

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	I ² C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN-EP (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12zvlslf0cfm

Chapter 3

5V Analog Comparator (ACMPV2)

3.1	Introduction	107
3.2	Features	107
3.3	Block Diagram	108
3.4	External Signals	108
3.4.1	Internal Signals	108
3.5	Modes of Operation	108
3.6	Memory Map and Register Definition	109
3.6.1	Register Map	109
3.6.2	Register Descriptions	110
3.7	Functional Description	113
3.8	Interrupts	114

Chapter 4

Memory Mapping Control (S12ZMMCV1)

4.1	Introduction	115
4.1.1	Glossary	116
4.1.2	Overview	116
4.1.3	Features	116
4.1.4	Modes of Operation	116
4.1.5	Block Diagram	117
4.2	External Signal Description	117
4.3	Memory Map and Register Definition	117
4.3.1	Memory Map	117
4.3.2	Register Descriptions	118
4.4	Functional Description	123
4.4.1	Global Memory Map	123
4.4.2	Illegal Accesses	125
4.4.3	Uncorrectable ECC Faults	126

Chapter 5

Background Debug Controller (S12ZBDCV2)

5.1	Introduction	127
5.1.1	Glossary	128
5.1.2	Features	128
5.1.3	Modes of Operation	128
5.1.4	Block Diagram	131
5.2	External Signal Description	131
5.3	Memory Map and Register Definition	132
5.3.1	Module Memory Map	132
5.3.2	Register Descriptions	132
5.4	Functional Description	136
5.4.1	Security	136

Chapter 14

Supply Voltage Sensor (BATSV3)

14.1	Introduction	429
14.1.1	Features	429
14.1.2	Modes of Operation	429
14.1.3	Block Diagram	430
14.2	External Signal Description	430
14.2.1	VSUP — Voltage Supply Pin	430
14.3	Memory Map and Register Definition	431
14.3.1	Register Summary	431
14.3.2	Register Descriptions	431
14.4	Functional Description	435
14.4.1	General	435
14.4.2	Interrupts	435

Chapter 15

Timer Module (TIM16B6CV3)

15.1	Introduction	439
15.1.1	Features	439
15.1.2	Modes of Operation	440
15.1.3	Block Diagrams	440
15.2	External Signal Description	441
15.2.1	IOC5 - IOC0 — Input Capture and Output Compare Channel 5-0	441
15.3	Memory Map and Register Definition	441
15.3.1	Module Memory Map	441
15.3.2	Register Descriptions	441
15.4	Functional Description	453
15.4.1	Prescaler	454
15.4.2	Input Capture	455
15.4.3	Output Compare	455
15.5	Resets	456
15.6	Interrupts	456
15.6.1	Channel [5:0] Interrupt (C[5:0]F)	456
15.6.2	Timer Overflow Interrupt (TOF)	456

Chapter 16

Timer Module (TIM16B2CV3)

16.1	Introduction	457
16.1.1	Features	457
16.1.2	Modes of Operation	458
16.1.3	Block Diagrams	458
16.2	External Signal Description	459
16.2.1	IOC1 - IOC0 — Input Capture and Output Compare Channel 1-0	459
16.3	Memory Map and Register Definition	459

Similarly the STEP1 command issued from a WAI instruction cannot be completed by the CPU until the CPU leaves wait mode due to an interrupt. The first STEP1 into wait mode sets the BDCCSR WAIT bit.

If the part is still in Wait mode and a further STEP1 is carried out then the NORESP and ILLCMD bits are set because the device is no longer in active BDM for the duration of WAI execution.

5.1.4 Block Diagram

A block diagram of the BDC is shown in [Figure 5-1](#).

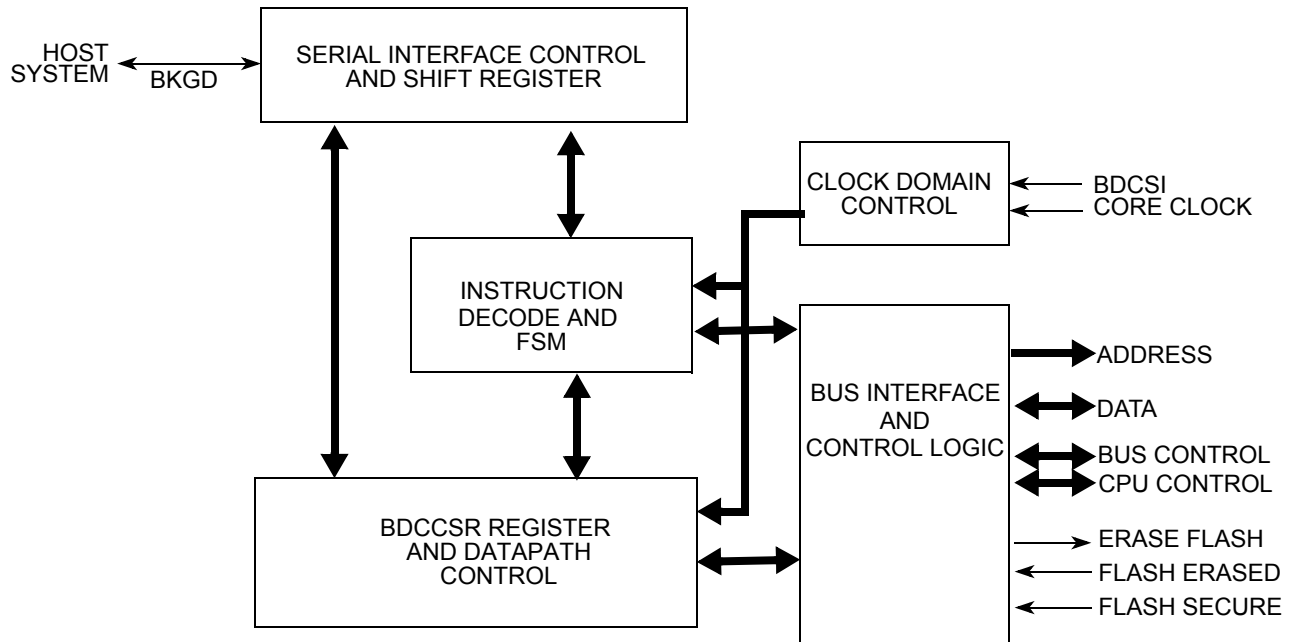


Figure 5-1. BDC Block Diagram

5.2 External Signal Description

A single-wire interface pin (BKGD) is used to communicate with the BDC system. During reset, this pin is a device mode select input. After reset, this pin becomes the dedicated serial interface pin for the BDC.

BKGD is a pseudo-open-drain pin with an on-chip pull-up. Unlike typical open-drain pins, the external RC time constant on this pin due to external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speed-up pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to [Section 5.4.6, “BDC Serial Interface”](#) for more details.

which cannot be disabled, are always handled by the CPU and have a fixed priority levels. A priority level of 0 effectively disables the associated I-bit maskable interrupt request.

If more than one interrupt request is configured to the same interrupt priority level 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 setup in the configuration register associated with the interrupt request channel must meet the following conditions:
 - a) The priority level must be set to non zero.
 - b) The priority level must be greater than the current interrupt processing level in the condition code register (CCW) of the CPU ($PRIOLVL[2:0] > IPL[2:0]$).
3. The I-bit in the condition code register (CCW) of the CPU must be cleared.
4. There is no access violation interrupt request pending.
5. There is no SYS, SWI, SPARE, TRAP, Machine Exception or \overline{XIRQ} request pending.

NOTE

All non I-bit maskable interrupt requests always have higher priority than I-bit maskable interrupt requests. If an I-bit maskable interrupt request is interrupted by a non I-bit maskable interrupt request, the currently active interrupt processing level (IPL) remains unaffected. It is possible to nest non I-bit maskable interrupt requests, e.g., by nesting SWI, SYS or TRAP calls.

6.4.2.1 Interrupt Priority Stack

The current interrupt processing level (IPL) is stored in the condition code register (CCW) of the CPU. This way the current IPL is automatically pushed to the stack by the standard interrupt stacking procedure. The new IPL is copied to the CCW from the priority level of the highest priority active interrupt request channel which is configured to be handled by the CPU. The copying takes place when the interrupt vector is fetched. The previous IPL is automatically restored from the stack by executing the RTI instruction.

6.4.3 Priority Decoder

The S12ZINTV0 module contains a priority decoder to determine the relative priority for all interrupt requests pending for the CPU.

A CPU interrupt vector is not supplied until the CPU requests it. Therefore, it is possible that a higher priority interrupt request could override the original exception which caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this exception first instead of the original request.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the vector request), the vector address supplied to the CPU defaults to that of the spurious interrupt vector.

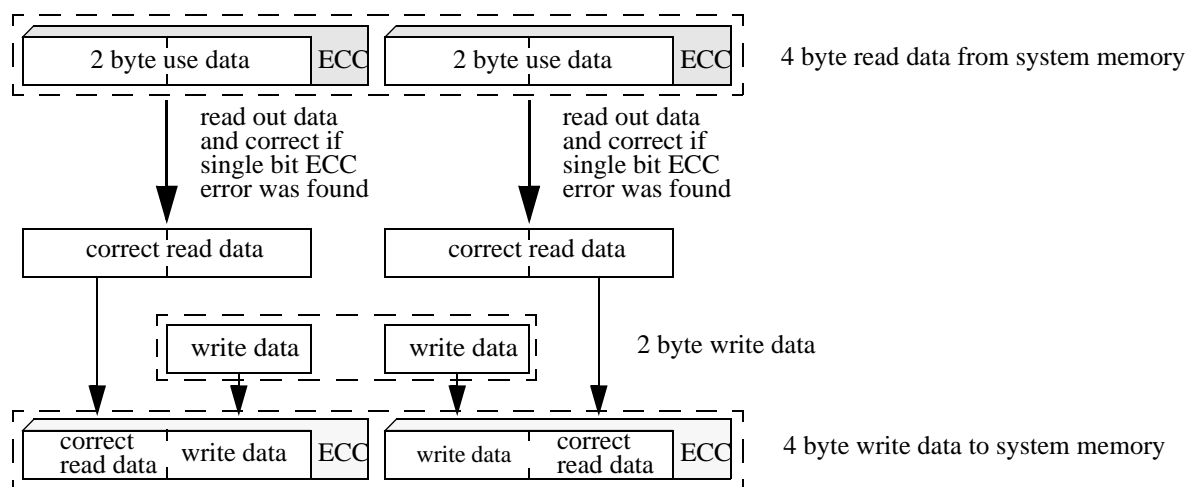


Figure 8-9. 2 byte non-aligned write access

8.3.3 Memory Read Access

During each memory read access an ECC check is performed. If the logic detects a single bit ECC error, then the module corrects the data, so that the access initiator module receives correct data. In parallel, the logic writes the corrected data back to the memory, so that this read access repairs the single bit ECC error. This automatic ECC read repair function is disabled by setting the ECCDRR bit.

If a single bit ECC error was detected, then the SBEEIF flag is set.

If the logic detects a double bit ECC error, then the data word is flagged as invalid, so that the access initiator module can ignore the data.

8.3.4 Memory Initialization

To avoid spurious ECC error reporting, memory operations that allow a read before a first write (like the read-modify-write operation of the non-aligned access) require that the memory contains valid ECC values before the first read-modify-write access is performed. The ECC module provides logic to initialize the complete memory content with zero during the power up phase. During the initialization process the access to the SRAM is disabled and the RDY status bit is cleared. If the initialization process is done, SRAM access is possible and the RDY status bit is set.

8.3.5 Interrupt Handling

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

access is always a 2 byte aligned memory access, so that no ECC check is performed and no single or double bit ECC error indication is activated.

8.3.7.2 ECC Debug Memory Read Access

Writing one to the ECCDR bit performs a debug read access from the memory address defined by register DPTR. If the ECCDR bit is cleared then the register DDATA contains the uncorrected read data from the memory. The register DECC contains the ECC value read from the memory. Independent of the ECCDRR register bit setting, the debug read access will not perform an automatic ECC repair during read access. During the debug read access no ECC check is performed, so that no single or double bit ECC error indication is activated.

If the ECCDW and the ECCDR bits are set at the same time, then only the debug write access is performed.

9.3.2.26 **Reserved Register CPMUTEST2**

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU_UHV’s functionality.

Module Base + 0x001C

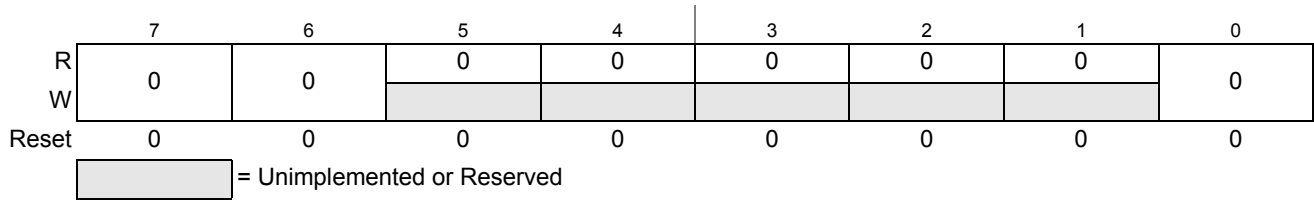


Figure 9-35. Reserved Register CPMUTEST2

Read: Anytime

Write: Only in Special Mode

10.2 Introduction

The ADC12B_LBA is an n-channel multiplexed input successive approximation analog-to-digital converter. Refer to device electrical specifications for ADC parameters and accuracy.

The List Based Architecture (LBA) provides flexible conversion sequence definition as well as flexible oversampling. The order of channels to be converted can be freely defined. Also, multiple instantiations of the module can be triggered simultaneously (matching sampling point across multiple module instantiations).

There are four register bits which control the conversion flow (please refer to the description of register ADCFLWCTL).

The four conversion flow control bits of register ADCFLWCTL can be modified in two different ways:

- Via data bus accesses
- Via internal interface Signals (Trigger, Restart, LoadOK, and Seq_Abort; see also [Figure 10-2](#)). Each Interface Signal is associated with one conversion flow control bit.

For information regarding internal interface connectivity related to the conversion flow control please refer to the device overview of the reference manual.

The ADCFLWCTL register can be controlled via internal interface only or via data bus only or by both depending on the register access configuration bits ACC_CFG[1:0].

The four bits of register ADCFLWCTL reflect the captured request and status of the four internal interface Signals (LoadOK, Trigger, Restart, and Seq_abort; see also [Figure 10-2](#)) if access configuration is set accordingly and indicate event progress (when an event is processed and when it is finished).

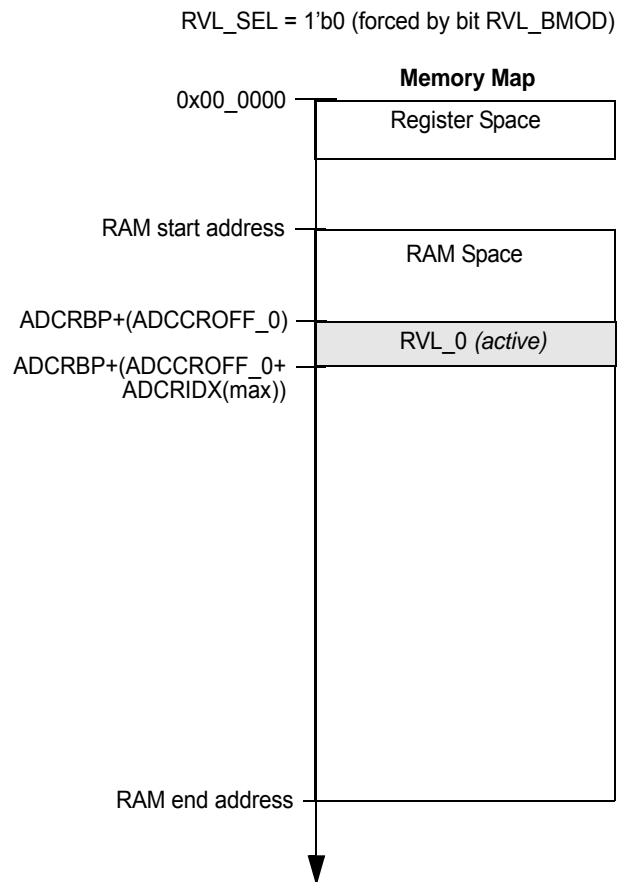
Conversion flow error situations are captured by corresponding interrupt flags in the ADCEIF register.

There are two conversion flow control modes (Restart Mode, Trigger Mode). Each mode causes a certain behavior of the conversion flow control bits which can be selected according to the application needs.

Please refer to [Section 10.5.2.1, “ADC Control Register 0 \(ADCCTL_0\)”](#) and [Section 10.6.3.2.4, “The two conversion flow control Mode Configurations”](#) for more information regarding conversion flow control.

Because internal components of the ADC are turned on/off with bit ADC_EN, the ADC requires a recovery time period (t_{REC}) after ADC is enabled until the first conversion can be launched via a trigger.

When bit ADC_EN gets cleared (transition from 1'b1 to 1'b0) any ongoing conversion sequence will be aborted and pending results, or the result of current conversion, gets discarded (not stored). The ADC cannot be re-enabled before any pending action or action in process is finished respectively aborted, which could take up to a maximum latency time of $t_{DISABLE}$ (see device level specification for more details).



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 10-34. Result Value List Schema in Single Buffer Mode

While ADC is enabled, one Result Value List is active (indicated by bit RVL_SEL). The conversion Result Value List can be read anytime. When the ADC is enabled the conversion result address registers (ADCRBP, ADCCROFF_0/1, ADCRIDX) are read only and register ADCRIDX is under control of the ADC.

A conversion result is always stored as 16bit entity in unsigned data representation. Left and right justification inside the entity is selected via the DJM control bit. Unused bits inside an entity are stored zero.

Table 10-33. Conversion Result Justification Overview

Conversion Resolution (SRES[1:0])	Left Justified Result (DJM = 1'b0)	Right Justified Result (DJM = 1'b1)
8 bit	{Result[7:0],8'b00000000}	{8'b00000000,Result[7:0]}
10 bit	{Result[9:0],6'b0000000}	{6'b0000000,Result[9:0]}
12 bit	{Result[11:0],4'b0000}	{4'b0000,Result[11:0]}

- ¹ Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 13-14. CANTIER Register Field Descriptions

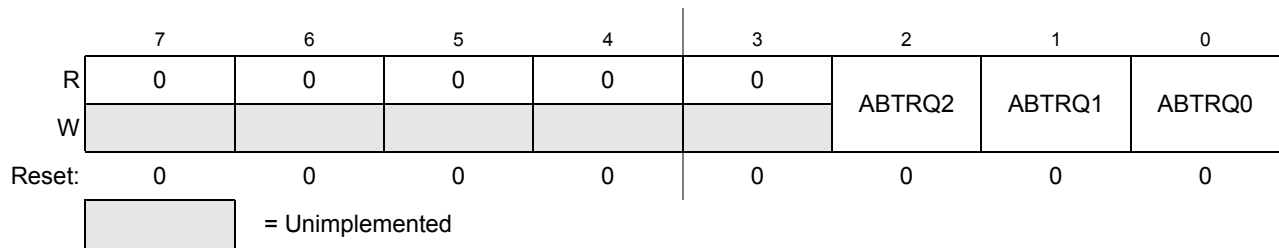
Field	Description
2-0 TXEIE[2:0]	Transmitter Empty Interrupt Enable 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

13.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008

Access: User read/write¹

**Figure 13-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)**

- ¹ Read: Anytime
Write: Anytime when not in initialization mode

NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 13-15. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	Abort Request — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 13.3.2.7, “MSCAN Transmitter Flag Register (CANTFLG)”) and abort acknowledge flags (ABTAK, see Section 13.3.2.10, “MSCAN Transmitter Message Abort Acknowledge Register (CANTAACK)”) are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

15.1.2 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.

15.1.3 Block Diagrams

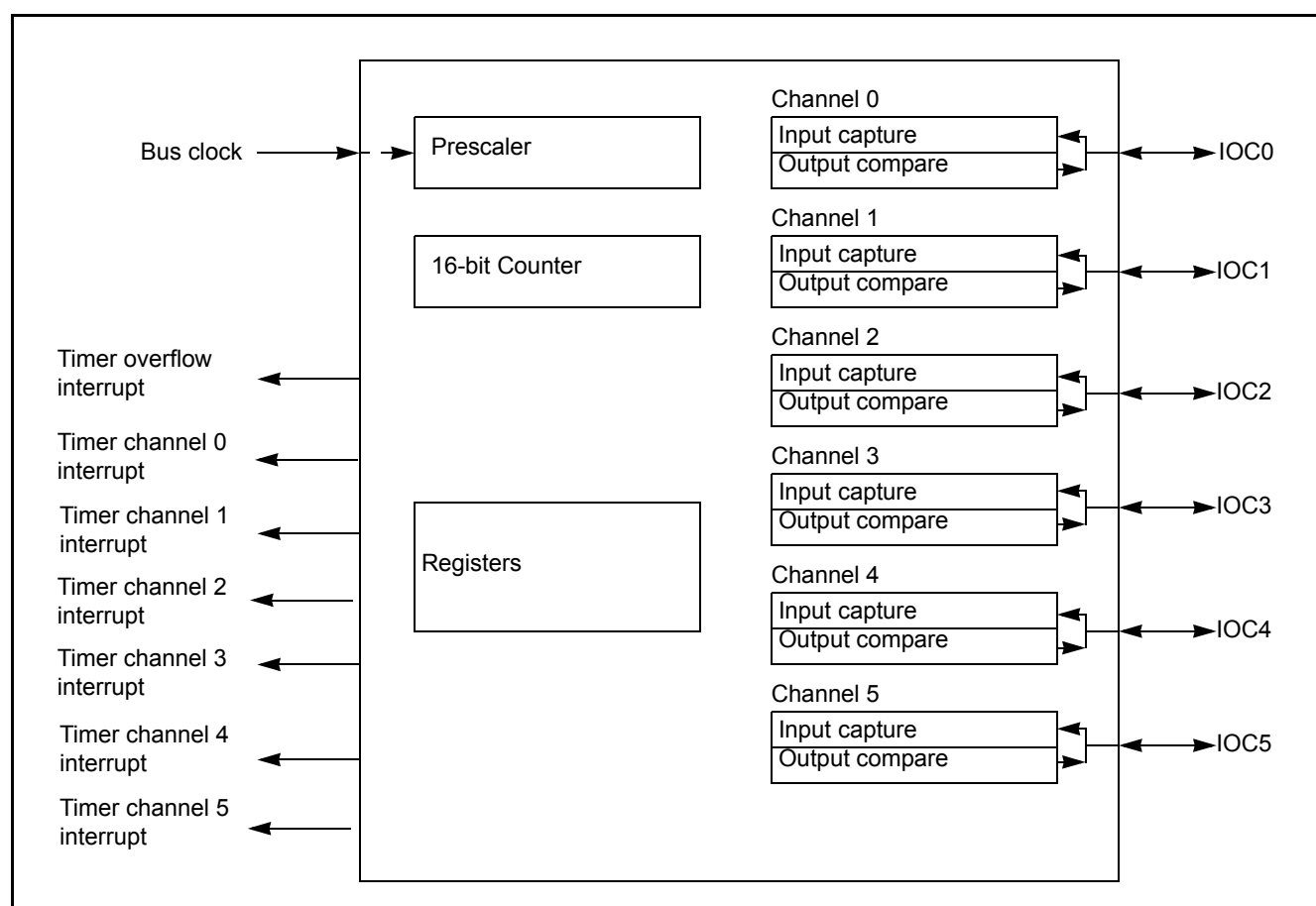


Figure 15-1. TIM16B6CV3 Block Diagram

15.3.2.8 Timer Interrupt Enable Register (TIE)

Module Base + 0x000C

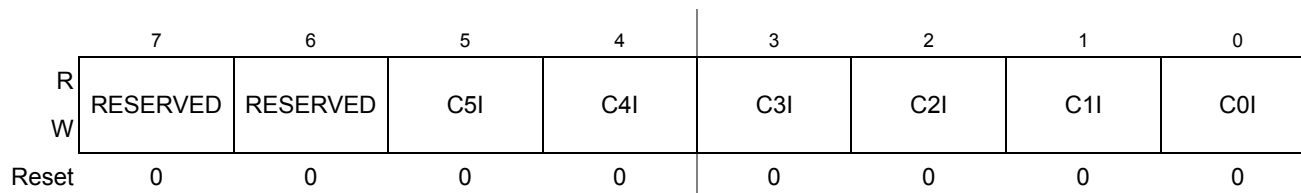


Figure 15-14. Timer Interrupt Enable Register (TIE)

Read: Anytime

Write: Anytime.

Table 15-10. TIE Field Descriptions

Note: Writing to unavailable bits has no effect. Reading from unavailable bits return a zero

Field	Description
5:0 C5I:C0I	Input Capture/Output Compare “x” Interrupt Enable — The bits in TIE correspond bit-for-bit with the bits in the TFLG1 status register. If cleared, the corresponding flag is disabled from causing a hardware interrupt. If set, the corresponding flag is enabled to cause a interrupt.

15.3.2.9 Timer System Control Register 2 (TSCR2)

Module Base + 0x000D

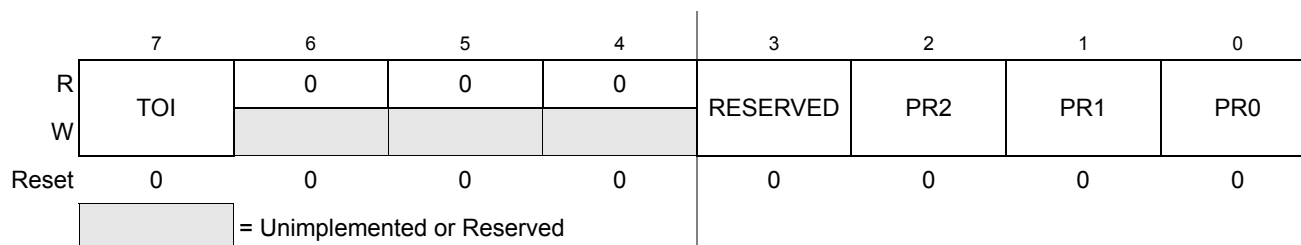


Figure 15-15. Timer System Control Register 2 (TSCR2)

Read: Anytime

Write: Anytime.

Table 15-11. TSCR2 Field Descriptions

Field	Description
7 TOI	Timer Overflow Interrupt Enable 0 Interrupt inhibited. 1 Hardware interrupt requested when TOF flag set.
2:0 PR[2:0]	Timer Prescaler Select — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 15-12 .

Chapter 16

Timer Module (TIM16B2CV3)

Table 16-1. Revision History Table

V03.00	Jan. 28, 2009		Initial version
V03.01	Aug. 26, 2009	16.1.2/16-458 16.3.2.2/16-461 , 16.4.3/16-473	- Correct typo: TSCR ->TSCR1; - Correct typo: ECTxxx->TIMxxx - Add description, "a counter overflow when TTOV[7] is set", to be the condition of channel 7 override event. - Phrase the description of OC7M to make it more explicit
V03.02	Apri,12,2010	16.3.2.6/16-464 16.3.2.9/16-466 16.4.3/16-473	-update TCRE bit description
V03.03	Jan,14,2013		-single source generate different channel guide

16.1 Introduction

The basic scalable timer consists of a 16-bit, software-programmable counter driven by a flexible programmable prescaler.

This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform.

This timer could contain up to 2 input capture/output compare channels . The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays.

A full access for the counter registers or the input capture/output compare registers should take place in one clock cycle. Accessing high byte and low byte separately for all of these registers may not yield the same result as accessing them in one word.

16.1.1 Features

The TIM16B2CV3 includes these distinctive features:

- Up to 2 channels available. (refer to device specification for exact number)
- All channels have same input capture/output compare functionality.
- Clock prescaling.
- 16-bit counter.

17.3.2.10 PWM Channel Counter Registers (PWMCNTx)

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

Any value written to the counter causes the counter to reset to \$00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see [Section 17.4.2.5, “Left Aligned Outputs”](#) and [Section 17.4.2.6, “Center Aligned Outputs”](#) for more details). When the channel is disabled ($PWMEx = 0$), the PWMCNTx register does not count. When a channel becomes enabled ($PWMEx = 1$), the associated PWM counter starts at the count in the PWMCNTx register. For more detailed information on the operation of the counters, see [Section 17.4.2.4, “PWM Timer Counters”](#).

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

NOTE

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

Module Base + 0x000C = PWMCNT0, 0x000D = PWMCNT1, 0x000E = PWMCNT2, 0x000F = PWMCNT3
Module Base + 0x0010 = PWMCNT4, 0x0011 = PWMCNT5, 0x0012 = PWMCNT6, 0x0013 = PWMCNT7

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

Figure 17-12. PWM Channel Counter Registers (PWMCNTx)

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 (any value written causes PWM counter to be reset to \$00).

17.3.2.11 PWM Channel Period Registers (PWMPERx)

There is a dedicated period register for each channel. The value in this register determines the period of the associated PWM channel.

The period 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)
- The channel is disabled

Table 18-12. SCISR2 Field Descriptions (continued)

Field	Description
3 RXPOL	Receive Polarity — This bit control the polarity of the received data. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity. 0 Normal polarity 1 Inverted polarity
2 BRK13	Break Transmit Character Length — This bit determines whether the transmit break character is 10 or 11 bit respectively 13 or 14 bits long. The detection of a framing error is not affected by this bit. 0 Break character is 10 or 11 bit long 1 Break character is 13 or 14 bit long
1 TXDIR	Transmitter Pin Data Direction in Single-Wire Mode — This bit determines whether the TXD pin is going to be used as an input or output, in the single-wire mode of operation. This bit is only relevant in the single-wire mode of operation. 0 TXD pin to be used as an input in single-wire mode 1 TXD pin to be used as an output in single-wire mode
0 RAF	Receiver Active Flag — RAF is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character. 0 No reception in progress 1 Reception in progress

18.3.2.9 SCI Data Registers (SCIDRH, SCIDRL)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	R8	T8	0	0	0	Reserved	Reserved	Reserved
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 18-12. SCI Data Registers (SCIDRH)

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Figure 18-13. SCI Data Registers (SCIDRL)

Read: Anytime; reading accesses SCI receive data register

Write: Anytime; writing accesses SCI transmit data register; writing to R8 has no effect

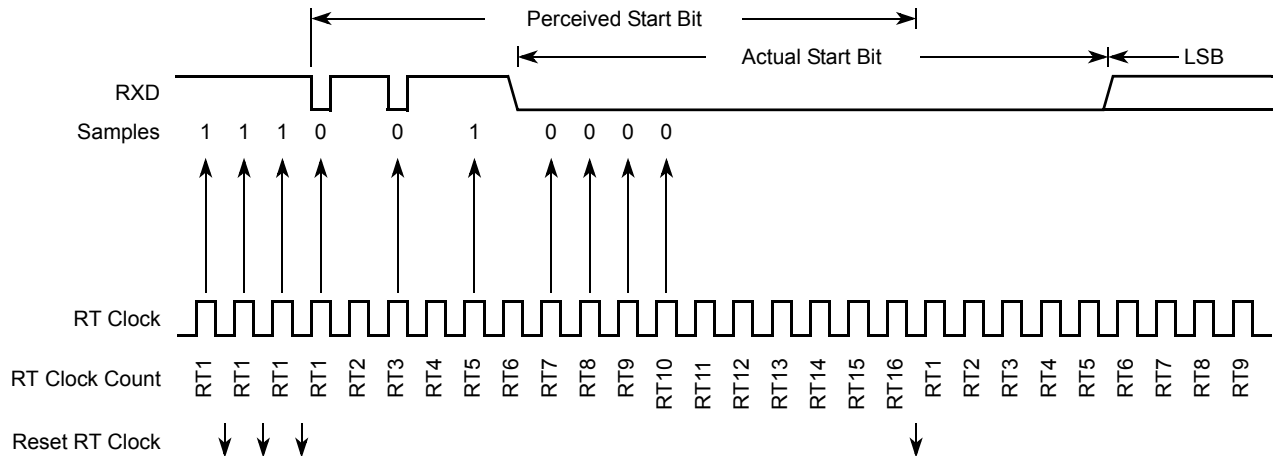


Figure 18-24. Start Bit Search Example 3

Figure 18-25 shows the effect of noise early in the start bit time. Although this noise does not affect proper synchronization with the start bit time, it does set the noise flag.

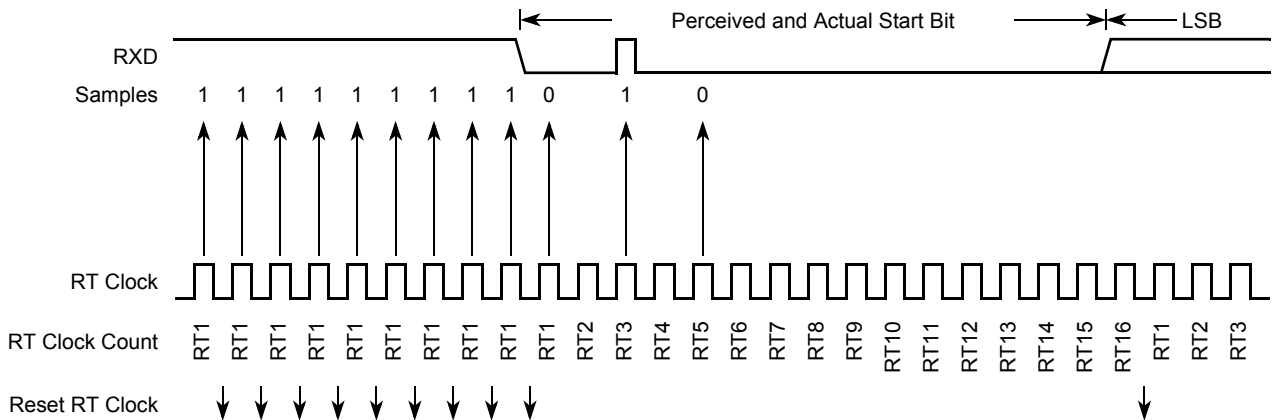


Figure 18-25. Start Bit Search Example 4

Figure 18-26 shows a burst of noise near the beginning of the start bit that resets the RT clock. The sample after the reset is low but is not preceded by three high samples that would qualify as a falling edge. Depending on the timing of the start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

```

MASTX      TST      TXCNT      ;GET VALUE FROM THE TRANSMITING COUNTER
           BEQ      END        ;END IF NO MORE DATA
           BRSET    IBSR,#$01,END ;END IF NO ACK
           MOVB     DATABUF,IBDR ;TRANSMIT NEXT BYTE OF DATA
           DEC      TXCNT      ;DECREASE THE TXCNT
           BRA      EMASTX     ;EXIT
END         BCLR     IBCR,#$20  ;GENERATE A STOP CONDITION
EMASTX     RTI          ;RETURN FROM INTERRUPT

```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data which can be done by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. The following is an example showing how a STOP signal is generated by a master receiver.

```

MASR      DEC      RXCNT      ;DECREASE THE RXCNT
           BEQ      ENMASR     ;LAST BYTE TO BE READ
           MOVB     RXCNT,D1   ;CHECK SECOND LAST BYTE
           DEC      D1        ;TO BE READ
           BNE      NXMAR      ;NOT LAST OR SECOND LAST
LAMAR     BSET      IBCR,#$08  ;SECOND LAST, DISABLE ACK
                               ;TRANSMITTING
           BRA      NXMAR
ENMASR    BCLR     IBCR,#$20   ;LAST ONE, GENERATE 'STOP' SIGNAL
NXMAR     MOVB     IBDR,RXBUF  ;READ DATA AND STORE
           RTI

```

20.7.1.5 Generation of Repeated START

At the end of data transfer, if the master continues to want to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. A program example is as shown.

```

RESTART   BSET      IBCR,#$04  ;ANOTHER START (RESTART)
           MOVB     CALLING,IBDR ;TRANSMIT THE CALLING ADDRESS;D0=R/W

```

20.7.1.6 Slave Mode

In the slave interrupt service routine, the module addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the transmit/receive mode select bit (Tx/Rx bit of IBCR) according to the R/W command bit (SRW). Writing to the IBCR clears the IAAS automatically. Note that the only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred, interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer may now be initiated by writing information to IBDR, for slave transmits, or dummy reading from IBDR, in slave receive mode. The slave will drive SCL low in-between byte transfers, SCL is released when the IBDR is accessed in the required mode.

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. Setting RXAK means an 'end of data' signal from the master receiver, after which it must be switched from transmitter mode to receiver mode by software. A dummy read then releases the SCL line so that the master can generate a STOP signal.

Table 22-16. FSTAT Field Descriptions (continued)

Field	Description
2 RSVD	Reserved Bit — This bit is reserved and always reads 0.
1–0 MGSTAT[1:0]	Memory Controller Command Completion Status Flag — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. The MGSTAT bits are cleared automatically at the start of the execution of a Flash command. See Section 22.4.7, “Flash Command Description,” and Section 22.6, “Initialization” for details.

22.3.2.8 Flash Error Status Register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

Offset Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	DFDF	SFDIF
W								
Reset	0	0	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 22-12. Flash Error Status Register (FERSTAT)

All flags in the FERSTAT register are readable and only writable to clear the flag.

Table 22-17. FERSTAT Field Descriptions

Field	Description
1 DFDF	Double Bit Fault Detect Flag — The setting of the DFDF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation. ¹ The DFDF flag is cleared by writing a 1 to DFDF. Writing a 0 to DFDF has no effect on DFDF. ² 0 No double bit fault detected 1 Double bit fault detected or a Flash array read operation returning invalid data was attempted while command running. See Section 22.4.3, “Flash Block Read Access” for details
0 SFDIF	Single Bit Fault Detect Interrupt Flag — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation. The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or a Flash array read operation returning invalid data was attempted while command running

¹ In case of ECC errors the corresponding flag must be cleared for the proper setting of any further error, i.e. any new error will only be indicated properly when DFDF and/or SFDIF are clear at the time the error condition is detected.

² There is a one cycle delay in storing the ECC DFDF and SFDIF fault flags in this register. At least one NOP is required after a flash memory read before checking FERSTAT for the occurrence of ECC errors.

O.10 0x0500-x052F PWM1 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0519	PWMPER5	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051A	PWMPER6	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051B	PWMPER7	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051C	PWMDTY0	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051D	PWMDTY1	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051E	PWMDTY2	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x051F	PWMDTY32	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0520	PWMDTY42	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0521	PWMDTY52	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0522	PWMDTY62	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0523	PWMDTY72	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0524 - 0x052F	RESERVED	R W	0	0	0	0	0	0	0	0

O.11 0x05C0-0x05FF TIM0

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x05C0	TIM0TIOS	R W	RESERVE D	RESERVE D	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
0x05C1	TIM0CFORC	R W	RESERVE D	RESERVE D	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x05C2- 0x05C3	Reserved	R W								
0x05C4	TIM0TCNTH	R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8