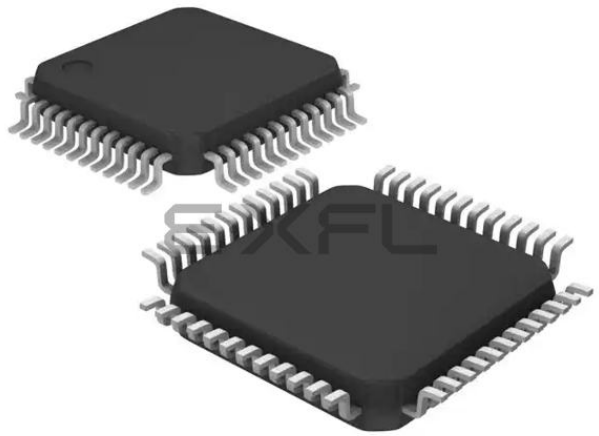


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	S12Z
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12zvl32f0vlfr

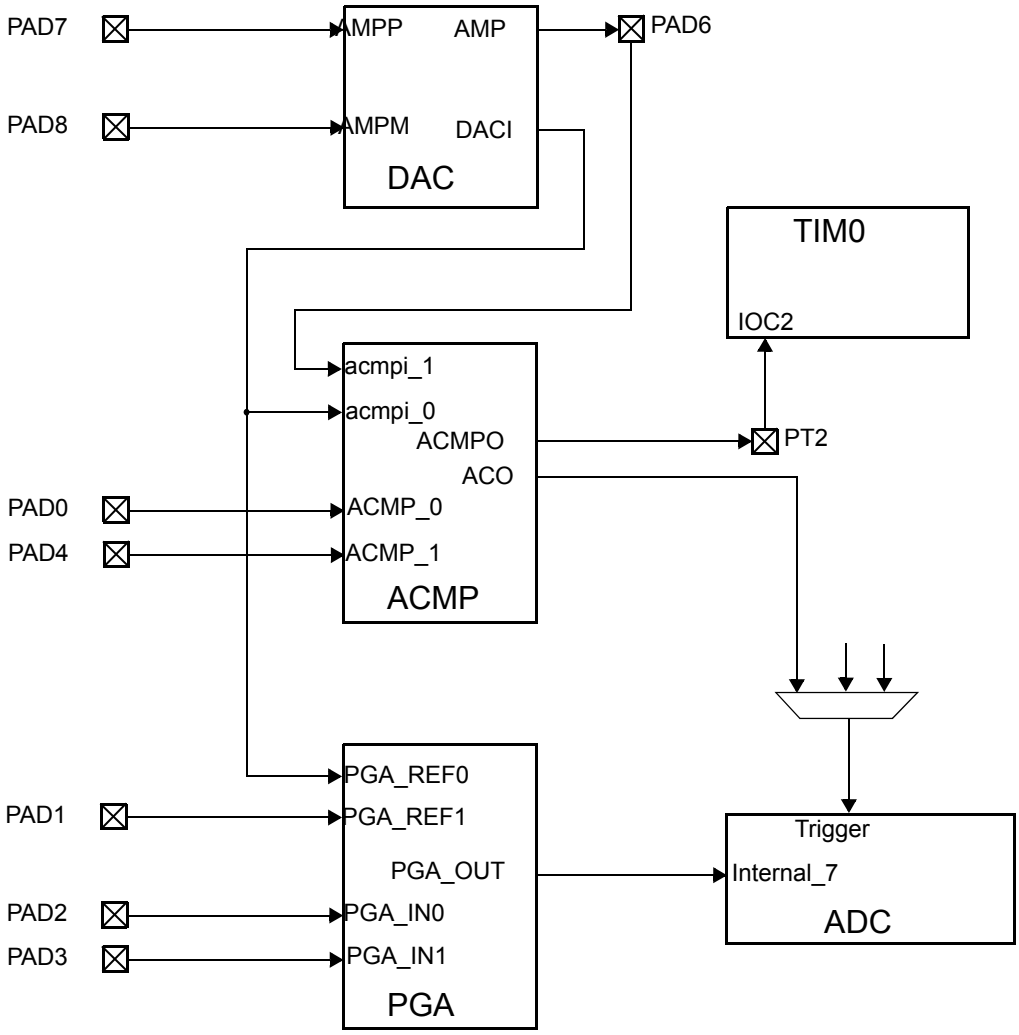


Figure 1-6. MC9S12ZVLA ACMP - PGA - DAC - ADC Connectivity

On the MC9S12ZVLA device follow ADC option bit decoding is used.

Table 1-8. ADC option bit decoding

ADC option bit OPT[1:0]	PGA input source selection
2’b00	no input selected
2’b01	PGA_IN0 is used as input voltage
2’b10	PGA_IN1 is used as input voltage
other	Reserved

- All illegal accesses performed by the ADC module trigger error interrupts. See ADC section for details.

NOTE

Illegal accesses caused by S12ZCPU opcode prefetches will also trigger machine exceptions, even if those opcodes might not be executed in the program flow. To avoid these machine exceptions, S12ZCPU instructions must not be executed from the last (high addresses) 8 bytes of RAM, EEPROM, and Flash.

4.4.3 Uncorrectable ECC Faults

RAM and flash use error correction codes (ECC) to detect and correct memory corruption. Each uncorrectable memory corruption, which is detected during a S12ZCPU or ADC access triggers a machine exception. Uncorrectable memory corruptions which are detected during a S12ZBDC access, are captured in the RAMWF or the RDINV bit of the BDCCSRL register.

Table 5-8. BDC Command Summary (continued)

Command Mnemonic	Command Classification	ACK	Command Structure	Description
DUMP_MEM.sz	Non-Intrusive	Yes	(0x32+4 x sz)/dack/rd.sz	Dump (read) memory based on operand size (sz). Used with READ_MEM to dump large blocks of memory. An initial READ_MEM is executed to set up the starting address of the block and to retrieve the first result. Subsequent DUMP_MEM commands retrieve sequential operands.
DUMP_MEM.sz_WS	Non-Intrusive	No	(0x33+4 x sz)/d/ss/rd.sz	Dump (read) memory based on operand size (sz) and report status. Used with READ_MEM{_WS} to dump large blocks of memory. An initial READ_MEM{_WS} is executed to set up the starting address of the block and to retrieve the first result. Subsequent DUMP_MEM{_WS} commands retrieve sequential operands.
FILL_MEM.sz	Non-Intrusive	Yes	(0x12+4 x sz)/wd.sz/dack	Fill (write) memory based on operand size (sz). Used with WRITE_MEM to fill large blocks of memory. An initial WRITE_MEM is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM commands write sequential operands.
FILL_MEM.sz_WS	Non-Intrusive	No	(0x13+4 x sz)/wd.sz/d/ss	Fill (write) memory based on operand size (sz) and report status. Used with WRITE_MEM{_WS} to fill large blocks of memory. An initial WRITE_MEM{_WS} is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM{_WS} commands write sequential operands.
GO	Active Background	Yes	0x08/dack	Resume CPU user code execution
GO_UNTIL[2]	Active Background	Yes	0x0C/dack	Go to user program. ACK is driven upon returning to active background mode.
NOP	Non-Intrusive	Yes	0x00/dack	No operation
READ_Rn	Active Background	Yes	(0x60+CRN)/dack/rd32	Read the requested CPU register
READ_MEM.sz	Non-Intrusive	Yes	(0x30+4 x sz)/ad24/dack/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address
READ_MEM.sz_WS	Non-Intrusive	No	(0x31+4 x sz)/ad24/d/ss/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address and report status
READ_DBGTB	Non-Intrusive	Yes	(0x07)/dack/rd32/dack/rd32	Read 64-bits of DBG trace buffer

Table 6-2. Terminology

Term	Meaning
MCU	Micro-Controller Unit
$\overline{\text{IRQ}}$	refers to the interrupt request associated with the $\overline{\text{IRQ}}$ pin
$\overline{\text{XIRQ}}$	refers to the interrupt request associated with the $\overline{\text{XIRQ}}$ pin

6.1.2 Features

- Interrupt vector base register (IVBR)
- One system reset vector (at address 0xFFFFFC).
- One non-maskable unimplemented page1 op-code trap (SPARE) vector (at address vector base[1] + 0x0001F8).
- One non-maskable unimplemented page2 op-code trap (TRAP) vector (at address vector base[1] + 0x0001F4).
- One non-maskable software interrupt request (SWI) vector (at address vector base[1] + 0x0001F0).
- One non-maskable system call interrupt request (SYS) vector (at address vector base[1] + 0x00001EC).
- One non-maskable machine exception vector request (at address vector base[1] + 0x0001E8).
- One spurious interrupt vector (at address vector base[1] + 0x0001DC).
- One X-bit maskable interrupt vector request associated with $\overline{\text{XIRQ}}$ (at address vector base[1] + 0x0001D8).
- One I-bit maskable interrupt vector request associated with $\overline{\text{IRQ}}$ (at address vector base[1] + 0x0001D4).
- up to 113 additional I-bit maskable interrupt vector requests (at addresses vector base[1] + 0x000010 .. vector base + 0x0001D0).
- Each I-bit maskable interrupt request has a configurable priority level.
- I-bit maskable interrupts can be nested, depending on their priority levels.
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs or whenever $\overline{\text{XIRQ}}$ is asserted, even if X interrupt is masked.

6.1.3 Modes of Operation

- Run mode

  This is the basic mode of operation.
- Wait mode

  In wait mode, the S12ZINTV0 module is capable of waking up the CPU if an eligible CPU exception occurs. Please refer to Section 6.5.3, “Wake Up from Stop or Wait Mode” for details.
- Stop Mode

1. The vector base is a 24-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as the upper 15 bits of the address) and 0x000 (used as the lower 9 bits of the address).

Address: 0x000019

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

= Unimplemented or Reserved

Figure 6-6. Interrupt Request Configuration Data Register 1 (INT_CFDATA1)

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

Address: 0x00001A

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

= Unimplemented or Reserved

Figure 6-7. Interrupt Request Configuration Data Register 2 (INT_CFDATA2)

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

Address: 0x00001B

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

= Unimplemented or Reserved

Figure 6-8. Interrupt Request Configuration Data Register 3 (INT_CFDATA3)

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

Address: 0x00001C

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

= Unimplemented or Reserved

Figure 6-9. Interrupt Request Configuration Data Register 4 (INT_CFDATA4)

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

Chapter 7
S12Z DebugLite (S12ZDBGV3)

Table 7-1. Revision History Table

Revision Number	Revision Date	Sections Affected	Description Of Changes
3.02	05.JUL.2012	Section 7.3.2.6, “Debug Event Flag Register (DBGEFR)	Removed ME2 flag from DBGEFR
3.03	16.NOV.2012	Section 7.5.1, “Avoiding Unintended Breakpoint Re-triggering	Modified step over breakpoint information
3.04	19.DEC.2012	General	Formatting corrections
3.05	19.APR.2013	General	Specified DBGC1[0] reserved bit as read only
3.06	15.JUL.2013	Section 7.3.2, “Register Descriptions	Added explicit names to state control register bit fields

7.1 Introduction

The DBG module provides on-chip breakpoints with flexible triggering capability to allow non-intrusive debug of application software. The DBG module is optimized for the S12Z architecture and allows debugging of CPU module operations.

Typically the DBG module is used in conjunction with the BDC module, whereby the user configures the DBG module for a debugging session over the BDC interface. Once configured the DBG module is armed and the device leaves active BDM returning control to the user program, which is then monitored by the DBG module. Alternatively the DBG module can be configured over a serial interface using SWI routines.

7.1.1 Glossary

Table 7-2. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
PC	Program Counter

- Through comparator matches via Final State.
- Through software writing to the TRIG bit in the DBGC1 register via Final State.
- Through the external event input (DBGEEV) via Final State.

Breakpoints are not generated by software writes to DBGC1 that clear the ARM bit.

7.4.5.1 Breakpoints From Comparator Matches or External Events

Breakpoints can be generated when the state sequencer transitions to State0 following a comparator match or an external event.

7.4.5.2 Breakpoints Generated Via The TRIG Bit

When TRIG is written to “1”, the Final State is entered. In the next cycle TRIG breakpoints are possible even if the DBG module is disarmed.

7.4.5.3 DBG Breakpoint Priorities

7.4.5.3.1 DBG Breakpoint Priorities And BDC Interfacing

Breakpoint operation is dependent on the state of the S12ZBDC module. BDM cannot be entered from a breakpoint unless the BDC is enabled (ENBDC bit is set in the BDC). If BDM is already active, breakpoints are disabled. In addition, while executing a BDC STEP1 command, breakpoints are disabled.

When the DBG breakpoints are mapped to BDM (BDMBP set), then if a breakpoint request, either from a BDC BACKGROUND command or a DBG event, coincides with an SWI instruction in application code, (i.e. the DBG requests a breakpoint at the next instruction boundary and the next instruction is an SWI) then the CPU gives priority to the BDM request over the SWI request.

On returning from BDM, the SWI from user code gets executed. Breakpoint generation control is summarized in Table 7-32.

Table 7-32. Breakpoint Mapping Summary

BRKCPU	BDMBP Bit (DBGC1[4])	BDC Enabled	BDM Active	Breakpoint Mapping
0	X	X	X	No Breakpoint
1	0	X	0	Breakpoint to SWI
1	0	1	1	No Breakpoint
1	1	0	X	No Breakpoint
1	1	1	0	Breakpoint to BDM
1	1	1	1	No Breakpoint

9.3.2.9 S12CPMU_UHV Clock Select Register (CPMUCLKS)

This register controls S12CPMU_UHV clock selection.

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R W	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
Reset	1	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-12. S12CPMU_UHV Clock Select Register (CPMUCLKS)

Read: Anytime

Write:

- Only possible if PROT=0 (CPMUPROT register) in all MCU Modes (Normal and Special Mode).
- All bits in Special Mode (if PROT=0).
- PLLSEL, PSTP, PRE, PCE, RTIOSCSEL: In Normal Mode (if PROT=0).
- CSAD: In Normal Mode (if PROT=0) until CPMUCOP write once has taken place.
- COPOSCSEL0: In Normal Mode (if PROT=0) until CPMUCOP write once has taken place. If COPOSCSEL0 was cleared by UPOSC=0 (entering Full Stop Mode with COPOSCSEL0=1 or insufficient OSCCLK quality), then COPOSCSEL0 can be set once again.
- COPOSCSEL1: In Normal Mode (if PROT=0) until CPMUCOP write once has taken place. COPOSCSEL1 will not be cleared by UPOSC=0 (entering Full Stop Mode with COPOSCSEL1=1 or insufficient OSCCLK quality if OSCCLK is used as clock source for other clock domains: for instance core clock etc.).

NOTE

After writing CPMUCLKS register, it is strongly recommended to read back CPMUCLKS register to make sure that write of PLLSEL, RTIOSCSEL and COPOSCSEL was successful. This is because under certain circumstances writes have no effect or bits are automatically changed (see CPMUCLKS register and bit descriptions).

NOTE

When using the oscillator clock as system clock (write PLLSEL = 0) it is highly recommended to enable the oscillator clock monitor reset feature (write OMRE = 1 in CPMUOSC2 register). If the oscillator monitor reset feature is disabled (OMRE = 0) and the oscillator clock is used as system clock, the system might stall in case of loss of oscillation.

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.

9.7 Initialization/Application Information

9.7.1 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.

9.7.2 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.

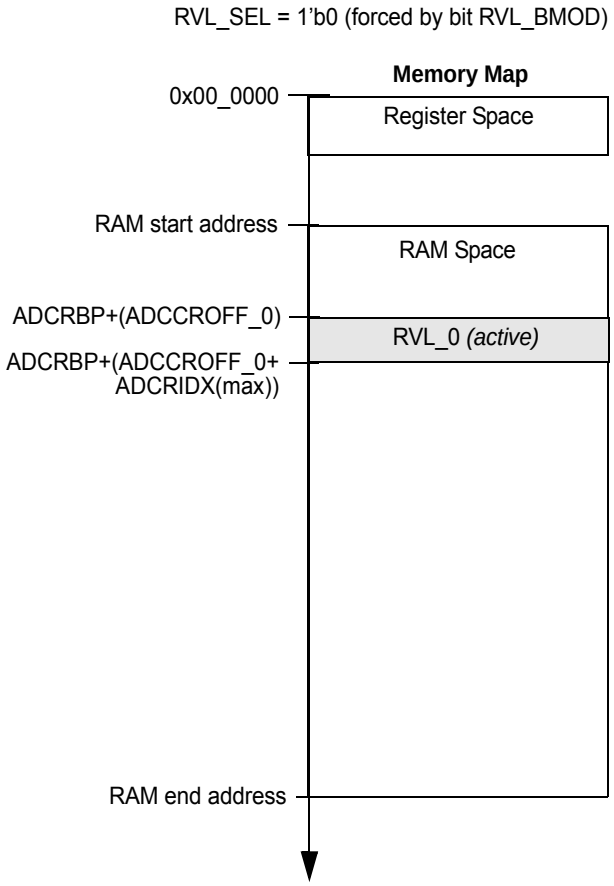
The Interrupt Service Routine (ISR) of the Autonomous Periodic Interrupt API should contain the write instruction to the CPMUARMCOP register. The value (byte) written is derived from the “main routine” (alternating sequence of $55 and $AA) of the application software.

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.

9.7.3 Application Information for PLL and Oscillator Startup

The following C-code example shows a recommended way of setting up the system clock system using the PLL and Oscillator:



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

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

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

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

Table 10-33. Conversion Result Justification Overview

Conversion Resolution (SRES[1:0])	Left Justified Result (DJM = 1’b0)	Right Justified Result (DJM = 1’b1)
8 bit	{Result[7:0],8’b00000000}	{8’b00000000,Result[7:0]}
10 bit	{Result[9:0],6’b000000}	{6’b000000,Result[9:0]}
12 bit	{Result[11:0],4’b0000}	{4’b0000,Result[11:0]}

10.9 Use Cases and Application Information

10.9.1 List Usage — CSL single buffer mode and RVL single buffer mode

In this use case both list types are configured for single buffer mode (CSL_BMOD=1’b0 and RVL_BMOD=1’b0, CSL_SEL and RVL_SEL are forced to 1’b0). The index register for the CSL and RVL are cleared to start from the top of the list with next conversion command and result storage in the following cases:

- The conversion flow reaches the command containing the “End-of-List” command type identifier
- A Restart Request occurs at a sequence boundary
- After an aborted conversion or conversion sequence

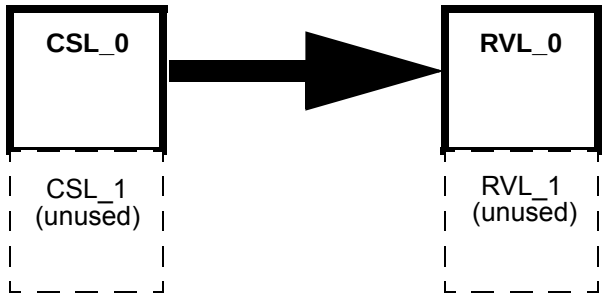


Figure 10-35. CSL Single Buffer Mode — RVL Single Buffer Mode Diagram

10.9.2 List Usage — CSL single buffer mode and RVL double buffer mode

In this use case the CSL is configured for single buffer mode (CSL_BMOD=1’b0) and the RVL is configured for double buffer mode (RVL_BMOD=1’b1). In this buffer configuration only the result list RVL is switched when the first conversion result of a CSL is stored after a CSL was successfully finished or a CSL got aborted.

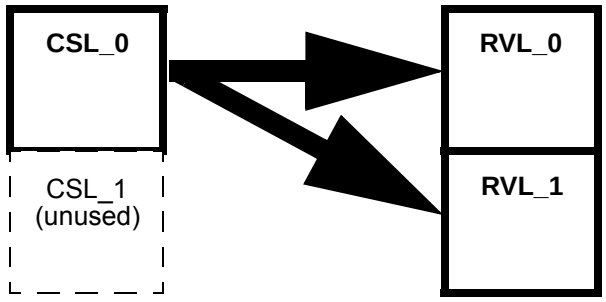


Figure 10-36. CSL Single Buffer Mode — RVL Single Buffer Mode Diagram

The last entirely filled RVL (an RVL where the corresponding CSL has been executed including the “End Of List “ command type) is shown by register ADCEOLRI.

The CSL is used in single buffer mode and bit CSL_SEL is forced to 1’b0.

Table 13-18. CANIDAC Register Field Descriptions

Field	Description
5-4 IDAM[1:0]	**Identifier Acceptance Mode** — The CPU sets these flags to define the identifier acceptance filter organization (see Section 13.4.3, “Identifier Acceptance Filter”). Table 13-19 summarizes the different settings. In filter closed mode, no message is accepted such that the foreground buffer is never reloaded.
2-0 IDHIT[2:0]	Identifier Acceptance Hit Indicator — The MSCAN sets these flags to indicate an identifier acceptance hit (see Section 13.4.3, “Identifier Acceptance Filter”). Table 13-20 summarizes the different settings.

Table 13-19. Identifier Acceptance Mode Settings

IDAM1	IDAM0	Identifier Acceptance Mode
0	0	Two 32-bit acceptance filters
0	1	Four 16-bit acceptance filters
1	0	Eight 8-bit acceptance filters
1	1	Filter closed

Table 13-20. Identifier Acceptance Hit Indication

IDHIT2	IDHIT1	IDHIT0	Identifier Acceptance Hit
0	0	0	Filter 0 hit
0	0	1	Filter 1 hit
0	1	0	Filter 2 hit
0	1	1	Filter 3 hit
1	0	0	Filter 4 hit
1	0	1	Filter 5 hit
1	1	0	Filter 6 hit
1	1	1	Filter 7 hit

The IDHITx indicators are always related to the message in the foreground buffer (RxFG). When a message gets shifted into the foreground buffer of the receiver FIFO the indicators are updated as well.

13.3.2.13 MSCAN Reserved Registers

These registers are reserved for factory testing of the MSCAN module and is not available in normal system operating modes.

16.3.2.13 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 16-20. Output Compare Pin Disconnect Register (OCPD)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 16-15. 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.

16.3.2.14 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 16-21. Precision Timer Prescaler Select Register (PTPSR)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 16-16. PTPSR Field Descriptions

Field	Description
7:0 PTPS[7:0]	**Precision Timer Prescaler Select Bits** — These eight bits specify the division rate of the main Timer prescaler. These are effective only when the PRNT bit of TSCR1 is set to 1. Table 16-17 shows some selection examples in this case. The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

18.5.3.1.5 IDLE Description

The IDLE interrupt is set when 10 consecutive logic 1s (if M = 0) or 11 consecutive logic 1s (if M = 1) appear on the receiver input. Once the IDLE is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag. Clear IDLE by reading SCI status register 1 (SCISR1) with IDLE set and then reading SCI data register low (SCIDRL).

18.5.3.1.6 RXEDGIF Description

The RXEDGIF interrupt is set when an active edge (falling if RXPOL = 0, rising if RXPOL = 1) on the RXD pin is detected. Clear RXEDGIF by writing a “1” to the SCIASR1 SCI alternative status register 1.

18.5.3.1.7 BERRIF Description

The BERRIF interrupt is set when a mismatch between the transmitted and the received data in a single wire application like LIN was detected. Clear BERRIF by writing a “1” to the SCIASR1 SCI alternative status register 1. This flag is also cleared if the bit error detect feature is disabled.

18.5.3.1.8 BKDIF Description

The BKDIF interrupt is set when a break signal was received. Clear BKDIF by writing a “1” to the SCIASR1 SCI alternative status register 1. This flag is also cleared if break detect feature is disabled.

18.5.4 Recovery from Wait Mode

The SCI interrupt request can be used to bring the CPU out of wait mode.

18.5.5 Recovery from Stop Mode

An active edge on the receive input can be used to bring the CPU out of stop mode.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0).

Table 22-6. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset
6 FDIVLCK	**Clock Divider Locked** 0 FDIV field is open for writing 1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field in normal mode.
5–0 FDIV[5:0]	**Clock Divider Bits** — FDIV[5:0] must be set to effectively divide BUSCLK down to 1 MHz to control timed events during Flash program and erase algorithms. Table 22-7 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Please refer to Section 22.4.5, “Flash Command Operations,” for more information.

Table 22-7. FDIV values for various BUSCLK Frequencies

BUSCLK Frequency (MHz)		FDIV[5:0]	BUSCLK Frequency (MHz)		FDIV[5:0]
MIN[1]	MAX[2]		MIN[1]	MAX[2]	
1.0	1.6	0x00	26.6	27.6	0x1A
1.6	2.6	0x01	27.6	28.6	0x1B
2.6	3.6	0x02	28.6	29.6	0x1C
3.6	4.6	0x03	29.6	30.6	0x1D
4.6	5.6	0x04	30.6	31.6	0x1E
5.6	6.6	0x05	31.6	32.6	0x1F
6.6	7.6	0x06	32.6	33.6	0x20
7.6	8.6	0x07	33.6	34.6	0x21
8.6	9.6	0x08	34.6	35.6	0x22
9.6	10.6	0x09	35.6	36.6	0x23
10.6	11.6	0x0A	36.6	37.6	0x24
11.6	12.6	0x0B	37.6	38.6	0x25
12.6	13.6	0x0C	38.6	39.6	0x26
13.6	14.6	0x0D	39.6	40.6	0x27
14.6	15.6	0x0E	40.6	41.6	0x28
15.6	16.6	0x0F	41.6	42.6	0x29
16.6	17.6	0x10	42.6	43.6	0x2A
17.6	18.6	0x11	43.6	44.6	0x2B

NOTE

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 22-59.

Table 22-59. Valid Set Field Margin Level Settings

FCCOB2	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[1]
0x0002	User Margin-0 Level[2]
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

1 Read margin to the erased state

2 Read margin to the programmed state

Table 22-60. Set Field Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 22-28)
		Set if an invalid global address [23:0] is supplied see Table 22-2)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level.

Table A-3. Absolute Maximum Ratings[1]

Num	Rating	Symbol	Min	Max	Unit
1	Voltage regulator and LINPHY supply voltage	V_{SUP}	-0.3	42	V
2	DC voltage on LIN	V_{LIN}	-32	42	V
3	Voltage Regulator Ballast Connection	V_{BCTL}	-0.3	42	V
4	Supplies VDDA, VDDX	V_{VDDACX}	-0.3	6	V
5	Voltage difference V_{DDX} to V_{DDA}[2]	Δ_{VDDX}	–0.3	0.3	V
6	Voltage difference V_{SSX} to V_{SSA}	Δ_{VSSX}	–0.3	0.3	V
7	Digital I/O input voltage	V_{IN}	–0.3	6.0	V
8	HVI PL0 input voltage	V_{Lx}	–27	42.0	V
9	EXTAL, XTAL [3]	V_{ILV}	–0.3	2.16	V
10	TEST input	V_{TEST}	–0.3	10.0	V
11	Instantaneous maximum current Single pin limit for all digital I/O pins[4]	I_D	–25	+25	mA
12	Continuous current on LIN	I_{LIN}		± 200 [5]	mA
13	Instantaneous maximum current on PP7	I_{PP7}	–80	+25	mA
14	Instantaneous maximum current on PP1, PP3[6] and PP5[6]	I_{PP135}	–30	+80	mA
15	Instantaneous maximum current Single pin limit for EXTAL, XTAL	I_{DL}	–25	+25	mA
16	Storage temperature range	T_{stg}	–65	155	°C

1 Beyond absolute maximum ratings device might be damaged.

2 VDDX and VDDA must be shorted

3 EXTAL, XTAL pins configured for external oscillator operation only

4 All digital I/O pins are internally clamped to V_{SSX} and V_{DDX}, or V_{SSA} and V_{DDA}.

5 The current on the LIN pin is internally limited. Therefore, it should not be possible to reach the 200mA anyway.

6 only applicable for PP3 and PP5 if pin VSSX2 is available

A.1.4 ESD Protection and Latch-up Immunity

All ESD testing is in conformity with CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification ESD stresses were performed for the Human Body Model (HBM) and the Charged-Device Model.

A device will be defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

Appendix D
LINPHY Electrical Specifications

D.1 Maximum Ratings

Table D-1. Maximum ratings of the LINPHY

Num	Ratings	Symbol	Value	Unit
1	DC voltage on LIN	V_{LIN}	-32 to +42	V
2	Continuous current on LIN	I_{LIN}	± 200 [1]	mA

[1]The current on the LIN pin is internally limited. Therefore, it should not be possible to reach the 200mA anyway.

D.2 Static Electrical Characteristics

Table D-2. Static electrical characteristics of the LINPHY

Characteristics noted under conditions 5.5V ≤ V_{LINSUP} ≤ 18V unless otherwise noted[1 2 3]. Typical values noted reflect the approximate parameter mean at T_A = 25°C under nominal conditions unless otherwise noted.						
Num	**Ratings**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	V_{LINSUP} operating range	V_{LINSUP_LIN}	5.5[1 2]	12	18	V
2	Current limitation into the LIN pin in dominant state[4] V_{LIN} = $V_{LINSUP\_LIN\_MAX}$	I_{LIN_LIM}	40		200	mA
3	Input leakage current in dominant state, driver off, internal pull-up on V_{LIN} = 0V, V_{LINSUP} = 12V	$I_{LIN\_PAS\_dom}$	-1			mA
4	Input leakage current in recessive state, driver off 5.5V<V_{LINSUP}<18V, 5.5V<V_{LIN}<18V, V_{LIN} > V_{LINSUP}	$I_{LIN\_PAS\_rec}$			20	µA
5	Input leakage current when ground disconnected -40°C < TJ < 175°C GND_{Device} = V_{LINSUP}, 0V<V_{LIN}<18V, V_{LINSUP} = 12V	$I_{LIN\_NO\_GND}$	-1		1	mA
6	Input leakage current when battery disconnected -40°C < TJ < 175°C V_{LINSUP} = GND_{Device}, 0<V_{LIN}<18V	$I_{LIN\_NO\_BAT}$			30	µA
7	Receiver dominant state	V_{LINdom}			0.4	V_{LINSUP}
8	Receiver recessive state	V_{LINrec}	0.6			V_{LINSUP}
9	V_{LIN_CNT} =(V_{th_dom}+ V_{th_rec})/2	V_{LIN_CNT}	0.475	0.5	0.525	V_{LINSUP}
10	V_{HYS} = V_{th_rec} -V_{th_dom}	V_{HYS}			0.175	V_{LINSUP}
11	Maximum capacitance allowed on slave node including external components	C_{slave}		220	250	pF
12a	Capacitance of the LIN pin, Recessive state	C_{LIN}		20		pF

O.5 0x0200-0x037F PIM (continued)

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x031F	WOMJ	R	0	0	0	0	0	0	WOMJ1	WOMJ0
		W								
0x0320–0x032F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0330	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0331	PTIL	R	0	0	0	0	0	0	0	PTIL0
		W								
0x0332–0x0333	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0334	PPSL	R	0	0	0	0	0	0	0	PPSL0
		W								
0x0335	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0336	PIEL	R	0	0	0	0	0	0	0	PIEL0
		W								
0x0337	PIFL	R	0	0	0	0	0	0	0	PIFL0
		W								
0x0338–0x033B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x033C	DIENL	R	0	0	0	0	0	0	0	DIENL0
		W								
0x033D	PTAL	R	PTTEL	PTPSL	PTABYPL	PTADIRL	PTAENL	0	0	0
		W								
0x033E	PIRL	R	0	0	0	0	0	0	0	PIRL0
		W								
0x033F-0x037F	Reserved	R	0	0	0	0	0	0	0	0
		W								

O.6 0x0380-0x039F FTMRZ

Address	Name		7	6	5	4	3	2	1	0
0x0380	FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
		W								
0x0381	FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
		W								

O.9 0x0480-x04AF PWM0 (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x048A - 0x048B	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x048C	PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048D	PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048E	PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x048F	PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0490	PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0491	PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0492	PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0493	PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x0494	PWMPER0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0495	PWMPER1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0496	PWMPER2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0497	PWMPER3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0498	PWMPER4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0499	PWMPER5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x049A	PWMPER6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x049B	PWMPER7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								