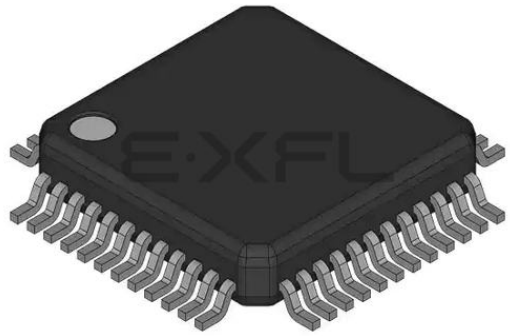


Welcome to E-XFL.COM

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"

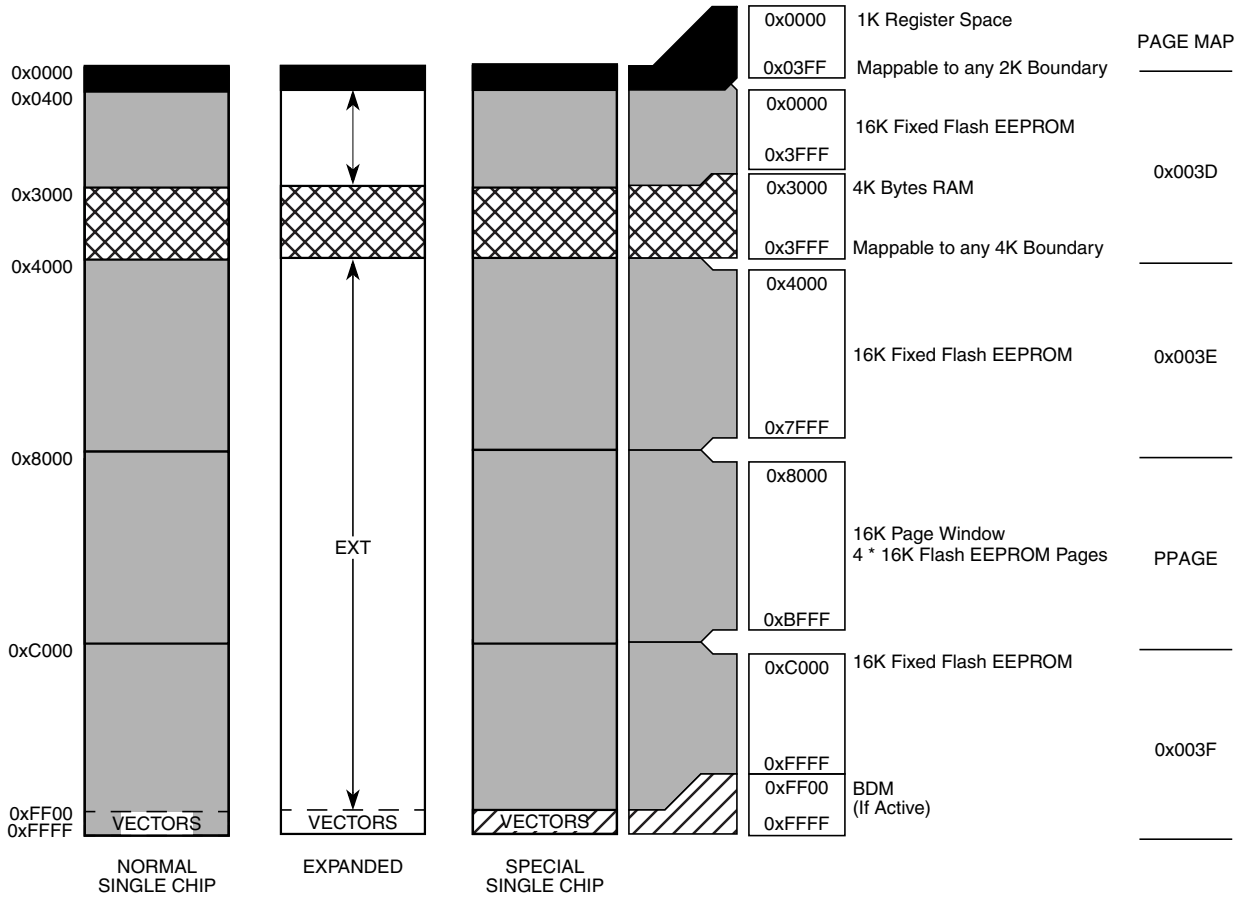
Details	
Product Status	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	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/pro/item?MUrl=&PartUrl=mc9s12c64vfae

- Operating frequency:
  - 32MHz equivalent to 16MHz bus speed for single chip
  - 32MHz equivalent to 16MHz bus speed in expanded bus modes
  - Option of 9S12C Family: 50MHz equivalent to 25MHz bus speed
  - All 9S12GC Family members allow a 50MHz operating frequency.
- Internal 2.5V regulator:
  - Supports an input voltage range from 2.97V to 5.5V
  - Low power mode capability
  - Includes low voltage reset (LVR) circuitry
  - Includes low voltage interrupt (LVI) circuitry
- 48-pin LQFP, 52-pin LQFP, or 80-pin QFP package:
  - Up to 58 I/O lines with 5V input and drive capability (80-pin package)
  - Up to 2 dedicated 5V input only lines (IRQ, XIRQ)
  - 5V 8 A/D converter inputs and 5V I/O
- Development support:
  - Single-wire background debug™ mode (BDM)
  - On-chip hardware breakpoints
  - Enhanced DBG12 debug features

1.1.2 Modes of Operation

User modes (expanded modes are only available in the 80-pin package version).

- Normal and emulation operating modes:
  - Normal single-chip mode
  - Normal expanded wide mode
  - Normal expanded narrow mode
  - Emulation expanded wide mode
  - Emulation expanded narrow mode
- Special operating modes:
  - Special single-chip mode with active background debug mode
  - Special test mode **(Freescale use only)**
  - Special peripheral mode **(Freescale use only)**
- Low power modes:
  - Stop mode
  - Pseudo stop mode
  - Wait mode



The figure shows a useful map, which is not the map out of reset. After reset the map is:

0x0000–0x03FF: Register space

0x0000–0x0FFF: 4K RAM (only 3K visible 0x0400–0x0FFF)

Flash erase sector size is 1024 Bytes

Figure 1-4. MC9S12C64 and MC9S12GC64 User Configurable Memory Map

0x0010–0x0014 MMC Map 1 of 4 (HCS12 Module Mapping Control)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	INITRM	Read:	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL
		Write:								
0x0011	INITRG	Read:	0	REG14	REG13	REG12	REG11	0	0	0
		Write:								
0x0012	INITEE	Read:	EE15	EE14	EE13	EE12	EE11	0	0	EEON
		Write:								
0x0013	MISC	Read:	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON
		Write:								
0x0014	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

0x0015–0x0016 INT Map 1 of 2 (HCS12 Interrupt)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0015	ITCR	Read:	0	0	0	WRINT	ADR3	ADR2	ADR1	ADR0
		Write:								
0x0016	ITEST	Read:	INTE	INTC	INTA	INT8	INT6	INT4	INT2	INT0
		Write:								

0x0017–0x0017 MMC Map 2 of 4 (HCS12 Module Mapping Control)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0017	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

0x0018–0x0018 Miscellaneous Peripherals (Device User Guide)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0018	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

0x0019–0x0019 VREG3V3 (Voltage Regulator)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$0019	VREGCTRL	Read:	0	0	0	0	0	LVDS	LVIE	LVIF
		Write:								

4.3.2.8 Port E Assignment Register (PEAR)

Module Base + 0x000A
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	NOACCE	0	PIPOE	NECLK	LSTRE	RDWE	0	0
W								
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Special Test	0	0	1	0	1	1	0	0
Peripheral	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	1	0	1	0	1	1	0	0
Emulation Expanded Wide	1	0	1	0	1	1	0	0
Normal Single Chip	0	0	0	1	0	0	0	0
Normal Expanded Narrow	0	0	0	0	0	0	0	0
Normal Expanded Wide	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 4-12. Port E Assignment Register (PEAR)

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

Port E serves as general-purpose I/O or as system and bus control signals. The PEAR register is used to choose between the general-purpose I/O function and the alternate control functions. When an alternate control function is selected, the associated DDRE bits are overridden.

The reset condition of this register depends on the mode of operation because bus control signals are needed immediately after reset in some modes. In normal single-chip mode, no external bus control signals are needed so all of port E is configured for general-purpose I/O. In normal expanded modes, only the E clock is configured for its alternate bus control function and the other bits of port E are configured for general-purpose I/O. As the reset vector is located in external memory, the E clock is required for this access. R/$\overline{W}$ is only needed by the system when there are external writable resources. If the normal expanded system needs any other bus control signals, PEAR would need to be written before any access that needed the additional signals. In special test and emulation modes, IPIPE1, IPIPE0, E, $\overline{LSTRB}$, and R/$\overline{W}$ are configured out of reset as bus control signals.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.

mode. Background debugging should not be used while the MCU is in special peripheral mode as internal bus conflicts between BDM and the external master can cause improper operation of both functions.

4.4.4 Internal Visibility

Internal visibility is available when the MCU is operating in expanded wide modes or emulation narrow mode. It is not available in single-chip, peripheral or normal expanded narrow modes. Internal visibility is enabled by setting the IVIS bit in the MODE register.

If an internal access is made while E, R/$\overline{W}$, and $\overline{LSTRB}$ are configured as bus control outputs and internal visibility is off (IVIS=0), E will remain low for the cycle, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

When internal visibility is enabled (IVIS=1), certain internal cycles will be blocked from going external. During cycles when the BDM is selected, R/$\overline{W}$ will remain high, data will maintain its previous state, and address and $\overline{LSTRB}$ pins will be updated with the internal value. During CPU no access cycles when the BDM is not driving, R/$\overline{W}$ will remain high, and address, data and the $\overline{LSTRB}$ pins will remain at their previous state.

NOTE

When the system is operating in a secure mode, internal visibility is not available (i.e., IVIS = 1 has no effect). Also, the IPIPE signals will not be visible, regardless of operating mode. IPIPE1–IPIPE0 will display 0es if they are enabled. In addition, the MOD bits in the MODE control register cannot be written.

4.4.5 Low-Power Options

The MEBI does not contain any user-controlled options for reducing power consumption. The operation of the MEBI in low-power modes is discussed in the following subsections.

4.4.5.1 Operation in Run Mode

The MEBI does not contain any options for reducing power in run mode; however, the external addresses are conditioned to reduce power in single-chip modes. Expanded bus modes will increase power consumption.

4.4.5.2 Operation in Wait Mode

The MEBI does not contain any options for reducing power in wait mode.

4.4.5.3 Operation in Stop Mode

The MEBI will cease to function after execution of a CPU STOP instruction.

6.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands, namely, hardware commands and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 6.4.3, “BDM Hardware Commands.” Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 6.4.4, “Standard BDM Firmware Commands.” The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted, see Section 6.4.3, “BDM Hardware Commands.” Firmware commands can only be executed when the system is in active background debug mode (BDM).

6.4.1 Security

If the user resets into special single-chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip EEPROM and FLASH EEPROM are erased. This being the case, the UNSEC bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the EEPROM or FLASH do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the EEPROM and FLASH. After execution of the secure firmware, regardless of the results of the erase tests, the CPU registers, INITEE and PPAGE, will no longer be in their reset state.

6.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[1]:

- Hardware BACKGROUND command
- BDM external instruction tagging mechanism
- CPU BGND instruction
- Breakpoint sub-block’s force or tag mechanism[2]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by the breakpoint sub-

1. BDM is enabled and active immediately out of special single-chip reset.
2. This method is only available on systems that have a a breakpoint or a debug sub-block.

DBGC2 being logic 1 or logic 0, respectively. BDM requests have a higher priority than SWI requests. No data breakpoints are allowed in this mode.

TAGAB in DBGC2 selects whether the breakpoint mode is forced or tagged. The BKxMBH:L bits in DBGC3 select whether or not the breakpoint is matched exactly or is a range breakpoint. They also select whether the address is matched on the high byte, low byte, both bytes, and/or memory expansion. The RWx and RWxEN bits in DBGC3 select whether the type of bus cycle to match is a read, write, or read/write when performing forced breakpoints.

7.4.1.2 Full Breakpoint Mode

Full breakpoint mode requires a match on address and data for a breakpoint to occur. Upon a successful match, the system will enter background debug mode or initiate a software interrupt based upon the state of BDM in DBGC2 being logic 1 or logic 0, respectively. BDM requests have a higher priority than SWI requests. R/W matches are also allowed in this mode.

TAGAB in DBGC2 selects whether the breakpoint mode is forced or tagged. When TAGAB is set in DBGC2, only addresses are compared and data is ignored. The BKAMBH:L bits in DBGC3 select whether or not the breakpoint is matched exactly, is a range breakpoint, or is in page space. The BKBMBH:L bits in DBGC3 select whether the data is matched on the high byte, low byte, or both bytes. RWA and RWAEN bits in DBGC2 select whether the type of bus cycle to match is a read or a write when performing forced breakpoints. RWB and RWBEN bits in DBGC2 are not used in full breakpoint mode.

NOTE

The full trigger mode is designed to be used for either a word access or a byte access, but not both at the same time. Confusing trigger operation (seemingly false triggers or no trigger) can occur if the trigger address occurs in the user program as both byte and word accesses.

7.4.1.3 Breakpoint Priority

Breakpoint operation is first determined by the state of the BDM module. If the BDM module is already active, meaning the CPU is executing out of BDM firmware, breakpoints are not allowed. In addition, while executing a BDM TRACE command, tagging into BDM is not allowed. If BDM is not active, the breakpoint will give priority to BDM requests over SWI requests. This condition applies to both forced and tagged breakpoints.

In all cases, BDM related breakpoints will have priority over those generated by the Breakpoint sub-block. This priority includes breakpoints enabled by the $\overline{\text{TAGLO}}$ and $\overline{\text{TAGHI}}$ external pins of the system that interface with the BDM directly and whose signal information passes through and is used by the breakpoint sub-block.

control (TBC) block. When PAGSEL = 01, registers DBGCAX, DBGCBX, and DBGCCX are used to match the upper addresses as shown in Table 7-11.

NOTE

If a tagged-type C breakpoint is set at the same address as an A/B tagged-type trigger (including the initial entry in an inside or outside range trigger), the C breakpoint will have priority and the trigger will not be recognized.

7.4.2.1.1 Read or Write Comparison

Read or write comparisons are useful only with TRGSEL = 0, because only opcodes should be tagged as they are “read” from memory. RWAEN and RWBEN are ignored when TRGSEL = 1.

In full modes (“A and B” and “A and not B”) RWAEN and RWA are used to select read or write comparisons for both comparators A and B. Table 7-24 shows the effect for RWAEN, RWA, and RW on the DBGCB comparison conditions. The RWBEN and RWB bits are not used and are ignored in full modes.

Table 7-24. Read or Write Comparison Logic Table

RWAEN bit	RWA bit	RW signal	Comment
0	x	0	Write data bus
0	x	1	Read data bus
1	0	0	Write data bus
1	0	1	No data bus compare since RW=1
1	1	0	No data bus compare since RW=0
1	1	1	Read data bus

7.4.2.1.2 Trigger Selection

The TRGSEL bit in DBGC1 is used to determine the triggering condition in DBG mode. TRGSEL applies to both trigger A and B except in the event only trigger modes. By setting TRGSEL, the comparators A and B will qualify a match with the output of opcode tracking logic and a trigger occurs before the tagged instruction executes (tagged-type trigger). With the TRGSEL bit cleared, a comparator match forces a trigger when the matching condition occurs (force-type trigger).

NOTE

If the TRGSEL is set, the address stored in the comparator match address registers must be an opcode address for the trigger to occur.

7.4.2.2 Trace Buffer Control (TBC)

The TBC is the main controller for the DBG module. Its function is to decide whether data should be stored in the trace buffer based on the trigger mode and the match signals from the comparator. The TBC also determines whether a request to break the CPU should occur.

8.3.2.12 Port Data Register (PORTAD)

The data port associated with the ATD is general purpose I/O. The port pins are shared with the analog A/D inputs AN7–AN0.

Module Base + 0x000F

	7	6	5	4	3	2	1	0
R	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
W								
Reset	1	1	1	1	1	1	1	1
Pin Function	AN7	AN6	AN5	AN4	AN3‘	AN2	AN1	AN0

= Unimplemented or Reserved

Figure 8-14. Port Data Register (PORTAD)

Read: Anytime

Write: Anytime, no effect

The A/D input channels may be used for general-purpose digital I/0.

Table 8-18. PORTAD Field Descriptions

Field	Description
7 PTAD[7:0]	**A/D Channel x (ANx) Digital Input (x = 7, 6, 5, 4, 3, 2, 1, 0)** — If the digital input buffer on the ANx pin is enabled (IENx = 1) read returns the logic level on ANx pin (signal potentials not meeting V_{IL} or V_{IH} specifications will have an indeterminate value)). If the digital input buffers are disabled (IENx = 0), read returns a “1”. Reset sets all PORTAD bits to “1”.

8.3.2.13 ATD Conversion Result Registers (ATDDRHx/ATDDRLx)

The A/D conversion results are stored in 8 read-only result registers ATDDRHx/ATDDRLx. The result data is formatted in the result registers based on two criteria. First there is left and right justification; this selection is made using the DJM control bit in ATDCTL5. Second there is signed and unsigned data; this selection is made using the DSGN control bit in ATDCTL5. Signed data is stored in 2’s complement format and only exists in left justified format. Signed data selected for right justified format is ignored.

Read: Anytime

Write: Anytime, no effect in normal modes

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.

9.3.2 Register Descriptions

This section describes in address order all the CRGV4 registers and their individual bits.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SYNR	R	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
	W								
0x0001 REFDV	R	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
	W								
0x0002 CTFLG	R	0	0	0	0	0	0	0	0
	W								
0x0003 CRGFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	TRACK	SCMIF	SCM
	W								
0x0004 CRGINT	R	RTIE	0	0	LOCKIE	0	0	SCMIE	0
	W								
0x0005 CLKSEL	R	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
	W								
0x0006 PLLCTL	R	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME
	W								
0x0007 RTICTL	R	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
	W								
0x0008 COPCTL	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
	W								
0x0009 FORBYP	R	0	0	0	0	0	0	0	0
	W								
0x000A CTCTL	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 9-3. CRG Register Summary

Table 9-11. Outcome of Clock Loss in Wait Mode

CME	SCME	SCMIE	CRG Actions
0	X	X	Clock failure --> No action, clock loss not detected.
1	0	X	Clock failure --> CRG performs Clock Monitor Reset immediately
1	1	0	Clock failure --> Scenario 1: OSCCLK **recovers** prior to exiting Wait Mode. – MCU remains in Wait Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag. *Some time later OSCCLK recovers.* – CM no longer indicates a failure, – 4096 OSCCLK cycles later Clock Quality Check indicates clock o.k., – SCM deactivated, – PLL disabled depending on PLLWAI, – VREG remains enabled *(never gets disabled in Wait Mode)*. – MCU remains in Wait Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit Wait Mode using OSCCLK as system clock (SYSCLK), – Continue normal operation. *or an External Reset is applied.* – Exit Wait Mode using OSCCLK as system clock, – Start reset sequence. Scenario 2: OSCCLK **does not recover** prior to exiting Wait Mode. – MCU remains in Wait Mode, – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – Set SCMIF interrupt flag, – Keep performing Clock Quality Checks (could continue infinitely) while in Wait Mode. *Some time later either a wakeup interrupt occurs (no SCM interrupt)* – Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform additional Clock Quality Checks until OSCCLK is o.k. again. or an External RESET is applied. – Exit Wait Mode in SCM using PLL clock (f_{SCM}) as system clock, – Start reset sequence, – Continue to perform additional Clock Quality Checks until OSCCLK is o.k.again.

The internal reset of the MCU remains asserted while the reset generator completes the 192 SYSCLK long reset sequence. The reset generator circuitry always makes sure the internal reset is deasserted synchronously after completion of the 192 SYSCLK cycles. In case the RESET pin is externally driven low for more than these 192 SYSCLK cycles (external reset), the internal reset remains asserted too.

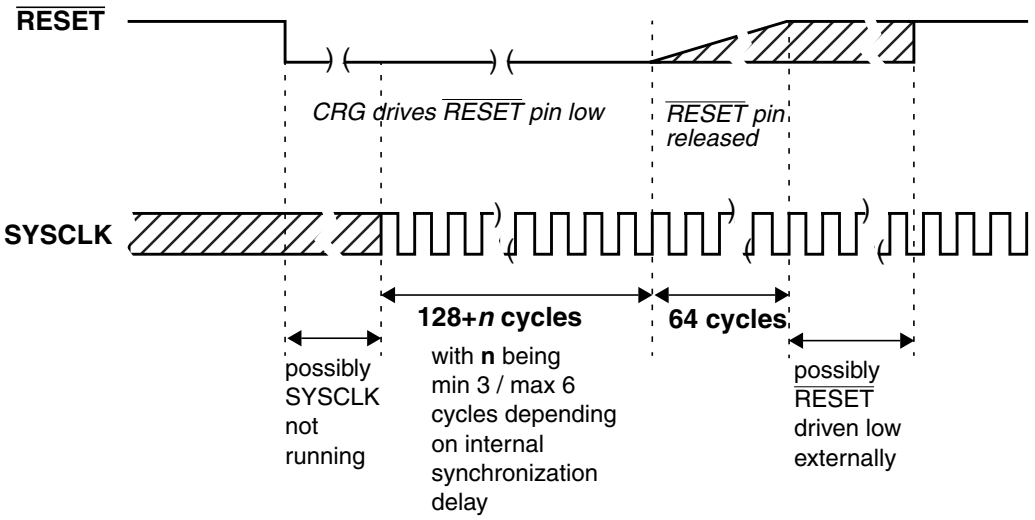


Figure 9-25. RESET Timing

9.5.1 Clock Monitor Reset

The CRGV4 generates a clock monitor reset in case all of the following conditions are true:

- Clock monitor is enabled (CME=1)
- Loss of clock is detected
- Self-clock mode is disabled (SCME=0)

The reset event asynchronously forces the configuration registers to their default settings (see Section 9.3, “Memory Map and Register Definition”). In detail the CME and the SCME are reset to logical ‘1’ (which doesn’t change the state of the CME bit, because it has already been set). As a consequence, the CRG immediately enters self-clock mode and starts its internal reset sequence. In parallel the clock quality check starts. As soon as clock quality check indicates a valid oscillator clock the CRG switches to OSCCLK and leaves self-clock mode. Because the clock quality checker is running in parallel to the reset generator, the CRG may leave self-clock mode while completing the internal reset sequence. When the reset sequence is finished the CRG checks the internally latched state of the clock monitor fail circuit. If a clock monitor fail is indicated processing begins by fetching the clock monitor reset vector.

9.5.2 Computer Operating Properly Watchdog (COP) Reset

When COP is enabled, the CRG expects sequential write of 0x0055 and 0x00AA (in this order) to the ARMCOP register during the selected time-out period. As soon as this is done, the COP time-out period restarts. If the program fails to do this the CRG will generate a reset. Also, if any value other than 0x0055 or 0x00AA is written, the CRG immediately generates a reset. In case windowed COP operation is enabled

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

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	0	0	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-7. PWM Center Align Enable Register (PWMCAE)

Read: anytime

Write: anytime

NOTE

Write these bits only when the corresponding channel is disabled.

Table 12-8. PWMCAE Field Descriptions

Field	Description
5 CAE5	**Center Aligned Output Mode on Channel 5** 0 Channel 5 operates in left aligned output mode. 1 Channel 5 operates in center aligned output mode.
4 CAE4	**Center Aligned Output Mode on Channel 4** 0 Channel 4 operates in left aligned output mode. 1 Channel 4 operates in center aligned output mode.
3 CAE3	**Center Aligned Output Mode on Channel 3** 1 Channel 3 operates in left aligned output mode. 1 Channel 3 operates in center aligned output mode.
2 CAE2	**Center Aligned Output Mode on Channel 2** 0 Channel 2 operates in left aligned output mode. 1 Channel 2 operates in center aligned output mode.
1 CAE1	**Center Aligned Output Mode on Channel 1** 0 Channel 1 operates in left aligned output mode. 1 Channel 1 operates in center aligned output mode.
0 CAE0	**Center Aligned Output Mode on Channel 0** 0 Channel 0 operates in left aligned output mode. 1 Channel 0 operates in center aligned output mode.

12.3.2.6 PWM Control Register (PWMCTL)

The PWMCTL register provides for various control of the PWM module.

12.3.2.11 Reserved Registers (PWMSCNTx)

The registers PWMSCNTA and PWMSCNTB are reserved for factory testing of the PWM module and are not available in normal modes.

Module Base + 0x000A

	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 12-13. Reserved Register (PWMSCNTA)

Module Base + 0x000B

	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 12-14. Reserved Register (PWMSCNTB)

Read: always read 0x0000 in normal modes

Write: unimplemented in normal modes

NOTE

Writing to these registers when in special modes can alter the PWM functionality.

writing another byte to the Transmitter buffer (SCIDRH/SCIDRL), while the shift register is still shifting out the first byte.

To initiate an SCI transmission:

1. Configure the SCI:
   a) Select a baud rate. Write this value to the SCI baud registers (SCIBDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the SCIBDH has no effect without also writing to SCIBDL.
   b) Write to SCICR1 to configure word length, parity, and other configuration bits (LOOPS,RSRC,M,WAKE,ILT,PE,PT).
   c) Enable the transmitter, interrupts, receive, and wake up as required, by writing to the SCICR2 register bits (TIE,TCIE,RIE,ILIE,TE,RE,RWU,SBK). A preamble or idle character will now be shifted out of the transmitter shift register.
2. Transmit Procedure for Each Byte:
   a. Poll the TDRE flag by reading the SCISR1 or responding to the TDRE interrupt. Keep in mind that the TDRE bit resets to one.
   d) If the TDRE flag is set, write the data to be transmitted to SCIDRH/L, where the ninth bit is written to the T8 bit in SCIDRH if the SCI is in 9-bit data format. A new transmission will not result until the TDRE flag has been cleared.
3. Repeat step 2 for each subsequent transmission.

NOTE

The TDRE flag is set when the shift register is loaded with the next data to be transmitted from SCIDRH/L, which happens, generally speaking, a little over half-way through the stop bit of the previous frame. Specifically, this transfer occurs 9/16ths of a bit time AFTER the start of the stop bit of the previous frame.

Writing the TE bit from 0 to a 1 automatically loads the transmit shift register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic transfers the data from the SCI data register into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

Hardware supports odd or even parity. When parity is enabled, the most significant bit (msb) of the data character is the parity bit.

The transmit data register empty flag, TDRE, in SCI status register 1 (SCISR1) becomes set when the SCI data register transfers a byte to the transmit shift register. The TDRE flag indicates that the SCI data register can accept new data from the internal data bus. If the transmit interrupt enable bit, TIE, in SCI control register 2 (SCICR2) is also set, the TDRE flag generates a transmitter interrupt request.

When the transmit shift register is not transmitting a frame, the **Tx output** signal goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

13.4.4.2 Character Reception

During an SCI reception, the receive shift register shifts a frame in from the **Rx input** signal. The SCI data register is the read-only buffer between the internal data bus and the receive shift register.

After a complete frame shifts into the receive shift register, the data portion of the frame transfers to the SCI data register. The receive data register full flag, RDRF, in SCI status register 1 (SCISR1) becomes set, indicating that the received byte can be read. If the receive interrupt enable bit, RIE, in SCI control register 2 (SCICR2) is also set, the RDRF flag generates an RDRF interrupt request.

13.4.4.3 Data Sampling

The receiver samples the **Rx input** signal at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock (see Figure 13-13) is re-synchronized:

- After every start bit
- After the receiver detects a data bit change from logic 1 to logic 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid logic 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid logic 0)

To locate the start bit, data recovery logic does an asynchronous search for a logic 0 preceded by three logic 1s.When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.

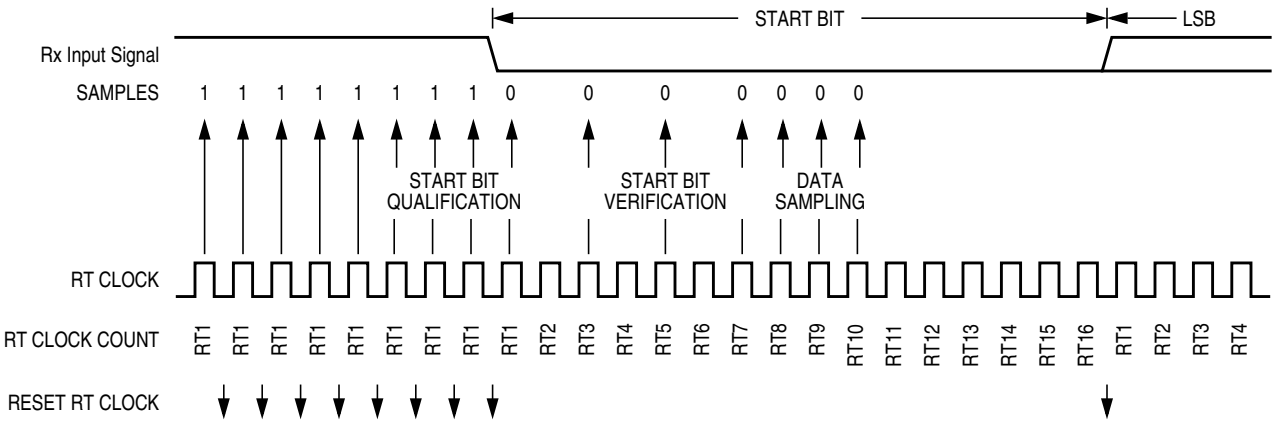


Figure 13-13. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 13-11 summarizes the results of the start bit verification samples.

Table 13-11. Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0

Table 15-5. OC7M Field Descriptions

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

15.3.2.4 Output Compare 7 Data Register (OC7D)

Module Base + 0x0003

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

Figure 15-9. Output Compare 7 Data Register (OC7D)

Read: Anytime

Write: Anytime

Table 15-6. OC7D Field Descriptions

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

15.3.2.5 Timer Count Register (TCNT)

Module Base + 0x0004

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

Figure 15-10. Timer Count Register High (TCNTH)

Module Base + 0x0005

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

Figure 15-11. Timer Count Register Low (TCNTL)

FPHDIS is cleared. The FPROT register is loaded from Flash address 0xFF0D during the reset sequence, indicated by F in Figure 20-10.

To change the Flash protection that will be loaded on reset, the upper sector of the Flash array must be unprotected, then the Flash protection byte located at Flash address 0xFF0D must be written to.

A protected Flash sector is disabled by FPHDIS and FPLDIS while the size of the protected sector is defined by FPHS[1:0] and FPLS[1:0] in the FPROT register.

Trying to alter any of the protected areas will result in a protect violation error and the PVIOL flag will be set in the FSTAT register (see Section 20.3.2.6). A mass erase of the whole Flash array is only possible when protection is fully disabled by setting the FPOPEN, FPLDIS, and FPHDIS bits. An attempt to mass erase a Flash array while protection is enabled will set the PVIOL flag in the FSTAT register.

Table 20-9. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Protection Function for Program or Erase** — It is possible using the FPOPEN bit to either select address ranges to be protected using FPHDIS, FPLDIS, FPHS[1:0] and FPLS[1:0] or to select the same ranges to be unprotected. When FPOPEN is set, FPxDIS enables the ranges to be protected, whereby clearing FPxDIS enables protection for the range specified by the corresponding FPxS[1:0] bits. When FPOPEN is cleared, FPxDIS defines unprotected ranges as specified by the corresponding FPxS[1:0] bits. In this case, setting FPxDIS enables protection. Thus the effective polarity of the FPxDIS bits is swapped by the FPOPEN bit as shown in Table 20-10. This function allows the main part of the Flash array to be protected while a small range can remain unprotected for EEPROM emulation. 0 The FPHDIS and FPLDIS bits define Flash address ranges to be unprotected 1 The FPHDIS and FPLDIS bits define Flash address ranges to be protected
6 NV6	**Nonvolatile Flag Bit** — The NV6 bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in the higher space of the Flash address map. 0 Protection/unprotection enabled 1 Protection/unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 20-11. The FPHS[1:0] bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected sector in the lower space of the Flash address map. 0 Protection/unprotection enabled 1 Protection/unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 20-12. The FPLS[1:0] bits can only be written to while the FPLDIS bit is set.