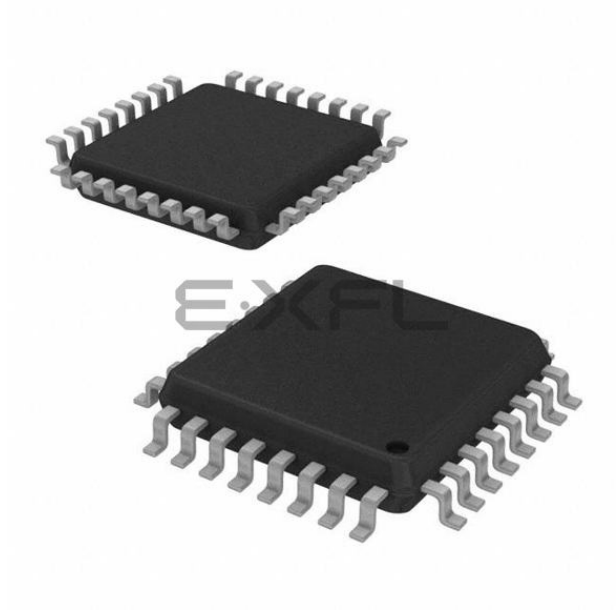

Welcome to E-XFL.COM

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	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12vr64af0mlc

1.6 Family Memory Map

Table 1-2 shows the MC9S12VR-Family register memory map.

Table 1-2. Device Register Memory Map

Address	Module	Size (Bytes)
0x0000–0x0009	PIM (port integration module)	10
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	Reserved	2
0x0010–0x0017	MMC (memory map control)	8
0x0018–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001F	PIM (port integration module)	4
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (clock and power management)	12
0x0040–0x006F	TIM (timer module <= 4channels)	48
0x0070–0x009F	ADC (analog to digital converter <= 6 channels)	48
0x00A0–0x00C7	PWM (pulse-width modulator <= 2channels)	40
0x00C8–0x00CF	SCI0 (serial communication interface)	8
0x00D0–0x00D7	SCI1 (serial communication interface)	8
0x00D8–0x00DF	SPI (serial peripheral interface)	8
0x00E0–0x00FF	Reserved	32
0x0100–0x0113	FTMRG control registers	20
0x0114–0x011F	Reserved	12
0x0120	INT (interrupt module)	1
0x0121–0x013F	Reserved	31
0x0140-0x0147	HSDRV (high-side driver)	8
0x0148-0x014F	Reserved	8
0x0150-0x0157	LSDRV (low-side driver)	8
0x0158-0x015F	Reserved	8
0x0160-0x0167	LINPHY (LIN physical layer)	8
0x0168-0x016F	Reserved	8
0x0170-0x0177	BATS (Supply Voltage Sense)	8
0x0178–0x023F	Reserved	200
0x0240–0x027F	PIM (port integration module)	64

1.7.2.8 PT[3:0] — Port T I/O Signals

PT[3:0] are general-purpose input or output signals. They can have a pull-up or pull-down device selected and enabled on per signal basis. Out of reset the pull devices are disabled.

1.7.2.9 PL[3:0] / KWL[3:0] — Port L Input Signals

PL[3:0] are high voltage input ports. The signals can be configured on per signal basis as interrupt inputs with wake-up capability (KWL[3:0]).

1.7.2.10 LIN — LIN Physical Layer Signal

This pad is connected to the single-wire LIN data bus.

1.7.2.11 HS[1:0] — High-Side Drivers Output Signals

Outputs of the two high-side drivers intended to drive incandescent bulbs or LEDs.

1.7.2.12 LS[1:0] — Low-Side Drivers Output Signals

Outputs of the two low-side drivers intended to drive inductive loads (relays).

1.7.2.13 VSENSE — Voltage Sensor Input

This pin can be connected to the supply (Battery) line for voltage measurements. The voltage present at this input is scaled down by an internal voltage divider, and can be routed to the internal ADC via an analog multiplexer. The pin itself is protected against reverse battery connections. To protect the pin from external fast transients an external resistor is needed.

1.7.2.14 AN[5:0] — ADC Input Signals

AN[5:0] are the analog inputs of the Analog-to-Digital Converter.

1.7.2.15 SPI Signals

1.7.2.15.1 $\overline{SS}$ Signal

This signal is associated with the slave select SS functionality of the serial peripheral interface SPI.

1.7.2.15.2 SCK Signal

This signal is associated with the serial clock SCK functionality of the serial peripheral interface SPI.

1.7.2.15.3 MISO Signal

This signal is associated with the MISO functionality of the serial peripheral interface SPI. This signal acts as master input during master mode or as slave output during slave mode.

1.11.5 Flash IFR Mapping

Table 1-13. Flash IFR Mapping

IFR Byte Address	Target															
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0x405A - 0x405B							ADC Bandgap Reference[1]									
0x40B8 - 0x40B9			ACLKTR[5:0][2]										HTTR[3:0][3]			
0x40BA - 0x40BB		TCTRIM[4:0][4]					IRCTRIM[9:0][4]									

1 see Section 1.12.1 ADC Calibration

2 see Section 4.3.2.16 Autonomous Clock Trimming Register (CPMUACLKTR)

3 see Section 4.3.2.19 High Temperature Trimming Register (CPMUHTTR)

4 see Section 4.3.2.20 S12CPMU_UHV_V8 IRC1M Trim Registers (CPMUIRCTRIMH / CPMUIRCTRIML)

1.11.6 ADC External Trigger Input Connection

The ADC module includes external trigger inputs ETRIG0, ETRIG1, ETRIG2, and ETRIG3. The external trigger allows the user to synchronize ADC conversion to external trigger events. On MC9S12VR64/48 ETRIG0 is connected to PP0 / PWM0 and ETRIG1 is connected to PP1 / PWM1. ETRIG2 and ETRIG3 are not used. ETRIG0 can be routed to PS2 and ETRIG1 can be routed to PS3. On MC9S12VR32/16 ETRIG0 is connected to PS2 and ETRIG1 is connected to PS3.

1.11.7 ADC Special Conversion Channels

Whenever the ADC's Special Channel Conversion Bit (SC) in 8.3.2.6 ATD Control Register 5 (ATDCTL5) is set, it is capable of running conversion on a number of internal channels. Table 1-14 lists the internal sources which are connected to these special conversion channels.

Global Address	Register Name[1]		Bit 7	6	5	4	3	2	1	Bit 0
0x025A	DDRP	R	0	0	**DDRP5**	**DDRP4**	**DDRP3**	DDRP2	DDRP1	**DDRP0**
		W								
0x025B	RDRP	R	0	0	0	0	0	RDRP2	RDRP1	**RDRP0**
		W								
0x025C	PERP	R	0	0	**PERP5**	**PERP4**	**PERP3**	PERP2	PERP1	**PERP0**
		W								
0x025D	PPSP	R	0	0	**PPSP5**	**PPSP4**	**PPSP3**	PPSP2	PPSP1	**PPSP0**
		W								
0x025E	PIEP	R	OCIE	0	**PIEP5**	**PIEP4**	**PIEP3**	PIEP2	PIEP1	**PIEP0**
		W								
0x025F	PIFP	R	OCIF	0	**PIFP5**	**PIFP4**	**PIFP3**	PIFP2	PIFP1	**PIFP0**
		W								
0x0260–0x0268	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0269	PTIL	R	0	0	0	0	PTIL3	PTIL2	PTIL1	PTIL0
		W								
0x026A	DIENL	R	0	0	0	0	DIENL3	DIENL2	DIENL1	DIENL0
		W								
0x026B	PTAL	R	PTTEL	PTPSL	PTABYPL	PTADIRL	PTAENL	0	PTAL1	PTAL0
		W								
0x026C	PIRL	R	0	0	0	0	PIRL3	PIRL2	PIRL1	PIRL0
		W								
0x026D	PPSL	R	0	0	0	0	PPSL3	PPSL2	PPSL1	PPSL0
		W								
0x026E	PIEL	R	0	0	0	0	PIEL3	PIEL2	PIEL1	PIEL0
		W								
0x026F	PIFL	R	0	0	0	0	PIFL3	PIFL2	PIFL1	PIFL0
		W								
0x0270	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0271	PT1AD	R	0	0	**PT1AD5**	**PT1AD4**	**PT1AD3**	**PT1AD2**	PT1AD1	PT1AD0
		W								

The configuration bit PPS is used for two purposes:

1. Configure the sensitive interrupt edge (rising or falling), if interrupt is enabled.
2. Select either a pullup or pulldown device if PER is active.

Table 2-3. Pin Configuration Summary[1]

DDR	PORT PT	PER	PPS[1]	PIE[2]	Function	Pull Device	Interrupt
0	x	0	x	0	Input	Disabled	Disabled
0	x	1	0	0	Input	Pullup	Disabled
0	x	1	1	0	Input	Pulldown	Disabled
0	x	0	0	1	Input	Disabled	Falling edge
0	x	0	1	1	Input	Disabled	Rising edge
0	x	1	0	1	Input	Pullup	Falling edge
0	x	1	1	1	Input	Pulldown	Rising edge
1	0	x	x	0	Output, drive to 0	Disabled	Disabled
1	1	x	x	0	Output, drive to 1	Disabled	Disabled
1	0	x	0	1	Output, drive to 0	Disabled	Falling edge
1	1	x	1	1	Output, drive to 1	Disabled	Rising edge

1 Always “0” on Port E

2 Applicable only on Port P and AD

NOTE

- All register bits in this module are completely synchronous to internal clocks during a register read.
- Figure of port data registers also display the alternative functions if applicable on the related pin as defined in Table 2-2. Names in parentheses denote the availability of the function when using a specific routing option.
- Figures of module routing registers also display the module instance or module channel associated with the related routing bit.

1. Not applicable for Port L. Refer to register descriptions.

Table 2-9. IRQCR Register Field Descriptions

Field	Description
7 IRQE	**IRQ select Edge sensitive only** — 1 $\overline{\text{IRQ}}$ pin configured to respond only to falling edges. Falling edges on the $\overline{\text{IRQ}}$ pin will be detected anytime IRQE=1 and will be cleared only upon a reset or the servicing of the $\overline{\text{IRQ}}$ interrupt. 0 $\overline{\text{IRQ}}$ pin configured for low level recognition
6 IRQEN	**IRQ ENable** — 1 $\overline{\text{IRQ}}$ pin is connected to interrupt logic 0 $\overline{\text{IRQ}}$ pin is disconnected from interrupt logic

2.3.9 Reserved Register

Address 0x001E (S12VR32/16) Access: User read/write[1]

	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

Figure 2-7. Reserved Register

[1] Read: Anytime
Write:Never

2.3.10 Reserved Register

Address 0x001F Access: User read/write[1]

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

Figure 2-8. Reserved Register

[1] Read: Anytime
Write: Only in special mode

NOTE

These reserved registers are designed for factory test purposes only and are not intended for general user access. Writing to these registers when in special modes can alter the module's functionality.

2.3.21 Port S Data Direction Register (DDRS)

Address 0x024A (S12VR64/48) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-22. Port S Data Direction Register (DDRS - S12VR64/48)

1 Read: Anytime
Write: Anytime

Address 0x024A (S12VR32/16) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	DDRS3	DDRS2	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-23. Port S Data Direction Register (DDRS - S12VR32/16)

1 Read: Anytime
Write: Anytime

Table 2-19. DDRS Register Field Descriptions

Field	Description
5 DDRS	**Data Direction Register port S —** This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SPI the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input
4 DDRS	**Data Direction Register port S —** This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SPI the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin is configured as output 0 Associated pin is configured as input
3 DDRS	**Data Direction Register port S —** This bit determines whether the associated pin is an input or output. The ECLK output function, routed SCI1 and routed PWM function forces the I/O state to output if enabled. Depending on the configuration of the enabled SPI the I/O state will be forced to be input or output. In these cases the data direction bit will not change. The routed ETRIG function has no effect on the I/O state. 1 Associated pin is configured as output 0 Associated pin is configured as input

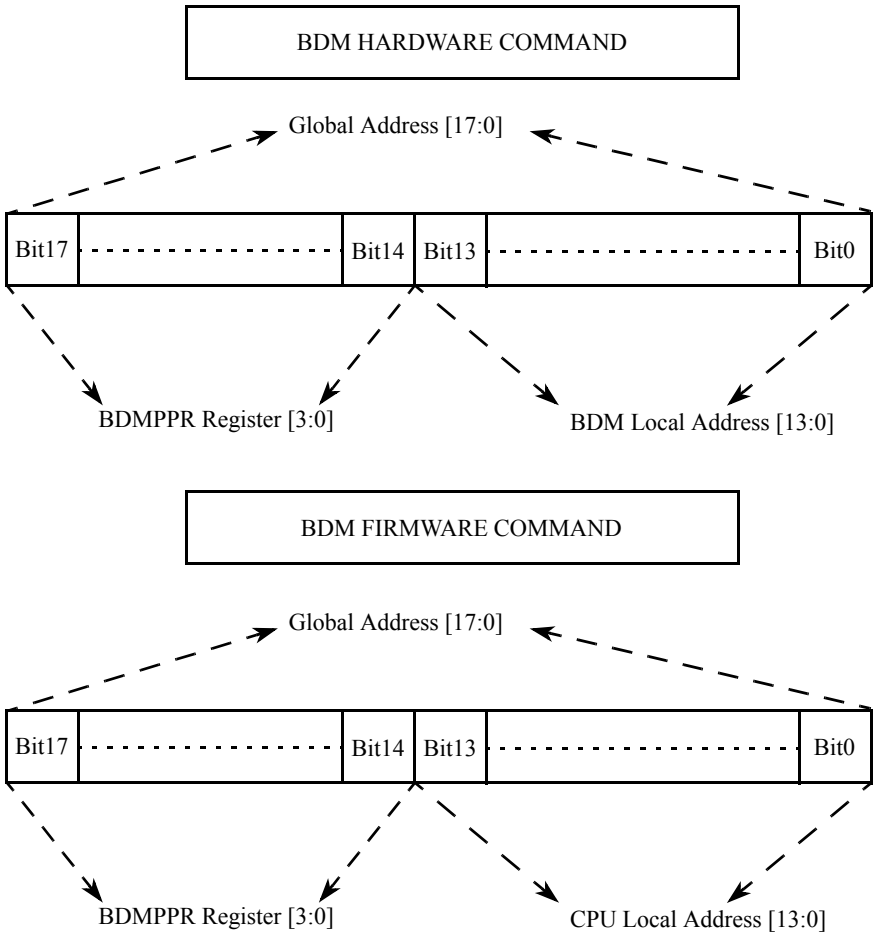
BDM HARDWARE COMMAND
Global Address [17:0]
Bit17
Bit14
Bit13
Bit0
BDMPPR Register [3:0]
BDM Local Address [13:0]
BDM FIRMWARE COMMAND
Global Address [17:0]
Bit17
Bit14
Bit13
Bit0
BDMPPR Register [3:0]
CPU Local Address [13:0]

Figure 3-10.

4.4.5 External Oscillator

4.4.5.1 Enabling the External Oscillator

An example of how to use the oscillator as source of the Bus Clock is shown in Figure 4-38.

Figure 4-38. Enabling the external oscillator

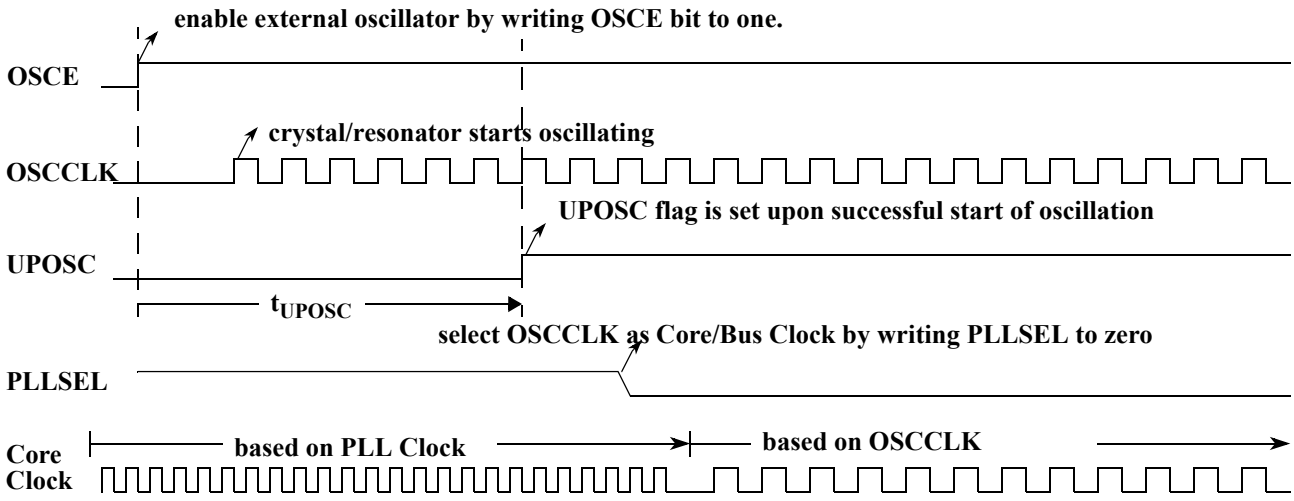

Table 5-6. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT[2]	62	16-bit data out	Increment X index register by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT[2]	42	16-bit data in	Increment X index register by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[3]	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

1 If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

2 When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

3 System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.4.7, “Serial Interface Hardware Handshake Protocol” last note).

5.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word, depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, only one byte of which contains valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

Table 7-3. IVBR Field Descriptions

Field	Description
7–0 IVB_ADDR[7:0]	**Interrupt Vector Base Address Bits** — These bits represent the upper byte of all vector addresses. Out of reset these bits are set to 0xFF (that means vectors are located at 0xFF80–0xFFFE) to ensure compatibility to HCS12. **Note:** A system reset will initialize the interrupt vector base register with "0xFF" before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the three reset vectors (0xFFFA–0xFFFE). **Note:** If the BDM is active (that means the CPU is in the process of executing BDM firmware code), the contents of IVBR are ignored and the upper byte of the vector address is fixed as "0xFF". This is done to enable handling of all non-maskable interrupts in the BDM firmware.

7.4 Functional Description

The INT module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

7.4.1 S12S Exception Requests

The CPU handles both reset requests and interrupt requests. A priority decoder is used to evaluate the priority of pending interrupt requests.

7.4.2 Interrupt Prioritization

The INT module contains a priority decoder to determine the priority for all interrupt requests pending for the CPU. If more than one interrupt request is pending, the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The I bit in the condition code register (CCR) of the CPU must be cleared.
3. There is no SWI, TRAP, or X bit maskable request pending.

NOTE

All non I bit maskable interrupt requests always have higher priority than the I bit maskable interrupt requests. If the X bit in the CCR is cleared, it is possible to interrupt an I bit maskable interrupt by an X bit maskable interrupt. It is possible to nest non maskable interrupt requests, for example by nesting SWI or TRAP calls.

Since an interrupt vector is only supplied at the time when the CPU requests it, it is possible that a higher priority interrupt request could override the original interrupt request that caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this interrupt request first, before the original interrupt request is processed.

If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works following the same rules like any interrupt request, that is care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

1. The capability of the $\overline{\text{XIRQ}}$ pin to wake-up the MCU with the X bit set may not be available if, for example, the $\overline{\text{XIRQ}}$ pin is shared with other peripheral modules on the device. Please refer to the Device section of the MCU reference manual for details.

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.

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

Table 9-13 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 9-13. 16-bit Concatenation Mode Summary

Note: Bits related to available channels have functional significance.

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON67	PWME7	PPOL7	PCLK7	CAE7	PWM7
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

9.4.2.8 PWM Boundary Cases

Table 9-14 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation).

Table 9-14. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always low
$00 (indicates no duty)	>$00	0	Always high
XX	$00[1] (indicates no period)	1	Always high
XX	$00[1] (indicates no period)	0	Always low
>= PWMPERx	XX	1	Always high
>= PWMPERx	XX	0	Always low

[1] Counter = $00 and does not count.

9.5 Resets

The reset state of each individual bit is listed within the Section 9.3.2, “Register Descriptions” 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.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters do not count.

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.

10.4.6.3 Data Sampling

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 10-21) is re-synchronized immediatelly at bus clock edge:

- 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.

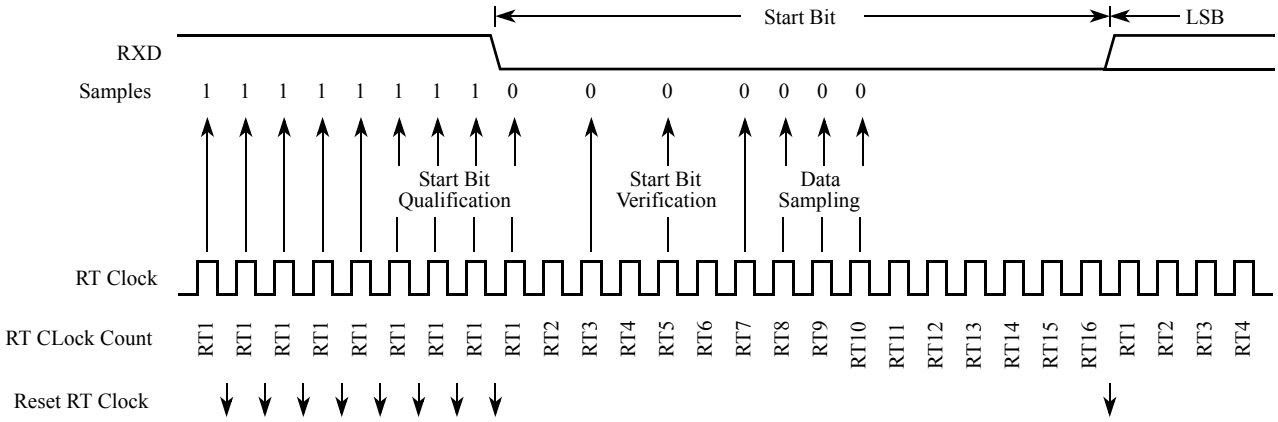

Figure 10-21. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Figure 10-17 summarizes the results of the start bit verification samples.

Table 10-17. 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
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

In Figure 10-22 the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

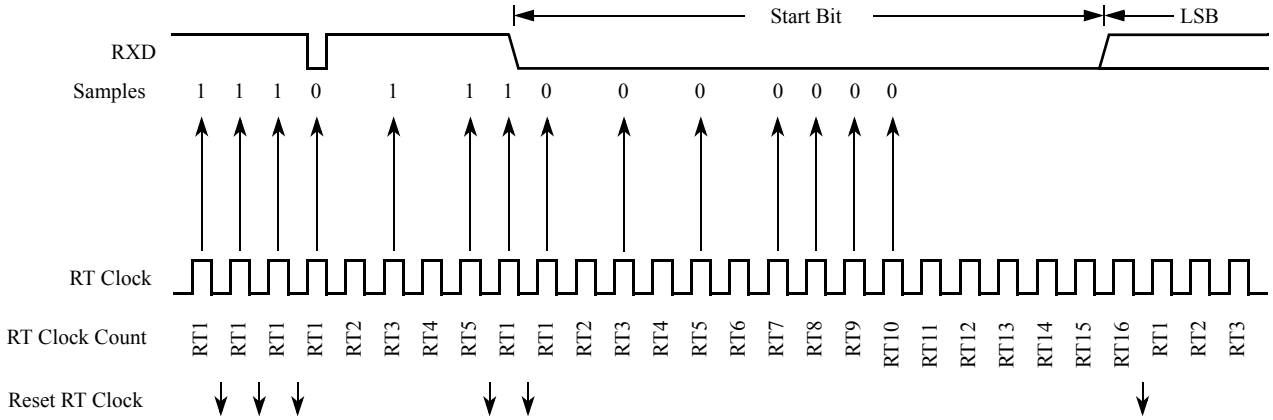

Figure 10-22. Start Bit Search Example 1

In Figure 10-23, verification sample at RT3 is high. The RT3 sample sets the noise flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

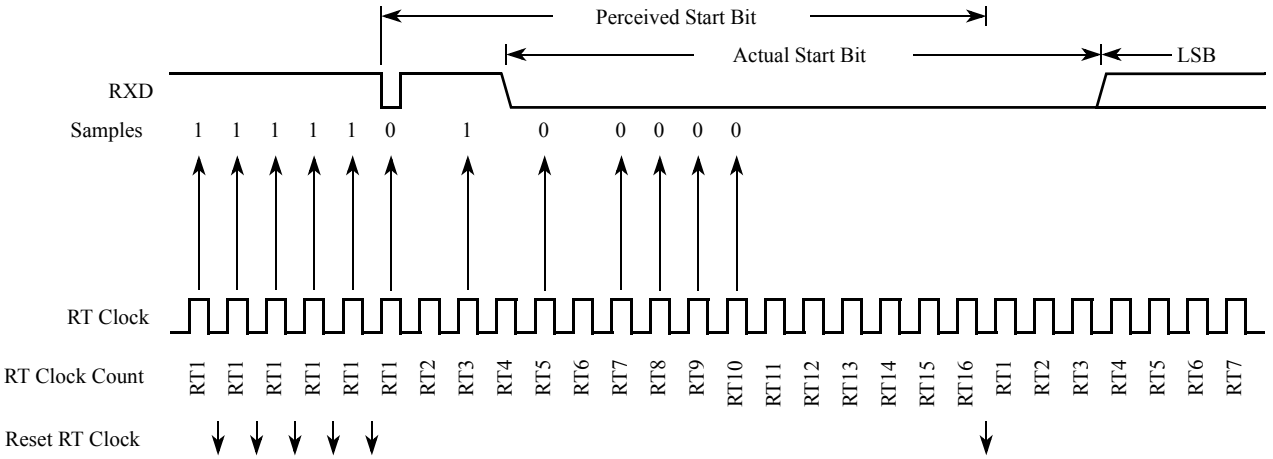

Figure 10-23. Start Bit Search Example 2

The main element of the SPI system is the SPI data register. The n-bit[1] data register in the master and the n-bit[1] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[1] register. When a data transfer operation is performed, this 2n-bit[1] register is serially shifted n[1] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 11.4.3, “Transmission Formats”).

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

11.4.1 Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{SS}$ pin

  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.

1. n depends on the selected transfer width, please refer to Section 11.3.2.2, “SPI Control Register 2 (SPICR2)

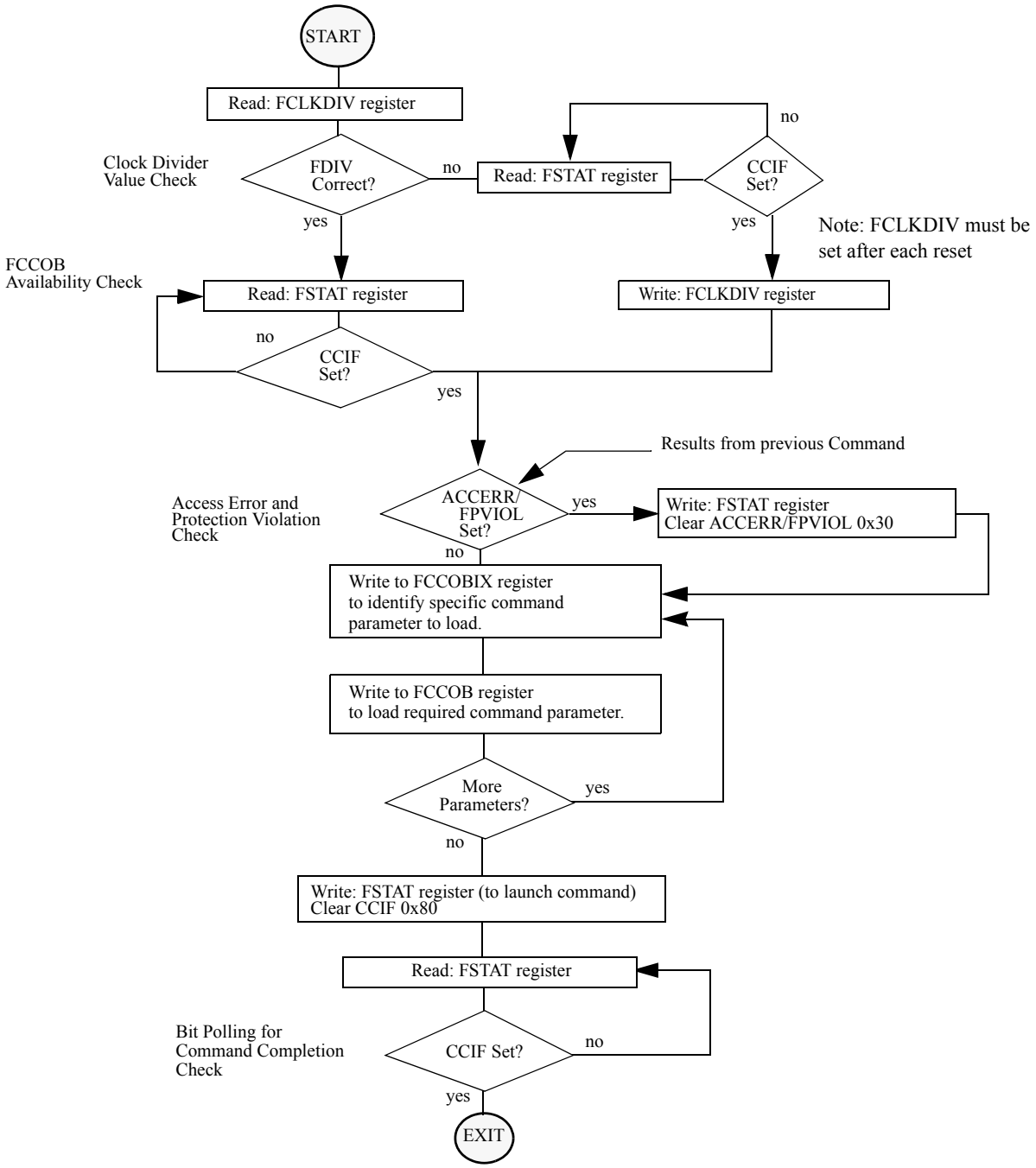

Figure 18-25. Generic Flash Command Write Sequence Flowchart

Address & Name		7	6	5	4	3	2	1	0
FRSV5	R	0	0	0	0	0	0	0	0
	W								
FRSV6	R	0	0	0	0	0	0	0	0
	W								
FRSV7	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 19-4. FTMRG32K128 Register Summary (continued)

19.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field in normal mode. In special mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0).

Table 19-7. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset

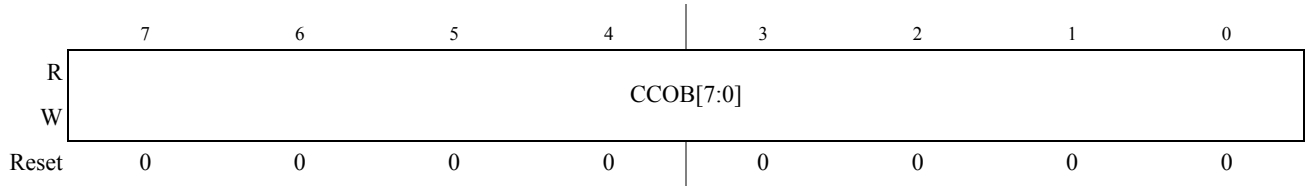

Figure 19-17. Flash Common Command Object Low Register (FCCOBLO)

19.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 19-24. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 19-24 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 19.4.6 Flash Command Description.

Table 19-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

Table 19-41. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 19-27)
		Set if an invalid global address [17:0] is supplied see Table 19-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

19.4.6.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 19.4.6.4 Read Once Command. The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 19-42. Program Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.