0x4FFF	4 Kbytes
11	0x4000–0x5FFF	8 Kbytes

Figure 21-9 illustrates all possible protection scenarios. Although the protection scheme is loaded from the Flash array after reset, it is allowed to change in normal modes. This protection scheme can be used by applications requiring re-programming in single chip mode while providing as much protection as possible if no re-programming is required.

then FCLKDIV register bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 21-21.

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 $(1/FCLK + T_{bus}) < 5\mu s$ 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.

Table A-7. 3.3V I/O Characteristics

Conditions are $V_{DDX}=3.3V \pm 10\%$, Temperature from $-40^{\circ}C$ to $+140^{\circ}C$, unless otherwise noted							
Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	P	Input High Voltage	V_{IH}	$0.65 \cdot V_{DD5}$	—	—	V
	T	Input High Voltage	V_{IH}	—	—	$V_{DD5} + 0.3$	V
2	P	Input Low Voltage	V_{IL}	—	—	$0.35 \cdot V_{DD5}$	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	—	$\pi\Phi$
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 passed ³	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.

A.4.1.3 SRAM Data Retention

Provided an appropriate external reset signal is applied to the MCU, preventing the CPU from executing code when V_{DD5} is out of specification limits, the SRAM contents integrity is guaranteed if after the reset the PORF bit in the CRG Flags Register has not been set.

A.4.1.4 External Reset

When external reset is asserted for a time greater than PW_{RSTL} the CRG module generates an internal reset, and the CPU starts fetching the reset vector without doing a clock quality check, if there was an oscillation before reset.

A.4.1.5 Stop Recovery

Out of STOP the controller can be woken up by an external interrupt. A clock quality check as after POR is performed before releasing the clocks to the system.

A.4.1.6 Pseudo Stop and Wait Recovery

The recovery from Pseudo STOP and Wait are essentially the same since the oscillator was not stopped in both modes. In Pseudo Stop Mode the voltage regulator is switched to reduced performance mode to reduce power consumption. The returning out of pseudo stop to full performance takes t_{vup} . The controller can be woken up by internal or external interrupts. After t_{wrs} in Wait or $t_{vup} + t_{wrs}$ in Pseudo Stop the CPU starts fetching the interrupt vector.

A.4.2 Oscillator

The device features an internal Colpitts and Pierce oscillator. The selection of Colpitts oscillator or Pierce oscillator/external clock depends on the XCLKS signal which is sampled during reset. Pierce oscillator/external clock mode allows the input of a square wave. Before asserting the oscillator to the internal system clocks the quality of the oscillation is checked for each start from either power-on, STOP or oscillator fail. t_{CQOUT} specifies the maximum time before switching to the internal self clock mode after POR or STOP if a proper oscillation is not detected. The quality check also determines the minimum oscillator start-up time t_{UPOSC} . The device also features a clock monitor. A Clock Monitor Failure is asserted if the frequency of the incoming clock signal is below the Assert Frequency f_{CMFA} .

The loop bandwidth f_C should be chosen to fulfill the Gardner's stability criteria by at least a factor of 10, typical values are 50. $\zeta = 0.9$ ensures a good transient response.

$$f_C < \frac{2 \cdot \zeta \cdot f_{\text{ref}}}{\pi \cdot (\zeta + \sqrt{1 + \zeta^2})} \cdot \frac{1}{10} \rightarrow f_C < \frac{f_{\text{ref}}}{4 \cdot 10}; (\zeta = 0.9)$$

$$f_C < 25\text{kHz}$$

And finally the frequency relationship is defined as

$$n = \frac{f_{\text{VCO}}}{f_{\text{ref}}} = 2 \cdot (\text{synr} + 1) = 50$$

With the above values the resistance can be calculated. The example is shown for a loop bandwidth $f_C=10\text{kHz}$:

$$R = \frac{2 \cdot \pi \cdot n \cdot f_C}{K_\Phi} = 2 \cdot \pi \cdot 50 \cdot 10\text{kHz} / (316.7\text{Hz}/\Omega) = 9.9\text{k}\Omega \approx 10\text{k}\Omega$$

The capacitance C_s can now be calculated as:

$$C_s = \frac{2 \cdot \zeta^2}{\pi \cdot f_C \cdot R} \approx \frac{0.516}{f_C \cdot R}; (\zeta = 0.9) = 5.19\text{nF} \approx 4.7\text{nF}$$

The capacitance C_p should be chosen in the range of:

$$C_s / 20 \leq C_p \leq C_s / 10 \quad C_p = 470\text{pF}$$

A.4.3.2 Jitter Information

The basic functionality of the PLL is shown in [Figure A-3](#). With each transition of the clock f_{cmp} , the deviation from the reference clock f_{ref} is measured and input voltage to the VCO is adjusted accordingly. The adjustment is done continuously with no abrupt changes in the clock output frequency. Noise, voltage, temperature and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in [Figure A-4](#).

A.5.1.1 Single Word Programming

The programming time for single word programming is dependant on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

A.5.1.2 Row Programming

Generally the time to program a consecutive word can be calculated as:

$$t_{bwp gm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

For the C16, GC16, C32 and GC32 device flash arrays, where up to 32 words in a row can be programmed consecutively by keeping the command pipeline filled, the time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 31 \cdot t_{bwp gm}$$

For the C64, GC64, C96, C128 and GC128 device flash arrays, where up to 64 words in a row can be programmed consecutively by keeping the command pipeline filled, the time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 63 \cdot t_{bwp gm}$$

Row programming is more than 2 times faster than single word programming.

A.5.1.3 Sector Erase

Erasing either a 512 byte or 1024 byte Flash sector takes:

$$t_{era} \approx 4000 \cdot \frac{1}{f_{NVMOP}}$$

The setup times can be ignored for this operation.

A.5.1.4 Mass Erase

Erasing a NVM block takes:

$$t_{mass} \approx 20000 \cdot \frac{1}{f_{NVMOP}}$$

This is independent of sector size.

The setup times can be ignored for this operation.