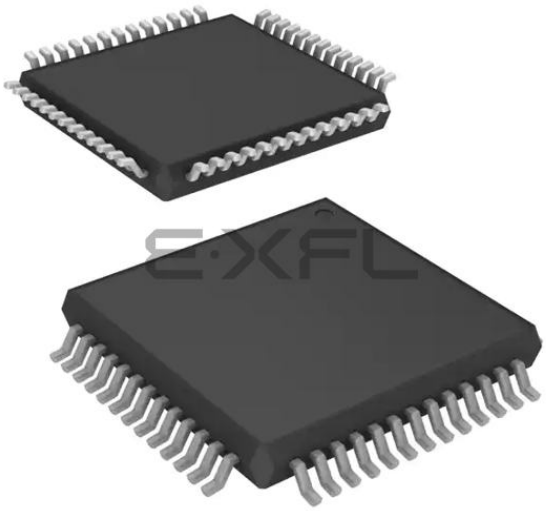


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

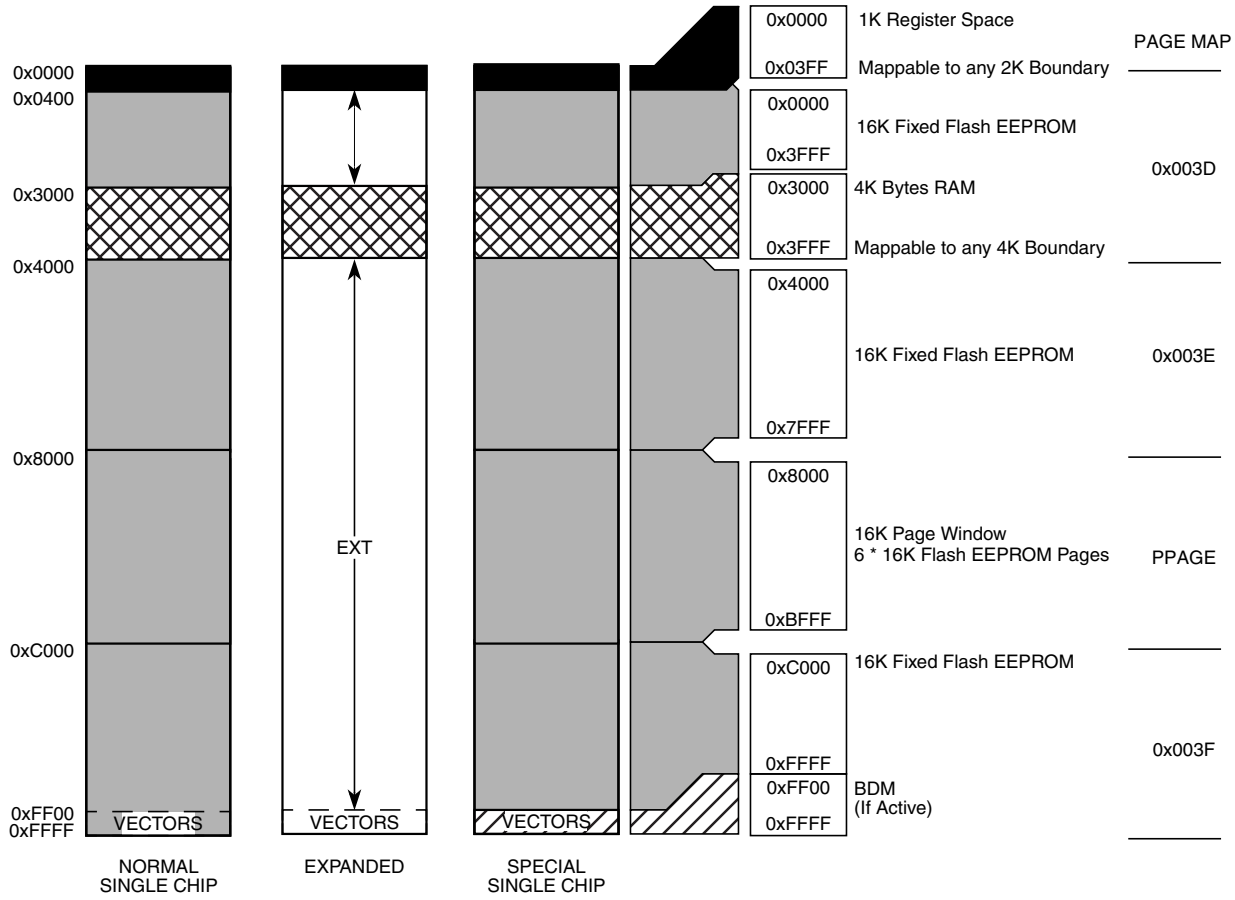
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	52-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12c128vpber

Chapter 14
Serial Peripheral Interface (SPIV3) Block Description

Chapter 15
Timer Module (TIM16B8CV1) Block Description



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 1-3. MC9S12C96 and MC9S12GC96 User Configurable Memory Map

0x0034–0x003F CRG (Clock and Reset Generator)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0034	SYNR	Read:	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
		Write:								
0x0035	REFDV	Read:	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
		Write:								
0x0036	CTFLG TEST ONLY	Read:	TOUT7	TOUT6	TOUT5	TOUT4	TOUT3	TOUT2	TOUT1	TOUT0
		Write:								
0x0037	CRGFLG	Read:	RTIF	PORF	LVRF	LOCKIF	LOCK	TRACK	SCMIF	SCM
		Write:								
0x0038	CRGINT	Read:	RTIE	0	0	LOCKIE	0	0	SCMIE	0
		Write:								
0x0039	CLKSEL	Read:	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
		Write:								
0x003A	PLLCTL	Read:	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME
		Write:								
0x003B	RTICTL	Read:	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		Write:								
0x003C	COPCTL	Read:	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		Write:								
0x003D	FORBYP TEST ONLY	Read:	RTIBYP	COPBYP	0	PLLBYP	0	0	FCM	0
		Write:								
0x003E	CTCTL TEST ONLY	Read:	TCTL7	TCTL6	TCTL5	TCTL4	TCLT3	TCTL2	TCTL1	TCTL0
		Write:								
0x003F	ARMCOP	Read:	0	0	0	0	0	0	0	0
		Write:	Bit 7	6	5	4	3	2	1	Bit 0

0x0040-0x006F TIM

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0040	TIOS	Read:	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
		Write:								
0x0041	CFORC	Read:	0	0	0	0	0	0	0	0
		Write:	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0042	OC7M	Read:	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
		Write:								
0x0043	OC7D	Read:	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
		Write:								
0x0044	TCNT (hi)	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
0x0045	TCNT (lo)	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
0x0046	TSCR1	Read:	TEN	TSWAI	TSFRZ	TFFCA	0	0	0	0
		Write:								
0x0047	TTOV	Read:	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
		Write:								

1.3.4.13 PE2 / R/$\overline{W}$ — Port E I/O Pin [2] / Read/Write

In all modes this pin can be used as a general-purpose I/O and is an input with an active pull-up out of reset. If the read/write function is required it should be enabled by setting the RDWE bit in the PEAR register. External writes will not be possible until enabled. This pin is not available in the 48- / 52-pin package versions.

1.3.4.14 PE1 / $\overline{IRQ}$ — Port E Input Pin [1] / Maskable Interrupt Pin

The $\overline{IRQ}$ input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge-sensitive triggering or level-sensitive triggering is program selectable (INTCR register). $\overline{IRQ}$ is always enabled and configured to level-sensitive triggering out of reset. It can be disabled by clearing IRQEN bit (INTCR register). When the MCU is reset the $\overline{IRQ}$ function is masked in the condition code register. This pin is always an input and can always be read. There is an active pull-up on this pin while in reset and immediately out of reset. The pull-up can be turned off by clearing PUPEE in the PUCR register.

1.3.4.15 PE0 / $\overline{XIRQ}$ — Port E input Pin [0] / Non Maskable Interrupt Pin

The $\overline{XIRQ}$ input provides a means of requesting a non-maskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. Because the $\overline{XIRQ}$ input is level sensitive, it can be connected to a multiple-source wired-OR network. This pin is always an input and can always be read. There is an active pull-up on this pin while in reset and immediately out of reset. The pull-up can be turned off by clearing PUPEE in the PUCR register.

1.3.4.16 PAD[7:0] / AN[7:0] — Port AD I/O Pins [7:0]

PAD7–PAD0 are general purpose I/O pins and also analog inputs for the analog to digital converter. In order to use a PAD pin as a standard input, the corresponding ATDDIEN register bit must be set. These bits are cleared out of reset to configure the PAD pins for A/D operation.

When the A/D converter is active in multi-channel mode, port inputs are scanned and converted irrespective of Port AD configuration. Thus Port AD pins that are configured as digital inputs or digital outputs are also converted in the A/D conversion sequence.

1.3.4.17 PP[7] / KWP[7] — Port P I/O Pin [7]

PP7 is a general purpose input or output pin, shared with the keypad interrupt function. When configured as an input, it can generate interrupts causing the MCU to exit stop or wait mode. This pin is not available in the 48- / 52-pin package versions.

1.3.4.18 PP[6] / KWP[6]/ROMCTL — Port P I/O Pin [6]

PP6 is a general purpose input or output pin, shared with the keypad interrupt function. When configured as an input, it can generate interrupts causing the MCU to exit stop or wait mode. This pin is not available in the 48- / 52-pin package versions. During MCU expanded modes of operation, this pin is used to enable

2.3.2.5.3 Port J Data Direction Register (DDRJ)

Module Base + 0x002A

	7	6	5	4	3	2	1	0
R	DDRJ7	DDRJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-34. Port J Data Direction Register (DDRJ)

Read: Anytime.

Write: Anytime.

Table 2-28. DDRJ Field Descriptions

Field	Description
7–6 DDRJ[7:6]	**Data Direction Port J** — This register configures port pins J[7:6] as either input or output. DDRJ[7:6] — Data Direction Port J 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTJ or PTIJ registers, when changing the DDRJ register.

2.3.2.5.4 Port J Reduced Drive Register (RDRJ)

Module Base + 0x002B

	7	6	5	4	3	2	1	0
R	RDRJ7	RDRJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-35. Port J Reduced Drive Register (RDRJ)

Read: Anytime.

Write: Anytime.

Table 2-29. RDRJ Field Descriptions

Field	Description
7–6 RDRJ[7:6]	**Reduced Drive Port J** — This register configures the drive strength of each port J output pin as either full or reduced. If the port is used as input this bit is ignored. 0 Full drive strength at output. 1 Associated pin drives at about 1/3 of the full drive strength.

2.3.2.6.5 Port AD Pull Device Enable Register (PERAD)

Module Base + 0x0034

	7	6	5	4	3	2	1	0
R W	PERAD7	PERAD6	PERAD5	PERAD4	PERAD3	PERAD2	PERAD1	PERAD0
Reset	0	0	0	0	0	0	0	0

Figure 2-44. Port AD Pull Device Enable Register (PERAD)

Read: Anytime.

Write: Anytime.

Table 2-36. PERAD Field Descriptions

Field	Description
7–0 PERAD[7:0]	**Pull Device Enable Port AD** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled. It is not possible to enable pull devices when a associated ATD channel is enabled simultaneously. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.6.6 Port AD Polarity Select Register (PPSAD)

Module Base + 0x0035

	7	6	5	4	3	2	1	0
R W	PPSAD7	PPSAD6	PPSAD5	PPSAD4	PPSAD3	PPSAD2	PPSAD1	PPSAD0
Reset	0	0	0	0	0	0	0	0

Figure 2-45. Port AD Polarity Select Register (PPSAD)

Read: Anytime.

Write: Anytime.

Table 2-37. PPSAD Field Descriptions

Field	Description
7–0 PPSAD[7:0]	**Pull Select Port AD** — This register selects whether a pull-down or a pull-up device is connected to the pin. 0 A pull-up device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input. 1 A pull-down device is connected to the associated port AD pin, if enabled by the associated bit in register PERAD and if the port is used as input.

4.3.2.3 Data Direction Register A (DDRA)

Module Base + 0x0002
Starting address location affected by INITRG register setting.

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

Figure 4-4. Data Direction Register A (DDRA)

Read: Anytime when register is in the map

Write: Anytime when register is in the map

This register controls the data direction for port A. When port A is operating as a general-purpose I/O port, DDRA determines the primary direction for each port A pin. A 1 causes the associated port pin to be an output and a 0 causes the associated pin to be a high-impedance input. The value in a DDR bit also affects the source of data for reads of the corresponding PORTA register. If the DDR bit is 0 (input) the buffered pin input state is read. If the DDR bit is 1 (output) the associated port data register bit state is read.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally. It is reset to 0x00 so the DDR does not override the three-state control signals.

Table 4-3. DDRA Field Descriptions

Field	Description
7:0 DDRA	**Data Direction Port A** 0 Configure the corresponding I/O pin as an input 1 Configure the corresponding I/O pin as an output

NOTE

To ensure that you read the value present on the PORTE pins, always wait at least one cycle after writing to the DDRE register before reading from the PORTE register.

4.3.2.7 Data Direction Register E (DDRE)

Module Base + 0x0009
Starting address location affected by INITRG register setting.

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

= Unimplemented or Reserved

Figure 4-11. Data Direction Register E (DDRE)

Read: Anytime when register is in the map

Write: Anytime when register is in the map

Data direction register E is associated with port E. For bits in port E that are configured as general-purpose I/O lines, DDRE determines the primary direction of each of these pins. A 1 causes the associated bit to be an output and a 0 causes the associated bit to be an input. Port E bit 1 (associated with $\overline{\text{IRQ}}$) and bit 0 (associated with $\overline{\text{XIRQ}}$) cannot be configured as outputs. Port E, bits 1 and 0, can be read regardless of whether the alternate interrupt function is enabled. The value in a DDR bit also affects the source of data for reads of the corresponding PORTE register. If the DDR bit is 0 (input) the buffered pin input state is read. If the DDR bit is 1 (output) the associated port data register bit state is read.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally. Also, it is not in the map in expanded modes while the EME control bit is set.

Table 4-5. DDRE Field Descriptions

Field	Description
7:2 DDRE	**Data Direction Port E** 0 Configure the corresponding I/O pin as an input 1 Configure the corresponding I/O pin as an output **Note:** It is unwise to write PORTE and DDRE as a word access. If you are changing port E pins from inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTE before enabling as outputs.

4.3.2.15 Port K Data Register (PORTK)

Module Base + 0x0032
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0
Alternate Pin Function	$\overline{ECS}$	$\overline{XCS}$	XAB19	XAB18	XAB17	XAB16	XAB15	XAB14

Figure 4-19. Port K Data Register (PORTK)

Read: Anytime

Write: Anytime

This port is associated with the internal memory expansion emulation pins. When the port is not enabled to emulate the internal memory expansion, the port pins are used as general-purpose I/O. When port K is operating as a general-purpose I/O port, DDRK determines the primary direction for each port K pin. A 1 causes the associated port pin to be an output and a 0 causes the associated pin to be a high-impedance input. The value in a DDR bit also affects the source of data for reads of the corresponding PORTK register. If the DDR bit is 0 (input) the buffered pin input is read. If the DDR bit is 1 (output) the output of the port data register is read.

This register is not in the map in peripheral or expanded modes while the EMK control bit in MODE register is set. Therefore, these accesses will be echoed externally.

When inputs, these pins can be selected to be high impedance or pulled up, based upon the state of the PUPKE bit in the PUCR register.

Table 4-13. PORTK Field Descriptions

Field	Description
7 Port K, Bit 7	**Port K, Bit 7** — This bit is used as an emulation chip select signal for the emulation of the internal memory expansion, or as general-purpose I/O, depending upon the state of the EMK bit in the MODE register. While this bit is used as a chip select, the external bit will return to its de-asserted state (V_{DD}) for approximately 1/4 cycle just after the negative edge of ECLK, unless the external access is stretched and ECLK is free-running (ESTR bit in EBICTL = 0). See the MMC block description chapter for additional details on when this signal will be active.
6 Port K, Bit 6	**Port K, Bit 6** — This bit is used as an external chip select signal for most external accesses that are not selected by $\overline{ECS}$ (see the MMC block description chapter for more details), depending upon the state the of the EMK bit in the MODE register. While this bit is used as a chip select, the external pin will return to its de-asserted state (V_{DD}) for approximately 1/4 cycle just after the negative edge of ECLK, unless the external access is stretched and ECLK is free-running (ESTR bit in EBICTL = 0).
5:0 Port K, Bits 5:0	**Port K, Bits 5:0** — These six bits are used to determine which FLASH/ROM or external memory array page is being accessed. They can be viewed as expanded addresses XAB19–XAB14 of the 20-bit address used to access up to1M byte internal FLASH/ROM or external memory array. Alternatively, these bits can be used for general-purpose I/O depending upon the state of the EMK bit in the MODE register.

7.3.2.4 Debug Count Register (DBGCNT)

Module Base + 0x0024
Starting address location affected by INITRG register setting.

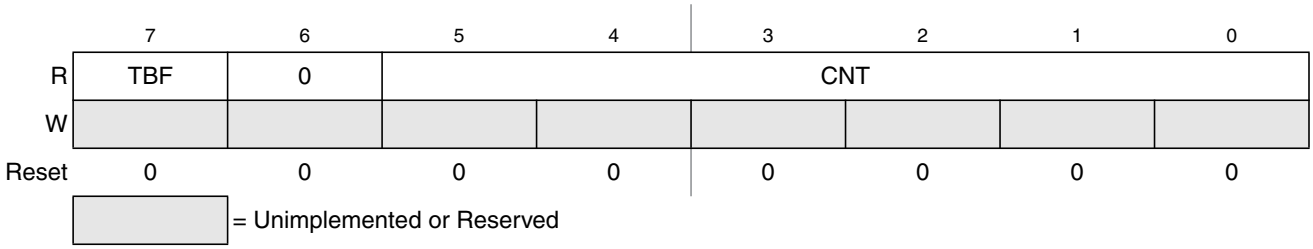


Figure 7-8. Debug Count Register (DBGCNT)

Table 7-8. DBGCNT 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 DBGC1 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 DBGCNT register is cleared when ARM in DBGC1 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 111110	2 words valid .. .. 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 111111	64 words valid, oldest data has been overwritten by most recent data

Table 9-12. Outcome of Clock Loss in Pseudo-Stop Mode (continued)

CME	SCME	SCMIE	CRG Actions
1	1	1	Clock failure --> – VREG enabled, – PLL enabled, – SCM activated, – Start Clock Quality Check, – SCMIF set. SCMIF generates Self-Clock Mode wakeup interrupt. – Exit Pseudo-Stop Mode in SCM using PLL clock (f_{SCM}) as system clock, – Continue to perform a additional Clock Quality Checks until OSCCLK is o.k. again.

9.4.10.2 Wake-up from Full Stop (PSTP=0)

The MCU requires an external interrupt or an external reset in order to wake-up from stop mode.

If the MCU gets an external reset during full stop mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and will perform a maximum of 50 clock *check_windows* (see Section 9.4.4, “Clock Quality Checker”). After completing the clock quality check the CRG starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Full stop mode is exited and the MCU is in run mode again.

If the MCU is woken-up by an interrupt, the CRG will also perform a maximum of 50 clock *check_windows* (see Section 9.4.4, “Clock Quality Checker”). If the clock quality check is successful, the CRG will release all system and core clocks and will continue with normal operation. If all clock checks within the timeout-window are failing, the CRG will switch to self-clock mode or generate a clock monitor reset (CMRESET) depending on the setting of the SCME bit.

Because the PLL has been powered-down during stop mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving stop mode. The software must manually set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

NOTE

In full stop mode, the clock monitor is disabled and any loss of clock will not be detected.

9.5 Resets

This section describes how to reset the CRGV4 and how the CRGV4 itself controls the reset of the MCU. It explains all special reset requirements. Because the reset generator for the MCU is part of the CRG, this section also describes all automatic actions that occur during or as a result of individual reset conditions. The reset values of registers and signals are provided in Section 9.3, “Memory Map and Register

10.3.2.3 MSCAN Bus Timing Register 0 (CANBTR0)

The CANBTR0 register configures various CAN bus timing parameters of the MSCAN module.

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R W	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
Reset:	0	0	0	0	0	0	0	0

Figure 10-6. MSCAN Bus Timing Register 0 (CANBTR0)

Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 10-3. CANBTR0 Register Field Descriptions

Field	Description
7:6 SJW[1:0]	**Synchronization Jump Width** — The synchronization jump width defines the maximum number of time quanta (Tq) clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the CAN bus (see Table 10-4).
5:0 BRP[5:0]	**Baud Rate Prescaler** — These bits determine the time quanta (Tq) clock which is used to build up the bit timing (see Table 10-5).

Table 10-4. Synchronization Jump Width

SJW1	SJW0	Synchronization Jump Width
0	0	1 Tq clock cycle
0	1	2 Tq clock cycles
1	0	3 Tq clock cycles
1	1	4 Tq clock cycles

Table 10-5. Baud Rate Prescaler

BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	Prescaler value (P)
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
1	1	1	1	1	1	64

With the misaligned character shown in Figure 13-20, the receiver counts 167 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

((167 – 160) / 167) X 100 = 4.19%

13.4.4.5.2 Fast Data Tolerance

Figure 13-21 shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

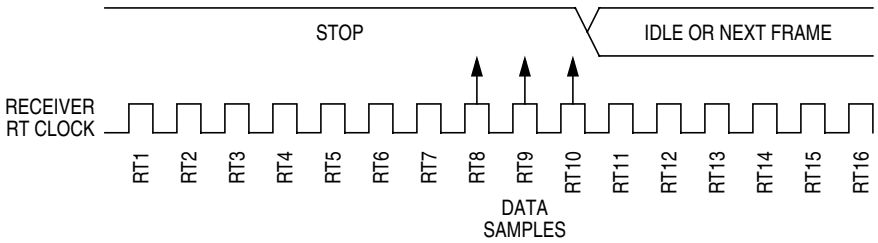


Figure 13-21. Fast Data

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles + 10 RTr cycles = 154 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 13-21, the receiver counts 154 RTr cycles at the point when the count of the transmitting device is 10 bit times x 16 RTt cycles = 160 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

((160 – 154) / 160) x 100 = 3.75%

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 10 RTr cycles = 170 RTr cycles to finish data sampling of the stop bit.

With the misaligned character shown in Figure 13-21, the receiver counts 170 RTr cycles at the point when the count of the transmitting device is 11 bit times x 16 RTt cycles = 176 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

((176 – 170) / 176) x 100 = 3.40%

13.4.4.6 Receiver Wakeup

To enable the SCI to ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCI control register 2 (SCICR2) puts the receiver into standby state during which receiver interrupts are disabled.The SCI will still load the receive data into the SCIDRH/L registers, but it will not set the RDRF flag.

13.4.5 Single-Wire Operation

Normally, the SCI uses two pins for transmitting and receiving. In single-wire operation, the RXD pin is disconnected from the SCI. The SCI uses the TXD pin for both receiving and transmitting.

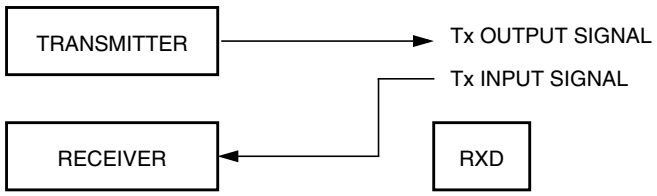


Figure 13-22. Single-Wire Operation (LOOPS = 1, RSRC = 1)

Enable single-wire operation by setting the LOOPS bit and the receiver source bit, RSRC, in SCI control register 1 (SCICR1). Setting the LOOPS bit disables the path from the **Rx Input** signal to the receiver. Setting the RSRC bit connects the receiver input to the output of the TXD pin driver. Both the transmitter and receiver must be enabled (TE = 1 and RE = 1).The TXDIR bit (SCISR2[1]) determines whether the TXD pin is going to be used as an input (TXDIR = 0) or an output (TXDIR = 1) in this mode of operation.

13.4.6 Loop Operation

In loop operation the transmitter output goes to the receiver input. The **Rx Input** signal is disconnected from the SCI

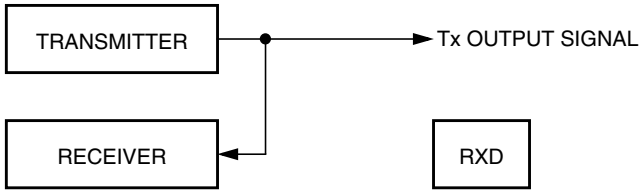


Figure 13-23. Loop Operation (LOOPS = 1, RSRC = 0)

Enable loop operation by setting the LOOPS bit and clearing the RSRC bit in SCI control register 1 (SCICR1). Setting the LOOPS bit disables the path from the **Rx Input** signal to the receiver. Clearing the RSRC bit connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (TE = 1 and RE = 1).

13.5 Initialization Information

13.5.1 Reset Initialization

The reset state of each individual bit is listed in Section 13.3, “Memory Map and Registers” which details the registers and their bit fields. All special functions or modes which are initialized during or just following reset are described within this section.

14.4.7 Operation in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

14.4.8 Operation in Wait Mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI Control Register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of wait mode (i.e. If the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e. a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. A SPIF flag and SPIDR copy is only generated if wait mode is entered or exited during a tranmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

14.4.9 Operation in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

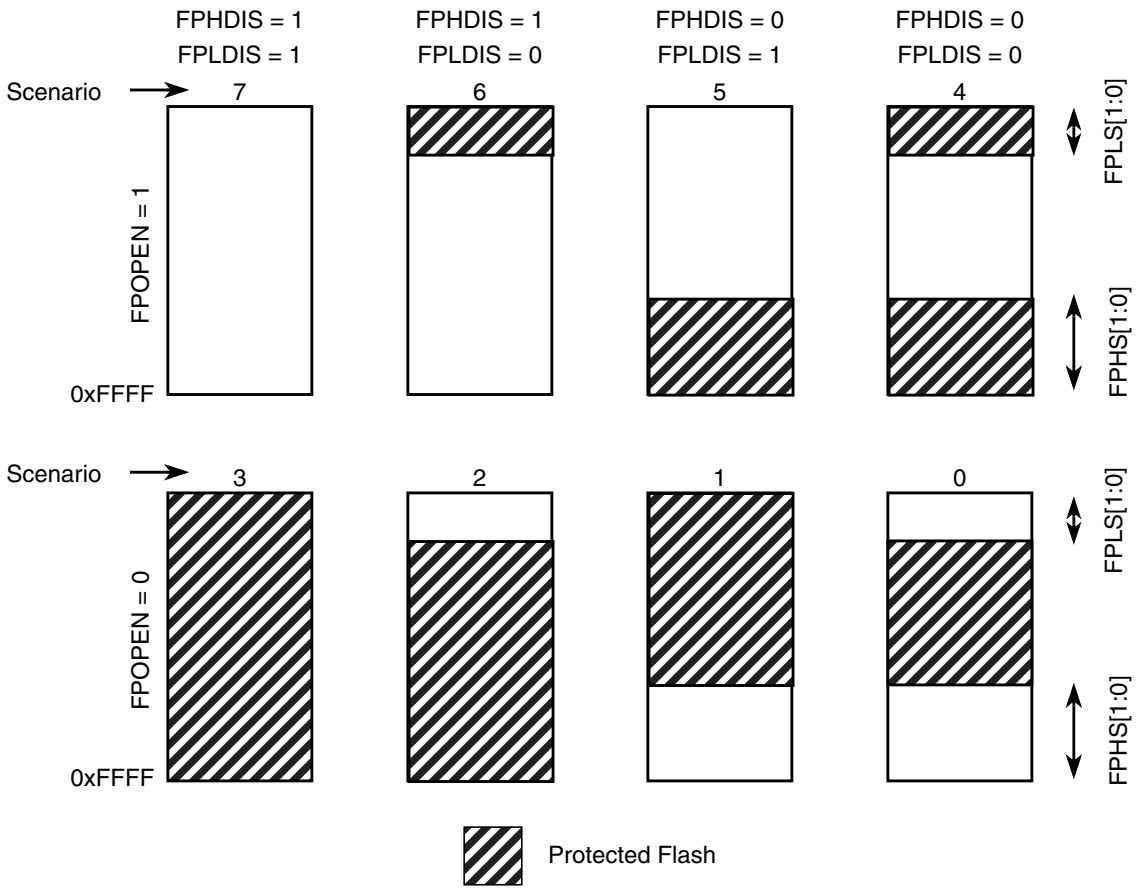


Figure 19-11. Flash Protection Scenarios

19.3.2.5.1 Flash Protection Restrictions

The general guideline is that protection can only be added, not removed. All valid transitions between Flash protection scenarios are specified in Table 19-13. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario.

Table 19-13. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		

addresses sequentially staring with 0xFF00-0xFF01 and ending with 0xFF06–0xFF07. The values 0x0000 and 0xFFFF are not permitted as keys. When the KEYACC bit is set, reads of the Flash array will return invalid data.

The user code stored in the Flash array must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If KEYEN[1:0] = 1:0 in the FSEC register, the MCU can be unsecured by the backdoor key access sequence described below:

1. Set the KEYACC bit in the FCNFG register
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00
3. Clear the KEYACC bit in the FCNFG register
4. If all four 16-bit words match the backdoor key stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and bits SEC[1:0] in the FSEC register are forced to the unsecure state of 1:0

The backdoor key access sequence is monitored by the internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following illegal operations will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor key programmed in the Flash array
2. If the four 16-bit words are written in the wrong sequence
3. If more than four 16-bit words are written
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF
5. If the KEYACC bit does not remain set while the four 16-bit words are written

After the backdoor key access sequence has been correctly matched, the MCU will be unsecured. The Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the four word backdoor key by programming bytes 0xFF00–0xFF07 of the Flash configuration field.

The security as defined in the Flash security/options byte at address 0xFF0F is not changed by using the backdoor key access sequence to unsecure. The backdoor key stored in addresses 0xFF00–0xFF07 is unaffected by the backdoor key access sequence. After the next reset sequence, the security state of the Flash module is determined by the Flash security/options byte at address 0xFF0F. The backdoor key access sequence has no effect on the program and erase protection defined in the FPROT register.

It is not possible to unsecure the MCU in special single chip mode by executing the backdoor key access sequence in background debug mode.

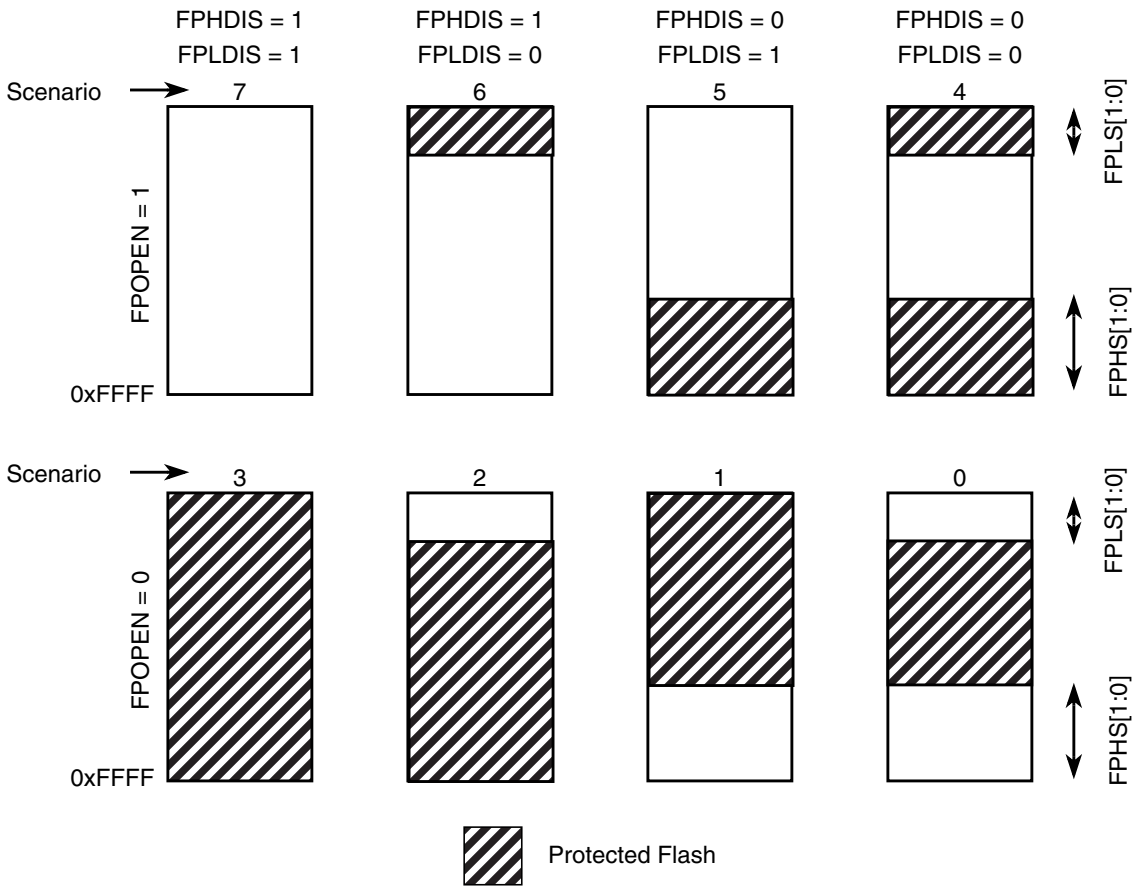


Figure 20-11. Flash Protection Scenarios

20.3.2.5.1 Flash Protection Restrictions

The general guideline is that protection can only be added, not removed. All valid transitions between Flash protection scenarios are specified in Table 20-13. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario.

Table 20-13. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		

21.4.1.3.2 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 21-23. 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.

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 $f_{CMFA.}$