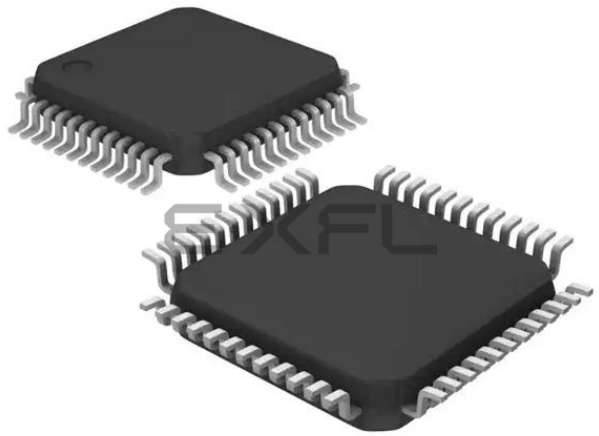


Welcome to E-XFL.COM

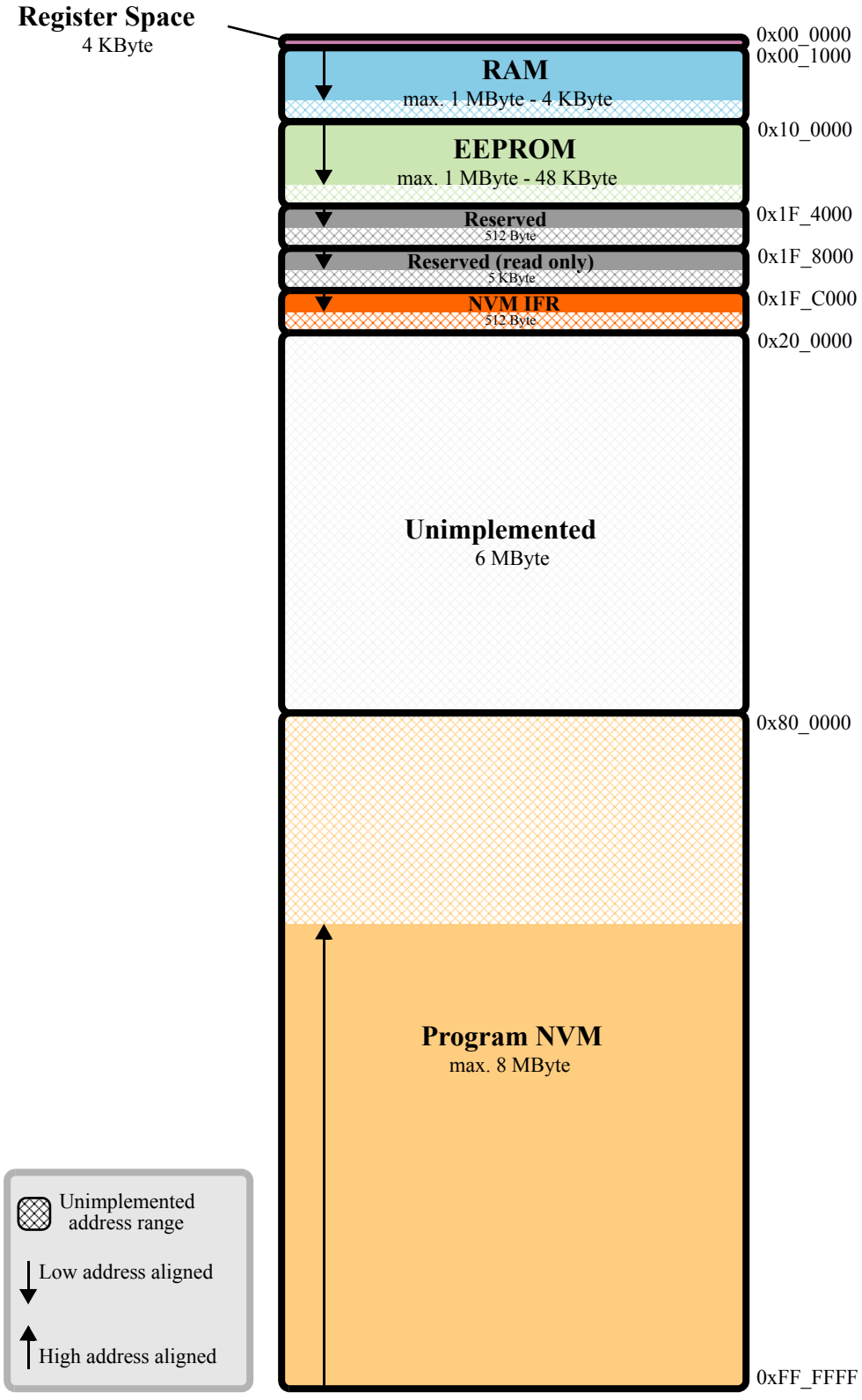
What is "Embedded - Microcontrollers"?

"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	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, I²C, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	28
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 10x12b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvca96f0mlf

Figure 1-2. shows S12Z CPU global memory map as a graphical representation.



1.7.1 Pin Assignment Overview

Table 1-5. - Port Availability by package option

Port	64 LQFP-EP	48 LQFP
Port AD	PAD[15:0]	PAD[9:0]
Port E	PE[1:0]	PE[1:0]
Port L (HVI)	PL[1:0]	PL[1:0]
Port J	PJ[1:0]	—
Port P	PP[7:0]	PP[6:4,2, 0]
Port S	PS[7:0]	PS[7,3:0]
Port T	PT[7:0]	PT[7,4:0]
sum of ports	46	30

1.7.2 Detailed Signal Descriptions

This section describes the signal properties.

1.7.2.1 $\overline{\text{RESET}}$ — External Reset signal

The $\overline{\text{RESET}}$ signal is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state, and an output when an internal MCU function causes a reset. The $\overline{\text{RESET}}$ pin has an internal pull-up device.

1.7.2.2 TEST — Test pin

This input only pin is reserved for factory test. This pin has an internal pull-down device.

NOTE

The TEST pin must be tied to ground in all applications.

1.7.3 MODC — Mode C signal

The MODC signal is used as a MCU operating mode select during reset. The state of this signal is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The signal has an internal pull-up device.

1.7.4 PAD[15:0] / KWAD[15:0] — Port AD, input pins of ADC

PAD[15:0] are general-purpose input or output signals. The signals can be configured on per signal basis as interrupt inputs with wake-up capability (KWAD[15:0]). These signals can have a pull-up or pull-down device selected and enabled on per signal basis. Out of reset the pull devices are disabled.

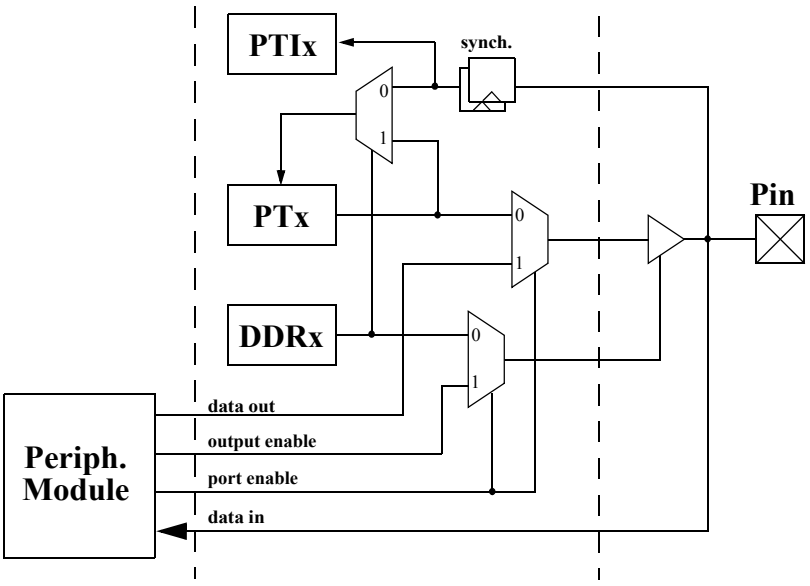


Figure 2-33. Illustration of I/O pin functionality

2.4.4 Interrupts

This section describes the interrupts generated by the PIM and their individual sources. Vector addresses and interrupt priorities are defined at MCU level.

Table 2-35. PIM Interrupt Sources

Module Interrupt Sources	Local Enable
$\overline{\text{XIRQ}}$	None
$\overline{\text{IRQ}}$	IRQCR[IRQEN]
Port AD pin interrupt	PIEADH[PIEADH] PIEADL[PIEADL]
Port S pin interrupt	PIES[PIES]
Port P pin interrupt	PIEP[PIEP]
Port L pin interrupt	PIEL[PIEL]
Port P over-current interrupt	OCIEP[OCIEP]

2.4.4.1 XIRQ, IRQ Interrupts

The $\overline{\text{XIRQ}}$ pin allows requesting non-maskable interrupts after reset initialization. During reset, the X bit in the condition code register is set and any interrupts are masked until software enables them.

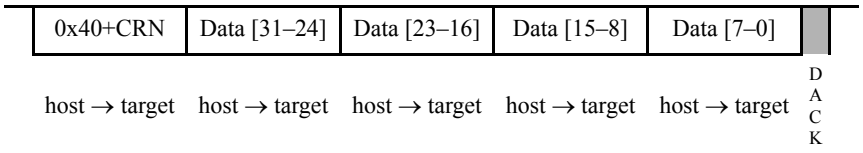
The $\overline{\text{IRQ}}$ pin allows requesting asynchronous interrupts. The interrupt input is disabled out of reset. To enable the interrupt the IRQCR[IRQEN] bit must be set and the I bit cleared in the condition code register. The interrupt can be configured for level-sensitive or falling-edge-sensitive triggering. If IRQCR[IRQEN] is cleared while an interrupt is pending, the request will deassert.

Table 3-6. BDCCSRL Field Descriptions (continued)

Field	Description
4 OVRUN	**Overrun Flag** — Indicates unexpected host activity before command completion. This occurs if a new command is received before the current command completion. With ACK enabled this also occurs if the host drives the BKGD pin low whilst a target ACK pulse is pending To protect internal resources from misinterpreted BDC accesses following an overrun, internal accesses are suppressed until a SYNC clears this bit. A SYNC clears the bit. 0 No overrun detected. 1 Overrun detected when issuing a BDC command.
3 NORESP	**No Response Flag** — Indicates that the BDC internal action or data access did not complete. This occurs in the following scenarios: a) If no free cycle for an access is found within 512 core clock cycles. This could typically happen if a code loop without free cycles is executing with ACK enabled and STEAL clear. b) With ACK disabled or STEAL set, when an internal access is not complete before the host starts data/BDCCSRL retrieval or an internal write access is not complete before the host starts the next BDC command. c) Attempted internal memory or SYNC_PC accesses during STOP mode set NORESP if BDCCIS is clear. In the above cases, on setting NORESP, the BDC aborts the access if permitted. (For devices supporting EWAIT, BDC external accesses with EWAIT assertions, prevent a command from being aborted until EWAIT is deasserted). d) If a BACKGROUND command is issued whilst the device is in wait mode the NORESP bit is set but the command is not aborted. The active BDM request is completed when the device leaves wait mode. Furthermore subsequent CPU register access commands during wait mode set the NORESP bit, should it have been cleared. e) If a command is issued whilst awaiting return from Wait mode. This can happen when using STEP1 to step over a CPU WAI instruction, if the CPU has not returned from Wait mode before the next BDC command is received. f) If STEP1 is issued with the BDC enabled as the device enters Wait mode regardless of the BDMACT state. When NORESP is set a value of 0xEE is returned for each data byte associated with the current access. Writing a “1” to this bit, clears the bit. 0 Internal action or data access completed. 1 Internal action or data access did not complete.
2 RDINV	**Read Data Invalid Flag** — Indicates invalid read data due to an ECC error during a BDC initiated read access. The access returns the actual data read from the location. Writing a “1” to this bit, clears the bit. 0 No invalid read data detected. 1 Invalid data returned during a BDC read access.
1 ILLACC	**Illegal Access Flag** — Indicates an attempted illegal access. This is set in the following cases: When the attempted access addresses unimplemented memory When the access attempts to write to the flash array When a CPU register access is attempted with an invalid CRN (Section 3.4.5.1, “BDC Access Of CPU Registers). Illegal accesses return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal access detected. 1 Illegal BDC access detected.
0 ILLCMD	**Illegal Command Flag** — Indicates an illegal BDC command. This bit is set in the following cases: When an unimplemented BDC command opcode is received. When a DUMP_MEM{_WS}, FILL_MEM{_WS} or READ_SAME{_WS} is attempted in an illegal sequence. When an active BDM command is received whilst BDM is not active When a non Always-available command is received whilst the BDC is disabled or a flash mass erase is ongoing. When a non Always-available command is received whilst the device is secure Read commands return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal command detected. 1 Illegal BDC command detected.

3.4.4.17 WRITE_Rn

Write general-purpose CPU register **Active Background**

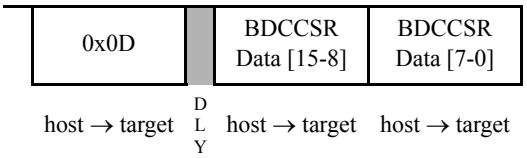


If the device is in active BDM, this command writes the 32-bit operand to the selected CPU general-purpose register. See Section 3.4.5.1, “BDC Access Of CPU Registers for the CRN details. Accesses to CPU registers are always 32-bits wide, regardless of implemented register width. If enabled an ACK pulse is generated after the internal write access has been completed or aborted.

If the device is not in active BDM, this command is rejected as an illegal operation, the ILLCMD bit is set and no operation is performed.

3.4.4.18 WRITE_BDCCSR

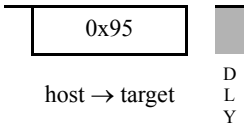
Write BDCCSR **Always Available**



16-bit write to the BDCCSR register. No ACK pulse is generated. Writing to this register can be used to configure control bits or clear flag bits. Refer to the register bit descriptions.

3.4.4.19 ERASE_FLASH

Erase FLASH **Always Available**



Mass erase the internal flash. This command can always be issued. On receiving this command twice in succession, the BDC sets the ERASE bit in BDCCSR and requests a flash mass erase. Any other BDC command following a single ERASE_FLASH initializes the sequence, such that thereafter the ERASE_FLASH must be applied twice in succession to request a mass erase. If 512 BDCSI clock cycles elapse between the consecutive ERASE_FLASH commands then a timeout occurs, which forces a soft reset and initializes the sequence. The ERASE bit is cleared when the mass erase sequence has been completed. No ACK is driven.

4.1.1 Glossary

Table 4-2. Glossary Of Terms

Term	Definition
MCU	Microcontroller Unit
CPU	S12Z Central Processing Unit
BDC	S12Z Background Debug Controller
ADC	Analog-to-Digital Converter
unmapped address range	Address space that is not assigned to a memory
reserved address range	Address space that is reserved for future use cases
illegal access	Memory access, that is not supported or prohibited by the S12ZMMC, e.g. a data store to NVM
access violation	Either an illegal access or an uncorrectable ECC error
byte	8-bit data
word	16-bit data

4.1.2 Overview

The S12ZMMC provides access to on-chip memories and peripherals for the S12ZCPU, the S12ZBDC, and the ADC. It arbitrates memory accesses and determines all of the MCU memory maps. Furthermore, the S12ZMMC is responsible for selecting the MCUs functional mode.

4.1.3 Features

- S12ZMMC mode operation control
- Memory mapping for S12ZCPU, S12ZBDC, and ADC
  - Maps peripherals and memories into a 16 MByte address space for the S12ZCPU, the S12ZBDC, and the ADC
  - Handles simultaneous accesses to different on-chip resources (NVM, RAM, and peripherals)
- Access violation detection and logging
  - Triggers S12ZCPU machine exceptions upon detection of illegal memory accesses and uncorrectable ECC errors
  - Logs the state of the S12ZCPU and the cause of the access error

4.1.4 Modes of Operation

4.1.4.1 Chip configuration modes

The S12ZMMC determines the chip configuration mode of the device. It captures the state of the MODC pin at reset and provides the ability to switch from special-single chip mode to normal single chip-mode.

- One I-bit maskable interrupt vector request associated with $\overline{\text{IRQ}}$ (at address vector base[1] + 0x0001D4).
- up to 113 additional I-bit maskable interrupt vector requests (at addresses vector base[1] + 0x000010 .. vector base + 0x0001D0).
- Each I-bit maskable interrupt request has a configurable priority level.
- I-bit maskable interrupts can be nested, depending on their priority levels.
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs or whenever $\overline{\text{XIRQ}}$ is asserted, even if X interrupt is masked.

5.1.3 Modes of Operation

- Run mode

  This is the basic mode of operation.
- Wait mode

  In wait mode, the INT module is capable of waking up the CPU if an eligible CPU exception occurs. Please refer to Section 5.5.3, “Wake Up from Stop or Wait Mode” for details.
- Stop Mode

  In stop mode, the INT module is capable of waking up the CPU if an eligible CPU exception occurs. Please refer to Section 5.5.3, “Wake Up from Stop or Wait Mode” for details.

5.1.4 Block Diagram

Figure 5-1 shows a block diagram of the INT module.

9.5.3.2.6 Conversion flow control in case of conversion sequence control bit overrun scenarios

Restart Request Overrun:
If a legal Restart Request is detected and no Restart Event is in progress, the RSTA bit is set due to the request. The set RSTA bit indicates that a Restart Request was detected and the Restart Event is in process. In case further Restart Requests occur while the RSTA bit is set, this is defined a overrun situation. This scenario is likely to occur when bit STR_SEQA is set or when a Restart Event causes a Sequence Abort Event. The request overrun is captured in a background register that always stores the last detected overrun request. Hence if the overrun situation occurs more than once while a Restart Event is in progress, only the latest overrun request is pending. When the RSTA bit is cleared, the latest overrun request is processed and RSTA is set again one cycle later.

LoadOK Overrun:
Simultaneously at any Restart Request overrun situation the LoadOK input is evaluated and the status is captured in a background register which is alternated anytime a Restart Request Overrun occurs while Load OK Request is asserted. The Load OK background register is cleared as soon as the pending Restart Request gets processed.

Trigger Overrun:
If a Trigger occurs whilst bit TRIG is already set, this is defined as a Trigger overrun situation and causes the ADC to cease conversion at the next conversion boundary and to set bit TRIG_EIF. A overrun is also detected if the Trigger Event occurs automatically generated by hardware in “Trigger Mode” due to a Restart Event and simultaneously a Trigger Event is generated via data bus or internal interface. In this case the ADC ceases operation before conversion begins to sample. In “Trigger Mode” a Restart Request Overrun does not cause a Trigger Overrun (bit TRIG_EIF not set).

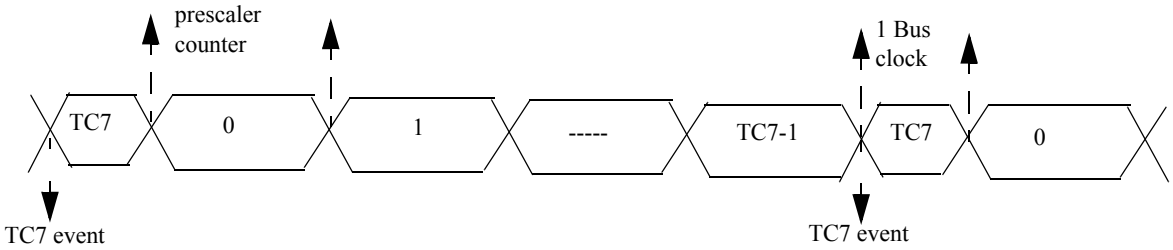
Sequence Abort Request Overrun:
If a Sequence Abort Request occurs whilst bit SEQA is already set, this is defined as a Sequence Abort Request Overrun situation and the overrun request is ignored.

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

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

Note: in Figure 11-31,if PR[2:0] is equal to 0, one prescaler counter equal to one Bus clock

Figure 11-31. The TCNT cycle diagram under TCRE=1 condition



11.4.3.1 OC Channel Initialization

The internal register whose output drives OCx can be programmed before the timer drives OCx. The desired state can be programmed to this internal register by writing a one to CFORCx bit with TIOSx, OCPDx and TEN bits set to one.

Set OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=1 and OCPDx=1

Clear OCx: Write a 1 to FOCx while TEN=1, IOSx=1, OMx=1, OLx=0 and OCPDx=1

Setting OCPDx to zero allows the internal register to drive the programmed state to OCx. This allows a glitch free switch over of port from general purpose I/O to timer output once the OCPDx bit is set to zero.

11.4.4 Pulse Accumulator

The pulse accumulator (PACNT) is a 16-bit counter that can operate in two modes:

Event counter mode — Counting edges of selected polarity on the pulse accumulator input pin, PAI.

Gated time accumulation mode — Counting pulses from a divide-by-64 clock. The PAMOD bit selects the mode of operation.

The minimum pulse width for the PAI input is greater than two Bus clocks.

- 16-bit counter.

12.1.2 Modes of Operation

Stop: Timer is off because clocks are stopped.

Freeze: Timer counter keeps on running, unless TSFRZ in TSCR1 is set to 1.

Wait: Counters keeps on running, unless TSWAI in TSCR1 is set to 1.

Normal: Timer counter keep on running, unless TEN in TSCR1 is cleared to 0.

12.1.3 Block Diagrams

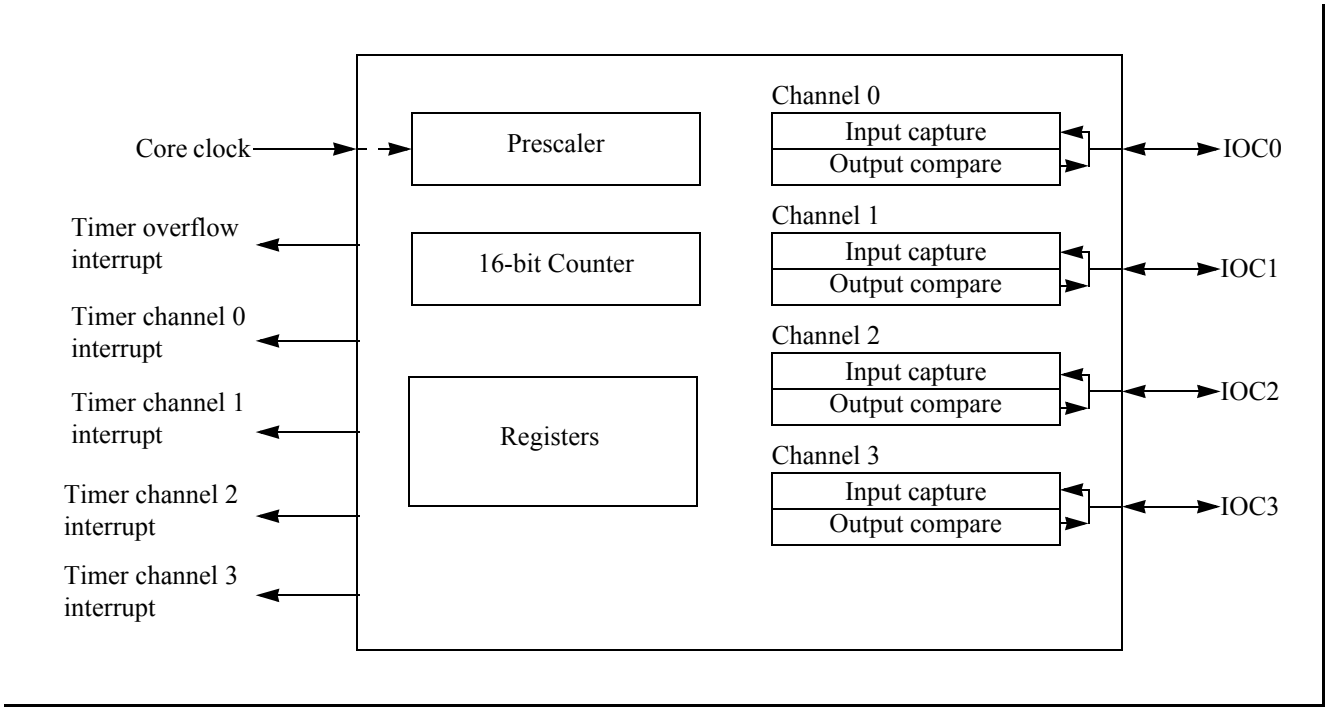


Figure 12-1. TIM16B4CV3 Block Diagram

- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period register will go directly to the latches as well as the buffer.

NOTE

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.

See Section 13.4.2.3, “PWM Period and Duty” for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for that channel:

- Left aligned output (CAEx = 0)

  PWMx Period = Channel Clock Period * PWMPERx
- Center Aligned Output (CAEx = 1)

  PWMx Period = Channel Clock Period * (2 * PWMPERx)

For boundary case programming values, please refer to Section 13.4.2.8, “PWM Boundary Cases”.

Module Base + 0x0014 = PWMPER0, 0x0015 = PWMPER1, 0x0016 = PWMPER2, 0x0017 = PWMPER3
Module Base + 0x0018 = PWMPER4, 0x0019 = PWMPER5, 0x001A = PWMPER6, 0x001B = PWMPER7

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

Figure 13-13. PWM Channel Period Registers (PWMPERx)

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Read: Anytime

Write: Anytime

13.3.2.12 PWM Channel Duty Registers (PWMDTYx)

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)

15.4.3.1 Clock Phase and Polarity Controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

15.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After $2n$[1] (last) SCK edges:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Figure 15-12 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

1. n depends on the selected transfer width, please refer to Section 15.3.2.2, "SPI Control Register 2 (SPICR2)

Table 18-25. Message Buffer Organization

Offset Address	Register	Access
0x00X0	IDR0 — Identifier Register 0	R/W
0x00X1	IDR1 — Identifier Register 1	R/W
0x00X2	IDR2 — Identifier Register 2	R/W
0x00X3	IDR3 — Identifier Register 3	R/W
0x00X4	DSR0 — Data Segment Register 0	R/W
0x00X5	DSR1 — Data Segment Register 1	R/W
0x00X6	DSR2 — Data Segment Register 2	R/W
0x00X7	DSR3 — Data Segment Register 3	R/W
0x00X8	DSR4 — Data Segment Register 4	R/W
0x00X9	DSR5 — Data Segment Register 5	R/W
0x00XA	DSR6 — Data Segment Register 6	R/W
0x00XB	DSR7 — Data Segment Register 7	R/W
0x00XC	DLR — Data Length Register	R/W
0x00XD	TBPR — Transmit Buffer Priority Register[1]	R/W
0x00XE	TSRH — Time Stamp Register (High Byte)	R
0x00XF	TSRL — Time Stamp Register (Low Byte)	R

[1] Not applicable for receive buffers

Figure 18-24 shows the common 13-byte data structure of receive and transmit buffers for extended identifiers. The mapping of standard identifiers into the IDR registers is shown in Figure 18-25.

All bits of the receive and transmit buffers are ‘x’ out of reset because of RAM-based implementation[1]. All reserved or unused bits of the receive and transmit buffers always read ‘x’.

1. Exception: The transmit buffer priority registers are 0 out of reset.

- Protection scheme to prevent accidental program or erase of EEPROM memory
- Ability to program up to four words in a burst sequence

22.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

22.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 22-1.

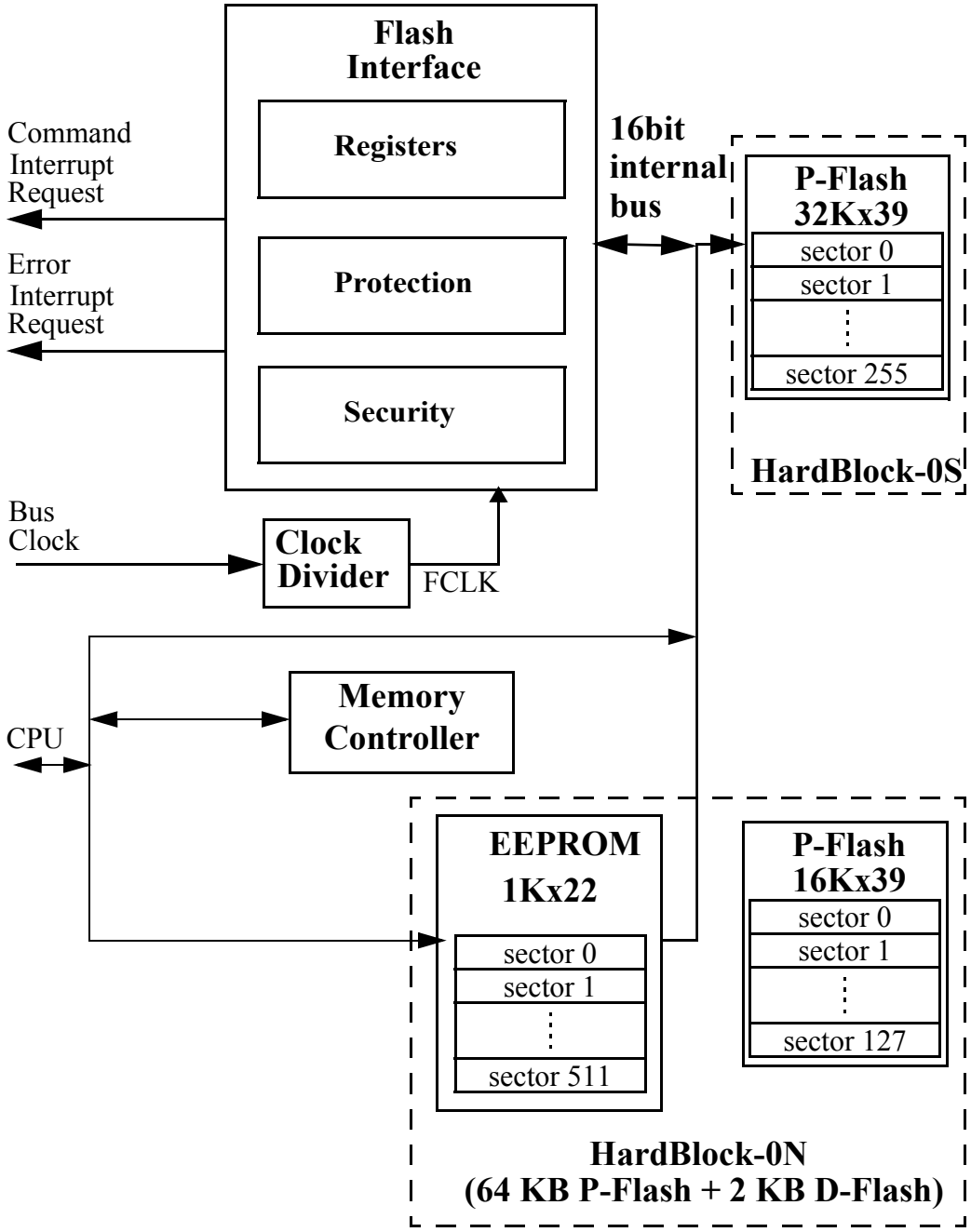


Figure 22-1. FTMRZ192K2K Block Diagram

22.2 External Signal Description

The Flash module contains no signals that connect off-chip.

Table 22-35. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if an invalid global address [23:0] is supplied see Table 22-2)
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

22.4.7.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Table 22-36. Erase Verify P-Flash Section Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x03	Global address [23:16] of a P-Flash block
FCCOB1	Global address [15:0] of the first phrase to be verified	
FCCOB2	Number of phrases to be verified	

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 22-37. Erase Verify P-Flash Section Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is supplied see Table 22-2)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if the requested section crosses a the P-Flash address boundary
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

22.4.7.14 Erase Verify EEPROM Section Command

The Erase Verify EEPROM Section command will verify that a section of code in the EEPROM is erased. The Erase Verify EEPROM Section command defines the starting point of the data to be verified and the number of words.

Table 22-61. Erase Verify EEPROM Section Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x10	Global address [23:16] to identify the EEPROM block
FCCOB1	Global address [15:0] of the first word to be verified	
FCCOB2	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify EEPROM Section command, the Memory Controller will verify the selected section of EEPROM memory is erased. The CCIF flag will set after the Erase Verify EEPROM Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 22-62. Erase Verify EEPROM Section Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested section breaches the end of the EEPROM block
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

Table A-10. CPMU Configuration for Pseudo Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUOSC	OSCE=1, Quartz oscillator f_{EXTAL}=4MHz
CPMURTI	RTDEC=0, RTR[6:4]=111, RTR[3:0]=1111
CPMUCOP	WCOP=1, CR[2:0]=111

Table A-11. CPMU Configuration for Run/Wait and Full Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUSYNR	VCOFRQ[1:0]= 1,SYNDIV[5:0] = 31
CPMUPOSTDIV	POSTDIV[4:0]=0
CPMUCLKS	PLLSEL=1, CSAD=0
CPMUOSC	OSCE=0, Reference clock for PLL is f_{ref}=f_{irc1m} trimmed to 1MHz
API settings for STOP current measurement	
CPMUAPICTL	APIEA=0, APIFE=1, APIE=0
CPMUACLKTR	trimmed to >=20Khz
CPMUAPIRH/RL	set to 0xFFFF

Table A-12. Peripheral Configurations for Run and Wait Current Measurement

Peripheral	Configuration
SCI	Continuously transmit data (0x55) at speed of 19200 baud
SPI	Configured to master mode, continuously transmit data (0x55) at 1Mbit/s
ACMP0 & 1	The module is enabled and the plus & minus inputs are toggling with 0-1 and 1-0 at 1MHz
DAC	The module is enabled in buffered mode at full voltage range
ADC	The peripheral is configured to operate at its maximum specified frequency and to continuously convert voltages on a single input channel
MSCAN	The module is connected to CANPHY and continously transmit data (0x55 or 0xAA) with a bit rate of 500kbit/s.
CANPHY	The module is enabled and connect to MSCAN module
IIC	Operate in master mode and contiously transmit data (0x55 or 0xAA) at the bit rate of 100Kbit/s
PWM0	The module is configured with a modulus rate of 10 kHz
PWM1	The module is configured with a modulus rate of 10 kHz

[2] T_A: Ambient Temperature

Input Offset and Hysteresis