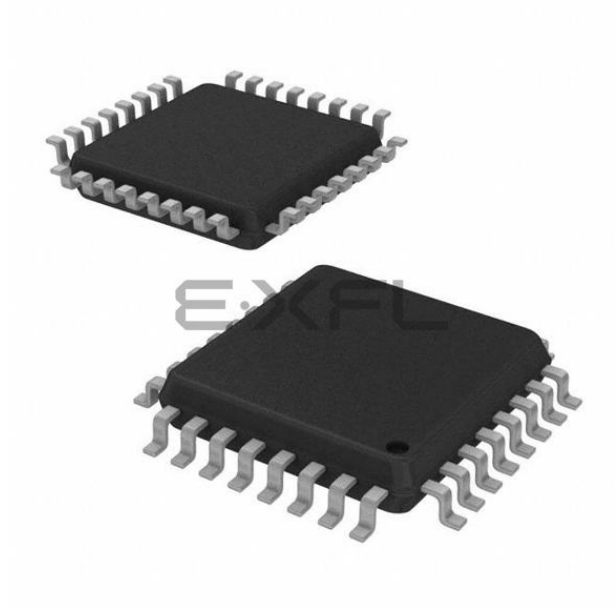


Welcome to E-XFL.COM

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	I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	32KB (32K 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
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12zvl32f0clcr

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.

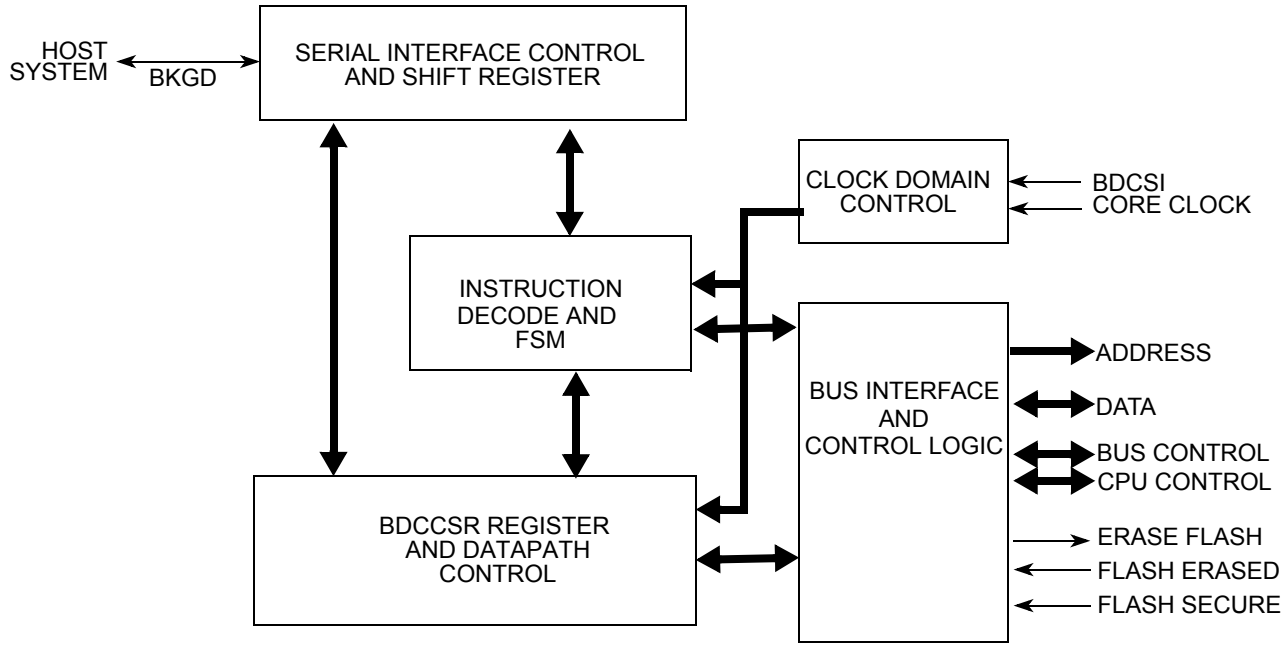


Figure 5-1. BDC Block Diagram

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.

5.4.5.2.2 READ_SAME Effects Of Variable Access Size

READ_SAME uses the unadjusted address given in the previous READ_MEM command as a base address for subsequent READ_SAME commands. When the READ_MEM and READ_SAME size parameters differ then READ_SAME uses the original base address buts aligns 32-bit and 16-bit accesses, where those accesses would otherwise cross the aligned 4-byte boundary. Table 5-12 shows some examples of this.

Table 5-12. Consecutive READ_SAME Accesses With Variable Size

Row	Command	Base Address	00	01	10	11
1	READ_MEM.32	0x004003	Accessed	Accessed	Accessed	Accessed
2	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
3	READ_SAME.16	—			Accessed	Accessed
4	READ_SAME.08	—				Accessed
5	READ_MEM.08	0x004000	Accessed			
6	READ_SAME.08	—	Accessed			
7	READ_SAME.16	—	Accessed	Accessed		
8	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
9	READ_MEM.08	0x004002			Accessed	
10	READ_SAME.08	—			Accessed	
11	READ_SAME.16	—			Accessed	Accessed
12	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
13	READ_MEM.08	0x004003				Accessed
14	READ_SAME.08	—				Accessed
15	READ_SAME.16	—			Accessed	Accessed
16	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
17	READ_MEM.16	0x004001		Accessed	Accessed	
18	READ_SAME.08	—		Accessed		
19	READ_SAME.16	—		Accessed	Accessed	
20	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
21	READ_MEM.16	0x004003			Accessed	Accessed
22	READ_SAME.08	—				Accessed
23	READ_SAME.16	—			Accessed	Accessed
24	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed

5.4.6 BDC Serial Interface

The BDC communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDC.

The BDC serial interface uses an internal clock source, selected by the CLKSW bit in the BDCCSR register. This clock is referred to as the target clock in the following explanation.

6.3.2.2 Interrupt Request Configuration Address Register (INT_CFADDR)

Address: 0x000017

	7	6	5	4	3	2	1	0
R	0	INT_CFADDR[6:3]				0	0	0
W								
Reset	0	0	0	0	1	0	0	0

= Unimplemented or Reserved

Figure 6-4. Interrupt Configuration Address Register (INT_CFADDR)

Read: Anytime

Write: Anytime

Table 6-5. INT_CFADDR Field Descriptions

Field	Description
6–3 INT_CFADDR[6:3]	**Interrupt Request Configuration Data Register Select Bits** — These bits determine which of the 128 configuration data registers are accessible in the 8 register window at INT_CFDATA0–7. The hexadecimal value written to this register corresponds to the upper 4 bits of the vector number (multiply with 4 to get the vector address offset). If, for example, the value 0x70 is written to this register, the configuration data register block for the 8 interrupt vector requests starting with vector at address (vector base + (0x70*4 = 0x0001C0)) is selected and can be accessed as INT_CFDATA0–7.

6.3.2.3 Interrupt Request Configuration Data Registers (INT_CFDATA0–7)

The eight register window visible at addresses INT_CFDATA0–7 contains the configuration data for the block of eight interrupt requests (out of 128) selected by the interrupt configuration address register (INT_CFADDR) in ascending order. INT_CFDATA0 represents the interrupt configuration data register of the vector with the lowest address in this block, while INT_CFDATA7 represents the interrupt configuration data register of the vector with the highest address, respectively.

Address: 0x000018

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	[11]

= Unimplemented or Reserved

Figure 6-5. Interrupt Request Configuration Data Register 0 (INT_CFDATA0)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

10.5.2.4 ADC Timing Register (ADCTIM)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	PRS[6:0]						
W								
Reset	0	0	0	0	0	1	0	1

= Unimplemented or Reserved

Figure 10-7. ADC Timing Register (ADCTIM))

Read: Anytime

Write: These bits are writable if bit ADC_EN is clear or bit SMOD_ACC is set

Table 10-7. ADCTIM Field Descriptions

Field	Description
6-0 PRS[6:0]	**ADC Clock Prescaler** — These 7bits are the binary prescaler value PRS. The ADC conversion clock frequency is calculated as follows: $f_{ATDCLK} = \frac{f_{BUS}}{2 \times (PRS + 1)}$ Refer to Device Specification for allowed frequency range of f_{ATDCLK}.

10.5.2.5 ADC Format Register (ADCFMT)

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	DJM	0	0	0	0	SRES[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-8. ADC Format Register (ADCFMT)

Read: Anytime

Write: Bits DJM and SRES[2:0] are writable if bit ADC_EN clear or bit SMOD_ACC set

Table 10-8. ADCFMT Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — Conversion result data format is always unsigned. This bit controls justification of conversion result data in the conversion result list. 0 Left justified data in the conversion result list. 1 Right justified data in the conversion result list.
2-0 SRES[2:0]	**ADC Resolution Select** — These bits select the resolution of conversion results. See Table 10-9 for coding.

Table 10-9. Selectable Conversion Resolution

SRES[2]	SRES[1]	SRES[0]	ADC Resolution
0	0	0	8-bit data
0	0	1	Reserved[1]
0	1	0	10-bit data
0	1	1	Reserved[1]
1	0	0	12-bit data
1	x	x	Reserved[1]

1 Reserved settings cause a severe error at ADC conversion start whereby the CMD_EIF flag is set and ADC ceases operation

10.5.2.12 ADC Conversion Interrupt Flag Register (ADCCONIF)

After being set any of these bits can be cleared by writing a value of 1’b1. All bits are cleared if bit ADC_EN is clear or via ADC soft-reset (bit ADC_SR set). Writing any flag with value 1’b0 does not clear the flag. Writing any flag with value 1’b1 does not set the flag.

Module Base + 0x000C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CON_IF[15:1]															EOL_IF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 10-15. ADC Conversion Interrupt Flag Register (ADCCONIF)

Read: Anytime

Write: Anytime

Table 10-17. ADCCONIF Field Descriptions

Field	Description
15-1 CON_IF[15:1]	**Conversion Interrupt Flags** — These bits could be set by the binary coded interrupt select bits INTFLG_SEL[3:0] when the corresponding conversion command has been processed and related data has been stored to RAM. See also notes below.
0 EOL_IF	**End Of List Interrupt Flag** — This bit is set by the binary coded conversion command type select bits CMD_SEL[1:0] for “end of list” type of commands and after such a command has been processed and the related data has been stored RAM. See also second note below

NOTE

These bits can be used to indicate if a certain packet of conversion results is available. Clearing a flag indicates that conversion results have been retrieved by software and the flag can be used again (see also Section 10.9.6, “RVL swapping in RVL double buffer mode and related registers ADCIMDRI and ADCEOLRI.

NOTE

Overrun situation of a flag CON_IF[15:1] and EOL_IF are indicated by flag CONIF_OIF.

Table 10-24. Analog Input Channel Select

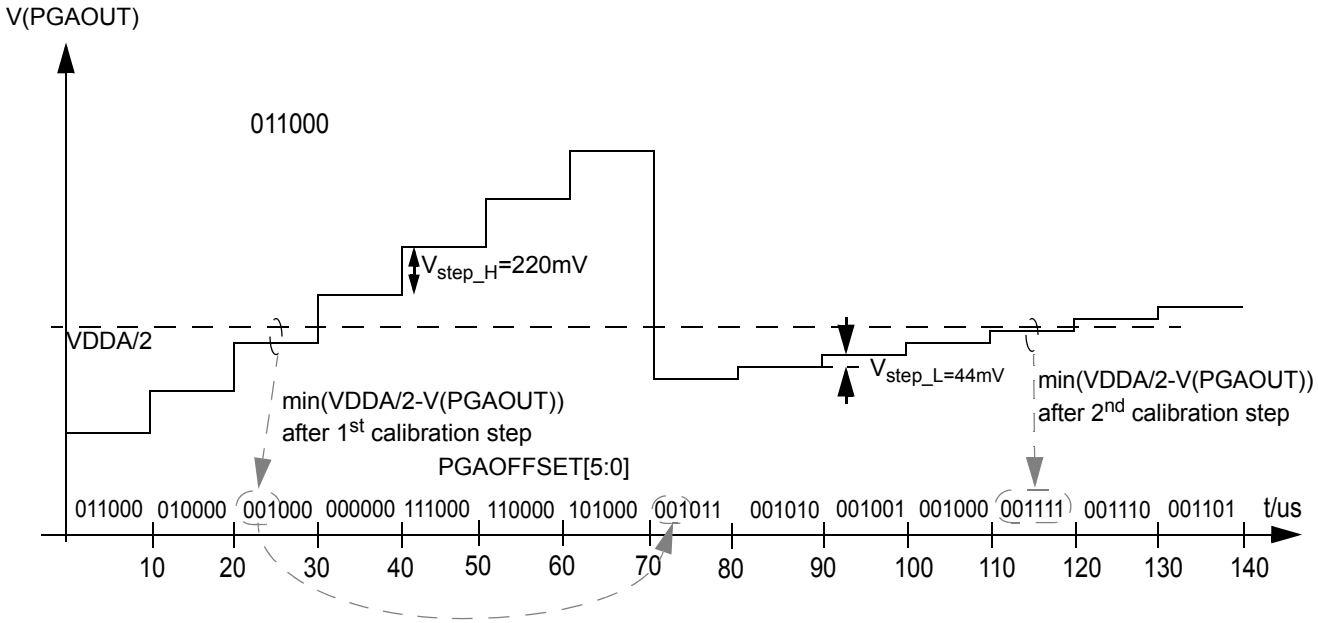
CH_SEL[5]	CH_SEL[4]	CH_SEL[3]	CH_SEL[2]	CH_SEL[1]	CH_SEL[0]	Analog Input Channel
0	0	0	0	0	0	VRL_0/1 (V1, V2, see Table 10-2) VRL_0 (V3, see Table 10-2)
0	0	0	0	0	1	VRH_0/1 (V1, V2, see Table 10-2) VRH_0/1/2 (V3, see Table 10-2)
0	0	0	0	1	0	(VRH_0/1 + VRL_0/1) / 2 (V1, V2, see Table 10-2) (VRH_0/1/2 + VRL_0) / 2 (V3, see Table 10-2)
0	0	0	0	1	1	Reserved
0	0	0	1	0	0	Reserved
0	0	0	1	0	1	Reserved
0	0	0	1	1	0	Reserved
0	0	0	1	1	1	Reserved
0	0	1	0	0	0	Internal_0 (ADC temperature sense)
0	0	1	0	0	1	Internal_1
0	0	1	0	1	0	Internal_2
0	0	1	0	1	1	Internal_3
0	0	1	1	0	0	Internal_4
0	0	1	1	0	1	Internal_5
0	0	1	1	1	0	Internal_6
0	0	1	1	1	1	Internal_7
0	1	0	0	0	0	AN0
0	1	0	0	0	1	AN1
0	1	0	0	1	0	AN2
0	1	0	0	1	1	AN3
0	1	0	1	0	0	AN4
0	1	x	x	x	x	ANx
1	x	x	x	x	x	Reserved

NOTE

ANx in Table 10-24 is the maximum number of implemented analog input channels on the device. Please refer to the device overview of the reference manual for details regarding number of analog input channels.

2. With the optimal PGAOFFSET[5:3] setting step through the offset compensation values PGAOFFSET[2:0]= {0x011, 0x010, 0x001, 0x000, 0x111, 0x110, 0x101} and measure the PGA_OUT value with the ADC. Select as optimal offset compensation value for the lower three bits the PGAOFFSET[2:0] which is closest to the expected ADC reading of VDDA/2.

Figure 12-8. Offset compensation timing diagram



12.4.3 Application Example for differential voltage measurement

For sensor applications it is often required to measure a small differential voltage V_{diff}. The PGA is not capable of amplifying a differential voltage, but an algorithm to calculate the differential voltage can be implemented. The PGA contains two input pins PGA_IN0 and PGA_IN1 which can be multiplexed by the ADC command list. By subtracting the ADC readings of the two pins the amplified differential voltage can be calculated.

For this algorithm two requirements must be met:

1. The minimum time for the input signal multiplexing is given by PGA to ADC settling time $t_{PGA\_settling}$. The rate of signal change within $t_{PGA\_settling}$ must be small.
2. The common mode input voltage range of the differential input signals must be limited that for a given gain A_{PGA} a reference voltage V_{ref} can be selected so that both amplified signals do not saturate.

If both requirements are met the algorithm can be implemented. The error calculation is the following:

$$V_{diff} = (ADC\_reading(PGA\_IN1) - ADC\_reading(PGA\_IN0))/A_{PGA} \qquad \textit{Eqn. 12-1}$$

13.1.3 Features

The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol — Version 2.0A/B
  - — Standard and extended data frames
  - — Zero to eight bytes data length
  - — Programmable bit rate up to 1 Mbps[1]
  - — Support for remote frames
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a “local priority” concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter
- Programmable loopback mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (warning, error passive, bus-off)
- Programmable MSCAN clock source either bus clock or oscillator clock
- Internal timer for time-stamping of received and transmitted messages
- Three low-power modes: sleep, power down, and MSCAN enable
- Global initialization of configuration registers

13.1.4 Modes of Operation

For a description of the specific MSCAN modes and the module operation related to the system operating modes refer to Section 13.4.4, “Modes of Operation”.

1. Depending on the actual bit timing and the clock jitter of the PLL.

Table 13-25. 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 13-23 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 13-24.

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.

13.4.5.7 Disabled Mode

The MSCAN is in disabled mode out of reset (CANE=0). All module clocks are stopped for power saving, however the register map can still be accessed as specified.

13.4.5.8 Programmable Wake-Up Function

The MSCAN can be programmed to wake up from sleep or power down mode as soon as CAN bus activity is detected (see control bit WUPE in MSCAN Control Register 0 (CANCTL0). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line (see control bit WUPM in Section 13.3.2.2, “MSCAN Control Register 1 (CANCTL1)”).

This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

13.4.6 Reset Initialization

The reset state of each individual bit is listed in Section 13.3.2, “Register Descriptions,” which details all the registers and their bit-fields.

13.4.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.

13.4.7.1 Description of Interrupt Operation

The MSCAN supports four interrupt vectors (see Table 13-38), any of which can be individually masked (for details see Section 13.3.2.6, “MSCAN Receiver Interrupt Enable Register (CANRIER)” to Section 13.3.2.8, “MSCAN Transmitter Interrupt Enable Register (CANTIER)”).

Refer to the device overview section to determine the dedicated interrupt vector addresses.

Table 13-38. Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Wake-Up Interrupt (WUPIF)	I bit	CANRIER (WUPIE)
Error Interrupts Interrupt (CSCIF, OVRIF)	I bit	CANRIER (CSCIE, OVRIE)
Receive Interrupt (RXF)	I bit	CANRIER (RXFIE)
Transmit Interrupts (TXE[2:0])	I bit	CANTIER (TXEIE[2:0])

13.4.7.2 Transmit Interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

14.3.2.3 BATSV3 Interrupt Enable Register (BATIE)

Module Base + 0x0002 — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	BVHIE	BVLIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-6. BATSV3 Interrupt Enable Register (BATIE)

[1] Read: Anytime
Write: Anytime

Table 14-4. BATIE Register Field Descriptions

Field	Description
1 BVHIE	**BATS Interrupt Enable High** — Enables High Voltage Interrupt **.** 0 No interrupt will be requested whenever BVHIF flag is set . 1 Interrupt will be requested whenever BVHIF flag is set
0 BVLIE	**BATS Interrupt Enable Low** — Enables Low Voltage Interrupt . 0 No interrupt will be requested whenever BVLIF flag is set . 1 Interrupt will be requested whenever BVLIF flag is set .

14.3.2.4 BATSV3 Interrupt Flag Register (BATIF)

Module Base + 0x0003 — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	BVHIF	BVLIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-7. BATSV3 Interrupt Flag Register (BATIF)

[1] Read: Anytime
Write: Anytime, write 1 to clear

If measured when

a) V_{HBI1} selected with BVHS = 0
$V_{measure} \geq V_{HBI1\_A}$ (rising edge) or $V_{measure} \geq V_{HBI1\_D}$ (falling edge)

or when

a) V_{HBI2} selected with BVHS = 1
$V_{measure} \geq V_{HBI2\_A}$ (rising edge) or $V_{measure} \geq V_{HBI2\_D}$ (falling edge)

then BVHC is set. BVHC status bit indicates that a high voltage at pin VSUP is present. The High Voltage Interrupt flag (BVHIF) is set to 1 when a Voltage High Condition (BVHC) changes state. The Interrupt flag BVHIF can only be cleared by writing a 1. If the interrupt is enabled by bit BVHIE the module requests an interrupt to MCU (BATI).

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 17.4.2.3, “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 17.4.2.8, “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 17-13. 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

17.3.2.12 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)
- The channel is disabled

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 19.4.3, “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.

19.4.1 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 19.3.2.2, “SPI Control Register 2 (SPICR2)

20.4.1.5 Repeated START Signal

As shown in Figure 20-10, a repeated START signal is a START signal generated without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in different mode (transmit/receive mode) without releasing the bus.

20.4.1.6 Arbitration Procedure

The Inter-IC bus is a true multi-master bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. The relative priority of the contending masters is determined by a data arbitration procedure, a bus master loses arbitration if it transmits logic 1 while another master transmits logic 0. The losing masters immediately switch over to slave receive mode and stop driving SDA output. In this case the transition from master to slave mode does not generate a STOP condition. Meanwhile, a status bit is set by hardware to indicate loss of arbitration.

20.4.1.7 Clock Synchronization

Because wire-AND logic is performed on SCL line, a high-to-low transition on SCL line affects all the devices connected on the bus. The devices start counting their low period and as soon as a device's clock has gone low, it holds the SCL line low until the clock high state is reached.However, the change of low to high in this device clock may not change the state of the SCL line if another device clock is within its low period. Therefore, synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see Figure 20-11). When all devices concerned have counted off their low period, the synchronized clock SCL line is released and pulled high. There is then no difference between the device clocks and the state of the SCL line and all the devices start counting their high periods.The first device to complete its high period pulls the SCL line low again.

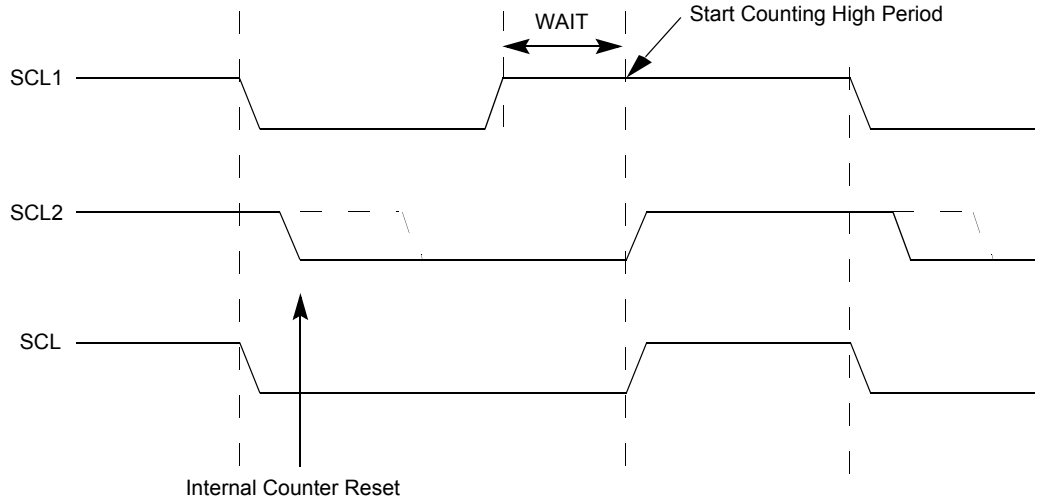


Figure 20-12. IIC-Bus Clock Synchronization

```
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.

21.2.2 LGND — LIN Ground Pin

This pin is the device LIN ground connection. It is used to sink currents related to the LIN Bus pin. A de-coupling capacitor external to the device (typically 220 pF, X7R ceramic) between LIN and LGND can further improve the quality of this ground and filter noise.

21.2.3 VLINSUP — Positive Power Supply

External power supply to the chip. The VLINSUP supply mapping is described in device level documentation.

21.2.4 LPTxD — LIN Transmit Pin

This pin can be routed to the SCI, LPDR1 register bit, an external pin, or other options. Please refer to the PIM chapter of the device specification for the available routing options.

This input is only used in normal mode; in other modes the value of this pin is ignored.

21.2.5 LPRxD — LIN Receive Pin

This pin can be routed to the SCI, an external pin, or other options. Please refer to the PIM chapter of the device specification for the available routing options.

In standby mode this output is disabled, and sends only a short pulse in case the wake-up functionality is enabled and a valid wake-up pulse was received in the LIN Bus.

21.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the LIN Physical Layer.

21.3.1 Module Memory Map

A summary of the registers associated with the S12LINPHYV2 module is shown in Table 21-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

NOTE

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

NOTE

Please refer to the temperature rating of the device with regards to the ambient temperature T_A and the junction temperature T_J. For power dissipation calculations refer to Section A.1.6, “Power Dissipation and Thermal Characteristics”.

Table A-6. Operating Conditions

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Voltage regulator and LINPHY supply voltage	V_{SUP}	5.5	12	40[1]	V
2	Voltage difference V_{DDX} to V_{DDA}	Δ_{VDDX}	-0.1	—	0.1	V
3	Voltage difference V_{SSX} to V_{SSA}	Δ_{VSSX}	-0.3	—	0.3	V
4	Oscillator	f_{osc}	4	—	20	MHz
5	Bus frequency[2] $T_J \leq 150°C$ $150°C < T_J < 175°C$ (option W only)	f_{bus}	[3]	—	 32 25	MHz
6	Bus frequency without wait states	f_{WSTAT}	—	—	25	MHz
7a	Operating junction temperature range Operating ambient temperature range[4] (option C)	T_J T_A	–40 –40	— —	105 85	°C
7b	Operating junction temperature range Operating ambient temperature range[4] (option V)	T_J T_A	–40 –40	— —	125 105	°C
7c	Operating junction temperature range Operating ambient temperature range[4] (option M)	T_J T_A	–40 –40	— —	150 125	°C
7d	Operating junction temperature range Operating ambient temperature range[4] (option W)	T_J T_A	–40 –40	— —	175 150	°C

1 Normal operating range is 5.5 V - 18 V. Continuous operation at 40 V is not allowed. Only Transient Conditions (Load Dump) single pulse t_{max}<400 ms. Operation down to 3.5V is guaranteed without reset, however some electrical parameters are specified only in the range above 4.5 V. Operation in the range 20V<VSUP<26.5V is limited to 1 hour over lifetime of the device. In this range the device continues to function but electrical parameters are degraded.

2 The flash program and erase operations must configure f_{NVMOP} as specified in the NVM electrical section.

3 Refer to f_{ATDCLK} for minimum ADC operating frequency. This is derived from the bus clock.

4 Please refer to Section A.1.6, “Power Dissipation and Thermal Characteristics” for more details about the relation between ambient temperature T_A and device junction temperature T_J.

NOTE

Operation is guaranteed when powering down until low voltage reset assertion.

Table F-1. Static Electrical Characteristics - Supply Voltage Sense - (BATS)

Characteristics noted under conditions 5.5V ≤ V_{SUP} ≤ 18V, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25°C[1] under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
4	Low Voltage Warning (LBI 4)					
	Ratio = 9					
	Assert (Measured on selected pin, falling edge)	$V_{LBI4\_A\_9}$	8	9	10	V
	Deassert (Measured on selected pin, rising edge)	$V_{LBI4\_D\_9}$	–	–	10.5	V
	Hysteresis (measured on selected pin)	$V_{LBI4\_H\_9}$	–	0.4	–	V
	Ratio = 17[2]					
	Assert (Measured on selected pin, falling edge)	$V_{LBI4\_A\_17}$	8	9.5	11.0	V
	Deassert (Measured on selected pin, rising edge)	$V_{LBI4\_D\_17}$	–	–	12.0	V
	Hysteresis (measured on selected pin)	$V_{LBI4\_H\_17}$	–	1.0	–	V
5	High Voltage Warning (HBI 1)					
	Ratio = 9 (VDDX > 4.5V)					
	Assert (Measured on selected pin, rising edge)	$V_{HBI1\_A\_9}$	14.5	16.5	18	V
	Deassert (Measured on selected pin, falling edge)	$V_{HBI1\_D\_9}$	14	–	–	V
	Hysteresis (measured on selected pin)	$V_{HBI1\_H\_9}$	–	1.0	–	V
	Ratio = 17[2]					
	Assert (Measured on selected pin, rising edge)	$V_{HBI1\_A\_17}$	14.5	16.5	18	V
	Deassert (Measured on selected pin, falling edge)	$V_{HBI1\_D\_17}$	12.5	–	–	V
	Hysteresis (measured on selected pin)	$V_{HBI1\_H\_17}$	–	2.0	–	V