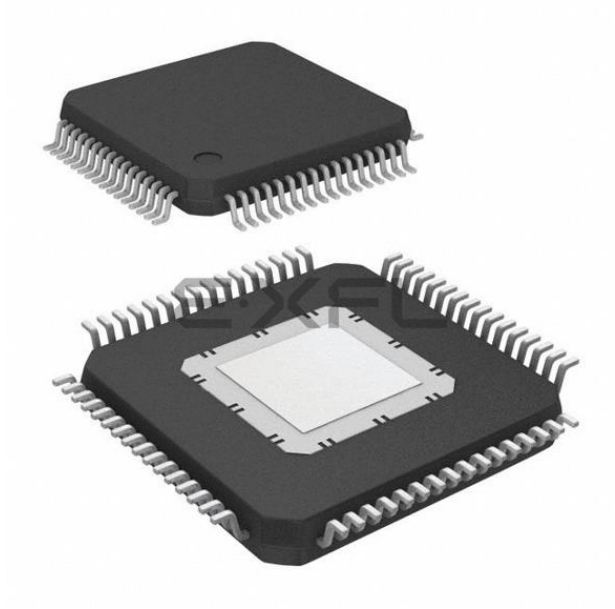


Welcome to E-XFL.COM

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	Obsolete
Core Processor	S12Z
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	512 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/s912zvml64f1wkhr

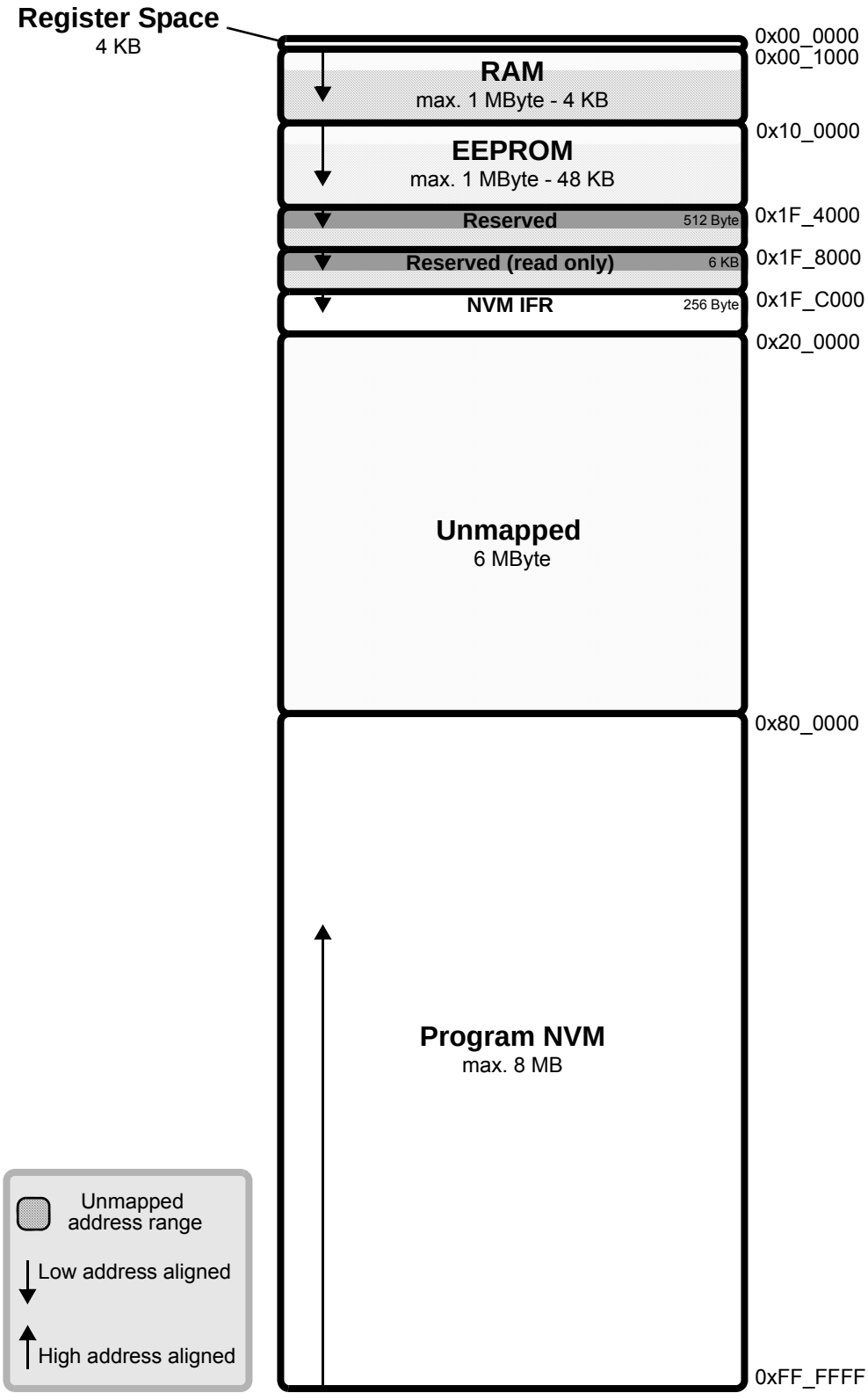


Figure 1-2. MC9S12ZVM-Family Global Memory Map. (See Table 1-3 for individual device details)

Table 2-4. Port AD Pin Functions and Priorities

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
AD	PAD15	✔					PDOCLK	O	DBG profiling clock	—	GPIO
		✔					AN0_7	I	ADC0 analog input	—	
		✔					PTADH[7]/ KWADH[7]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD14	✔					PDO	O	DBG profiling data output	—	
		✔					AN0_6	I	ADC0 analog input	—	
		✔					PTADH[6]/ KWADH[6]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD13	✔					PTURE	O	PTU reload event	—	
		✔					AN0_5	I	ADC0 analog input	—	
		✔					PTADH[5]/ KWADH[5]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD12	✔					AN1_7	I	ADC1 analog input	—	
		✔					PTADH[4]/ KWADH[4]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD11	✔					AN1_6	I	ADC1 analog input	—	
		✔					PTADH[3]/ KWADH[3]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD10	✔					AN1_5	I	ADC1 analog input	—	
		✔					PTADH[2]/ KWADH[2]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD9	✔					AN1_4	I	ADC1 analog input	—	
		✔					PTADH[1]/ KWADH[1]	I/O	General-purpose; with interrupt and wakeup	—	
	PAD8		✔	✔	✔	✔	VRH	I	ADC0&1 voltage reference high	—	
		✔	✔	✔	✔	✔	AN1_3	I	ADC1 analog input	—	
		✔	✔	✔	✔	✔	PTADH[0]/ KWADH[0]	I/O	General-purpose; with interrupt and wakeup	—	

2.3 Memory Map and Register Definition

This section provides a detailed description of all port integration module registers.

Subsection 2.3.1 shows all registers and bits at their related addresses within the global SoC register map. A detailed description of every register bit is given in subsection 2.3.2 to 2.3.4.

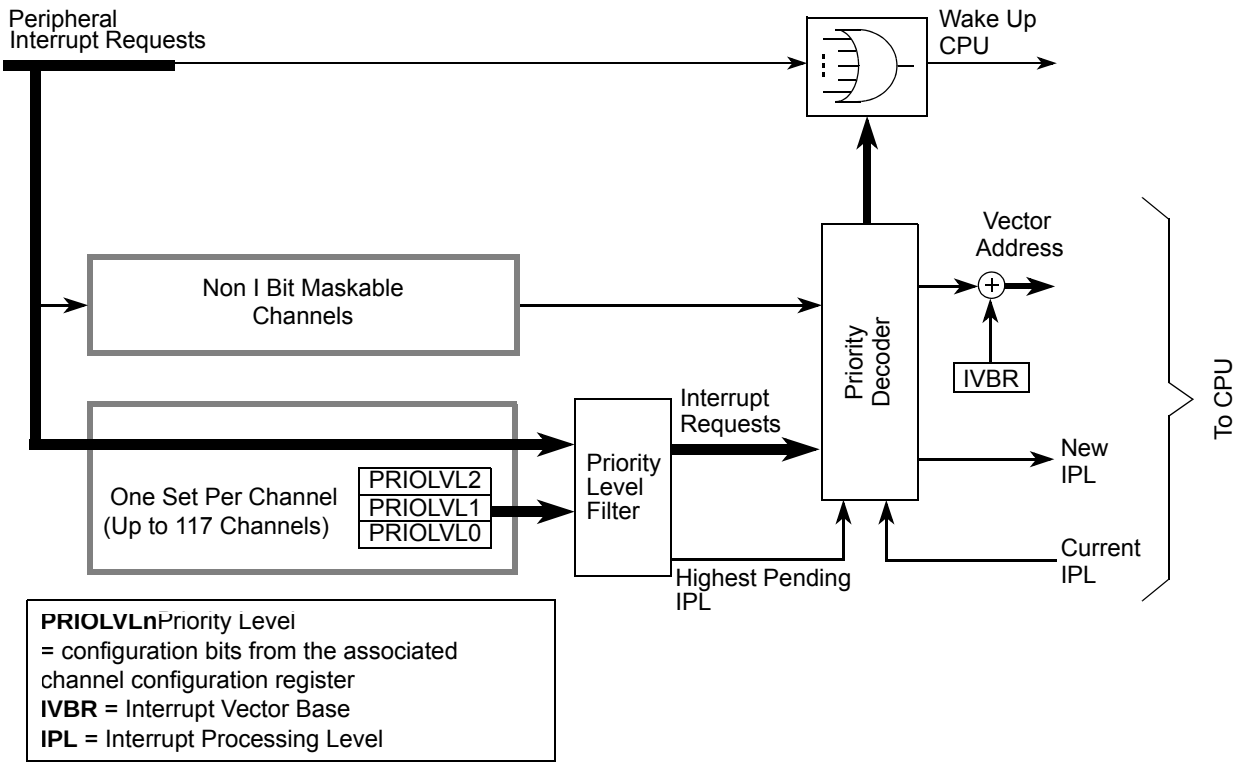


Figure 4-1. INT Block Diagram

4.2 External Signal Description

The INT module has no external signals.

4.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the INT module.

4.3.1 Module Memory Map

Table 4-3 gives an overview over all INT module registers.

Table 4-3. INT Memory Map

Address	Use	Access
0x000010–0x000011	Interrupt Vector Base Register (IVBR)	R/W
0x000012–0x000016	RESERVED	—
0x000017	Interrupt Request Configuration Address Register (INT_CFADDR)	R/W
0x000018	Interrupt Request Configuration Data Register 0 (INT_CFDATA0)	R/W

Chapter 5
Background Debug Controller (S12ZBDCV2)

Table 5-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V2.04	03.Dec.2012	Section 5.1.3.3	Included BACKGROUND/ Stop mode dependency
V2.05	22.Jan.2013	Section 5.3.2.2	Improved NORESP description and added STEP1/ Wait mode dependency
V2.06	22.Mar.2013	Section 5.3.2.2	Improved NORESP description of STEP1/ Wait mode dependency
V2.07	11.Apr.2013	Section 5.1.3.3.1	Improved STOP and BACKGROUND interdepency description
V2.08	31.May.2013	Section 5.4.4.4 Section 5.4.7.1	Removed misleading WAIT and BACKGROUND interdepency description Added subsection dedicated to Long-ACK
V2.09	29.Aug.2013	Section 5.4.4.12	Noted that READ_DBGTB is only available for devices featuring a trace buffer.
V2.10	21.Oct.2013	Section 5.1.3.3.2	Improved description of NORESP dependence on WAIT and BACKROUND
V2.11	02.Feb.2015	Section 5.1.3.3.1 Section 5.3.2	Corrected name of clock that can stay active in Stop mode

5.1 Introduction

The background debug controller (BDC) is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. The device BKGD pin interfaces directly to the BDC.

The S12ZBDC maintains the standard S12 serial interface protocol but introduces an enhanced handshake protocol and enhanced BDC command set to support the linear instruction set family of S12Z devices and offer easier, more flexible internal resource access over the BDC serial interface.

5.1.1 Glossary

Table 5-2. Glossary Of Terms

Term	Definition
DBG	On chip Debug Module
BDM	Active Background Debug Mode
CPU	S12Z CPU
SSC	Special Single Chip Mode (device operating mode
NSC	Normal Single Chip Mode (device operating mode)
BDCSI	Background Debug Controller Serial Interface. This refers to the single pin BKGD serial interface.
EWAIT	Optional S12 feature which allows external devices to delay external accesses until deassertion of EWAIT

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.

in bytes[6:4], the other payload bytes may be compressed or complete addresses as indicated by the info byte bits.

Table 6-57. Pure PC Mode Trace Buffer Format Single Source

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU	CXINF	BASE	BASE	BASE	PLB3	PLB2	PLB1	PLB0

If the info bit for byte3 indicates a full CPU PC address, whereby bytes[5:3] are used, then the info bit mapped to byte[4] is redundant and the byte[6] is unused because a line overflow has occurred. Similarly a base address stored in bytes[4:2] causes line overflow, so bytes[6:5] are unused.

CXINF[6:4] indicate how many bytes in a line contain valid data, since tracing may terminate before a complete line has been filled.

CXINF Information Byte Source Tracing

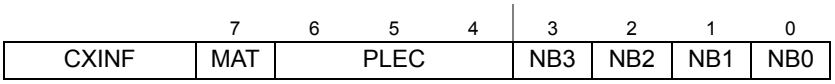


Figure 6-29. Pure PC Mode CXINF

Table 6-58. CXINF Field Descriptions

Field	Description
MAT	**Mid Aligned Trigger**— This bit indicates a mid aligned trigger position. When a mid aligned trigger occurs, the next trace buffer entry is a base address and the counter is incremented to a new line, independent of the number of bytes used on the current line. The MAT bit is set on the current line, to indicate the position of the trigger. When configured for begin or end aligned trigger, this bit has no meaning. NOTE: In the case when ARM and TRIG are simultaneously set together in the same cycle that a new PC value is registered, then this PC is stored to the same trace buffer line and MAT set. 0 Line filled without mid aligned trigger occurrence 1 Line last entry is the last PC entry before a mid aligned trigger
PLEC[2:0]	**Payload Entry Count**— This field indicates the number of valid bytes in the trace buffer line Binary encoding is used to indicate up to 7 valid bytes.
NBx	**Payload Compression Indicator**— This field indicates if the corresponding payload byte is the lowest byte of a base PC entry 0 Corresponding payload byte is a not the lowest byte of a base PC entry 1 Corresponding payload byte is the lowest byte of a base PC entry

Pure PC mode tracing does not support timestamps or external event entries.

6.4.5.3 Timestamp

When set, the STAMP bit in DBGTCRL configures the DBG to add a timestamp to trace buffer entries in Normal, Loop1 and Detail trace buffer modes. The timestamp is generated from a 16-bit counter and is stored to the trace buffer line each time a trace buffer entry is made.

If XOSCLCP is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

If XOSCLCP is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

The REFFRQ[1:0] bits are used to configure the internal PLL filter for optimal stability and lock time. For correct PLL operation the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Table 8-4.

If IRC1M is selected as REFCLK (OSCE=0) the PLL filter is fixed configured for the 1MHz <= f_{REF} <= 2MHz range. The bits can still be written but will have no effect on the PLL filter configuration.

For OSCE=1, setting the REFFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

Table 8-4. Reference Clock Frequency Selection if OSC_LCP is enabled

REFCLK Frequency Ranges (OSCE=1)	REFFRQ[1:0]
1MHz <= f_{REF} <= 2MHz	00
2MHz < f_{REF} <= 6MHz	01
6MHz < f_{REF} <= 12MHz	10
f_{REF} >12MHz	11

8.3.2.10 S12CPMU_UHV_V10_V6 PLL Control Register (CPMUPLL)

This register controls the PLL functionality.

Module Base + 0x000A

	7	6	5	4	3	2	1	0
R	0	0	FM1	FM0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Figure 8-15. S12CPMU_UHV_V10_V6 PLL Control Register (CPMUPLL)

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

NOTE

Write to this register clears the LOCK and UPOSC status bits.

NOTE

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

Table 8-9. CPMUPLL Field Descriptions

Field	Description
5, 4 FM1, FM0	PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is f_{ref} divided by 16. See Table 8-10 for coding.

Table 8-10. FM Amplitude selection

FM1	FM0	FM Amplitude / f_{VCO} Variation
0	0	FM off
0	1	±1%
1	0	±2%
1	1	±4%

The procedure to change APICLK or APIR[15:0] is first to clear APIFE, then write to APICLK or APIR[15:0], and afterwards set APIFE.

The API Trimming bits ACLKTR[5:0] must be set so the minimum period equals 0.2 ms if stable frequency is desired.

See Table 8-21 for the trimming effect of ACLKTR[5:0].

NOTE

The first period after enabling the counter by APIFE might be reduced by API start up delay t_{sdel}.

It is possible to generate with the API a waveform at the external pin API_EXTCLK by setting APIFE and enabling the external access with setting APIEA.

8.7 Initialization/Application Information

8.7.1 General Initialization Information

Usually applications run in MCU Normal Mode.

It is recommended to write the CPMUCOP register in any case from the application program initialization routine after reset no matter if the COP is used in the application or not, even if a configuration is loaded via the flash memory after reset. By doing a “controlled” write access in MCU Normal Mode (with the right value for the application) the write once for the COP configuration bits (WCOP,CR[2:0]) takes place which protects these bits from further accidental change. In case of a program sequencing issue (code runaway) the COP configuration can not be accidentally modified anymore.

8.7.2 Application information for COP and API usage

In many applications the COP is used to check that the program is running and sequencing properly. Often the COP is kept running during Stop Mode and periodic wake-up events are needed to service the COP on time and maybe to check the system status.

For such an application it is recommended to use the ACLK as clock source for both COP and API. This guarantees lowest possible IDD current during Stop Mode. Additionally it eases software implementation using the same clock source for both, COP and API.

The Interrupt Service Routine (ISR) of the Autonomous Periodic Interrupt API should contain the write instruction to the CPMUARMCOP register. The value (byte) written is derived from the “main routine” (alternating sequence of $55 and $AA) of the application software.

Using this method, then in the case of a runtime or program sequencing issue the application “main routine” is not executed properly anymore and the alternating values are not provided properly. Hence the COP is written at the correct time (due to independent API interrupt request) but the wrong value is written (alternating sequence of $55 and $AA is no longer maintained) which causes a COP reset.

If the COP is stopped during any Stop Mode it is recommended to service the COP shortly before Stop Mode is entered.

10.3 Memory Map and Register Definition

This section provides the detailed information of all registers for the BATS module.

10.3.1 Register Summary

Figure 10-2 shows the summary of all implemented registers inside the BATS module.

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 BATE	R	0	BVHS	BVLS[1:0]		BSUAE	BSUSE	0	0
	W								
0x0001 BATSR	R	0	0	0	0	0	0	BVHC	BVLC
	W								
0x0002 BATIE	R	0	0	0	0	0	0	BVHIE	BVLIE
	W								
0x0003 BATIF	R	0	0	0	0	0	0	BVHIF	BVLIF
	W								
0x0004 - 0x0005 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0006 - 0x0007 Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	W								

= Unimplemented

Figure 10-2. BATS Register Summary

10.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order. Unused bits read back zero.

10.3.2.3 BATS 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 10-6. BATS Interrupt Enable Register (BATIE)

1. Read: Anytime
   Write: Anytime

Table 10-4. 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 .

10.3.2.4 BATS 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 10-7. BATS Interrupt Flag Register (BATIF)

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

13.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the MSCAN.

13.3.1 Module Memory Map

Figure 13-3 gives an overview on all registers and their individual bits in the MSCAN memory map. The *register address* results from the addition of *base address* and *address offset*. The *base address* is determined at the MCU level and can be found in the MCU memory map description. The *address offset* is defined at the module level.

The MSCAN occupies 64 bytes in the memory space. The base address of the MSCAN module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset.

The detailed register descriptions follow in the order they appear in the register map.

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)”).

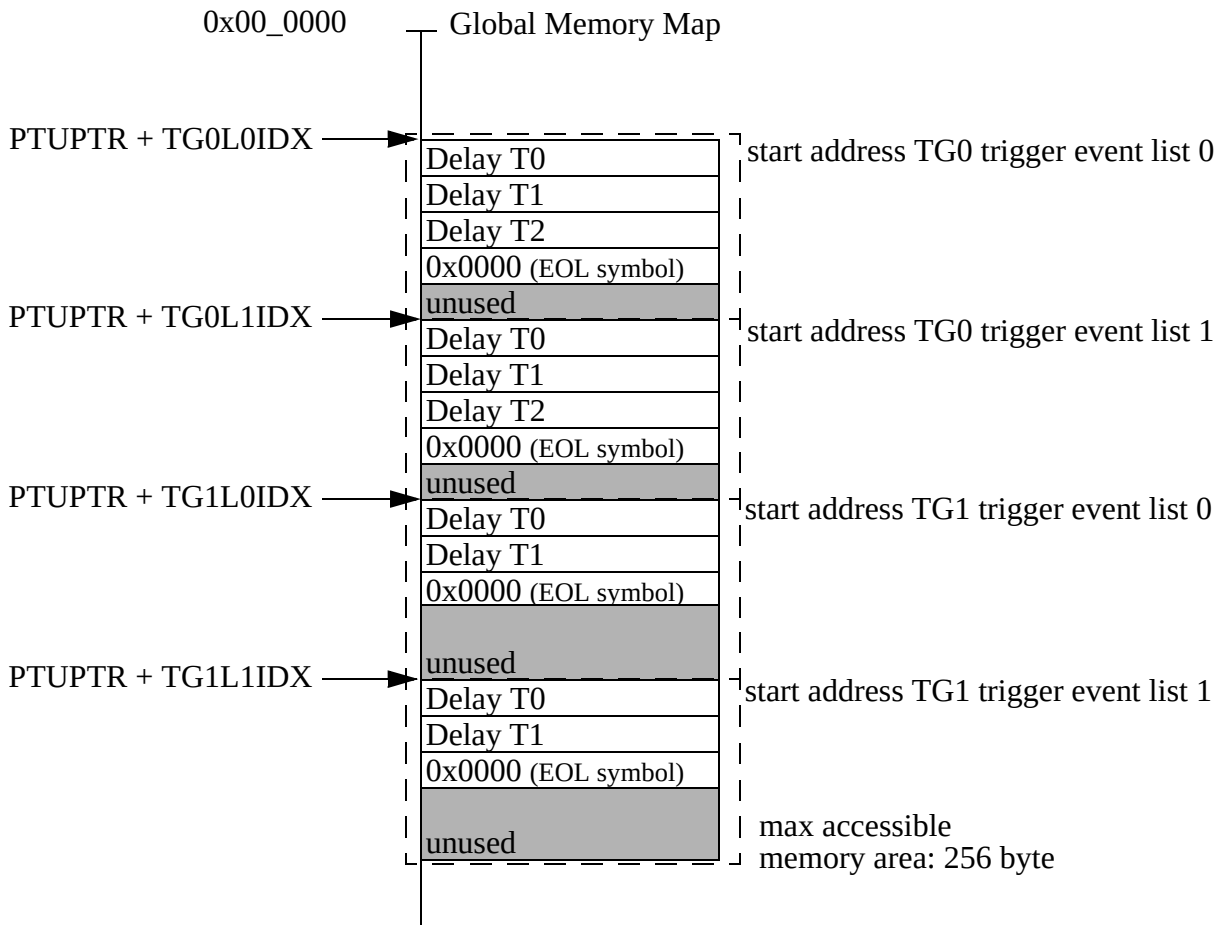
14.4.2 Memory based trigger event list

The lists with the trigger values are located inside the global memory map. The location of the trigger lists in the memory map is configured with registers PTUPTR and TGxLxIDX. If one of the TGs is enabled then the PTUPTR register is locked. If the TG is enabled then the associated TGxLxIDX registers are locked.

The trigger values inside the trigger list are 16 bit values. Each 16 bit value defines the delay between the reload event and the trigger event in bus clock cycles. A delay value of 0x0000 will be interpreted as End Of trigger List (EOL) symbol. The list must be sorted in ascending order. If a subsequent value is smaller than the previous value or the loaded trigger value is smaller than the current counter value then the TGxTEIF error indication is generated and the trigger generation of this list is stopped until the next reload event. For more information about these error scenario see Section 14.4.5.5, “Trigger Generator Timing Error”.

The module is not able to access memory area outside the 256 byte window starting at the memory address defined by PTUPTR.

Figure 14-23. Global Memory map usage



14.4.3 Reload mechanism

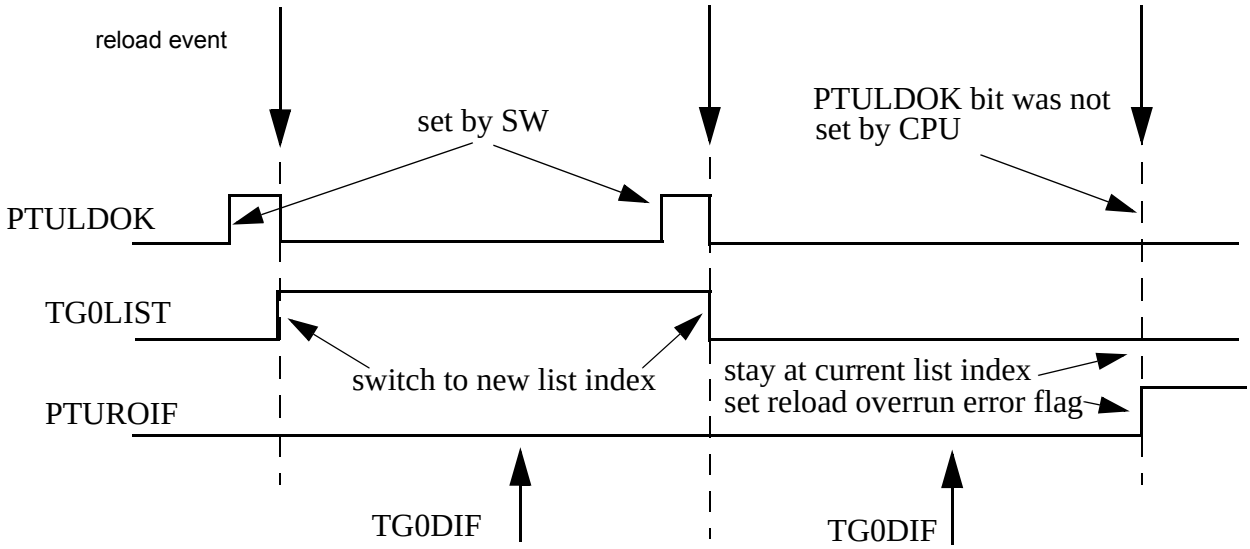
Each trigger generator uses two lists to load the trigger values from the memory. One list can be updated by the CPU while the other list is used to generate the trigger events. After enabling, the TG uses the lists in alternate order. When the update of alternate trigger list is done, the SW must set the PTULDOK bit. If the load OK bit is set at the time of reload event, the TG switches to the alternate list and loads the first trigger value from this trigger event list. The reload event clears the PTULDOK bit.

The TG0LIST and TG1LIST bits shows the currently use list number. These bits are writeable if the associated TG is disabled.

If the PTULDOK bit was not set before the reload event then the reload overrun error flag is set (PTUROIF)and both TGs do not switch to the alternative list. The current trigger list is used to load the trigger values. Figure 14-24 shows an example. The PTULDOK bit can be used by other modules as glb_ldok.

To reduce the used memory size, it is also possible to set TG0L0IDX equal to TG0L1IDX or to set TG1L0IDX equal to TG1L1IDX. In this case the trigger generator is using only one physical list of trigger events even if the trigger generator logic is switching between both pointers. The SW must make sure, that the CPU does not update the trigger list before the execution of the trigger list is done. The time window to update the trigger list starts at the trigger generator done interrupt flag (TGxDIF) and ends with the next reload event. Even if only one physical trigger event list is used the TGxLIST shows a swap between list 0 and 1 at every reload event with set PTULDOK bit.

Figure 14-24. TG0 Reload behavior with local PTULDOK



14.4.4 Async reload event

If the reload and reload_is_async are active at the same time then an async reload event happens. The PTU behavior on an async reload event is the same like on the reload event described in Section 14.4.3, “Reload

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0032	PMFCNTC	R	0	PMFCNTC						
		W								
0x0033	PMFCNTC	R	PMFCNTC							
		W								
0x0034	PMFMODC	R	0	PMFMODC						
		W								
0x0035	PMFMODC	R	PMFMODC							
		W								
0x0036	PMFDTMC	R	0	0	0	0	PMFDTMC			
		W								
0x0037	PMFDTMC	R	PMFDTMC							
		W								
0x0038	PMFDMP0	R	DMP05		DMP04		DMP03	DMP02	DMP01	DMP00
		W								
0x0039	PMFDMP1	R	DMP15		DMP14		DMP13	DMP12	DMP11	DMP10
		W								
0x003A	PMFDMP2	R	DMP25		DMP24		DMP23	DMP22	DMP21	DMP20
		W								
0x003B	PMFDMP3	R	DMP35		DMP34		DMP33	DMP32	DMP31	DMP30
		W								
0x003C	PMFDMP4	R	DMP45		DMP44		DMP43	DMP42	DMP41	DMP40
		W								
0x003D	PMFDMP5	R	DMP55		DMP54		DMP53	DMP52	DMP51	DMP50
		W								
0x003E	PMFOUTF	R	0	0	OUTF5	OUTF4	OUTF3	OUTF2	OUTF1	OUTF0
		W								
0x003F	Reserved	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

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

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI control register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the nth[1] shift, the transfer is considered complete and the received data is transferred into the SPI data register. To indicate transfer is complete, the SPIF flag in the SPI status register is set.

NOTE

A change of the bits CPOL, CPHA, SSOE, LSBFE, MODFEN, SPC0, or BIDIROE with SPC0 set in slave mode will corrupt a transmission in progress and must be avoided.

17.4.3 Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

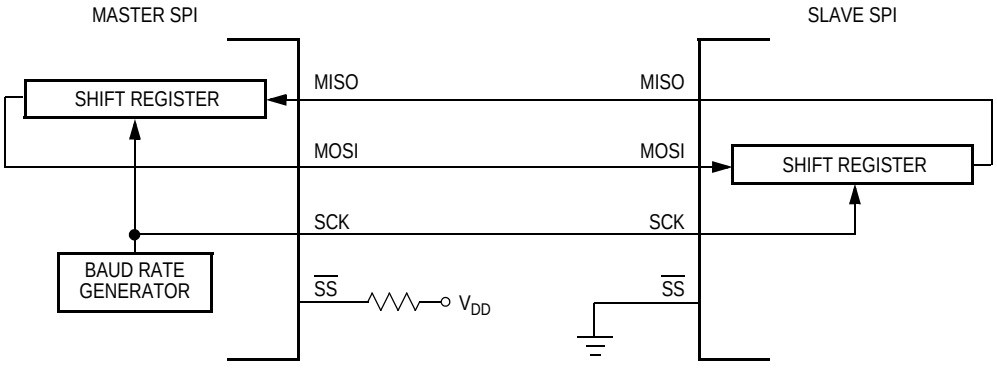


Figure 17-11. Master/Slave Transfer Block Diagram

17.4.3.1 Clock Phase and Polarity Controls

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

1. n depends on the selected transfer width, please refer to Section 17.3.2.2, "SPI Control Register 2 (SPICR2)

1. Outside of the given V_{HVI} range the error is significant. The ratio can be changed, if outside of the given range.

A.2.2 HV Physical Interface Characteristics

The HV Physical Interface specification is included in the LINPHY electrical section.

A.3 Supply Currents

This section describes the current consumption characteristics of the device as well as the conditions for the measurements.

A.3.1 Measurement Conditions

Current is measured on VSUP. VDDX is connected to VDDA. It does not include the current to drive external loads. Unless otherwise noted the currents are measured in special single chip mode and the CPU code is executed from RAM. For Run and Wait current measurements PLL is on and the reference clock is the IRC1M trimmed to 1MHz. For the junction temperature range from -40°C to +150°C the bus frequency is 50MHz. For the temperature range from +150°C to +175°C, the bus frequency is 40MHz. Table A-15, Table A-16 and Table A-17 show the configuration of the CPMU module and the peripherals for Run, Wait and Stop current measurement.

Table A-15. CPMU Configuration for Pseudo Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUCLKS	PLLSEL=0, PSTP=1, CSAD=0, PRE=PCE=RTIOSCSEL=1 COPOSCSEL[1:0]=01
CPMUOSC	OSCE=1, Quartz oscillator f_{EXTAL}=4MHz
CPMURTI	RTDEC=0, RTR[6:4]=111, RTR[3:0]=1111
CPMUCOP	WCOP=1, CR[2:0]=111

Table A-16. CPMU Configuration for Run/Wait and Full Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUSYNR	VCOFRQ[1:0]= 3,SYNDIV[5:0] = 49
CPMUPOSTDIV	POSTDIV[4:0]=0
CPMUCLKS	PLLSEL=1, CSAD=0
CPMUOSC	OSCE=0, Reference clock for PLL is f_{ref}=f_{irc1m} trimmed to 1MHz
CPMUVREGCTL	EXTXON=0, INTXON=1

How to Reach Us:

Home Page:

nxp.com

Web Support

nxp.com/support

Document Number: MC9S12ZVMRM
Rev. 2.11
October 28, 2016