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	50MHz
Connectivity	CANbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP Exposed Pad
Supplier Device Package	80-TQFP-EP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvmc25f1vkk

Table 0-1. Revision History

Date	Revision	Description
19 APR 2016	2.8	Added PAD pin leakage specification at 125°C Table A-12 Updated t_{HGON}, t_{HGOFF} parameter values Table E-1 Specified VRH drop when using VDDS1 or VDDS2 as VRH on ZVMC256 Section C.1.1.5 Added min. and max. desaturation comparator filter times to electrical spec. Table E-1 Updated 64LQFP-EP thermal parameters Table A-9, Table A-10
06 JUN 2016	2.9	Fixed corrupted symbol fonts Table A-3, Table A-5 Corrected wrong IFR reference Section 20.3.2.10 Clarified PAD8 leakage better Table A-12 Added I_{SUPR} and I_{SUPW} maximum values at T_J = 175°C for ZVMC256 Table A-18 Added Pseudo STOP maximum current for ZVMC256 Table A-20 Removed bandgap temperature dependency footnote, Table B-1 Changed ZVMC256 SNPS monitor threshold min/max values Table B-2 Changed VLS current limit threshold to 112mA Table E-1, Table E-2 Removed desaturation comparator filter times from GDU chapter. Added desaturation comparator levels to Table E-1, Table E-2 Added low side desaturation comparator functional range as footnote Table E-1, Table E-2
29 JUN 2016	2.10	Updated GDU VBS filter Figure 18-20 Removed incorrect reference to temperature sensor influencing GDU outputs Section 1.13.3.4 Changed Stop IDD (ISUPS) specifications for ZVMC256 Table A-19
28 OCT 2016	2.11	Added IOC0 signal mapping to 48LQFP package Figure 1-6 Fixed corrupted symbol fonts in PIM chapter Added diode to VDDC pin Figure 1-18 Updated Stop mode current ISUPS maximum values Table A-19 Updated tdelon, tdeloff values Table E-1

Table 3-6. MMCCCRH and MMCCCRL Field Descriptions

Field	Description
7 (MMCCCRH) CPUU	**S12ZCPU User State Flag** — This bit shows the state of the user/supervisor mode bit in the S12ZCPU's CCR at the time the access violation has occurred. The S12ZCPU user state flag is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC register. This bit is undefined if the error code registers (MMCECn) are cleared.
6 (MMCCCRL) CPUX	**S12ZCPU X-Interrupt Mask**— This bit shows the state of the X-interrupt mask in the S12ZCPU's CCR at the time the access violation has occurred. The S12ZCPU X-interrupt mask is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC register. This bit is undefined if the error code registers (MMCECn) are cleared.
4 (MMCCCRL) CPUI	**S12ZCPU I-Interrupt Mask**— This bit shows the state of the I-interrupt mask in the CPU's CCR at the time the access violation has occurred. The S12ZCPU I-interrupt mask is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC register. This bit is undefined if the error code registers (MMCECn) are cleared.

3.3.2.4 Captured S12ZCPU Program Counter (MMCPCH, MMCPCM, MMCPCL)

Address: 0x0085 (MMCPCH)

	7	6	5	4	3	2	1	0
R	CPUPC[23:16]							
W								
Reset	0	0	0	0	0	0	0	0

Address: 0x0086 (MMCPCM)

	7	6	5	4	3	2	1	0
R	CPUPC[15:8]							
W								
Reset	0	0	0	0	0	0	0	0

Address: 0x0087 (MMCPCL)

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

Figure 3-7. Captured S12ZCPU Program Counter (MMCPCH, MMCPCM, MMCPCL)

Read: Anytime

Write: Never

Table 3-7. MMCPCH, MMCPCM, and MMCPCL Field Descriptions

Field	Description
7–0 (MMCPCH) 7–0 (MMCPCM) 7–0 (MMCPCL) CPUPC[23:0]	**S12ZCPU Program Counter Value**— The CPUPC[23:0] stores the CPU's program counter value at the time the access violation occurred. CPUPC[23:0] always points to the instruction which triggered the violation. These bits are undefined if the error code registers (MMCECn) are cleared.

3.4 Functional Description

This section provides a complete functional description of the S12ZMMC module.

3.4.1 Global Memory Map

The S12ZMMC maps all on-chip resources into an 16MB address space, the global memory map. The exact resource mapping is shown in Figure 3-8. The global address space is used by the S12ZCPU, ADCs, PTU, and the S12ZBDC module.

5.1.3.3 Low-Power Modes

5.1.3.3.1 Stop Mode

The execution of the CPU STOP instruction leads to stop mode only when all bus masters (CPU, or others, depending on the device) have finished processing. The operation during stop mode depends on the ENBDC and BDCCIS bit settings as summarized in Table 5-3

Table 5-3. BDC STOP Operation Dependencies

ENBDC	BDCCIS	Description Of Operation
0	0	BDC has no effect on STOP mode.
0	1	BDC has no effect on STOP mode.
1	0	Only BDCCLK clock continues
1	1	All clocks continue

A disabled BDC has no influence on stop mode operation. In this case the BDCSI clock is disabled in stop mode thus it is not possible to enable the BDC from within stop mode.

STOP Mode With BDC Enabled And BDCCIS Clear

If the BDC is enabled and BDCCIS is clear, then the BDC prevents the BDCCLK clock (Figure 5-5) from being disabled in stop mode. This allows BDC communication to continue throughout stop mode in order to access the BDCCSR register. All other device level clock signals are disabled on entering stop mode.

NOTE

This is intended for application debugging, not for fast flash programming.
Thus the CLKSW bit must be clear to map the BDCSI to BDCCLK.

With the BDC enabled, an internal acknowledge delays stop mode entry and exit by 2 BDCSI clock + 2 bus clock cycles. If no other module delays stop mode entry and exit, then these additional clock cycles represent a difference between the debug and not debug cases. Furthermore if a BDC internal access is being executed when the device is entering stop mode, then the stop mode entry is delayed until the internal access is complete (typically for 1 bus clock cycle).

Accesses to the internal memory map are not possible when the internal device clocks are disabled. Thus attempted accesses to memory mapped resources are suppressed and the NORESP flag is set. Resources can be accessed again by the next command received following exit from Stop mode.

A BACKGROUND command issued whilst in stop mode remains pending internally until the device leaves stop mode. This means that subsequent active BDM commands, issued whilst BACKGROUND is pending, set the ILLCMD flag because the device is not yet in active BDM.

If ACK handshaking is enabled, then the first ACK, following a stop mode entry is long to indicate a stop exception. The BDC indicates a stop mode occurrence by setting the BDCCSR bit STOP. If the host attempts further communication before the ACK pulse generation then the OVRUN bit is set.

STOP Mode With BDC Enabled And BDCCIS Set

If the BDC is enabled and BDCCIS is set, then the BDC prevents core clocks being disabled in stop mode. This allows BDC communication, for access of internal memory mapped resources, but not CPU registers, to continue throughout stop mode.

A BACKGROUND command issued whilst in stop mode remains pending internally until the device leaves stop mode. This means that subsequent active BDM commands, issued whilst BACKGROUND is pending, set the ILLCMD flag because the device is not yet in active BDM.

If ACK handshaking is enabled, then the first ACK, following a stop mode entry is long to indicate a stop exception. The BDC indicates a stop mode occurrence by setting the BDCCSR bit STOP. If the host attempts further communication before the ACK pulse generation then the OVRUN bit is set.

5.1.3.3.2 Wait Mode

The device enters wait mode when the CPU starts to execute the WAI instruction. The second part of the WAI instruction (return from wait mode) can only be performed when an interrupt occurs. Thus on entering wait mode the CPU is in the middle of the WAI instruction and cannot permit access to CPU internal resources, nor allow entry to active BDM. Thus only commands classified as Non-Intrusive or Always-Available are possible in wait mode.

On entering wait mode, the WAIT flag in BDCCSR is set. If the ACK handshake protocol is enabled then the first ACK generated after WAIT has been set is a long-ACK pulse. Thus the host can recognize a wait mode occurrence. The WAIT flag remains set and cannot be cleared whilst the device remains in wait mode. After the device leaves wait mode the WAIT flag can be cleared by writing a “1” to it.

A BACKGROUND command issued whilst in wait mode sets the NORESP bit and the BDM active request remains pending internally until the CPU leaves wait mode due to an interrupt. The device then enters BDM with the PC pointing to the address of the first instruction of the ISR.

With ACK disabled, further Non-Intrusive or Always-Available commands are possible, in this pending state, but attempted Active-Background commands set NORESP and ILLCMD because the BDC is not in active BDM state.

With ACK enabled, if the host attempts further communication before the ACK pulse generation then the OVRUN bit is set.

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.

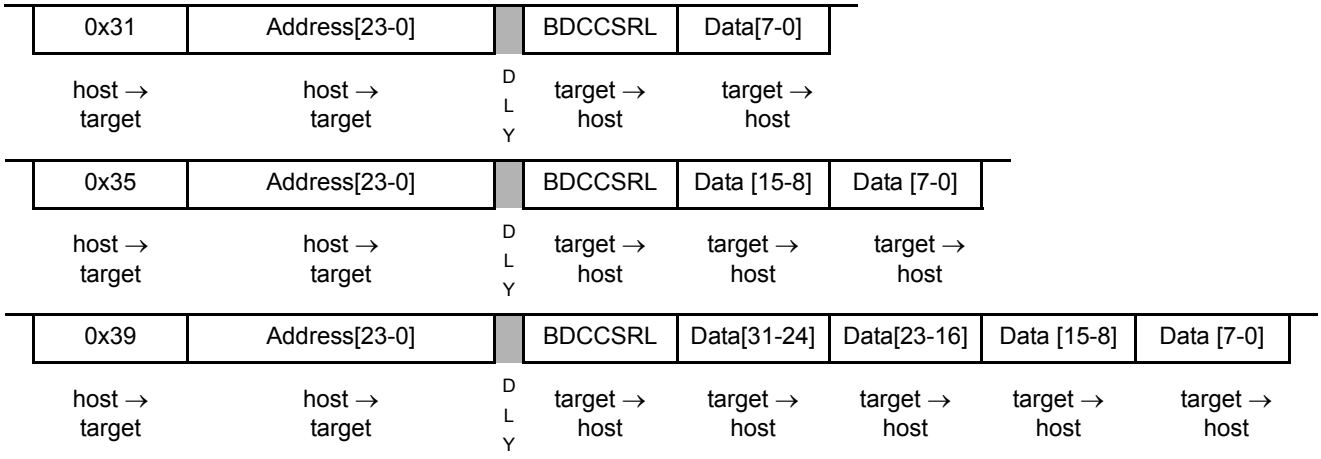
If the part is still in Wait mode and a further STEP1 is carried out then the NORESP and ILLCMD bits are set because the device is no longer in active BDM for the duration of WAI execution.

5.1.4 Block Diagram

A block diagram of the BDC is shown in Figure 5-1.

READ_MEM.sz_WS

Read memory at the specified address with status **Non-intrusive**



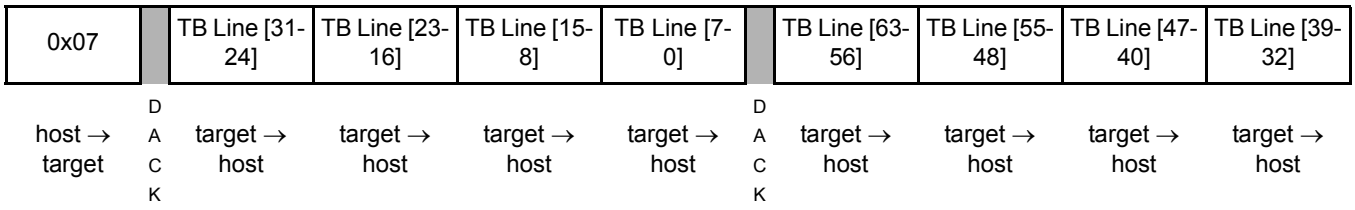
Read data at the specified memory address. The address is transmitted as three 8-bit packets (msb to lsb) immediately after the command.

The hardware forces low-order address bits to zero longword accesses to ensure these accesses are on 0-modulo-size alignments. Byte alignment details are described in Section 5.4.5.2”. If the with-status option is specified, the BDCCSR status byte is returned before the read data. This status byte reflects the state after the memory read was performed. If enabled, an ACK pulse is driven before the data bytes are transmitted.

The examples show the READ_MEM.B{_WS}, READ_MEM.W{_WS} and READ_MEM.L{_WS} commands.

5.4.4.12 READ_DBGTB

Read DBG trace buffer **Non-intrusive**



This command is only available on devices, where the DBG module includes a trace buffer. Attempted use of this command on devices without a traace buffer return 0x00.

Read 64 bits from the DBG trace buffer. Refer to the DBG module description for more detailed information. If enabled an ACK pulse is generated before each 32-bit longword is ready to be read by the host. After issuing the first ACK a timeout is still possible whilst accessing the second 32-bit longword, since this requires separate internal accesses. The first 32-bit longword corresponds to trace buffer line

Table 9-3. ADCCTL_0 Field Descriptions (continued)

Field	Description
11-10 ACC_CFG[1:0]	ADCFLWCTL Register Access Configuration — These bits define if the register ADCFLWCTL is controlled via internal interface only or data bus only or both. See Table 9-4. for more details.
9 STR_SEQA	Control Of Conversion Result Storage and RSTAR_EIF flag setting at Sequence Abort or Restart Event — This bit controls conversion result storage and RSTAR_EIF flag setting when a Sequence Abort Event or Restart Event occurs as follows: If STR_SEQA = 1'b0 and if a: • Sequence Abort Event or Restart Event is issued during a conversion the data of this conversion is not stored and the respective conversion complete flag is not set • Restart Event only is issued before the last conversion of a CSL is finished and no Sequence Abort Event is in process (SEQA clear) causes the RSTA_EIF error flag to be asserted and bit SEQA gets set by hardware If STR_SEQA = 1'b1 and if a: • Sequence Abort Event or Restart Event is issued during a conversion the data of this conversion is stored and the respective conversion complete flag is set and Intermediate Result Information Register is updated. • Restart Event only occurs during the last conversion of a CSL and no Sequence Abort Event is in process (SEQA clear) does not set the RSTA_EIF error flag • Restart Event only is issued before the CSL is finished and no Sequence Abort Event is in process (SEQA clear) causes the RSTA_EIF error flag to be asserted and bit SEQA gets set by hardware
8 MOD_CFG	(Conversion Flow Control) Mode Configuration — This bit defines the conversion flow control after a Restart Event and after execution of the "End Of List" command type: - Restart Mode - Trigger Mode (For more details please see also section Section 9.6.3.2, "Introduction of the Programmer's Model and following.) 0 "Restart Mode" selected. 1 "Trigger Mode" selected.

Table 9-4. ADCFLWCTL Register Access Configurations

ACC_CFG[1]	ACC_CFG[0]	ADCFLWCTL Access Mode
0	0	None of the access paths is enabled (default / reset configuration)
0	1	Single Access Mode - Internal Interface (ADCFLWCTL access via internal interface only)
1	0	Single Access Mode - Data Bus (ADCFLWCTL access via data bus only)
1	1	Dual Access Mode (ADCFLWCTL register access via internal interface and data bus)

NOTE

Each conversion flow control bit (SEQA, RSTA, TRIG, LDOK) must be controlled by software or internal interface according to the requirements described in Section 9.6.3.2.4, "The two conversion flow control Mode Configurations and overview summary in Table 9-11.

9.9.7 Conversion flow control application information

The ADC12B_LBA provides various conversion control scenarios to the user accomplished by the following features.

The ADC conversion flow control can be realized via the data bus only, the internal interface only, or by both access methods. The method used is software configurable via bits ACC_CFG[1:0].

The conversion flow is controlled via the four conversion flow control bits: SEQA, TRIG, RSTA, and LDOK.

Two different conversion flow control modes can be configured: Trigger Mode or Restart Mode

Single or double buffer configuration of CSL and RVL.

9.9.7.1 Initial Start of a Command Sequence List

At the initial start of a Command Sequence List after device reset all entries for at least one of the two CSL must have been completed and data must be valid. Depending on if the CSL_0 or the CSL_1 should be executed at the initial start of a Command Sequence List the following conversion control sequence must be applied:

If CSL_0 should be executed at the initial conversion start after device reset:
A Restart Event and a Trigger Event must occur (depending to the selected conversion flow control mode the events must occur one after the other or simultaneously) which causes the ADC to start conversion with commands loaded from CSL_0.

If CSL_1 should be executed at the initial conversion start after device reset:
Bit LDOK must be set simultaneously with the Restart Event followed by a Trigger Event (depending on the selected conversion flow control mode the Trigger events must occur simultaneously or after the Restart Event is finished). As soon as the Trigger Event gets executed the ADC starts conversion with commands loaded from CSL_1.

As soon as a new valid Restart Event occurs the flow for ADC register load at conversion sequence start as described in Section 9.6.3.3, “ADC List Usage and Conversion/Conversion Sequence Flow Description applies.

9.9.7.2 Restart CSL execution with currently active CSL

To restart a Command Sequence List execution it is mandatory that the ADC is idle (no conversion or conversion sequence is ongoing).

If necessary, a possible ongoing conversion sequence can be aborted by the Sequence Abort Event (setting bit SEQA). As soon as bit SEQA is cleared by the ADC, the current conversion sequence has been aborted and the ADC is idle (no conversion sequence or conversion ongoing).

After a conversion sequence abort is executed it is mandatory to request a Restart Event (bit RSTA set). After the Restart Event is finished (bit RSTA is cleared), the ADC accepts a new Trigger Event (bit TRIG can be set) and begins conversion from the top of the currently active CSL. In conversion flow control

Chapter 10
Supply Voltage Sensor - (BATSV3)

Table 10-1. Revision History Table

Rev. No. (Item No.)	Data	Sections Affected	Substantial Change(s)
V01.00	15 Dec 2010	all	Initial Version
V02.00	16 Mar 2011	10.3.2.1 10.4.2.1	- added BVLS[1] to support four voltage level - moved BVHS to register bit 6
V03.00	26 Apr 2011	all	- removed Vsense
V03.10	04 Oct 2011	10.4.2.1 and 10.4.2.2	- removed BSESE

10.1 Introduction

The BATS module provides the functionality to measure the voltage of the chip supply pin VSUP.

10.1.1 Features

The VSUP pin can be routed via an internal divider to the internal Analog to Digital Converter. Independent of the routing to the Analog to Digital Converter, it is possible to route this voltage to a comparator to generate a low or a high voltage interrupt to alert the MCU.

10.1.2 Modes of Operation

The BATS module behaves as follows in the system power modes:

1. Run mode

   The activation of the VSUP Level Sense Enable (BSUSE=1) or ADC connection Enable (BSUAE=1) closes the path from VSUP pin through the resistor chain to ground and enables the associated features if selected.

2. Stop mode

   During stop mode operation the path from the VSUP pin through the resistor chain to ground is opened and the low and high voltage sense features are disabled.
   The content of the configuration register is unchanged.

12.3.2.13 Output Compare Pin Disconnect Register(OCPD)

Module Base + 0x002C

	7	6	5	4	3	2	1	0
R W	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	OCPD1	OCPD0
Reset	0	0	0	0	0	0	0	0

Figure 12-20. Output Compare Pin Disconnect Register (OCPD)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 12-15. OCPD Field Description

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

Field	Description
1:0 OCPD[1:0]	Output Compare Pin Disconnect Bits 0 Enables the timer channel port. Output Compare action will occur on the channel pin. These bits do not affect the input capture . 1 Disables the timer channel port. Output Compare action will not occur on the channel pin, but the output compare flag still become set.

12.3.2.14 Precision Timer Prescaler Select Register (PTPSR)

Module Base + 0x002E

	7	6	5	4	3	2	1	0
R W	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
Reset	0	0	0	0	0	0	0	0

Figure 12-21. Precision Timer Prescaler Select Register (PTPSR)

Read: Anytime

Write: Anytime

All bits reset to zero.

NOTE

The CANRIER 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).

The RSTATE[1:0], TSTATE[1:0] bits are not affected by initialization mode.

Table 13-11. CANRIER Register Field Descriptions

Field	Description
7 WUPIE[1]	**Wake-Up Interrupt Enable** 0 No interrupt request is generated from this event. 1 A wake-up event causes a Wake-Up interrupt request.
6 CSCIE	**CAN Status Change Interrupt Enable** 0 No interrupt request is generated from this event. 1 A CAN Status Change event causes an error interrupt request.
5-4 RSTATE[1:0]	**Receiver Status Change Enable** — These RSTAT enable bits control the sensitivity level in which receiver state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level the RSTAT flags continue to indicate the actual receiver state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by receiver state changes. 01 Generate CSCIF interrupt only if the receiver enters or leaves “bus-off” state. Discard other receiver state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the receiver enters or leaves “RxErr” or “bus-off”[2] state. Discard other receiver state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
3-2 TSTATE[1:0]	**Transmitter Status Change Enable** — These TSTAT enable bits control the sensitivity level in which transmitter state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level, the TSTAT flags continue to indicate the actual transmitter state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by transmitter state changes. 01 Generate CSCIF interrupt only if the transmitter enters or leaves “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the transmitter enters or leaves “TxErr” or “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
1 OVRIE	**Overrun Interrupt Enable** 0 No interrupt request is generated from this event. 1 An overrun event causes an error interrupt request.
0 RXFIE	**Receiver Full Interrupt Enable** 0 No interrupt request is generated from this event. 1 A receive buffer full (successful message reception) event causes a receiver interrupt request.

1. WUPIE and WUPE (see Section 13.3.2.1, “MSCAN Control Register 0 (CANCTL0)”) must both be enabled if the recovery mechanism from stop or wait is required.

2. Bus-off state is only defined for transmitters by the CAN standard (see Bosch CAN 2.0A/B protocol specification). Because the only possible state change for the transmitter from bus-off to TxOK also forces the receiver to skip its current state to RxOK, the coding of the RXSTAT[1:0] flags define an additional bus-off state for the receiver (see Section 13.3.2.5, “MSCAN Receiver Flag Register (CANRFLG)”).

error condition the trigger generator reloads the new data from the trigger list and starts to generate the trigger. During an async reload event the TGxREIF interrupt flag is not set.

If the trigger value loaded from the memory contains double bit ECC errors (PTUDEEF flag is set) then the data is ignored and the trigger generator reload error flag (TGxREIF) is not set.

14.4.5.5 Trigger Generator Timing Error

The PTU module requires a defined number of bus clock cycle to load the next trigger value from the memory. This load time defines the minimum possible distance between consecutive trigger values within one trigger list or the distance between the reload event and the first trigger value. If a smaller distance is used then it is possible, depending on device conditions, that the TGxTEIF event is generated. To evaluate the TGxTEIF handling a distance of 1 should be used. This value will generate the TGxTEIF condition independent from the device conditions.

For the specification of this number, please see the Device Overview chapter.

The trigger generator timing error flag (TGxTEIF) is set if the loaded trigger value is smaller than the current counter value. The execution of this trigger list is stopped until the next reload event. There are different reasons for the trigger generator error condition:

- reload time exceeds time of next trigger event
- reload time exceeds the time between two consecutive trigger values
- a subsequent trigger value is smaller than the predecessor trigger value

If the trigger value loaded from the memory contains double bit ECC errors (PTUDEEF flag is set) then the data are ignored and the trigger generator timing error flag (TGxTEIF) is not set.

If enabled (TGxEIE is set) an interrupt will be generated.

14.4.5.6 Trigger Generator Done

The trigger generator done flag (TGxDIF) is set if the loaded trigger value contains 0x0000 or if the number of maximum trigger events (32) was reached. Please note, that the time which is required to load the next trigger value defines the delay between the generation of the last trigger and the assertion of the done flag. If enabled (TGxDIE is set) an interrupt is generated.If the trigger value loaded from the memory contains double bit ECC errors (PTUDEEF flag is set) then the data are ignored and the trigger generator done flag (TGxDIF) is not set.

14.4.6 Debugging

To see the internal status of the trigger generator the register TGxLIST, TGxTNUM, and TGxTV can be used. The TGxLIST register shows the number of currently used list. The TGxTNUM shows the number of generated triggers since the last reload event. If the maximum number of triggers was generated then this register shows zero. The trigger value loaded from the memory to generate the next trigger event is visible inside the TGxTV register. If the execution of the trigger list is done then these registers are unchanged until the next reload event. The next PWM reload event clears the TGxTNUM register and toggles the used trigger list if PTULDOK was set.

15.3 Memory Map and Registers

15.3.1 Module Memory Map

A summary of the registers associated with the PMF module is shown in Figure 15-2. 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.

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	PMFCFG0	R W	WP	MTG	EDGEC	EDGEB	EDGEA	INDEPC	INDEPB	INDEPA
0x0001	PMFCFG1	R W	0	ENCE	BOTNEGC	TOPNEGC	BOTNEGB	TOPNEGB	BOTNEGA	TOPNEGA
0x0002	PMFCFG2	R W	REV1	REV0	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x0003	PMFCFG3	R W	PMFWAI	PMFFRZ	0	VLMODE		PINVC	PINVB	PINVA
0x0004	PMFFEN	R W	0	FEN5	0	FEN4	FEN3	FEN2	FEN1	FEN0
0x0005	PMFFMOD	R W	0	FMOD5	0	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
0x0006	PMFFIE	R W	0	FIE5	0	FIE4	FIE3	FIE2	FIE1	FIE0
0x0007	PMFFIF	R W	0	FIF5	0	FIF4	FIF3	FIF2	FIF1	FIF0
0x0008	PMFQSMP0	R W	0	0	0	0	QSMP5		QSMP4	
0x0009	PMFQSMP1	R W	QSMP3		QSMP2		QSMP1		QSMP0	
0x000A- 0x000B	Reserved	R W	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-2. Quick Reference to PMF Registers (Sheet 1 of 5)

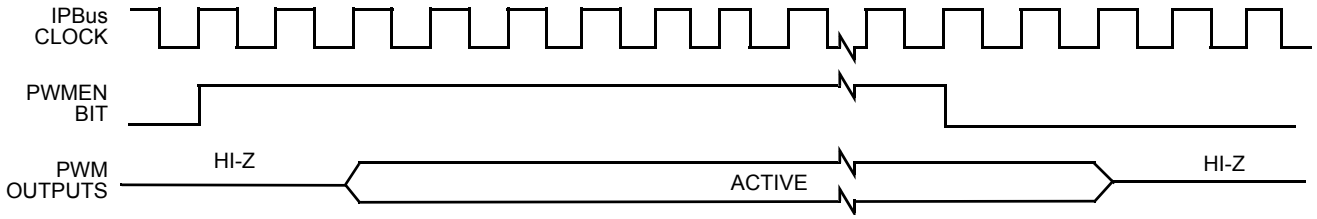


Figure 15-87. PWMEN and PWM Outputs in Independent Operation

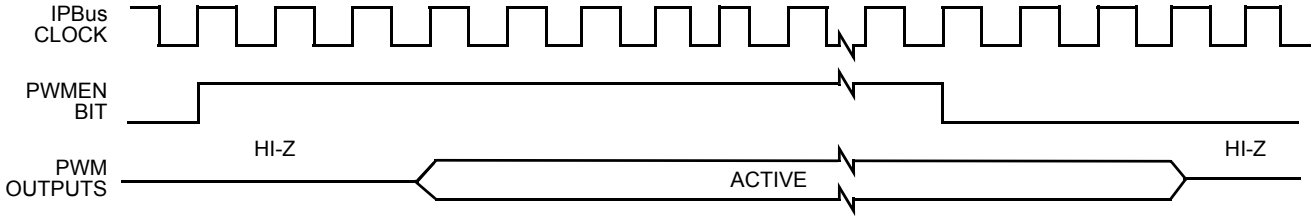


Figure 15-88. PWMEN and PWM Outputs in Complementary Operation

When the PWMEN bit is cleared:

- The PWM*n* outputs lose priority on associated outputs unless OUTCTL*n* = 1
- The PWM counter is cleared and does not count
- The PWM generator forces its outputs to zero
- The PWMRF flag and pending CPU interrupt requests are not cleared
- All fault circuitry remains active unless FEN*m* = 0
- Software output control remains active
- Deadtime insertion continues during software output control

15.8.1.1 Register Write Protection

The following configuration registers and bits can be write protected:

PMFCFG0, PMFCFG1, PMFCFG3, PMFFEN, PMFQSMP0-1, PMFENCA[RSTRTA,GLDOKA], PMFENCB[RSTRTB,GLDOKB], PMFENCC[RSTRTC,GLDOKC], PMFDTMA,B,C, PMFDMP0-5, PMFOUTF

NOTE

Make sure to set the write protection bit WP in PMFCFG0 after configuring and prior to enabling PWM outputs and fault inputs.

15.8.2 BLDC 6-Step Commutation

15.8.2.1 Unipolar Switching Mode

Unipolar switching mode uses registers PMFOUTC and PMFOUTB to perform commutation.

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.

17.4.3.2 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 17-12 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 17.3.2.2, "SPI Control Register 2 (SPICR2)

18.3.2.7 GDU Flag Register (GDUF)

Module Base + 0x0006 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	GSUF	GHHDF	GLVLSF	GOCIF[1:0]		0	GHHDIF	GLVLSIF
W								
Reset	X(2)	0	X(3)	0	0	0	0	X(3)

= Unimplemented

Figure 18-9. GDU Flag Register (GDUF)

1. Read: Anytime
   Write: Anytime, write 1 to clear flag

2. Loaded out of reset depending on mask set implementation as specified in the device overview

3. Out of power on reset the flags may be set.

Table 18-9. GDUF Register Field Descriptions

Field	Description
7 GSUF	GDU Start-up Flag — The start-up flag is cleared by reset and loaded depending on the device mask set implementation, as specified in the device overview, after reset de-asserts. Writing a logic “1” to the bit field clears the flag. If the flag is set all high-side FET pre-drivers are turned off and all low-side FET pre-drivers are turned on. If the flag is cleared and there is no error condition present all high-side and low-side FET pre-drivers are driven by the pwm channels. 0 High-side and low-side FET pre-drivers are driven by pwm channels 1 High-side FET pre-drivers turned off and low-side FET pre-drivers are turned on
6 GHHDF	GDU High V_{HD} Supply Flag — The flag controls the state of the FET pre-drivers. If the flag is set and GOCA1=0 the high-side pre-drivers are turned off and the low-side pre-drivers are turned on. If GOCA1=1 all high-side and low-side FET pre-drivers are turned off. If the flag is cleared and no other error condition is present the high-side and low-side pre-drivers are driven by the PWM channels. The flag is set by hardware if a high voltage condition on HD pin occurs. The flag is set if the voltage on pin HD is greater than the threshold voltage V_{HVHDLA} or V_{HVHDHA} . Writing a logic “1” to the bit field clears the flag. 0 Voltage on pin HD is less than V_{HVHDLD} or V_{HVHDHD} 1 Voltage on pin HD is greater than V_{HVHDLA} or V_{HVHDHA}
5 GLVLSF	GDU Low VLS Supply Flag — The flag controls the state of the FET pre-drivers. If the flag is set all high-side and low-side pre-drivers are turned off. If the flag is cleared and no other error condition is present the high-side and low-side pre-drivers are driven by the PWM channels. The flag is set by hardware if a low voltage condition on VLS_OUT pin occurs. Writing a logic “1” to the bit field clears the flag. 0 VLS_OUT pin voltage is above V_{LVLSD} 1 VLS_OUT pin voltage is below $V_{LVLSHA,}$ or V_{LVLSLA} all high-side and low-side FET pre-drivers are turned off
4-3 GOCIF[1:0]	GDU Overcurrent Interrupt Flag — The interrupt flags are set by hardware if an overcurrent condition occurs. The flags are set if the voltage on the overcurrent comparator input is greater than the threshold voltage V_{OCT}. If the GOCIE bit is set an interrupt is requested. Writing a logic “1” to the bit field clears the flag. If the GOCAx bit is cleared all high-side FET pre-drivers are turned off and fault[4] is asserted. If GOCAx is set all high-side and low-side FET pre-drivers are turned off and fault[2:0] are asserted. 0 Voltage on overcurrent comparator input is less than V_{OCT} 1 Voltage on overcurrent comparator is greater than V_{OCT}

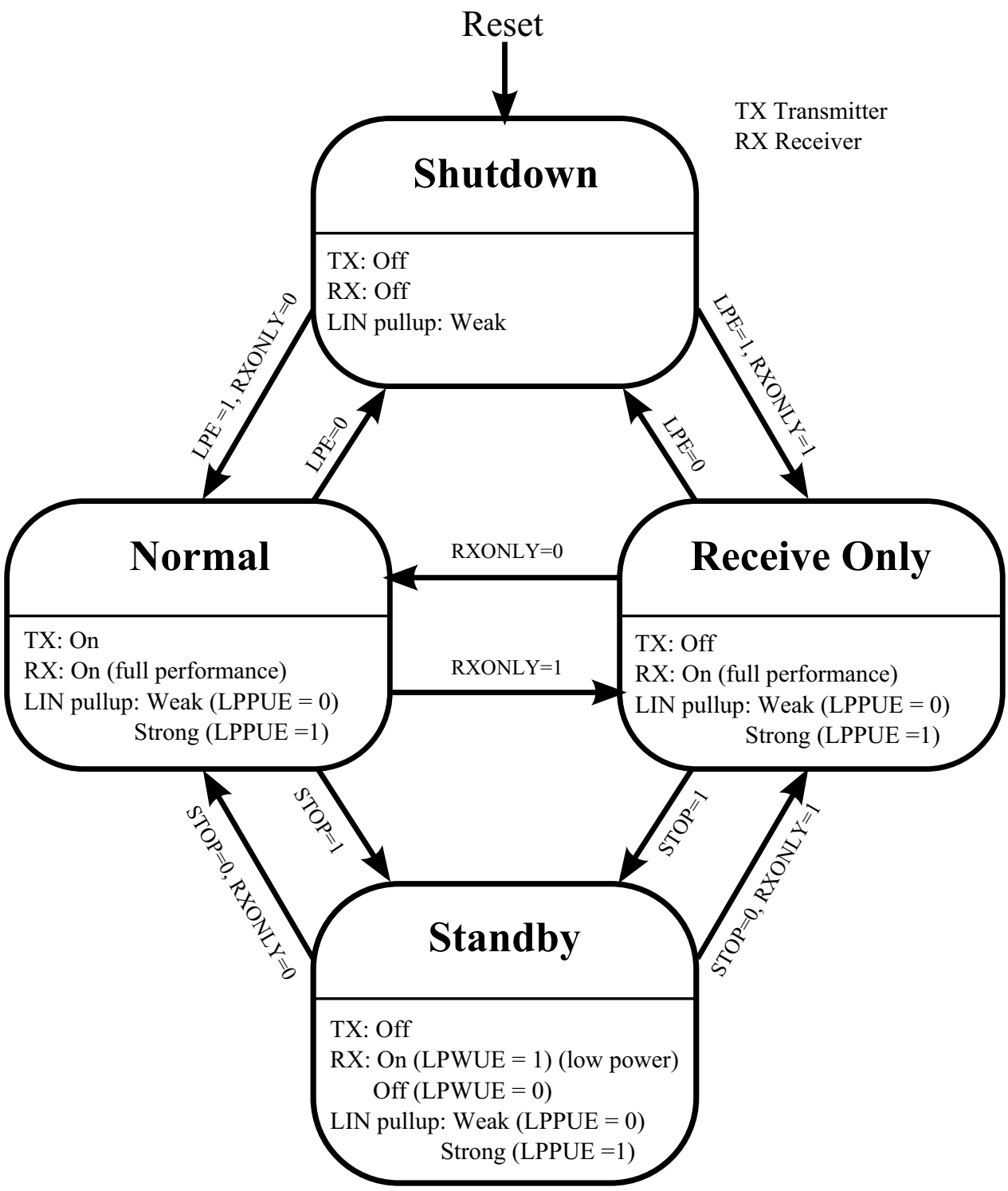


Figure 19-11. LIN/HV Physical Layer Mode Transitions

Address & Name		7	6	5	4	3	2	1	0
0x0003 FPSTAT	R	FPOVRD	0	0	0	0	0	0	WSTATACK
	W								
0x0004 FCNFG	R	CCIE	0	ERSAREQ	IGNSF	WSTAT[1:0]		FDFD	FSFD
	W								
0x0005 FERCNFG	R	0	0	0	0	0	0	0	SFDIE
	W								
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	0	0	0	0	0	0	DFDF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 DFPROT(1)	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
	W								
0x000A FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x000B FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x000C FCCOB0HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000D FCCOB0LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000E FCCOB1HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000F FCCOB1LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0010 FCCOB2HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								

Figure 20-4. FTMRZ128K512 Register Summary (continued)

20.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Offset Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	FPOPEN	RNV6	FPHDIS	FPHS[1:0]		FPLDIS	FPLS[1:0]	
W								
Reset	F(1)	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]	F[1]

= Unimplemented or Reserved

Figure 20-13. Flash Protection Register (FPROT)

1. Loaded from Flash configuration field, during reset sequence.

The (unreserved) bits of the FPROT register are writable Normal Single Chip Mode with the restriction that the size of the protected region can only be increased see Section 20.3.2.9.1, “P-Flash Protection Restrictions,” and Table 20-23). All (unreserved) bits of the FPROT register are writable without restriction in Special Single Chip Mode.

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0xFF_FE0C located in P-Flash memory (see Table 20-4) as indicated by reset condition ‘F’ in Figure 20-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 20-19. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 20-20 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0xFF_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 20-21. The FPHS bits can only be written to while the FPHDIS bit is set.

20.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.