



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	I ² C, IrDA, LINbus, SCI, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	5.5V ~ 18V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12zvl32f0clc

- Internal COP (watchdog) module
- analog-to-digital converter (ADC) with 10 -bit or 12 -bit resolution and up to 10 channels available on external pins and V_{bg} (bandgap) result reference
- PGA module with two input channels
- One 8-bit 5V digital-to-analog converter (DAC)
- One analog comparators (ACMP) with rail-to-rail inputs
- MSCAN (1 Mbit/s, CAN 2.0 A, B software compatible) module
- One serial peripheral interface (SPI) module
- One serial communication interface (SCI) module with interface to internal LIN physical layer transceiver (with RX connected to a timer channel for frequency calibration purposes, if desired)
- Up to one additional SCI (not connected to LIN physical layer)
- One on-chip LIN physical layer transceiver fully compliant with the LIN 2.2 standard
- 6-channel timer module (TIM0) with input capture/output compare
- 2-channel timer module (TIM1) with input capture/output compare
- Inter-IC (IIC) module
- 8-channel Pulse Width Modulation module (PWM)
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Autonomous periodic interrupt (API), supports cyclic wakeup from Stop mode
- Pins to support 25 mA drive strength to VSSX
- Pin to support 20 mA drive strength from VDDX (EVDD)
- High Voltage Input (HVI)
- Supply voltage sense with low battery warning
- On-chip temperature sensor, temperature value can be measured with ADC or can generate a high temperature warning
- Up to 23 pins can be used as keyboard wake-up interrupt (KWI)

1.4 Module Features

The following sections provide more details of the integrated modules.

1.4.1 S12Z Central Processor Unit (CPU)

The S12Z CPU is a revolutionary high-speed core, with code size and execution efficiencies over the S12X CPU. The S12Z CPU also provides a linear memory map eliminating the inconvenience and performance impact of page swapping.

- Harvard Architecture - parallel data and code access
- 3 stage pipeline
- 32-Bit wide instruction and databus
- 32-Bit ALU
- 24-bit addressing (16 MByte linear address space)

1.7.3 Power Supply Pins

The power and ground pins are described below. Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible.

NOTE

All ground pins must be connected together in the application.

1.7.3.1 VDDX, VSSX1, VSSX2 — Digital I/O Power and Ground Pins

VDDX is the voltage regulator output to supply the digital I/O drivers. It supplies the VDDX domain pads. The VSSX1 and VSSX2 pin are the ground pin for the digital I/O drivers.

Bypass requirements on VDDX, VSSX2 depend on how heavily the MCU pins are loaded.

1.7.3.2 VDDA, VSSA — Power Supply Pins for ADC

These are the power supply and ground pins for the analog-to-digital converter and the voltage regulator. These pins must be externally connected to the voltage regulator (VDDX, VSSX). A separate bypass capacitor for the ADC supply is recommended.

1.7.3.3 VSS — Core Ground Pin

The voltage supply of nominally 1.8V is generated by the internal voltage regulator. The return current path is through the VSS pin.

1.7.3.4 LGND — LINPHY Ground Pin

LGND is the ground pin for the LIN physical layer LINPHY. This pin must be connected to board ground, even if the LINPHY is not used.

1.7.3.5 VSUP — Voltage Supply Pin for Voltage Regulator

VSUP is the 12V/18V supply voltage pin for the on chip voltage regulator. This is the voltage supply input from which the voltage regulator generates the on chip voltage supplies. It must be protected externally against a reverse battery connection.

1.8 Device Pinouts

MC9S12ZVL-Family is available in 48-pin package and 32-pin package. Signals in parentheses in [Figure 1-3](#) to [Figure 1-5](#) denote alternative module routing options.

The exposed pad must be connected to a grounded contact pad on the PCB. The exposed pad has an electrical connection within the package to VSSFLAG (VSSX die connection).

5.4.5.2.2 READ_SAME Effects Of Variable Access Size

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

Table 5-12. Consecutive READ_SAME Accesses With Variable Size

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

5.4.6 BDC Serial Interface

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

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

The handshake protocol is enabled by the ACK_ENABLE command. The BDC sends an ACK pulse when the ACK_ENABLE command has been completed. This feature can be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol.

Unlike the normal bit transfer, where the host initiates the transmission by issuing a negative edge on the BKGD pin, the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. [Figure 5-9](#) specifies the timing when the BKGD pin is being driven. The host must follow this timing constraint in order to avoid the risk of an electrical conflict at the BKGD pin.

When the handshake protocol is enabled, the STEAL bit in BDCCSR selects if bus cycle stealing is used to gain immediate access. If STEAL is cleared, the BDC is configured for low priority bus access using free cycles, without stealing cycles. This guarantees that BDC accesses remain truly non-intrusive to not affect the system timing during debugging. If STEAL is set, the BDC gains immediate access, if necessary stealing an internal bus cycle.

NOTE

If bus steals are disabled then a loop with no free cycles cannot allow access. In this case the host must recognize repeated NORESP messages and then issue a BACKGROUND command to stop the target and access the data.

[Figure 5-10](#) shows the ACK handshake protocol without steal in a command level timing diagram. The READ_MEM.B command is used as an example. First, the 8-bit command code is sent by the host, followed by the address of the memory location to be read. The target BDC decodes the command. Then an internal access is requested by the BDC. When a free bus cycle occurs the READ_MEM.B operation is carried out. If no free cycle occurs within 512 core clock cycles then the access is aborted, the NORESP flag is set and the target generates a Long-ACK pulse.

Having retrieved the data, the BDC issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the data read part of the command.

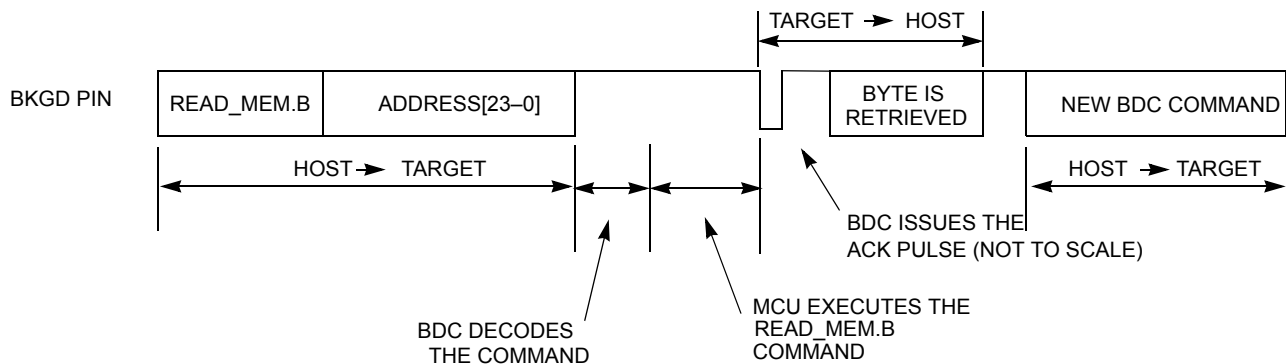


Figure 5-10. Handshake Protocol at Command Level

Alternatively, setting the STEAL bit configures the handshake protocol to make an immediate internal access, independent of free bus cycles.

Several examples of PLL divider settings are shown in [Table 9-33](#). The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF} .

Table 9-33. Examples of PLL Divider Settings

f_{osc}	REFDIV[3:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{bus}
off	\$00	1MHz	00	\$18	50MHz	01	\$03	12.5MHz	6.25MHz
off	\$00	1MHz	00	\$18	50MHz	01	\$00	50MHz	25MHz
4MHz	\$00	4MHz	01	\$05	48MHz	00	\$00	48MHz	24MHz

The phase detector inside the PLL compares the feedback clock ($FBCLK = VCOCLK / (SYNDIV + 1)$) with the reference clock ($REFCLK = (IRC1M \text{ or } OSCCLK) / (REFDIV + 1)$). Correction pulses are generated based on the phase difference between the two signals. The loop filter alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse which leads to a higher or lower VCO frequency.

The user must select the range of the REFCLK frequency (REFFRQ[1:0] bits) and the range of the VCOCLK frequency (VCOFRQ[1:0] bits) to ensure that the correct PLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK and the REFCLK. Therefore the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison. So e.g. a failure in the reference clock will cause the PLL not to lock.

If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and for instance check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during PLL start-up) or at periodic intervals. In either case, only when the LOCK bit is set, the VCOCLK will have stabilized to the programmed frequency.

- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within the tolerance, Δ_{Lock} , and is cleared when the VCO frequency is out of the tolerance, Δ_{unl} .
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

In case of loss of reference clock (e.g. IRCCLK) the PLL will not lock or if already locked, then it will unlock. The frequency of the VCOCLK will be very low and will depend on the value of the VCOFRQ[1:0] bits.

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:

10.5.2.2 ADC Control Register 1 (ADCCTL_1)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	CSL_BMOD	RVL_BMOD	SMOD_ACC	AUT_RSTA	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0


 = Unimplemented or Reserved

Figure 10-5. ADC Control Register 1 (ADCCTL_1)

Read: Anytime

Write:

- Bit CSL_BMOD and RVL_BMOD writable if bit ADC_EN clear or bit SMOD_ACC set
- Bit SMOD_ACC only writable in MCU Special Mode
- Bit AUT_RSTA writable anytime

Table 10-5. ADCCTL_1 Field Descriptions

Field	Description
7 CSL_BMOD	CSL Buffer Mode Select Bit — This bit defines the CSL buffer mode. This bit is only writable if ADC_EN is clear. 0 CSL single buffer mode. 1 CSL double buffer mode.
6 RVL_BMOD	RVL Buffer Mode Select Bit — This bit defines the RVL buffer mode. 0 RVL single buffer mode 1 RVL double buffer mode
5 SMOD_ACC	Special Mode Access Control Bit — This bit controls register access rights in MCU Special Mode. This bit is automatically cleared when leaving MCU Special Mode. Note: When this bit is set also the ADCCMD register is writeable via the data bus to allow modification of the current command for debugging purpose. But this is only possible if the current command is not already processed (conversion not started). Please see access details given for each register. Care must be taken when modifying ADC registers while bit SMOD_ACC is set to not corrupt a possible ongoing conversion. 0 Normal user access - Register write restrictions exist as specified for each bit. 1 Special access - Register write restrictions are lifted.
4 AUT_RSTA	Automatic Restart Event after exit from MCU Stop and Wait Mode (SWAI set) — This bit controls if a Restart Event is automatically generated after exit from MCU Stop Mode or Wait Mode with bit SWAI set. It can be configured for ADC conversion flow control mode “Trigger Mode” and “Restart Mode” (anytime during application runtime). 0 No automatic Restart Event after exit from MCU Stop Mode. 1 Automatic Restart Event occurs after exit from MCU Stop Mode.

10.9.7 Conversion flow control application information

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

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

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

Two different conversion flow control modes can be configured: Trigger Mode or Restart Mode
Single or double buffer configuration of CSL and RVL.

10.9.7.1 Initial Start of a Command Sequence List

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

If CSL_0 should be executed at the initial conversion start after device reset:

A Restart Event and a Trigger Event must occur (depending to the selected conversion flow control mode the events must occur one after the other or simultaneously) which causes the ADC to start conversion with commands loaded from CSL_0.

If CSL_1 should be executed at the initial conversion start after device reset:

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

As soon as a new valid Restart Event occurs the flow for ADC register load at conversion sequence start as described in [Section 10.6.3.3, “ADC List Usage and Conversion/Conversion Sequence Flow Description](#) applies.

10.9.7.2 Restart CSL execution with currently active CSL

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

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

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

13.4 Functional Description

13.4.1 General

This section provides a complete functional description of the MSCAN.

13.4.2 Message Storage

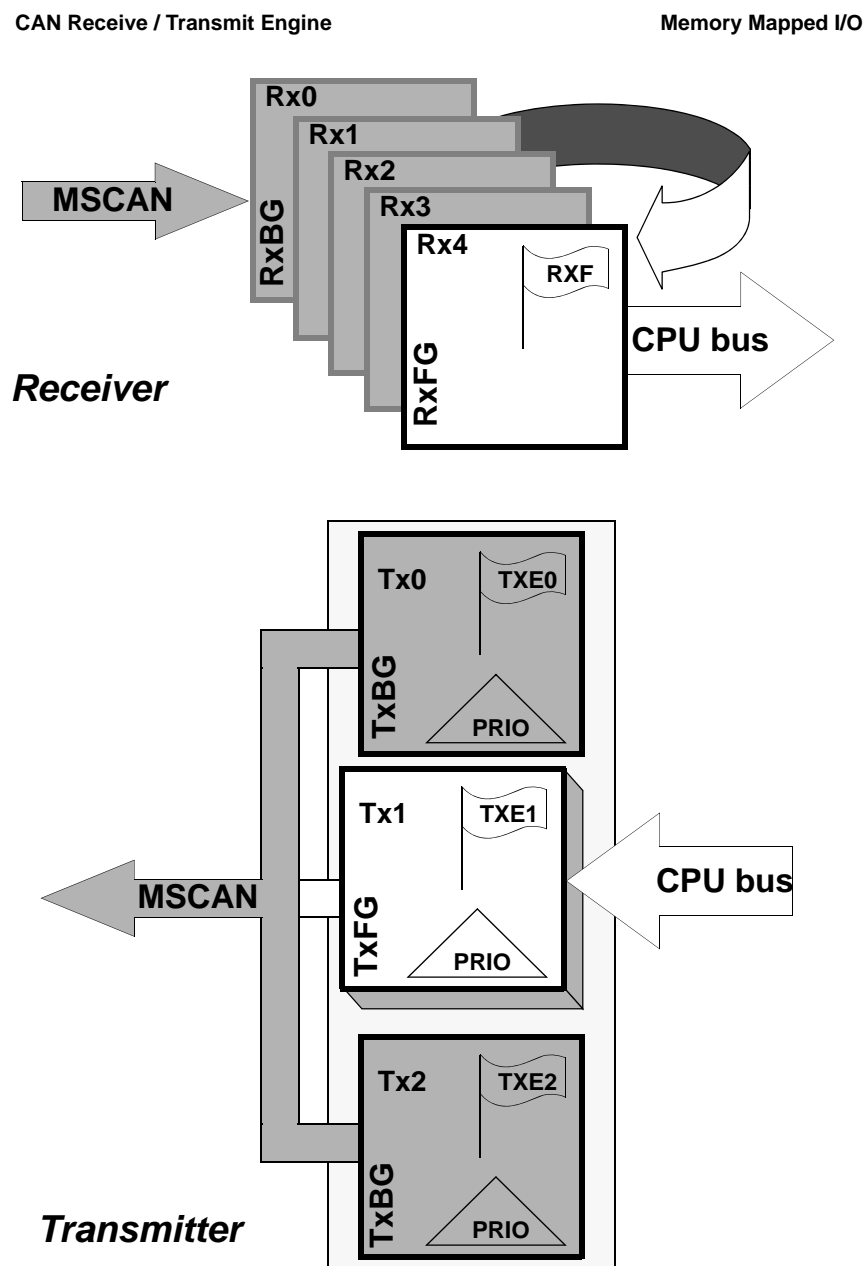


Figure 13-38. User Model for Message Buffer Organization

18.1.4 Block Diagram

Figure 18-1 is a high level block diagram of the SCI module, showing the interaction of various function blocks.

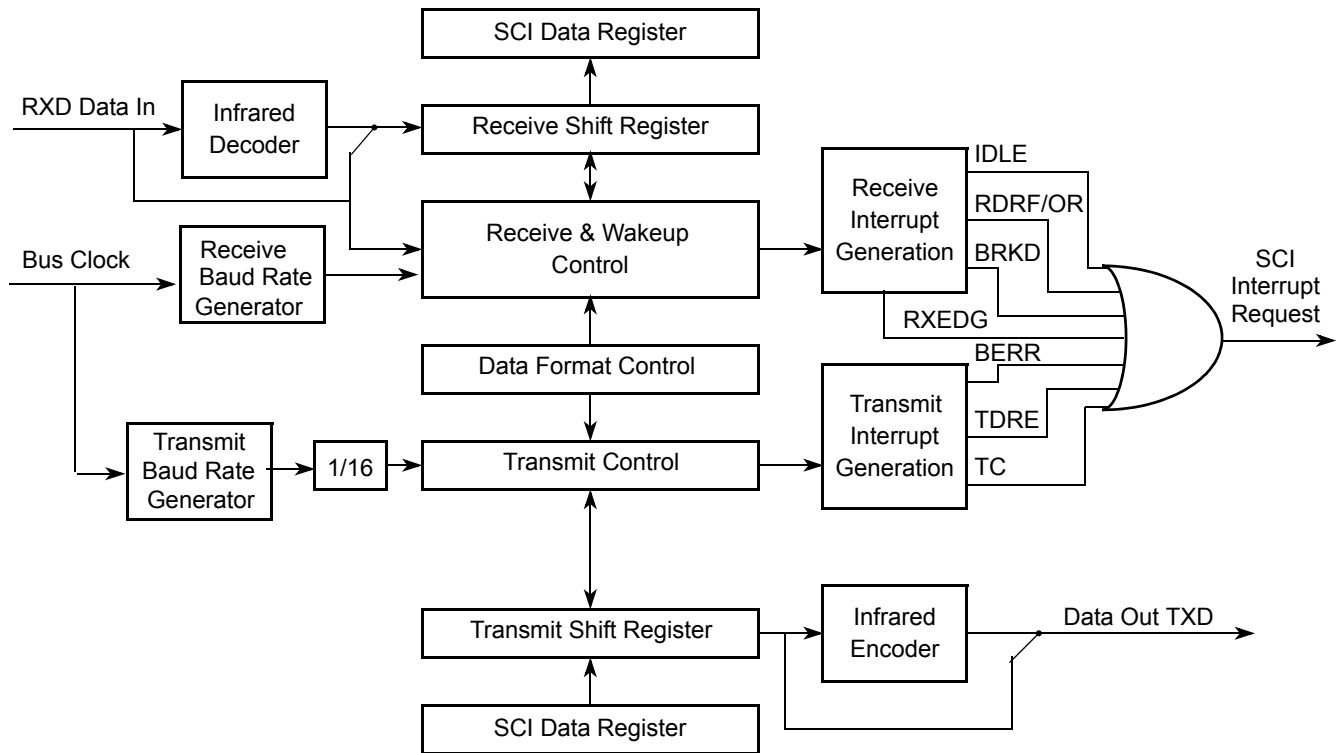


Figure 18-1. SCI Block Diagram

18.2 External Signal Description

The SCI module has a total of two external pins.

18.2.1 TXD — Transmit Pin

The TXD pin transmits SCI (standard or infrared) data. It will idle high in either mode and is high impedance anytime the transmitter is disabled.

18.2.2 RXD — Receive Pin

The RXD pin receives SCI (standard or infrared) data. An idle line is detected as a line high. This input is ignored when the receiver is disabled and should be terminated to a known voltage.

18.3 Memory Map and Register Definition

This section provides a detailed description of all the SCI registers.

Table 18-3. SCICR1 Field Descriptions (continued)

Field	Description
5 RSRC	Receiver Source Bit — When LOOPS = 1, the RSRC bit determines the source for the receiver shift register input. See Table 18-4. 0 Receiver input internally connected to transmitter output 1 Receiver input connected externally to transmitter
4 M	Data Format Mode Bit — MODE determines whether data characters are eight or nine bits long. 0 One start bit, eight data bits, one stop bit 1 One start bit, nine data bits, one stop bit
3 WAKE	Wakeup Condition Bit — WAKE determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit position of a received data character or an idle condition on the RXD pin. 0 Idle line wakeup 1 Address mark wakeup
2 ILT	Idle Line Type Bit — ILT determines when the receiver starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. 0 Idle character bit count begins after start bit 1 Idle character bit count begins after stop bit
1 PE	Parity Enable Bit — PE enables the parity function. When enabled, the parity function inserts a parity bit in the most significant bit position. 0 Parity function disabled 1 Parity function enabled
0 PT	Parity Type Bit — PT determines whether the SCI generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. 0 Even parity 1 Odd parity

Table 18-4. Loop Functions

LOOPS	RSRC	Function
0	x	Normal operation
1	0	Loop mode with transmitter output internally connected to receiver input
1	1	Single-wire mode with TXD pin connected to receiver input

18.3.2.3 SCI Alternative Status Register 1 (SCIASR1)

Module Base + 0x0000

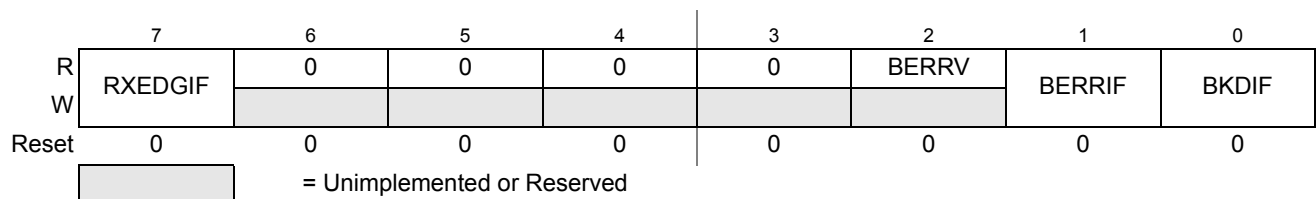


Figure 18-6. SCI Alternative Status Register 1 (SCIASR1)

Read: Anytime, if AMAP = 1

NOTE

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave's serial data output line.

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 n 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.

19.4.3 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.

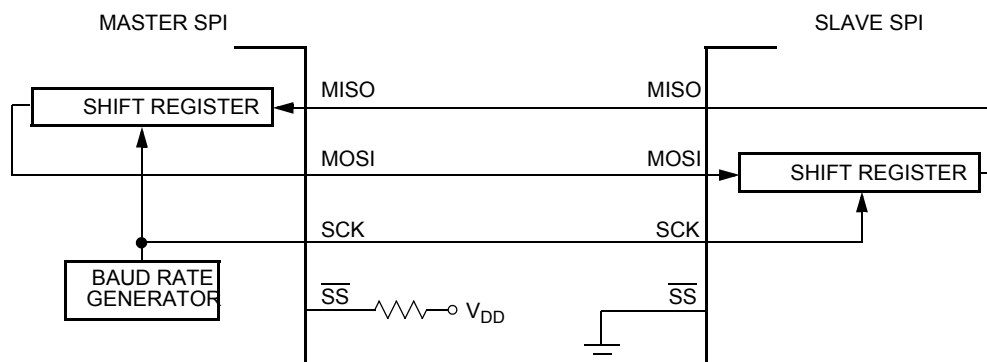


Figure 19-11. Master/Slave Transfer Block Diagram

1. n depends on the selected transfer width, please refer to [Section 19.3.2.2, "SPI Control Register 2 \(SPICR2\)"](#)

clock and the SCL period it may be necessary to wait until the IIC is busy after writing the calling address to the IBDR before proceeding with the following instructions. This is illustrated in the following example.

An example of a program which generates the START signal and transmits the first byte of data (slave address) is shown below:

```
CHFLAG      BRSET   IBSR,#$20,*      ;WAIT FOR IBB FLAG TO CLEAR
TXSTART     BSET    IBCR,#$30        ;SET TRANSMIT AND MASTER MODE;i.e. GENERATE START CONDITION
           MOV      CALLING,IBDR     ;TRANSMIT THE CALLING ADDRESS, D0=R/W
IBFREE      BRCLR   IBSR,#$20,*      ;WAIT FOR IBB FLAG TO SET
```

20.7.1.3 Post-Transfer Software Response

Transmission or reception of a byte will set the data transferring bit (TCF) to 1, which indicates one byte communication is finished. The IIC bus interrupt bit (IBIF) is set also; an interrupt will be generated if the interrupt function is enabled during initialization by setting the IBIE bit. Software must clear the IBIF bit in the interrupt routine first. The TCF bit will be cleared by reading from the IIC bus data I/O register (IBDR) in receive mode or writing to IBDR in transmit mode.

Software may service the IIC I/O in the main program by monitoring the IBIF bit if the interrupt function is disabled. Note that polling should monitor the IBIF bit rather than the TCF bit because their operation is different when arbitration is lost.

Note that when an interrupt occurs at the end of the address cycle the master will always be in transmit mode, i.e. the address is transmitted. If master receive mode is required, indicated by R/W bit in IBDR, then the Tx/Rx bit should be toggled at this stage.

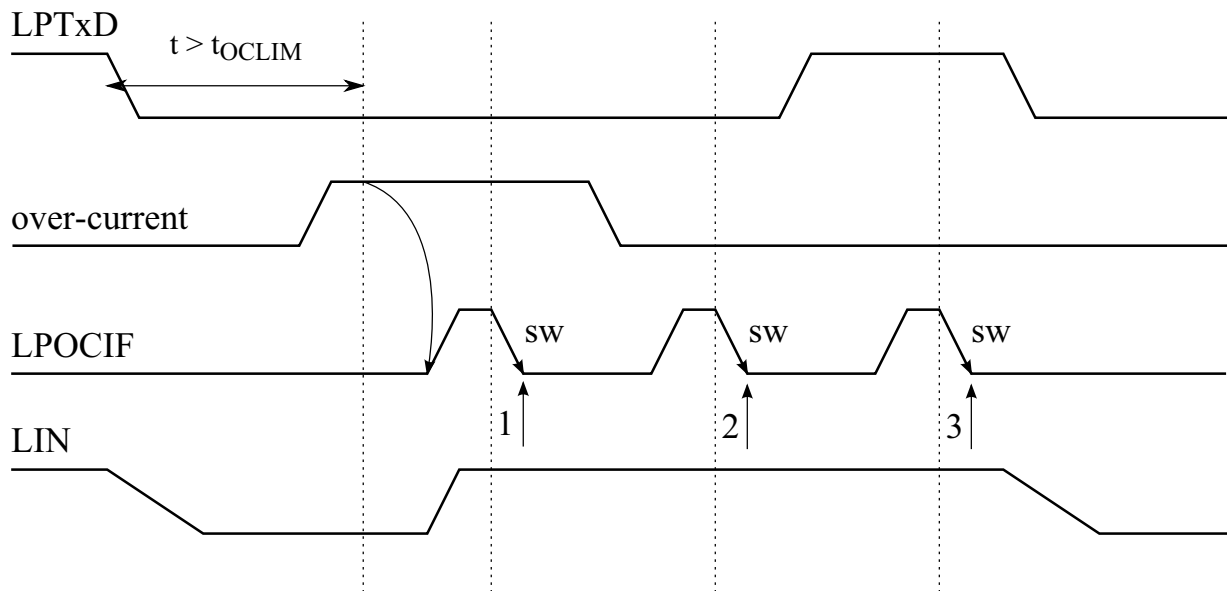
During slave mode address cycles (IAAS=1), the SRW bit in the status register is read to determine the direction of the subsequent transfer and the Tx/Rx bit is programmed accordingly. For slave mode data cycles (IAAS=0) the SRW bit is not valid, the Tx/Rx bit in the control register should be read to determine the direction of the current transfer.

The following is an example of a software response by a 'master transmitter' in the interrupt routine.

```
ISR          BCLR    IBSR,#$02        ;CLEAR THE IBIF FLAG
           BRCLR   IBCR,#$20,SLAVE    ;BRANCH IF IN SLAVE MODE
           BRCLR   IBCR,#$10,RECEIVE  ;BRANCH IF IN RECEIVE MODE
           BRSET   IBSR,#$01,END      ;IF NO ACK, END OF TRANSMISSION
TRANSMIT     MOV     DATABUF,IBDR     ;TRANSMIT NEXT BYTE OF DATA
```

20.7.1.4 Generation of STOP

A data transfer ends with a STOP signal generated by the 'master' device. A master transmitter can simply generate a STOP signal after all the data has been transmitted. The following is an example showing how a stop condition is generated by a master transmitter.



- 1: Flag cleared, transmitter re-enable not successful because over-current is still present
- 2: Flag cleared, transmitter re-enable not successful because LPTxD is dominant
- 3: Flag cleared, transmitter re-enable successful

Figure 21-12. Overcurrent interrupt handling

21.4.4.2 TxD-dominant timeout Interrupt

To protect the LIN bus from a network lock-up, the LIN Physical Layer implements a TxD-dominant timeout mechanism. When the LPTxD signal has been dominant for more than t_{DTLIM} the transmitter is disabled and the LPDT status flag and the LPDTIF interrupt flag are set.

In order to re-enable the transmitter again, the following prerequisites must be met:

- 1) TxD-dominant condition is over (LPDT=0)
- 2) LPTxD is recessive or the LIN Physical Layer is in shutdown or receive only mode for a minimum of a transmit bit time

To re-enable the transmitter then, the LPDTIF flag must be cleared (by writing a 1).

NOTE

Please make sure that LPDTIF=1 before trying to clear it. It is not allowed to try to clear LPDTIF if LPDTIF=0 already.

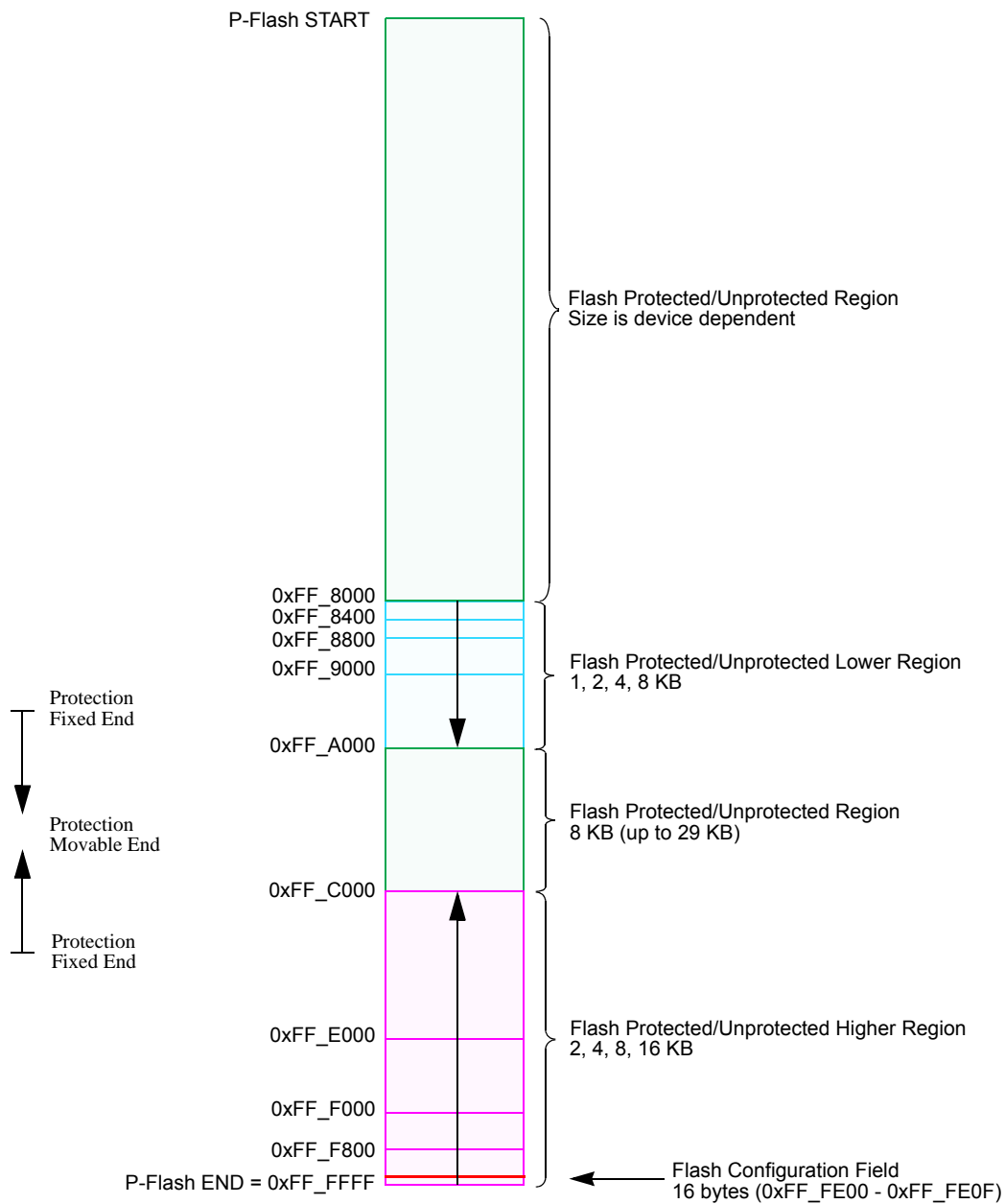


Figure 22-2. P-Flash Memory Map With Protection Alignment

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0xFF_FE00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 22-54. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 22.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

22.4.7.12 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

Table 22-55. Set User Margin Level Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0D	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	
FCCOB2	Margin level setting.	

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

NOTE

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

Valid margin level settings for the Set User Margin Level command are defined in [Table 22-56](#).

Table A-17. Run and Wait Current Characteristics for ZVL(A)128/96/64

Conditions are: $V_{SUP} = 18V$, see Table A-14 and Table A-15						
Num	Rating	Symbol	Min	Typ	Max	Unit
1	Run Current, $-40^{\circ}C < T_J \leq 150^{\circ}C$, $f_{bus}=32MHz$	I_{SUPR}	—	21	27	mA
2	Run Current, $150^{\circ}C < T_J < 175^{\circ}C$, $f_{bus}=25MHz$	I_{SUPR}	—	21	27	mA
3	Wait Current, $-40^{\circ}C < T_J \leq 150^{\circ}C$, $f_{bus}=32MHz$	I_{SUPW}	—	13	20	mA
4	Wait Current, $150^{\circ}C < T_J < 175^{\circ}C$, $f_{bus}=25MHz$	I_{SUPW}	—	13	20	mA

Table A-18. Stop Current Characteristics for ZVL(S)32/16/8

Conditions are: $V_{SUP} = 12V$						
Num	Rating ¹	Symbol	Min	Typ	Max	Unit
Stop Current all modules off						
1	$T_J = -40^{\circ}C$	I_{SUPS}	—	20	28	μA
2	$T_J = 25^{\circ}C$	I_{SUPS}	—	23	33	μA
3	$T_J = 85^{\circ}C$	I_{SUPS}	—	44	55	μA
4	$T_J = 105^{\circ}C$	I_{SUPS}	—	63	85	μA
5	$T_J = 125^{\circ}C$	I_{SUPS}	—	115	156	μA
Stop Current API enabled & LINPHY in standby						
6	$T_J = 25^{\circ}C$	I_{SUPS}	—	38	—	μA

¹ If MCU is in STOP long enough then $T_A = T_J$. Die self heating due to stop current can be ignored.

Table A-19. Stop Current Characteristics for ZVL(A)128/96/64

Conditions are: $V_{SUP} = 12V$						
Num	Rating ¹	Symbol	Min	Typ	Max	Unit
Stop Current all modules off						
1	$T_J = -40^{\circ}C$	I_{SUPS}	—	20	40	μA
2	$T_J = 25^{\circ}C$	I_{SUPS}	—	25	50	μA
3	$T_J = 85^{\circ}C$	I_{SUPS}	—	60	107	μA
4	$T_J = 105^{\circ}C$	I_{SUPS}	—	78	176	μA
5	$T_J = 125^{\circ}C$	I_{SUPS}	—	130	301	μA
Stop Current API enabled & LINPHY in standby						
6	$T_J = 25^{\circ}C$	I_{SUPS}	—	53	—	μA

¹ If MCU is in STOP long enough then $T_A = T_J$. Die self heating due to stop current can be ignored.

Table A-20. Pseudo Stop Current Characteristics for ZVL(S)32/16/8

Conditions are: $V_{SUP} = 12V$, API, COP & RTI enabled						
Num	Rating	Symbol	Min	Typ	Max	Unit
1	$T_J = 25^{\circ}C$	I_{SUPPS}	—	155	350	μA

B.2 IRC and OSC Electrical Specifications

Table B-3. IRC electrical characteristics

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Junction Temperature - 40° to 150° Celsius Internal Reference Frequency, factory trimmed	f_{IRC1M_TRIM}	0.987	1.000	1.013	MHz
2	Junction Temperature 150° to 175° Celsius Internal Reference Frequency, factory trimmed	f_{IRC1M_TRIM}	0.9855	—	1.0145	MHz

Table B-4. OSC electrical characteristics

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Nominal crystal or resonator frequency	f_{OSC}	4.0	—	20	MHz
2	Startup Current	i_{OSC}	100	—	—	μA
3a	Oscillator start-up time (4MHz) ¹	t_{UPOSC}	—	2	10	ms
3b	Oscillator start-up time (8MHz) ¹	t_{UPOSC}	—	1.6	8	ms
3c	Oscillator start-up time (16MHz) ¹	t_{UPOSC}	—	1	5	ms
3d	Oscillator start-up time (20MHz) ¹	t_{UPOSC}	—	1	4	ms
4	Clock Monitor Failure Assert Frequency	f_{CMFA}	200	450	1200	KHz
5	Input Capacitance (EXTAL, XTAL pins)	C_{IN}	—	7	—	pF
6	EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$	—	120	—	mV
7	EXTAL Pin oscillation amplitude (loop controlled Pierce)	$V_{PP,EXTAL}$	—	1.0	—	V
8	EXTAL Pin oscillation required amplitude ²	$V_{PP,EXTAL}$	0.8	—	1.5	V

¹ These values apply for carefully designed PCB layouts with capacitors that match the crystal/resonator requirements.

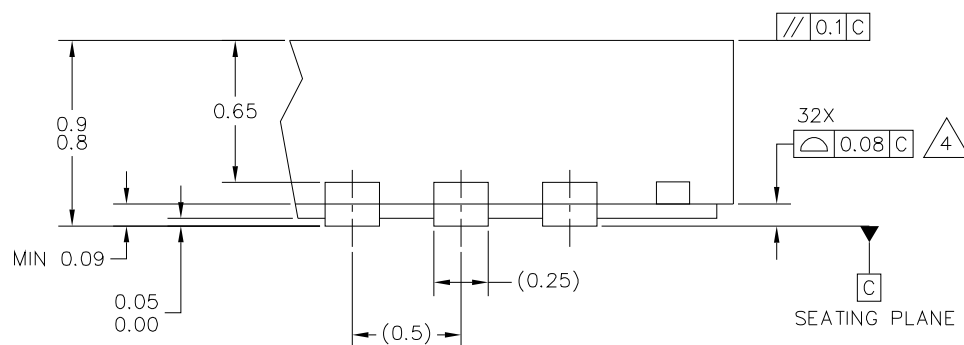
² Needs to be measured at room temperature on the application board using a probe with very low ($\leq 5pF$) input capacitance.

B.3 Phase Locked Loop

B.3.1 Jitter Information

With each transition of the feedback clock, the deviation from the reference clock is measured and the input voltage to the VCO is adjusted accordingly. The adjustment is done continuously with no abrupt changes in the VCOCLK frequency. Noise, voltage, temperature and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in [Figure B-2](#).

⁴ The structure of the HVI pins does not include diode structures shown in [Figure A-1](#) that inject current when the input voltage goes outside the supply-ground range. Thus the HVI pin current injection is limited to below 200uA within the absolute maximum pin voltage range. However if the HVI impedance converter bypass is enabled, then even currents in this range can corrupt ADC results from simultaneous conversions on other channels. This can be prevented by disabling the bypass, either by clearing the PTAENLx or PTABYPLx bit. Similarly when the ADC is converting a HVI pin voltage then the impedance converter bypass must be disabled to ensure that current injection on PADx pins does not impact the HVI ADC conversion result.



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: QFN, THERMALLY ENHANCED, 5 X 5 X 0.85, 0.5 PITCH, 32 TERMINAL	DOCUMENT NO: 98ASA00656D REV: 0	
	STANDARD: NON-JEDEC	
	10 DEC 2013	