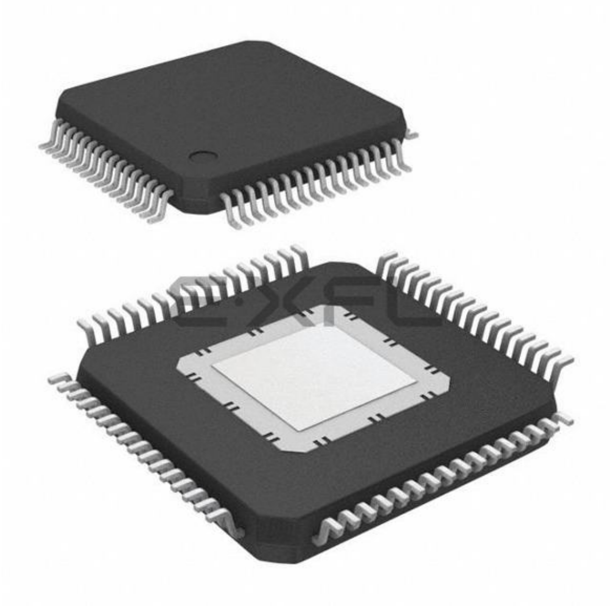
NXP USA Inc. - S912ZVML31F1WKHR Datasheet



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	40MHz
Connectivity	LINbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 40V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP Exposed Pad
Supplier Device Package	64-HLQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912zvml31f1wkhr

Table 0-1. Revision History

Date	Revision	Description
22 MAY2014	1.4	Updated family derivative table for S12ZVML32, S12ZVM32 and S12ZVM16 devices Added 64KB, 32KB and 16KB derivative information to flash module chapter Added pin routing options for S12ZVM32 and S12ZVM16 devices Added HV Phy information for the S12ZVM32 and S12ZVM16 derivatives Updated Part ID assignment table and ordering information for S12ZVM32 and S12ZVM16 Corrected PLL VCO maximum frequency specification Changed V_{LVLSA} maximum from 7V to 6.9V Added electrical parameter for HD division ratio through the phase multiplexer Corrected preferred VRL reference from VRL_1 to VRL_0 Included NVM timing parameters for the S12ZVM32 and S12ZVM16 devices Added GDU S12ZVM32 and S12ZVM16 specific differences and electrical specifications Added references to f_{WSTAT} Added VDDX short circuit fall back current and temperature/input dependency specs.
22 SEP 2014	1.5	Removed incorrect references to PACLK in TIM chapter Improved clarity of routing options in PIM chapter. Updated S12ZVM- Family derivative table. Added 48LQFP thermal package parameters Extended LINPHY specification range minimum to 5V Updated BKGD pin I/O specification Specified ADC accuracy for a range of VDDA and VREF.
20 MAR 2015	2.0	Added ZVMC256 information Added mask set 2N95G information Added more detailed PTU minimum trigger spacing description Updated CPMU, PIM and GDU chapters for ZVMC256 Improved CPMU specification clarity (see CPMU revision history) Removed electrical parameter classification Added reset startup timing parameter Updated BATS parameters Extended BKGD V_{IL} condition from 3.15V to 3.13V Extended GDU operating range from 26V to 26.6V Temperature sensor output at 150C changed from 2.25V to 2.33V. Added GDU VBS current parameter Updated package thermal information for ZVM32 and ZVM16 parts Added VBG temperature and voltage dependency parameters Added device stop current at 105C.
22 APR 2015	2.1	Updated Stop and Wait current parameter values (I_{SUPS}, I_{SUPW}) Corrected 80LQFP-EP pin name from VSS2 to VSS1 Updated ZVMC256 VDDS regulator parameters. Changed PL0 ESD specification Minor corrections to PIM, PMF, SRAM and ADC chapters (see module revision histories)
27 APR 2015	2.2	Updated Stop current parameter values (I_{SUPS}) Updated LINPHY parameter range limit to 5.5V Added more information about VDDS1, VDDS2, SNPS1, SNPS2 to CPMU chapter. Reintroduced EPRES bit for GDU V4 Added 80LQFP-EP mechanical package information

1.4.2.3 EEPROM

- Up to 1K byte EEPROM
  - 16 data bits plus 6 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 4 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

1.4.2.4 SRAM

- Up to 32 KB of general-purpose RAM with ECC
  - Single bit error correction and double bit error detection

1.4.3 Clocks, Reset & Power Management Unit (CPMU)

- Real time interrupt (RTI)
- Clock monitor, supervising the correct function of the oscillator (CM)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Can be initialized out of reset using option bits located in flash memory
- System reset generation
- Autonomous periodic interrupt (API) (combination with cyclic, watchdog)
- Low Power Operation
  - RUN mode is the main full performance operating mode with the entire device clocked.
  - WAIT mode when the internal CPU clock is switched off, so the CPU does not execute instructions.
  - Pseudo STOP - system clocks are stopped but the oscillator the RTI, the COP, and API modules can be enabled
  - STOP - the oscillator is stopped in this mode, all clocks are switched off and all counters and dividers remain frozen, with the exception of the COP and API which can optionally run from ACLK.

1.4.3.1 Internal Phase-Locked Loop (IPLL)

- Phase-locked-loop clock frequency multiplier
  - No external components required
  - Reference divider and multiplier allow large variety of clock rates
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
  - Reference clock sources:

Port	Pin Name	ZVMC256	ZVMC128\64	ZVML128/64/32	ZVML31	ZVM32/16	Pin Function & Priority[1]	I/O	Description	Routing Register Bit	Pin Function after Reset
	PS0	✔					$\overline{\text{SS0}}$	I/O	SPI0 slave select	SPI0RR SPI0SSRR	
		✔	✔	✔	✔	✔	PTUT0	O	PTU trigger 0	—	
		✔			✔	✔	(IOC0_1)	I/O	TIM0 channel 1	T0C1RR T0IC1RR T0IC1RR0	
				✔	✔	✔	(LPRXD0)	O	LINPHY0/HVPHY0 receive output	S0L0RR2-0	
		✔	✔	✔			RXCAN0[2]	I	MSCAN0 receive	M0C0RR2-0	
		✔	✔	✔	✔	✔	RXD1	I	SCI1 receive	SCI1RR	
		✔	✔	✔	✔	✔	PTS[0]/ KWS[0]	I/O	General-purpose; with interrupt and wakeup	—	

[1] Signals in parentheses denote alternative module routing pins.

[2] Routing option for ZVMC256

3.4.2 Illegal Accesses

The S12ZMMC module monitors all memory traffic for illegal accesses. See Table 3-9 for a complete list of all illegal accesses.

Table 3-9. Illegal memory accesses

		S12ZCPU	S12ZBDC	ADCs and PTU
Register space	Read access	ok	ok	illegal access
	Write access	ok	ok	illegal access
	Code execution	illegal access		
RAM	Read access	ok	ok	ok
	Write access	ok	ok	ok
	Code execution	ok		
EEPROM	Read access	ok[1]	ok[1]	ok[1]
	Write access	illegal access	illegal access	illegal access
	Code execution	ok[1]		
Reserved Space	Read access	ok	ok	illegal access
	Write access	only permitted in SS mode	ok	illegal access
	Code execution	illegal access		
Reserved Read-only Space	Read access	ok	ok	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		
NVM IFR	Read access	ok[1]	ok[1]	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		
Program NVM	Read access	ok[1]	ok[1]	ok[1]
	Write access	illegal access	illegal access	illegal access
	Code execution	ok[1]		
Unmapped Space	Read access	illegal access	illegal access	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		

[1] Unsupported NVM accesses during NVM command execution (“collisions”), are treated as illegal accesses.

Illegal accesses are reported in several ways:

- All illegal accesses performed by the S12ZCPU trigger machine exceptions.
- All illegal accesses performed through the S12ZBDC interface, are captured in the ILLACC bit of the BDCCSRL register.

Table 5-6. BDCCSRL Field Descriptions (continued)

Field	Description
4 OVRUN	**Overrun Flag** — Indicates unexpected host activity before command completion. This occurs if a new command is received before the current command completion. With ACK enabled this also occurs if the host drives the BKGD pin low whilst a target ACK pulse is pending To protect internal resources from misinterpreted BDC accesses following an overrun, internal accesses are suppressed until a SYNC clears this bit. A SYNC clears the bit. 0 No overrun detected. 1 Overrun detected when issuing a BDC command.
3 NORESP	**No Response Flag** — Indicates that the BDC internal action or data access did not complete. This occurs in the following scenarios: a) If no free cycle for an access is found within 512 core clock cycles. This could typically happen if a code loop without free cycles is executing with ACK enabled and STEAL clear. b) With ACK disabled or STEAL set, when an internal access is not complete before the host starts data/BDCCSRL retrieval or an internal write access is not complete before the host starts the next BDC command. c) Attempted internal memory or SYNC_PC accesses during STOP mode set NORESP if BDCCIS is clear. In the above cases, on setting NORESP, the BDC aborts the access if permitted. (For devices supporting EWAIT, BDC external accesses with EWAIT assertions, prevent a command from being aborted until EWAIT is deasserted). d) If a BACKGROUND command is issued whilst the device is in wait mode the NORESP bit is set but the command is not aborted. The active BDM request is completed when the device leaves wait mode. Furthermore subsequent CPU register access commands during wait mode set the NORESP bit, should it have been cleared. e) If a command is issued whilst awaiting return from Wait mode. This can happen when using STEP1 to step over a CPU WAI instruction, if the CPU has not returned from Wait mode before the next BDC command is received. f) If STEP1 is issued with the BDC enabled as the device enters Wait mode regardless of the BDMACT state. When NORESP is set a value of 0xEE is returned for each data byte associated with the current access. Writing a “1” to this bit, clears the bit. 0 Internal action or data access completed. 1 Internal action or data access did not complete.
2 RDINV	**Read Data Invalid Flag** — Indicates invalid read data due to an ECC error during a BDC initiated read access. The access returns the actual data read from the location. Writing a “1” to this bit, clears the bit. 0 No invalid read data detected. 1 Invalid data returned during a BDC read access.
1 ILLACC	**Illegal Access Flag** — Indicates an attempted illegal access. This is set in the following cases: When the attempted access addresses unimplemented memory When the access attempts to write to the flash array When a CPU register access is attempted with an invalid CRN (Section 5.4.5.1). Illegal accesses return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal access detected. 1 Illegal BDC access detected.

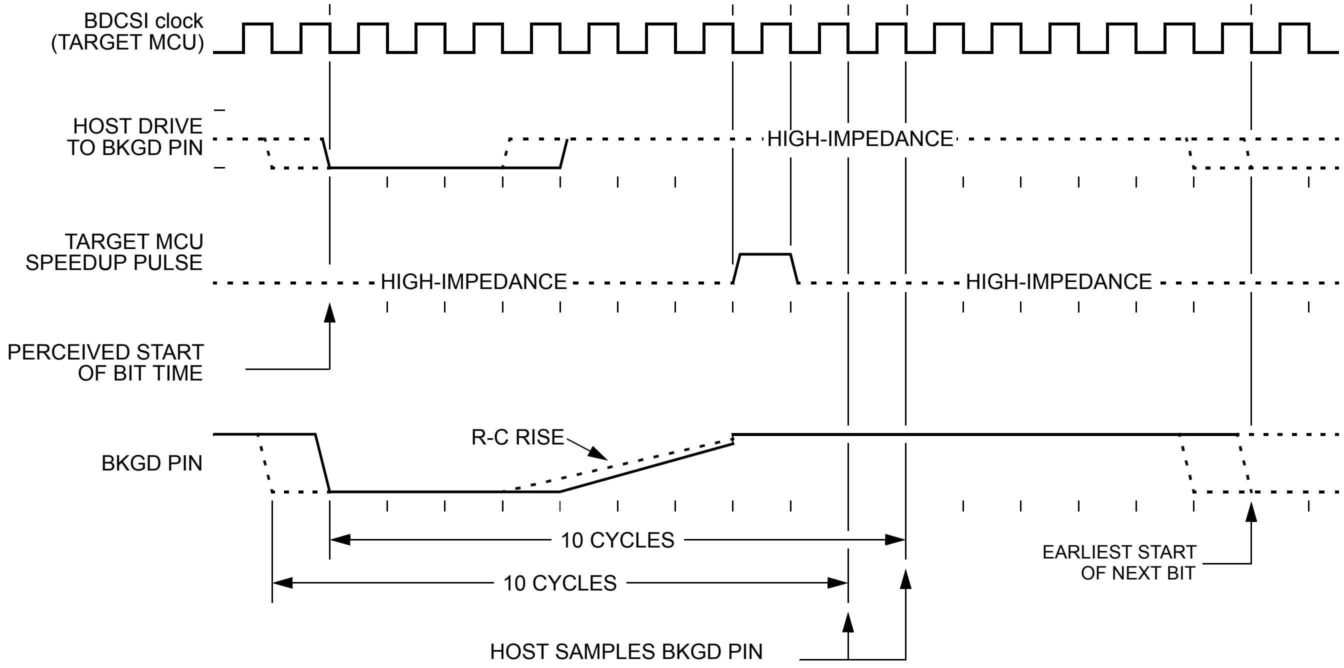


Figure 5-7. BDC Target-to-Host Serial Bit Timing (Logic 1)

Figure 5-8 shows the host receiving a logic 0 from the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

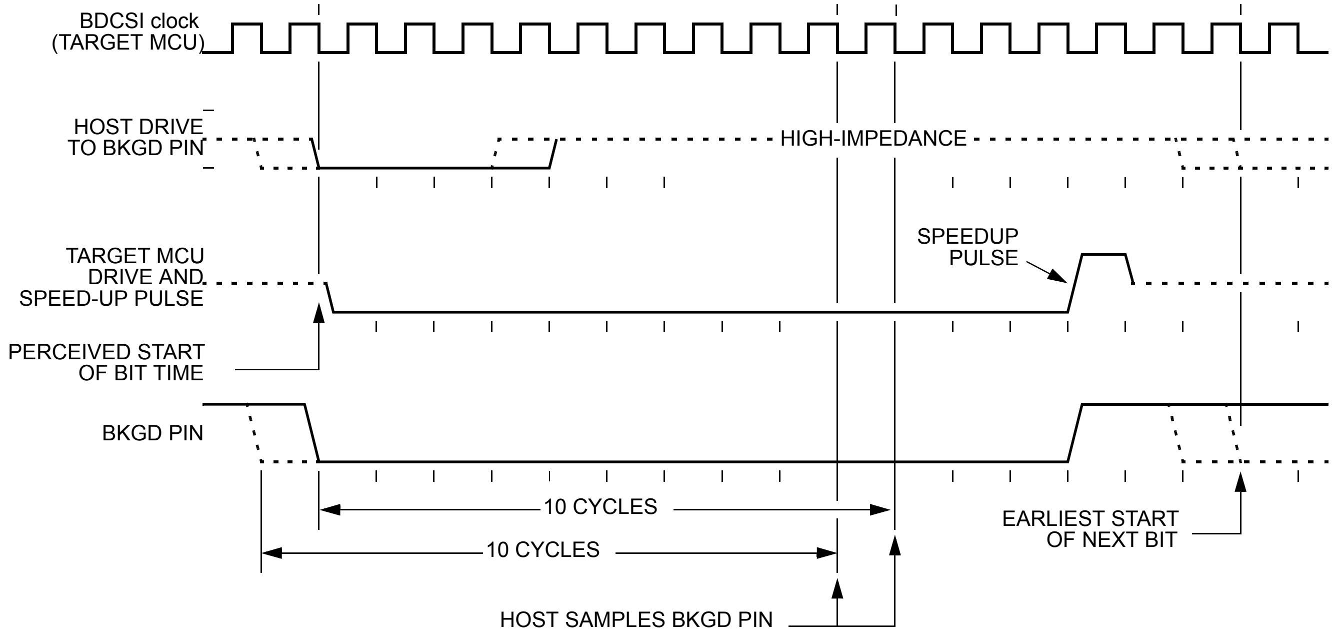


Figure 5-8. BDC Target-to-Host Serial Bit Timing (Logic 0)

Following a STOP or WAI instruction, if the BDC is enabled, the first ACK, following stop or wait mode entry is a long ACK to indicate an exception.

5.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. To abort a command that has not responded with an ACK pulse, the host controller generates a sync request (by driving BKGD low for at least 128 BDCSI clock cycles and then driving it high for one BDCSI clock cycle as a speedup pulse). By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 5.4.4.1”, and assumes that the pending command and therefore the related ACK pulse are being aborted. After the SYNC protocol has been completed the host is free to issue new BDC commands.

The host can issue a SYNC close to the 128 clock cycles length, providing a small overhead on the pulse length to assure the sync pulse is not misinterpreted by the target. See Section 5.4.4.1”.

Figure 5-11 shows a SYNC command being issued after a READ_MEM, which aborts the READ_MEM command. Note that, after the command is aborted a new command is issued by the host.

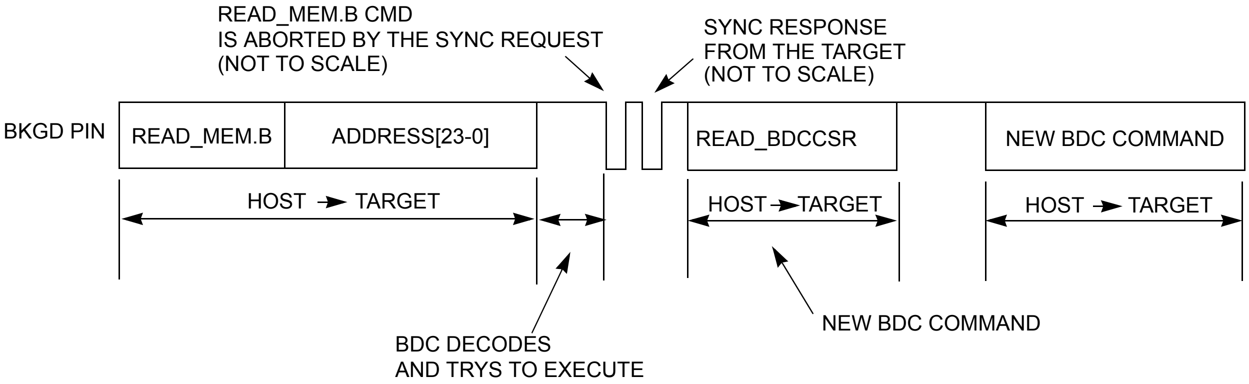


Figure 5-11. ACK Abort Procedure at the Command Level (Not To Scale)

Figure 5-12 shows a conflict between the ACK pulse and the SYNC request pulse. The target is executing a pending BDC command at the exact moment the host is being connected to the BKGD pin. In this case, an ACK pulse is issued simultaneously to the SYNC command. Thus there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. As this is not a probable situation, the protocol does not prevent this conflict from happening.

Table 8-14. CPMUCOP Field Descriptions

Field	Description
7 WCOP	**Window COP Mode Bit** — When set, a write to the CPMUARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period generates a COP reset. As long as all writes occur during this window, \$55 can be written as often as desired. Once \$AA is written after the \$55, the time-out logic restarts and the user must wait until the next window before writing to CPMUARMCOP. Table 8-15 shows the duration of this window for the seven available COP rates. 0 Normal COP operation 1 Window COP operation
6 RSBCK	COP and RTI Stop in Active BDM Mode Bit 0 Allows the COP and RTI to keep running in Active BDM mode. 1 Stops the COP and RTI counters whenever the part is in Active BDM mode.
5 WRTMASK	**Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP and CR[2:0] bits while writing the CPMUCOP register. It is intended for BDM writing the RSBCK without changing the content of WCOP and CR[2:0]. 0 Write of WCOP and CR[2:0] has an effect with this write of CPMUCOP 1 Write of WCOP and CR[2:0] has no effect with this write of CPMUCOP. (Does not count for “write once”.)
2–0 CR[2:0]	**COP Watchdog Timer Rate Select** — These bits select the COP time-out rate (see Table 8-15 and Table 8-16). Writing a nonzero value to CR[2:0] enables the COP counter and starts the time-out period. A COP counter time-out causes a System Reset. This can be avoided by periodically (before time-out) initializing the COP counter via the CPMUARMCOP register. While all of the following four conditions are true the CR[2:0], WCOP bits are ignored and the COP operates at highest time-out period (2^{24} cycles) in normal COP mode (Window COP mode disabled): 1) COP is enabled (CR[2:0] is not 000) 2) BDM mode active 3) RSBCK = 0 4) Operation in Special Mode

Table 8-15. COP Watchdog Rates if COPOSCSEL1=0. (default out of reset)

CR2	CR1	CR0	COPCLK Cycles to time-out (COPCLK is either IRCCLK or OSCCLK depending on the COPOSCSEL0 bit)
0	0	0	COP disabled
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

8.3.2.27 Voltage Regulator Control Register (CPMUVREGCTL)

The CPMUVREGCTL allows to enable or disable certain parts of the voltage regulator.This register must be configured after system startup.

Module Base + 0x001D

	7	6	5	4	3	2	1	0
R	VRH2EN[1]	VRH1EN[1]	EXTS2ON[1]	EXTS1ON[(1)]	0	EXTCON	EXTXON	INTXON
W					▨			
Reset	0	0	0	0	0	1	1	1

▨ = Unimplemented or Reserved

Figure 8-38. Voltage Regulator Control Register (CPMUVREGCTL)

[1] Only available in V10

Read: Anytime

Write: VRH2EN, VRH1EN, EXTS2ON, EXTS1ON anytime
Write: EXTCON, EXTXON, INTXON once in normal modes, anytime in special modes

Table 8-30. Effects of writing the EXTXON and INTXON bits

value of EXTXON to be written	value of INTXON to be written	Write Access
0	0	blocked, no effect
0	1	legal access
1	0	legal access
1	1	blocked, no effect

Table 8-31. CPMUVREGCTL Field Descriptions

Field	Description
7 VRH2EN	**VRH2 Enable Bit** — This bits switches VDDS2 pin to VRH2 of ADC. 0 VRH2 of ADC disconnected (open) 1 VRH2 of ADC connected to VDDS2. In RPM VRH2 is always disconnected from VDDS2 regardless of the value of the VRH2EN bit.
6 VRH1EN	**VRH1 Enable Bit** — This bits switches VDDS1 pin to VRH1 of ADC. 0 VRH1 of ADC disconnected (open) 1 VRH1 of ADC connected to VDDS1. In RPM VRH1 is always disconnected from VDDS1 regardless of the value of the VRH1EN bit.
5 EXTS2ON	**External voltage regulator Enable Bit for VDDS2 domain** — Should be enabled after system startup if VDDS2 is used. 0 VDDS2 domain disabled 1 VDDS2 domain enabled. BCTLS2 pin is active.

9.5.2.21 ADC Result Index Register (ADCRIDX)

It is important to note that these bits do not represent absolute addresses instead it is a sample index (object size 16bit).

Module Base + 0x0020

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

▨ = Unimplemented or Reserved

Figure 9-24. ADC Result Index Register (ADCRIDX)

Read: Anytime

Write: NA

Table 9-29. ADCRIDX Field Descriptions

Field	Description
5-0 RES_IDX[5:0]	**ADC Result Index Bits** — These read only bits represent the index value for the conversion results relative to the two RVL start addresses in the memory map. These bits do not represent absolute addresses instead it is a sample index (object size 16bit). See also Section 9.6.3.2.3, “Introduction of the two Result Value Lists (RVLs) for more details.

12.3.2.12 Timer Input Capture/Output Compare Registers High and Low 0–1(TCxH and TCxL)

Module Base + 0x0010 = TC0H 0x0018=RESERVD
0x0012 = TC1H 0x001A=RESERVD
0x0014=RESERVD 0x001C=RESERVD
0x0016=RESERVD 0x001E=RESERVD

	15	14	13	12	11	10	9	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 12-18. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L 0x0019 =RESERVD
0x0013 = TC1L 0x001B=RESERVD
0x0015 =RESERVD 0x001D=RESERVD
0x0017=RESERVD 0x001F=RESERVD

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

Figure 12-19. Timer Input Capture/Output Compare Register x Low (TCxL)

[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.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should take place before low byte otherwise it will give a different result.

- Global Load OK support, to guarantee coherent update of all control loop modules
- Trigger values stored inside the global memory map, basically inside system memory
- Software generated reload event and Trigger event generation for debugging

14.1.2 Modes of Operation

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

1. Run mode

   All PTU features are available.
2. Wait mode
   All PTU features are available.
3. Freeze Mode

   Depends on the PTUFRZ register bit setting the internal counter is stopped and no trigger events will be generated.
4. Stop mode

   The PTU is disabled and the internal counter is stopped; no trigger events will be generated. The content of the configuration register is unchanged.

Table 14-6. PTUIEL Register Field Descriptions

Field	Description
5 TG1TEIE	Trigger Generator 1 Timing Error Interrupt Enable — Enables trigger generator timing error interrupt. 0 No interrupt will be requested whenever TG1TEIF is set 1 Interrupt will be requested whenever TG1TEIF is set
4 TG1DIE	Trigger Generator 1 Done Interrupt Enable — Enables trigger generator done interrupt. 0 No interrupt will be requested whenever TG1DIF is set 1 Interrupt will be requested whenever TG1DIF is set
3 TG0AEIE	Trigger Generator 0 Memory Access Error Interrupt Enable — Enables trigger generator memory access error interrupt. 0 No interrupt will be requested whenever TG0AEIF is set 1 Interrupt will be requested whenever TG0AEIF is set
2 TG0REIE	Trigger Generator 0 Reload Error Interrupt Enable — Enables trigger generator reload error interrupt. 0 No interrupt will be requested whenever TG0REIF is set 1 Interrupt will be requested whenever TG0REIF is set
1 TG0TEIE	Trigger Generator 0 Timing Error Interrupt Enable — Enables trigger generator timing error interrupt. 0 No interrupt will be requested whenever TG0TEIF is set 1 Interrupt will be requested whenever TG0TEIF is set
0 TG0DIE	Trigger Generator 0 Done Interrupt Enable — Enables trigger generator done interrupt. 0 No interrupt will be requested whenever TG0DIF is set 1 Interrupt will be requested whenever TG0DIF is set

14.3.2.11 Trigger Generator 1 Trigger Number Register (TG1TNUM)

Module Base + 0x000B   Access: User read only[1]

	7	6	5	4	3	2	1	0
R	0	0	0	TG1TNUM[4:0]				
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 14-13. Trigger Generator 1 Trigger Number Register (TG1TNUM)

1. Read: Anytime
   Write: Never

Table 14-13. TG1TNUM Register Field Descriptions

Field	Description
4:0 TG1TNUM[4:0]	Trigger Generator 1 Trigger Number — This register shows the number of generated triggers since the last reload event. After the generation of 32 triggers this register shows zero. The next reload event clears this register. See also Figure 14-22.

14.3.2.12 Trigger Generator 1 Trigger Value (TG1TVH, TG1TVL)

Module Base + 0x000C   Access: User read only[1]

	7	6	5	4	3	2	1	0
R	TG1TV[15:8]							
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

Module Base + 0x000D   Access: User read only

	7	6	5	4	3	2	1	0
R	TG1TV[7:0]							
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 14-14. Trigger Generator 1 Trigger Value Register (TG1TVH, TG1TVL)

1. Read: Anytime
   Write: Never

15.1.3 Block Diagram

Figure 15-1 provides an overview of the PMF module.

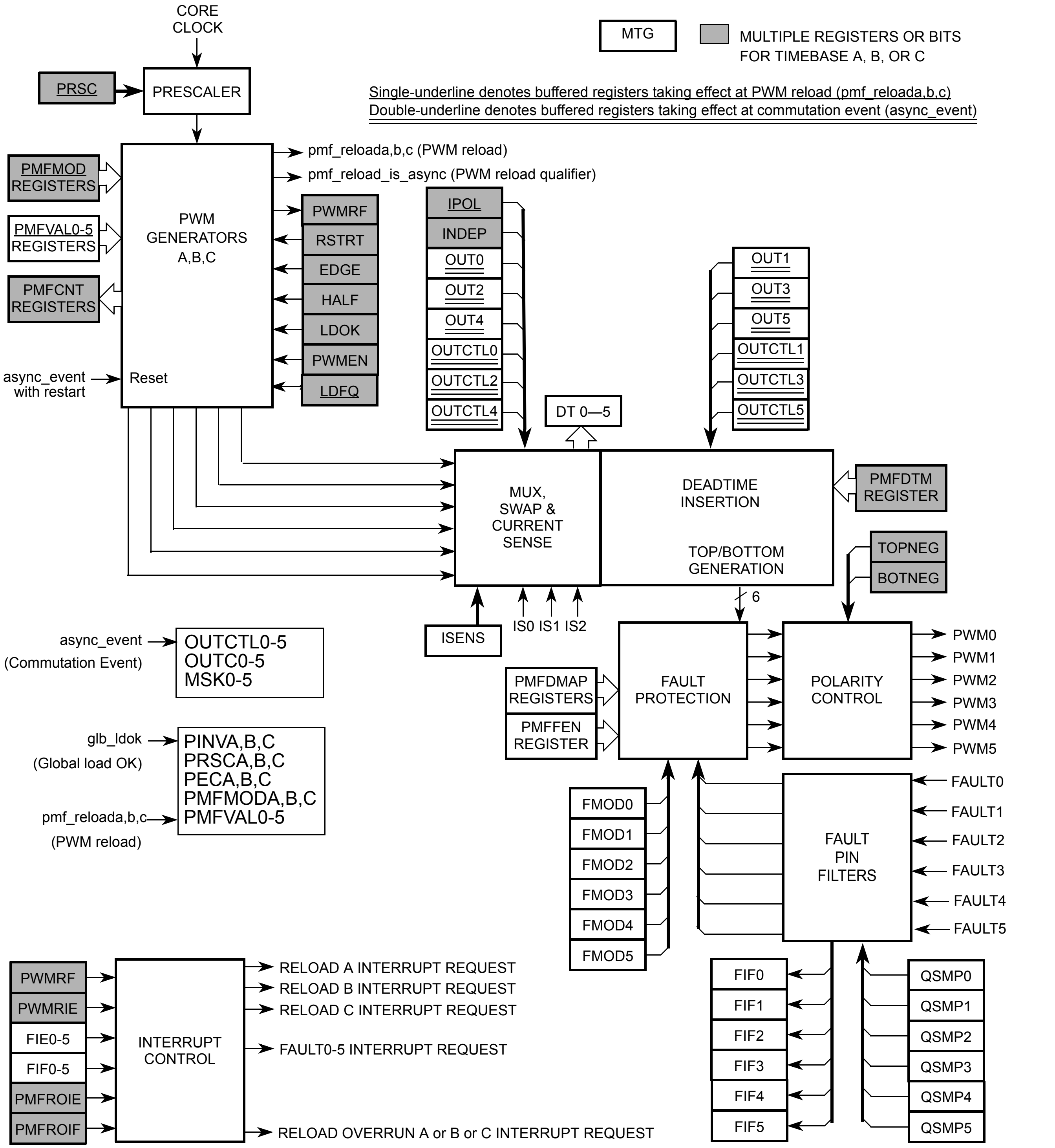


Figure 15-1. PMF Block Diagram

15.3.2.17 PMF Internal Correction Control Register (PMFICCTL)

Address: Module Base + 0x001E

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PECC	PECB	PECA	ICCC	ICCB	ICCA
W								
Reset	0	0	0	0	0	0	0	0

Figure 15-20. PMF Internal Correction Control Register (PMFICCTL)

1. Read: Anytime
   Write: Anytime

This register is used to control PWM pulse generation for various applications, such as a power-supply phase-shifting application.

ICC*x* bits apply only in center-aligned operation during complementary mode. These control bits determine whether values set in the IPOL*x* bits control or the whether PWM count direction controls which PWM value register is used.

NOTE

> The ICC*x* bits are buffered. The value written does not take effect until the next PWM load cycle begins regardless of the state of the LDOK bit or global load OK. Reading ICC*x* returns the value in a buffer and not necessarily the value the PWM generator is currently using.

The PEC*x* bits apply in edge-aligned and center-aligned operation during complementary mode. Setting the PEC*x* bits overrides the ICC*x* settings. This allows the PWM pulses generated by both the odd and even PWM value registers to be ANDed together prior to the complementary logic and deadtime insertion.

NOTE

> The PEC*x* bits are buffered. The value written does not take effect until the related LDOK bit or global load OK is set and the next PWM load cycle begins. Reading PEC*n* returns the value in a buffer and not necessarily the value the PWM generator is currently using.

Figure 15-21. PMF Internal Correction Control Register (PMFICCTL) Descriptions

Field	Description
5 PECC	Pulse Edge Control — This bit controls PWM4/PWM5 pair. 0 Normal operation 1 Allow one of PMFVAL4 and PMFVAL5 to activate the PWM pulse and the other to deactivate the pulse
4 PECB	Pulse Edge Control — This bit controls PWM2/PWM3 pair. 0 Normal operation 1 Allow one of PMFVAL2 and PMFVAL3 to activate the PWM pulse and the other to deactivate the pulse
3 PECA	Pulse Edge Control — This bit controls PWM0/PWM1 pair. 0 Normal operation 1 Allow one of PMFVAL0 and PMFVAL1 to activate the PWM pulse and the other to deactivate the pulse

17.3.2.3 SPI Baud Rate Register (SPIBR)

Module Base +0x0002

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W	▨				▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 17-5. SPI Baud Rate Register (SPIBR)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 17-6. SPIBR Field Descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 17-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 17-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$\mathbf{BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}}$$

Eqn. 17-1

The baud rate can be calculated with the following equation:

$$\mathbf{Baud\ Rate = BusClock / BaudRateDivisor}$$

Eqn. 17-2

NOTE

For maximum allowed baud rates, please refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

Table 17-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 1 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s
0	0	1	0	1	0	16	1.5625 Mbit/s

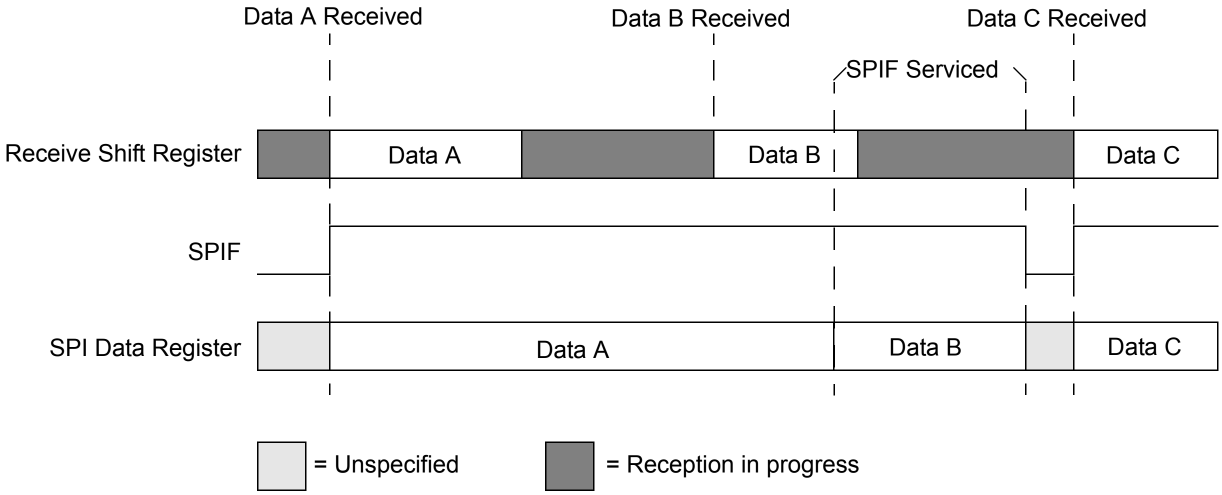


Figure 17-9. Reception with SPIF serviced in Time

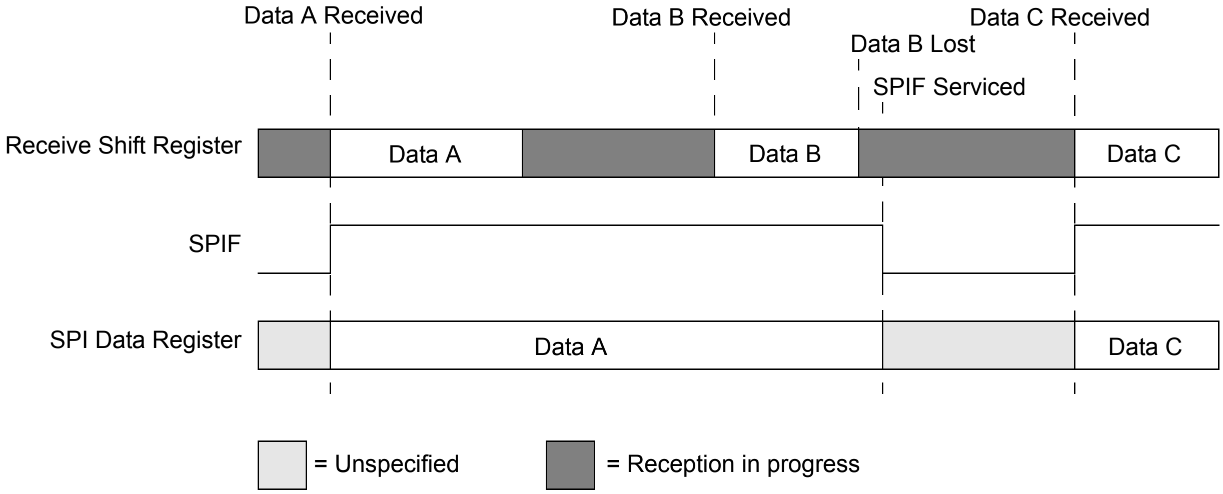


Figure 17-10. Reception with SPIF serviced too late

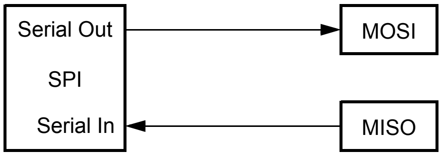
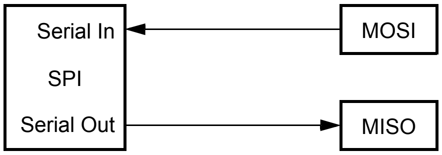
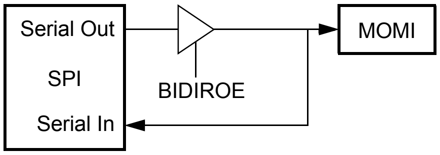
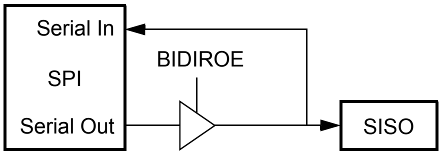
17.4 Functional Description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{\text{SS}}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

Table 17-11. Normal Mode and Bidirectional Mode

When SPE = 1	Master Mode MSTR = 1	Slave Mode MSTR = 0
Normal Mode SPC0 = 0		
Bidirectional Mode SPC0 = 1		

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{\text{SS}}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{\text{SS}}$ functions.

NOTE

In bidirectional master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode. In this case MISO becomes occupied by the SPI and MOSI is not used. This must be considered, if the MISO pin is used for another purpose.

17.4.6 Error Conditions

The SPI has one error condition:

- Mode fault error

17.4.6.1 Mode Fault Error

If the $\overline{\text{SS}}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{\text{SS}}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case

NOTE

> Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register will be lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a tranmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

17.4.7.3 SPI in Stop Mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

17.4.7.4 Reset

The reset values of registers and signals are described in Section 17.3, “Memory Map and Register Definition”, which details the registers and their bit fields.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it will transmit garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset will always read zeros.

17.4.7.5 Interrupts

The SPI only originates interrupt requests when SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the SPI makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

17.4.7.5.1 MODF

MODF occurs when the master detects an error on the $\overline{\text{SS}}$ pin. The master SPI must be configured for the MODF feature (see Table 17-3). After MODF is set, the current transfer is aborted and the following bit is changed:

- MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in Section 17.3.2.4, “SPI Status Register (SPISR)”.