Infineon Technologies - SAF-XC878M-16FFI 5V AA Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Discontinued at Digi-Key
Core Processor	XC800
Core Size	8-Bit
Speed	27MHz
Connectivity	SPI, SSI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	40
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	PG-LQFP-64-4
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/saf-xc878m-16ffi-5v-aa

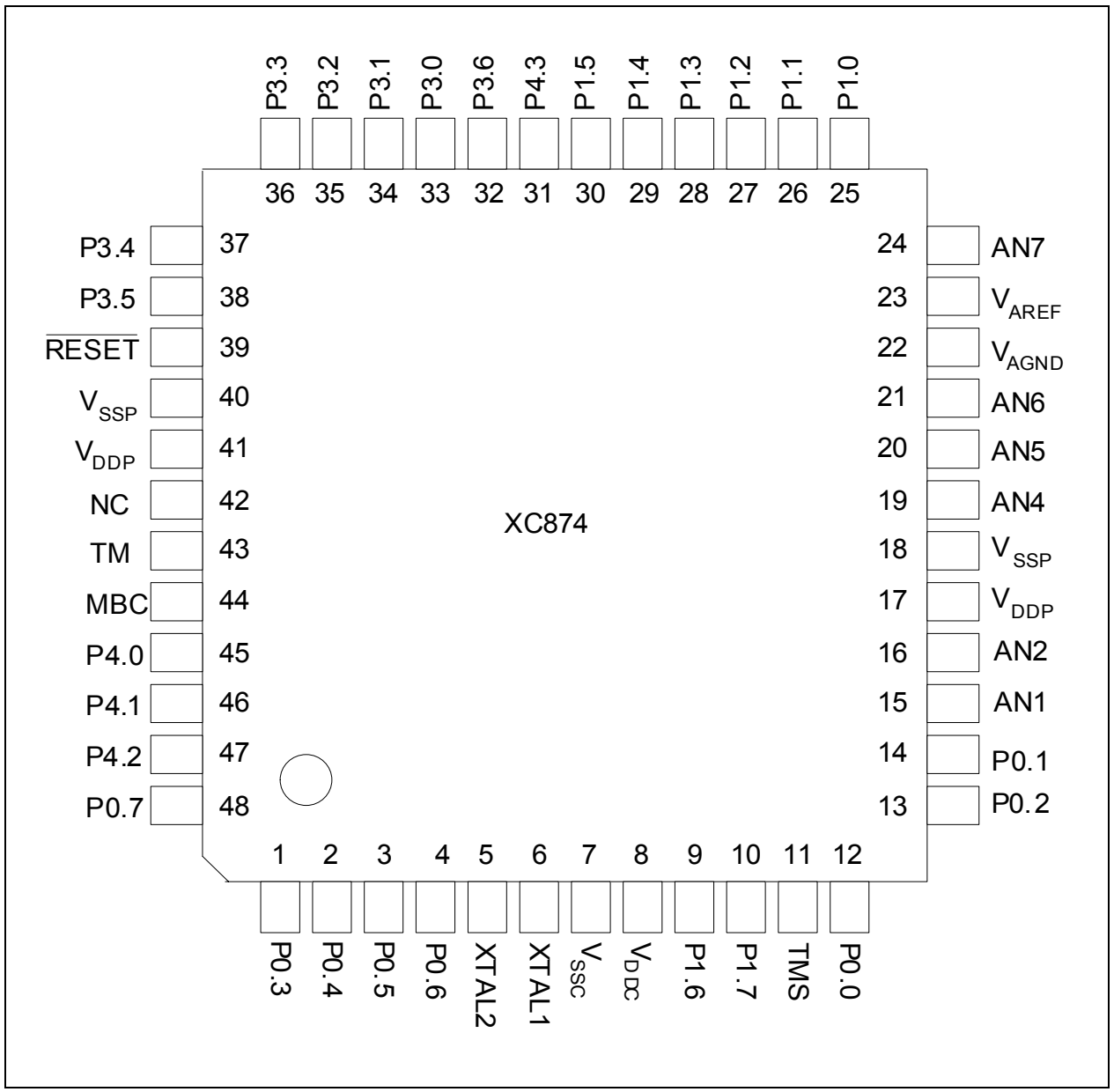

Figure 5 XC874 Pin Configuration, PG-VQFN-48 Package (top view)

Table 3 Pin Definitions and Functions (cont'd)

Symbol	Pin Number (LQFP-64 / VQFN-48)	Type	Reset State	Function	
P4		I/O		**Port 4** Port 4 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for CCU6, Timer 0, Timer 1, T2CCU, Timer 21, MultiCAN and External Bus Interface. *Note: External Bus Interface is not available in XC874.*	
P4.0	59/45		Hi-Z	RXDC0_3 CC60_1 T2CC0_0/ EXINT3_1 D0	MultiCAN Node 0 Receiver Input Output of Capture/Compare channel 0 External Interrupt Input 3/T2CCU Capture/Compare Channel 0 Data Line 0 Input/Output
P4.1	60/46		Hi-Z	TXDC0_3 COUT60_1 T2CC1_0/ EXINT4_1 D1	MultiCAN Node 0 Transmitter Output Output of Capture/Compare channel 0 External Interrupt Input 4/T2CCU Capture/Compare Channel 1 Data Line 1 Input/Output
P4.2	61/47		PU	EXINT6_1 T21_0 D2	External Interrupt Input 6 Timer 21 Input Data Line 2 Input/Output
P4.3	40/31		Hi-Z	T2EX_1 EXF21_1 COUT63_2 D3	Timer 2 External Trigger Input Timer 21 External Flag Output Output of Capture/Compare channel 3 Data Line 3 Input/Output
P4.4	45/-		Hi-Z	CCPOS0_3 T0_0 CC61_4 T2CC2_0/ EXINT5_1 D4	CCU6 Hall Input 0 Timer 0 Input Output of Capture/Compare channel 1 External Interrupt Input 5/T2CCU Capture/Compare Channel 2 Data Line 4 Input/Output

- Overwrite the contents of PAGE with the contents of STx, ignoring the value written to the bit positions of PAGE
  (this is done at the end of the interrupt routine to restore the previous page setting before the interrupt occurred)

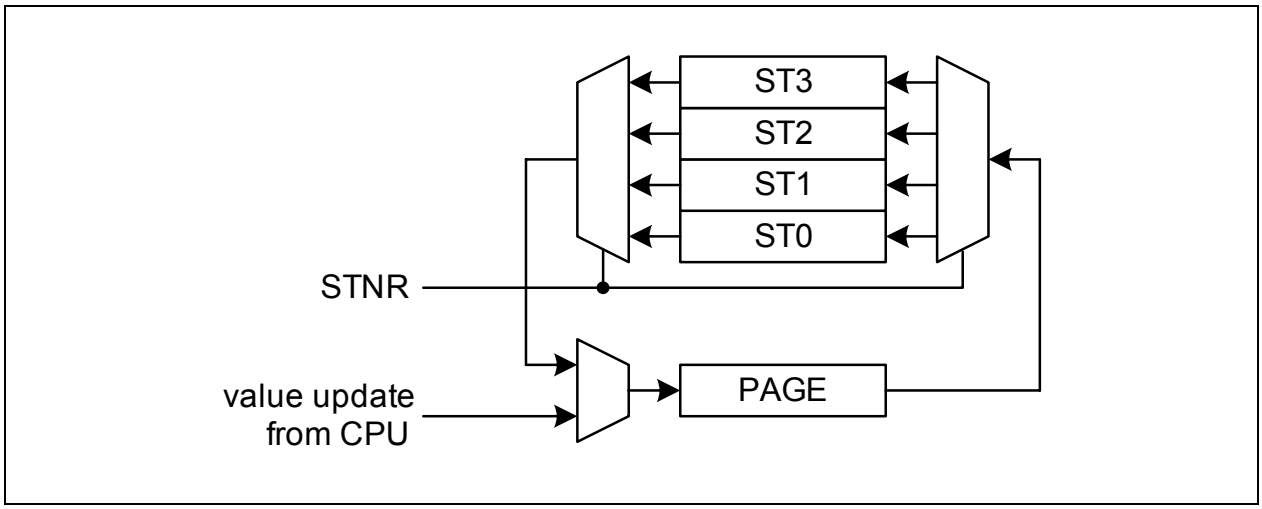

Figure 11 Storage Elements for Paging

With this mechanism, a certain number of interrupt routines (or other routines) can perform page changes without reading and storing the previously used page information. The use of only write operations makes the system simpler and faster. Consequently, this mechanism significantly improves the performance of short interrupt routines.

The XC87x supports local address extension for:

- Parallel Ports
- Analog-to-Digital Converter (ADC)
- Capture/Compare Unit 6 (CCU6)
- System Control Registers

Table 7 CORDIC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$9C_H$	**CD_CORDYL Reset: 00_H** CORDIC Y Data Low Byte	Bit Field	DATAL							
		Type	rw							
$9D_H$	**CD_CORDYH Reset: 00_H** CORDIC Y Data High Byte	Bit Field	DATAH							
		Type	rw							
$9E_H$	**CD_CORDZL Reset: 00_H** CORDIC Z Data Low Byte	Bit Field	DATAL							
		Type	rw							
$9F_H$	**CD_CORDZH Reset: 00_H** CORDIC Z Data High Byte	Bit Field	DATAH							
		Type	rw							
$A0_H$	**CD_STATC Reset: 00_H** CORDIC Status and Data Control Register	Bit Field	KEEPZ	KEEPY	KEEPX	DMAP	INT_EN	EOC	ERROR	BSY
		Type	rw	rw	rw	rw	rw	rwh	rh	rh
$A1_H$	**CD_CON Reset: 00_H** CORDIC Control Register	Bit Field	MPS		X_USIGN	ST_MODE	ROTVEC	MODE		ST
		Type	rw		rw	rw	rw	rw		rwh

3.2.4.4 System Control Registers

The system control SFRs can be accessed in the mapped memory area (RMAP = 0).

Table 8 SCU Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0 or 1										
$8F_H$	**SYSCON0 Reset: 04_H** System Control Register 0	Bit Field	0			IMODE	0	1	0	RMAP
		Type	r			rw	r	r	r	rw
RMAP = 0										
BF_H	**SCU_PAGE Reset: 00_H** Page Register	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, PAGE 0										
$B3_H$	**MODPISEL Reset: 00_H** Peripheral Input Select Register	Bit Field	0	URRISH	JTAGTDIS	JTAGTCKS	EXINT2IS	EXINT1IS	EXINT0IS	URRIS
		Type	r	rw	rw	rw	rw	rw	rw	rw
$B4_H$	**IRCON0 Reset: 00_H** Interrupt Request Register 0	Bit Field	0	EXINT6	EXINT5	EXINT4	EXINT3	EXINT2	EXINT1	EXINT0
		Type	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$B5_H$	**IRCON1 Reset: 00_H** Interrupt Request Register 1	Bit Field	0	CANSRC2	CANSRC1	ADCSR1	ADCSR0	RIR	TIR	EIR
		Type	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$B6_H$	**IRCON2 Reset: 00_H** Interrupt Request Register 2	Bit Field	0			CANSRC3	0			CANSRC0
		Type	r			rwh	r			rwh

3.2.4.5 WDT Registers

The WDT SFRs can be accessed in the mapped memory area (RMAP = 1).

Table 9 WDT Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 1										
BB_H	**WDTCON** **Reset: 00_H** Watchdog Timer Control Register	Bit Field	0		WINBEN	WDTPR	0	WDTEN	WDTRS	WDTIN
		Type	r		rw	rh	r	rw	rwh	rw
BC_H	**WDTREL** **Reset: 00_H** Watchdog Timer Reload Register	Bit Field	WDTREL							
		Type	rw							
BD_H	**WDTWINB** **Reset: 00_H** Watchdog Window-Boundary Count Register	Bit Field	WDTWINB							
		Type	rw							
BE_H	**WDTL** **Reset: 00_H** Watchdog Timer Register Low	Bit Field	WDT							
		Type	rh							
BF_H	**WDTH** **Reset: 00_H** Watchdog Timer Register High	Bit Field	WDT							
		Type	rh							

3.2.4.6 Port Registers

The Port SFRs can be accessed in the standard memory area (RMAP = 0).

Table 10 Port Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$B2_H$	**PORT_PAGE** **Reset: 00_H** Page Register	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0, PAGE 0										
80_H	**P0_DATA** **Reset: 00_H** P0 Data Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
86_H	**P0_DIR** **Reset: 00_H** P0 Direction Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
90_H	**P1_DATA** **Reset: 00_H** P1 Data Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
91_H	**P1_DIR** **Reset: 00_H** P1 Direction Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
92_H	**P5_DATA** **Reset: 00_H** P5 Data Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
93_H	**P5_DIR** **Reset: 00_H** P5 Direction Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw

Table 12 T2CCU Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$C5_H$	**T2CCU_CCTH Reset: 00_H** T2CCU Capture/Compare Timer Register High	Bit Field	CCT							
		Type	rwh							
$C6_H$	**T2CCU_CCTCON Reset: 00_H** T2CCU CaptureCcompare Timer Control Register	Bit Field	CCTPRE				CCTOVF	CCTOVEN	TIMSYN	CCTST
		Type	rw				rwh	rw	rw	rw
RMAP = 0, PAGE 2										
$C0_H$	**T2CCU_COSHDWReset: 00_H** T2CCU Capture/compare Enable Register	Bit Field	ENSHDW	TXOV	COOUT5	COOUT4	COOUT3	COOUT2	COOUT1	COOUT0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$C1_H$	**T2CCU_CC0L Reset: 00_H** T2CCU Capture/Compare Register 0 Low	Bit Field	CCVALL							
		Type	rwh							
$C2_H$	**T2CCU_CC0H Reset: 00_H** T2CCU Capture/compare Register 0 High	Bit Field	CCVALH							
		Type	rwh							
$C3_H$	**T2CCU_CC1L Reset: 00_H** T2CCU Capture/compare Register 1 Low	Bit Field	CCVALL							
		Type	rwh							
$C4_H$	**T2CCU_CC1H Reset: 00_H** T2CCU Capture/compare Register 1 High	Bit Field	CCVALH							
		Type	rwh							
$C5_H$	**T2CCU_CC2L Reset: 00_H** T2CCU Capture/compare Register 2 Low	Bit Field	CCVALL							
		Type	rwh							
$C6_H$	**T2CCU_CC2H Reset: 00_H** T2CCU Capture/compare Register 2 High	Bit Field	CCVALH							
		Type	rwh							
RMAP = 0, PAGE 3										
$C0_H$	**T2CCU_COCON Reset: 00_H** T2CCU Compare Control Register	Bit Field	CCM5	CCM4	CM5F	CM4F	POLB	POLA	COMOD	
		Type	rw	rw	rwh	rwh	rw	rw	rw	
$C1_H$	**T2CCU_CC3L Reset: 00_H** T2CCU Capture/compare Register 3 Low	Bit Field	CCVALL							
		Type	rwh							
$C2_H$	**T2CCU_CC3H Reset: 00_H** T2CCU Capture/compare Register 3 High	Bit Field	CCVALH							
		Type	rwh							
$C3_H$	**T2CCU_CC4L Reset: 00_H** T2CCU Capture/compare Register 4 Low	Bit Field	CCVALL							
		Type	rwh							
$C4_H$	**T2CCU_CC4H Reset: 00_H** T2CCU Capture/compare Register 4 High	Bit Field	CCVALH							
		Type	rwh							
$C5_H$	**T2CCU_CC5L Reset: 00_H** T2CCU Capture/compare Register 5 Low	Bit Field	CCVALL							
		Type	rwh							
$C6_H$	**T2CCU_CC5H Reset: 00_H** T2CCU Capture/compare Register 5 High	Bit Field	CCVALH							
		Type	rwh							

Table 14 CCU6 Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
FD_H	**CCU6_CC61RH Reset: 00_H** Capture/Compare Register for Channel CC61 High	Bit Field	CC61VH							
		Type	rh							
FE_H	**CCU6_CC62RL Reset: 00_H** Capture/Compare Register for Channel CC62 Low	Bit Field	CC62VL							
		Type	rh							
FF_H	**CCU6_CC62RH Reset: 00_H** Capture/Compare Register for Channel CC62 High	Bit Field	CC62VH							
		Type	rh							
RMAP = 0, PAGE 2										
$9A_H$	**CCU6_T12MSELL Reset: 00_H** T12 Capture/Compare Mode Select Register Low	Bit Field	MSEL61				MSEL60			
		Type	rw				rw			
$9B_H$	**CCU6_T12MSELH Reset: 00_H** T12 Capture/Compare Mode Select Register High	Bit Field	DBYP	HSYNC			MSEL62			
		Type	rw	rw			rw			
$9C_H$	**CCU6_IENL Reset: 00_H** Capture/Compare Interrupt Enable Register Low	Bit Field	ENT12PM	ENT12OM	ENCC62F	ENCC62R	ENCC61F	ENCC61R	ENCC60F	ENCC60R
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$9D_H$	**CCU6_IENH Reset: 00_H** Capture/Compare Interrupt Enable Register High	Bit Field	ENSTR	ENIDLE	ENWHE	ENCHE	0	ENTRPF	ENT13PM	ENT13CM
		Type	rw	rw	rw	rw	r	rw	rw	rw
$9E_H$	**CCU6_INPL Reset: 40_H** Capture/Compare Interrupt Node Pointer Register Low	Bit Field	INPCHE		INPCC62		INPCC61		INPCC60	
		Type	rw		rw		rw		rw	
$9F_H$	**CCU6_INPH Reset: 39_H** Capture/Compare Interrupt Node Pointer Register High	Bit Field	0		INPT13		INPT12		INPERR	
		Type	r		rw		rw		rw	
$A4_H$	**CCU6_ISSL Reset: 00_H** Capture/Compare Interrupt Status Set Register Low	Bit Field	ST12PM	ST12OM	SCC62F	SCC62R	SCC61F	SCC61R	SCC60F	SCC60R
		Type	w	w	w	w	w	w	w	w
$A5_H$	**CCU6_ISSH Reset: 00_H** Capture/Compare Interrupt Status Set Register High	Bit Field	SSTR	SIDLE	SWHE	SCHE	SWHC	STRPF	ST13PM	ST13CM
		Type	w	w	w	w	w	w	w	w
$A6_H$	**CCU6_PSLR Reset: 00_H** Passive State Level Register	Bit Field	PSL63	0	PSL					
		Type	rwh	r	rwh					
$A7_H$	**CCU6_MCMCTR Reset: 00_H** Multi-Channel Mode Control Register	Bit Field	0		SWSYN		0	SWSEL		
		Type	r		rw		r	rw		
FA_H	**CCU6_TCTR2L Reset: 00_H** Timer Control Register 2 Low	Bit Field	0	T13TED		T13TEC			T13SSC	T12SSC
		Type	r	rw		rw			rw	rw
FB_H	**CCU6_TCTR2H Reset: 00_H** Timer Control Register 2 High	Bit Field	0				T13RSEL		T12RSEL	
		Type	r				rw		rw	
FC_H	**CCU6_MODCTRL Reset: 00_H** Modulation Control Register Low	Bit Field	MCMEN	0	T12MODEN					
		Type	rw	r	rw					

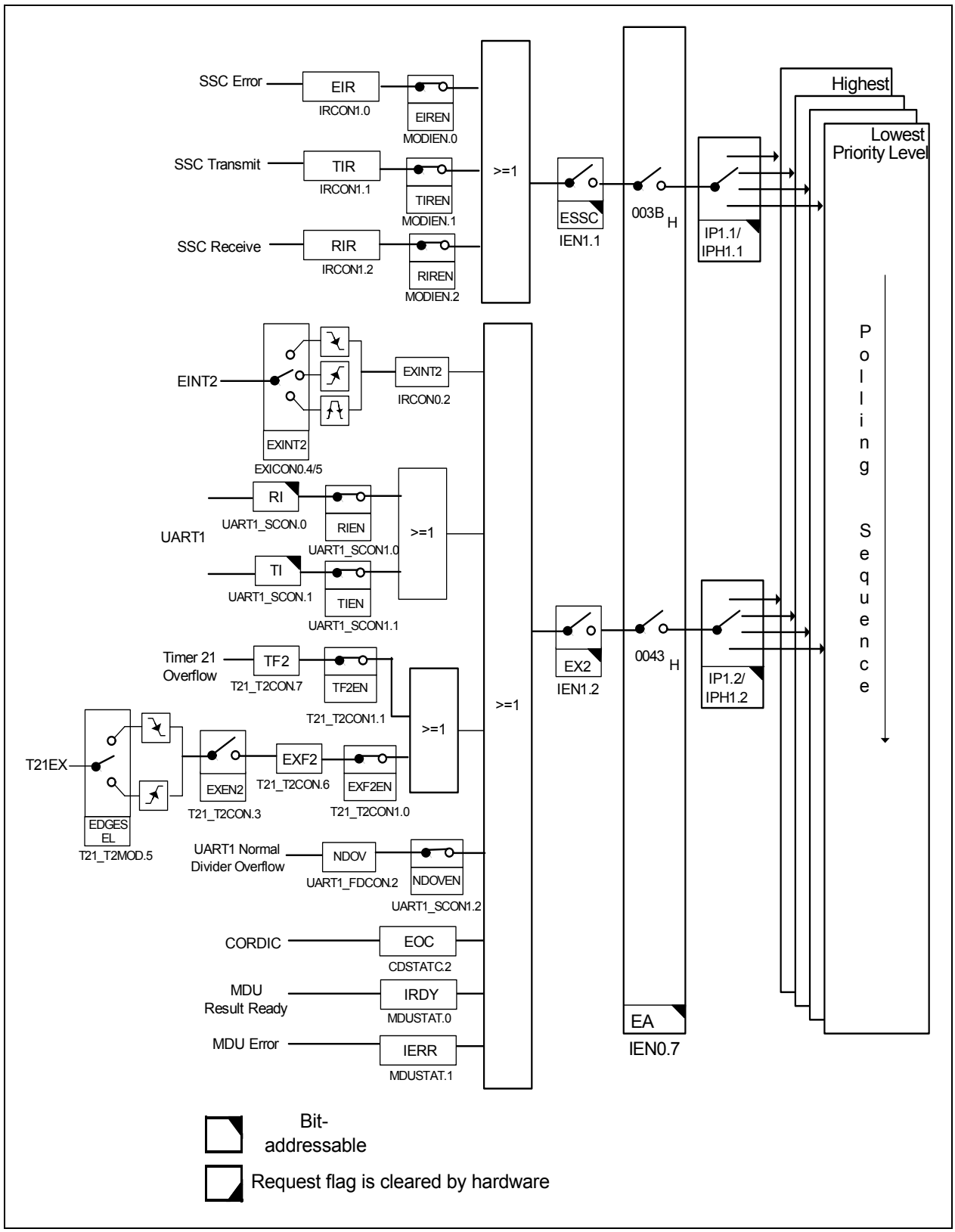

Figure 15 Interrupt Request Sources (Part 3)

3.5 Parallel Ports

The XC87x has 40 port pins organized into five parallel ports: Port 0 (P0), Port 1 (P1), Port 3 (P3), Port 4 (P4) and Port 5 (P5). Each pin has a pair of internal pull-up and pull-down devices that can be individually enabled or disabled. These ports are bidirectional and can be used as general purpose input/output (GPIO) or to perform alternate input/output functions for the on-chip peripherals. When configured as an output, the open drain mode can be selected.

Bidirectional Port Features

- Configurable pin direction
- Configurable pull-up/pull-down devices
- Configurable open drain mode
- Configurable drive strength
- Transfer of data through digital inputs and outputs (general purpose I/O)
- Alternate input/output for on-chip peripherals

3.7.2 Booting Scheme

When the XC87x is reset, it must identify the type of configuration with which to start the different modes once the reset sequence is complete. Thus, boot configuration information that is required for activation of special modes and conditions needs to be applied by the external world through input pins. After power-on reset or hardware reset, the pins MBC, TMS and P0.0 collectively select the different boot options. **Table 25** shows the available boot options in the XC87x.

Table 25 XC87x Boot Selection [1)]

MBC	TMS	P0.0	Type of Mode	PC Start Value
1	0	X	User Mode[2)]; on-chip OSC/PLL non-bypassed	0000_H
0	0	X	BSL Mode; (LIN Mode[3)], UART/ MultiCAN Mode[4)5)] and Alternate BSL Mode[6)]); on-chip OSC/PLL non-bypassed	0000_H
0	1	0	OCDS Mode; on-chip OSC/PLL non-bypassed	0000_H
1	1	0	User (JTAG) Mode[7)]; on-chip OSC/PLL non-bypassed (normal)	0000_H

1) In addition to the pins MBC, TMS and P0.0, TM pin also requires an external pull down for all the boot options.

2) BSL mode is automatically entered if no valid password is installed and data at memory address 0000H equals zero.

3) If a device is programmed as LIN, LIN BSL is always used instead of UART/MultiCAN.

4) UART or MultiCAN BSL is decoded by firmware based on the protocol for product variant with MultiCAN. If no MultiCAN and LIN variant, UART BSL is used.

5) In MultiCAN BSL mode, the clock source is switched to XTAL by firmware, bypassing the on-chip oscillator. This avoids any frequency invariance with the on-chip oscillator and allows other frequency clock input, thus ensuring accurate baud rate detection (especially at high bit rates).

6) Alternate BSL Mode is a user defined BSL code programmed in Flash. It is entered if the AltBSLPassword is valid.

7) Normal user mode with standard JTAG (TCK,TDI,TDO) pins for hot-attach purpose.

Note: The boot options are valid only with the default set of UART and JTAG pins.

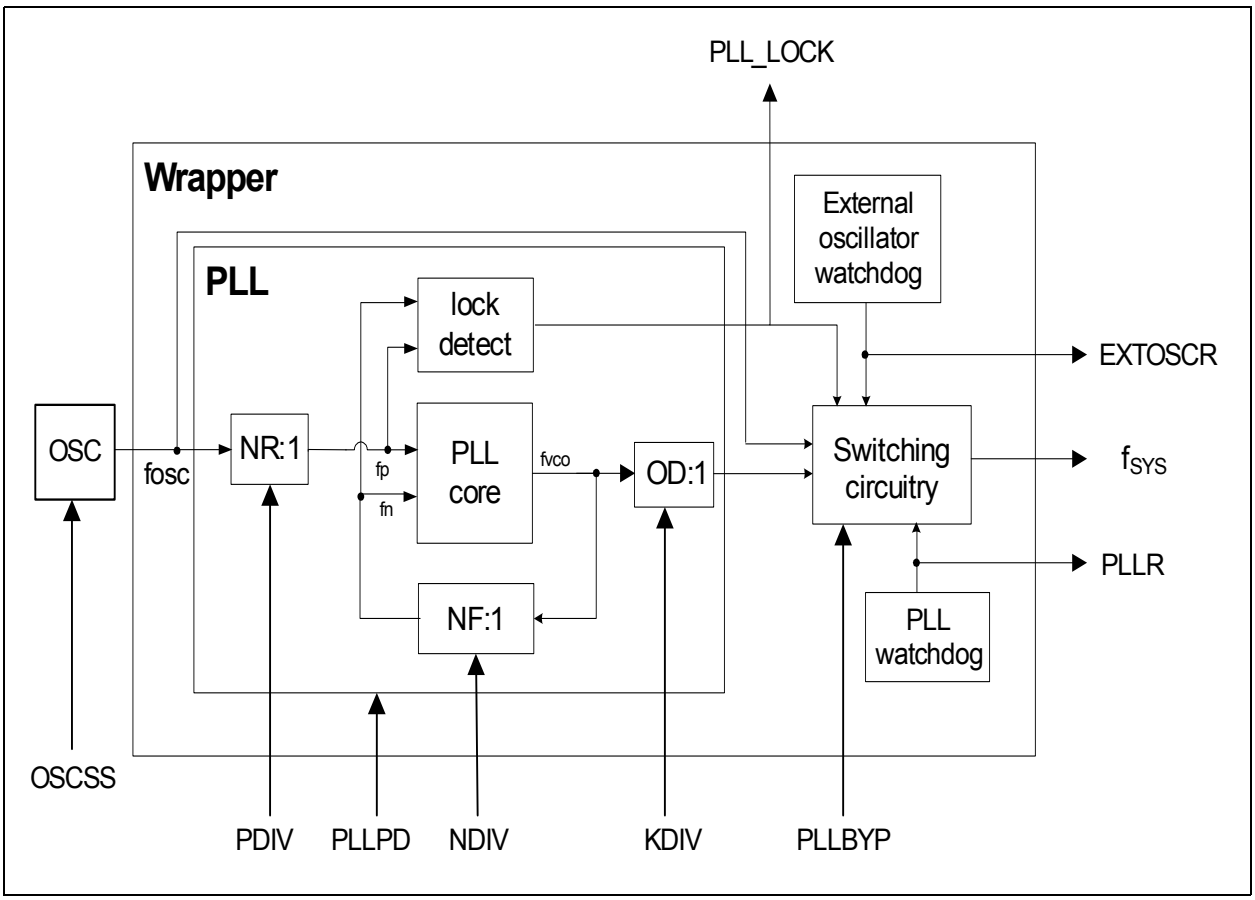

Figure 20 CGU Block Diagram

Direct Drive (PLL Bypass Operation)

During PLL bypass operation, the system clock has the same frequency as the external clock source.

(3.1)

$$f_{SYS} = f_{OSC}$$

PLL Mode

The CPU clock is derived from the oscillator clock, divided by the NR factor (PDIV), multiplied by the NF factor (NDIV), and divided by the OD factor (KDIV). PLL output must

3.9 Power Saving Modes

The power saving modes of the XC87x provide flexible power consumption through a combination of techniques, including:

- Stopping the CPU clock
- Stopping the clocks of individual system components
- Reducing clock speed of some peripheral components
- Power-down of the entire system with fast restart capability

After a reset, the active mode (normal operating mode) is selected by default (see **Figure 23**) and the system runs in the main system clock frequency. From active mode, different power saving modes can be selected by software. They are:

- Idle mode
- Slow-down mode
- Power-down mode

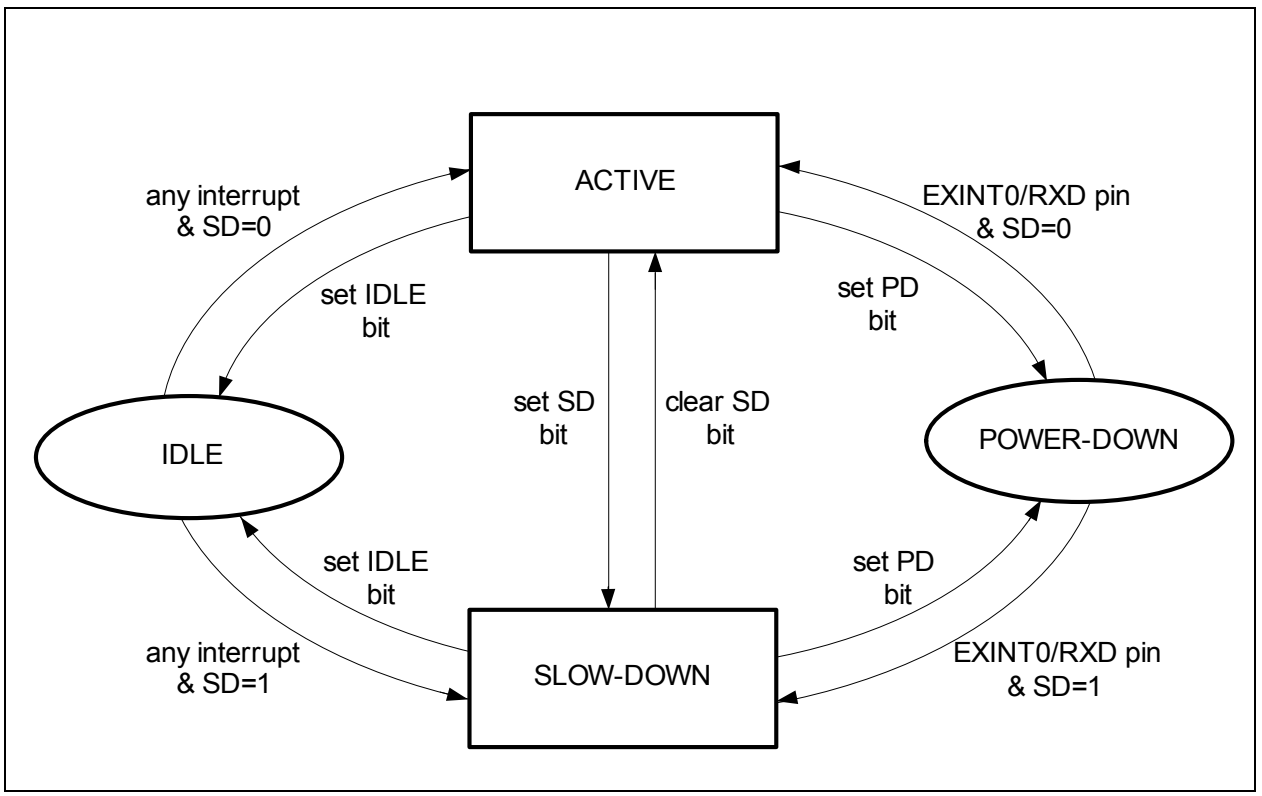

Figure 23 Transition between Power Saving Modes

Note: SAK product variant does not support power-down mode.

fractional divider) for generating a wide range of baud rates based on its input clock f_{PCLK}, see **Figure 26**.

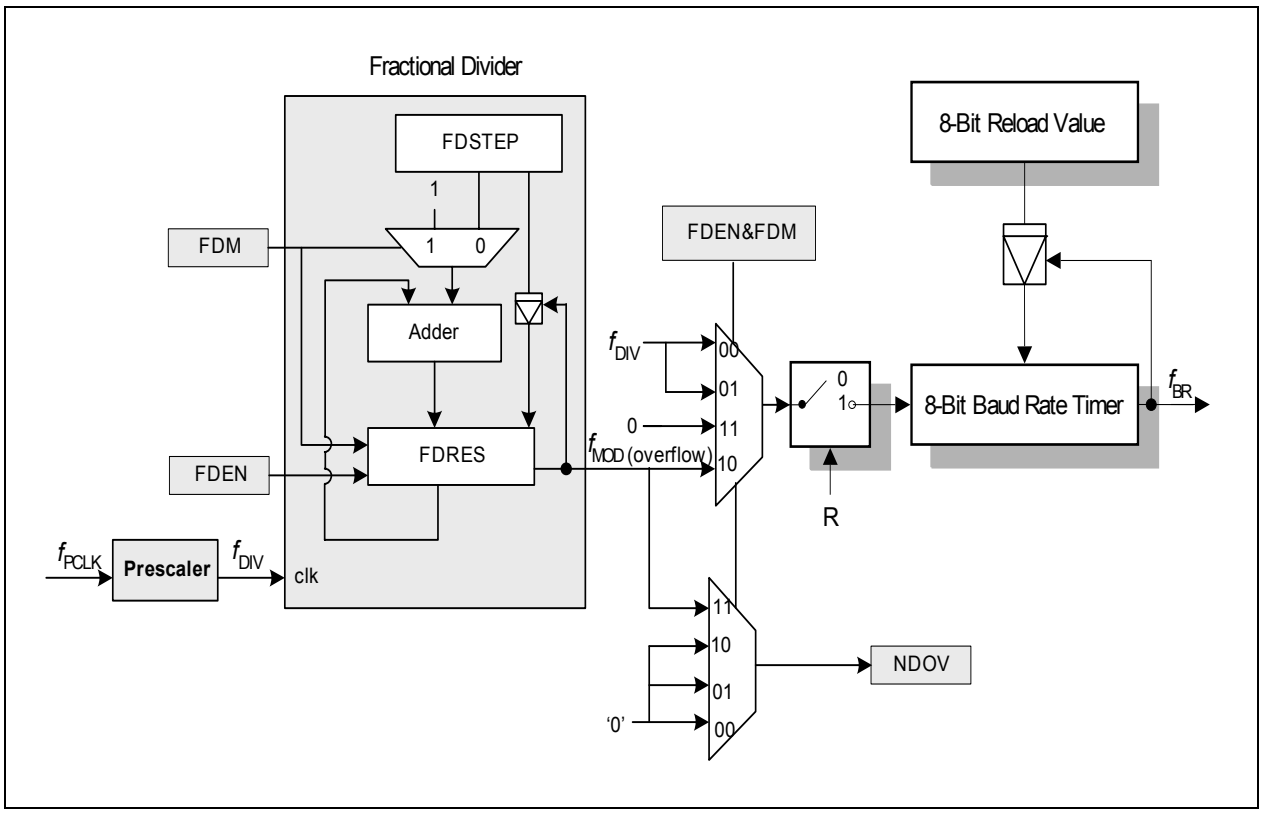

Figure 26 Baud-rate Generator Circuitry

The baud rate timer is a count-down timer and is clocked by either the output of the fractional divider (f_{MOD}) if the fractional divider is enabled (FDCON.FDEN = 1), or the output of the prescaler (f_{DIV}) if the fractional divider is disabled (FDEN = 0). For baud rate generation, the fractional divider must be configured to fractional divider mode (FDCON.FDM = 0). This allows the baud rate control run bit BCON.R to be used to start or stop the baud rate timer. At each timer underflow, the timer is reloaded with the 8-bit reload value in register BG and one clock pulse is generated for the serial channel.

Enabling the fractional divider in normal divider mode (FDEN = 1 and FDM = 1) stops the baud rate timer and nullifies the effect of bit BCON.R. See **Section 3.14**.

The baud rate (f_{BR}) value is dependent on the following parameters:

- Input clock f_{PCLK}
- Prescaling factor (2^{BRPRE}) defined by bit field BRPRE in register BCON
- Fractional divider (STEP/256) defined by register FDSTEP
  (to be considered only if fractional divider is enabled and operating in fractional divider mode)
- 8-bit reload value (BR_VALUE) for the baud rate timer defined by register BG

The following formulas calculate the final baud rate without and with the fractional divider respectively:

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR\_VALUE + 1)} \quad \text{where} \quad 2^{BRPRE} \times (BR\_VALUE + 1) > 1 \tag{3.4}$$

$$\text{baud rate} = \frac{f_{PCLK}}{16 \times 2^{BRPRE} \times (BR\_VALUE + 1)} \times \frac{STEP}{256} \tag{3.5}$$

The maximum baud rate that can be generated is limited to $f_{PCLK}/32$. Hence, for a module clock of 24 MHz, the maximum achievable baud rate is 0.75 MBaud.

Standard LIN protocol can support a maximum baud rate of 20 kHz, the baud rate accuracy is not critical and the fractional divider can be disabled. Only the prescaler is used for auto baud rate calculation. For LIN fast mode, which supports the baud rate of 20 kHz to 57.6 kHz, the higher baud rates require the use of the fractional divider for greater accuracy.

Table 31 lists the various commonly used baud rates with their corresponding parameter settings and deviation errors. The fractional divider is disabled and a module clock of 24 MHz is used.

Table 31 Typical Baud rates for UART with Fractional Divider disabled

Baud rate	Prescaling Factor (2BRPRE)	Reload Value (BR_VALUE + 1)	Deviation Error
19.2 kBaud	1 (BRPRE=000_B)	78 ($4E_H$)	0.17 %
9600 Baud	1 (BRPRE=000_B)	156 ($9C_H$)	0.17 %
4800 Baud	2 (BRPRE=001_B)	156 ($9C_H$)	0.17 %
2400 Baud	4 (BRPRE=010_B)	156 ($9C_H$)	0.17 %

The fractional divider allows baud rates of higher accuracy (lower deviation error) to be generated. **Table 32** lists the resulting deviation errors from generating a baud rate of 57.6 kHz, using different module clock frequencies. The fractional divider is enabled (fractional divider mode) and the corresponding parameter settings are shown.

needed for the handshaking between the master and slave tasks is provided by the master task through the header portion of the frame.

The header consists of a break and synch pattern followed by an identifier. Among these three fields, only the break pattern cannot be transmitted as a normal 8-bit UART data. The break must contain a dominant value of 13 bits or more to ensure proper synchronization of slave nodes.

In the LIN communication, a slave task is required to be synchronized at the beginning of the protected identifier field of frame. For this purpose, every frame starts with a sequence consisting of a break field followed by a synch byte field. This sequence is unique and provides enough information for any slave task to detect the beginning of a new frame and be synchronized at the start of the identifier field.

Upon entering LIN communication, a connection is established and the transfer speed (baud rate) of the serial communication partner (host) is automatically synchronized in the following steps:

STEP 1: Initialize interface for reception and timer for baud rate measurement

STEP 2: Wait for an incoming LIN frame from host

STEP 3: Synchronize the baud rate to the host

STEP 4: Enter for Master Request Frame or for Slave Response Frame

*Note: Re-synchronization and setup of baud rate are always done for **every** Master Request Header or Slave Response Header LIN frame.*

3.21 Controller Area Network (MultiCAN)

The MultiCAN module contains two Full-CAN nodes operating independently or exchanging data and remote frames via a gateway function. Transmission and reception of CAN frames is handled in accordance to CAN specification V2.0 B active. Each CAN node can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

Both CAN nodes share a common set of message objects, where each message object may be individually allocated to one of the CAN nodes. Besides serving as a storage container for incoming and outgoing frames, message objects may be combined to build gateways between the CAN nodes or to setup a FIFO buffer.

The message objects are organized in double chained lists, where each CAN node has it's own list of message objects. A CAN node stores frames only into message objects that are allocated to the list of the CAN node. It only transmits messages from objects of this list. A powerful, command driven list controller performs all list operations.

The bit timings for the CAN nodes are derived from the peripheral clock (f_{CAN}) and are programmable up to a data rate of 1 MBaud. A pair of receive and transmit pins connects each CAN node to a bus transceiver.

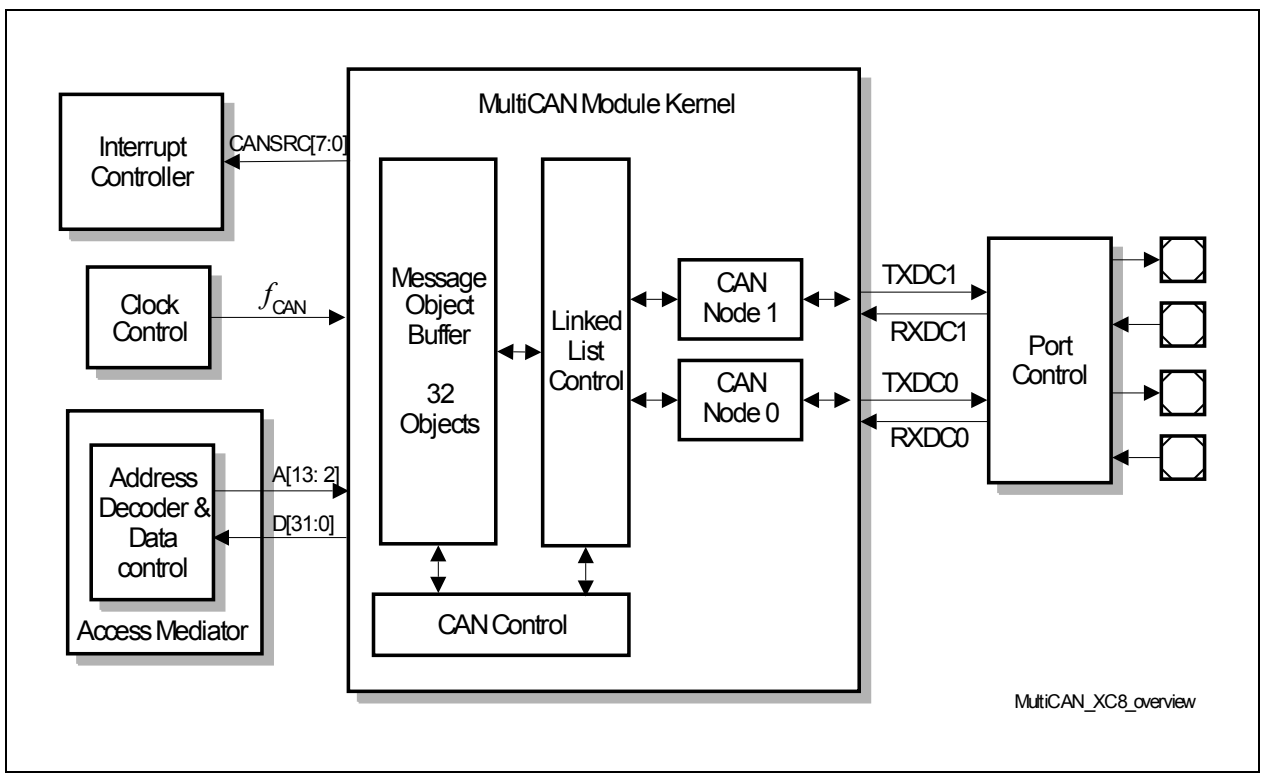

Figure 30 Overview of the MultiCAN

Features

- Compliant to ISO 11898.

4 Electrical Parameters

Chapter 4 provides the characteristics of the electrical parameters which are implementation-specific for the XC87x.

4.1 General Parameters

The general parameters are described here to aid the users in interpreting the parameters mainly in Section 4.2 and Section 4.3.

4.1.1 Parameter Interpretation

The parameters listed in this section represent partly the characteristics of the XC87x and partly its requirements on the system. To aid interpreting the parameters easily when evaluating them for a design, they are indicated by the abbreviations in the “Symbol” column:

- **CC**
  These parameters indicate **C**ontroller **C**haracteristics, which are distinctive features of the XC87x and must be regarded for a system design.
- **SR**
  These parameters indicate **S**ystem **R**equirements, which must be provided by the microcontroller system in which the XC87x is designed in.

4.3.2 Output Rise/Fall Times

Table 47 provides the characteristics of the output rise/fall times in the XC87x.

Table 47 Output Rise/Fall Times Parameters (Operating Conditions apply)

Parameter	Symbol	Limit Values		Unit	Test Conditions
		min.	max.		
V_{DDP} = 5V Range					
Rise/fall times	t_R, t_F	–	10	ns	20 pF.[1) 2)3)]
V_{DDP} = 3.3V Range					
Rise/fall times	t_R, t_F	–	10	ns	20 pF.[1) 2)4)]

1) Rise/Fall time measurements are taken with 10% - 90% of pad supply.
2) Not all parameters are 100% tested, but are verified by design/characterization and test correlation.
3) Additional rise/fall time valid for C_L = 20pF - 100pF @ 0.125 ns/pF.
4) Additional rise/fall time valid for C_L = 20pF - 100pF @ 0.225 ns/pF.

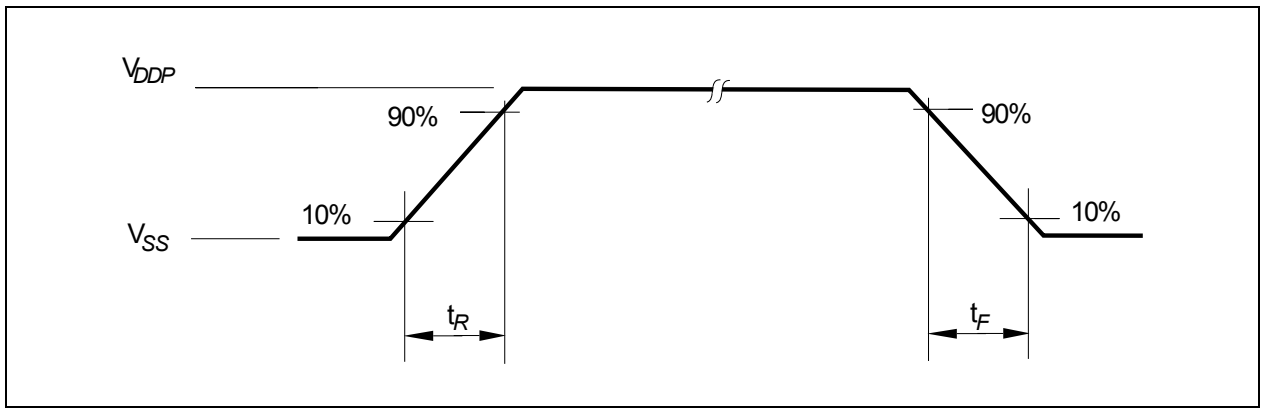

Figure 39 Rise/Fall Times Parameters

4.3.3 Power-on Reset and PLL Timing

Table 48 provides the characteristics of the power-on reset and PLL timing in the XC87x.

Table 48 Power-On Reset and PLL Timing (Operating Conditions apply)

Parameter	Symbol		Limit Values			Unit	Test Conditions
			min.	typ.	max.		
On-Chip Oscillator start-up time	t_{OSCST}	CC	–	–	500	ns	1)
PLL lock-in in time	t_{LOCK}	CC	–	–	200	µs	1)
PLL accumulated jitter	D_P		–	–	1.8	ns	1)2)

1) Not all parameters are 100% tested, but are verified by design/characterization and test correlation.

2) PLL lock at 144 MHz using a 4 MHz external oscillator. The PLL Divider settings are K = 2, N = 72 and P = 1.

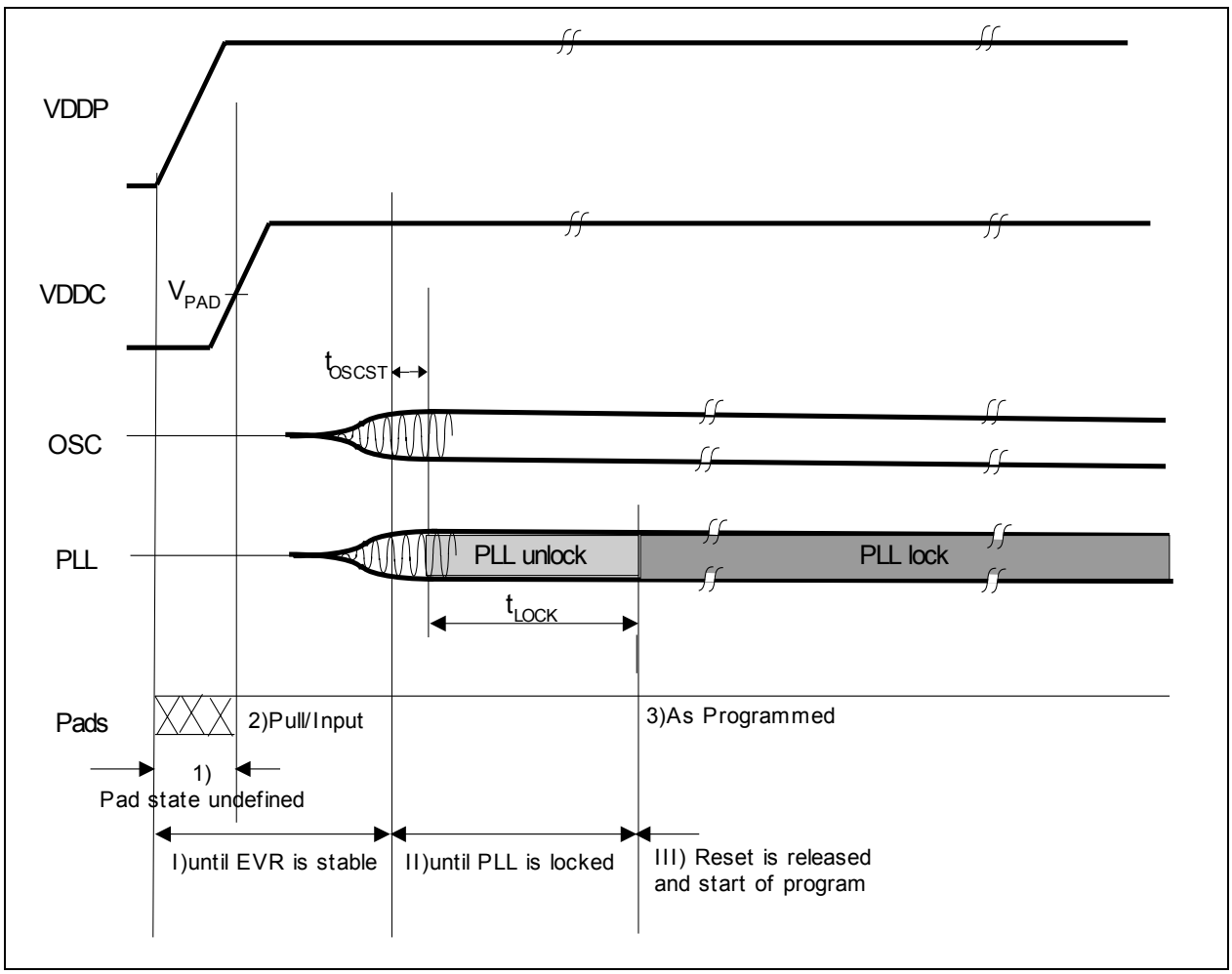

Figure 40 Power-on Reset Timing

5 Package and Quality Declaration

Chapter 5 provides the information of the XC87x package and reliability section.

5.1 Package Parameters

Table 56 provides the thermal characteristics of the package used in XC878 and XC874.

Table 56 Thermal Characteristics of the Packages

Parameter	Symbol		Limit Values		Unit	Notes
			Min.	Max.		
PG-LQFP-64-4 (XC878)						
Thermal resistance junction case[1)]	R_{TJC}	CC	-	13.8	K/W	-
Thermal resistance junction lead[1)]	R_{TJL}	CC	-	34.6	K/W	-
PG-VQFN-48-22 (XC874)						
Thermal resistance junction case[1)]	R_{TJC}	CC	-	16.6	K/W	-
Thermal resistance junction lead[1)]	R_{TJL}	CC	-	30.7	K/W	-

1) The thermal resistances between the case and the ambient (R_{TCA}) , the lead and the ambient (R_{TLA}) are to be combined with the thermal resistances between the junction and the case (R_{TJC}), the junction and the lead (R_{TJL}) given above, in order to calculate the total thermal resistance between the junction and the ambient (R_{TJA}). The thermal resistances between the case and the ambient (R_{TCA}), the lead and the ambient (R_{TLA}) depend on the external system (PCB, case) characteristics, and are under user responsibility.
The junction temperature can be calculated using the following equation: $T_J=T_A+R_{TJA} \times P_D$, where the R_{TJA} is the total thermal resistance between the junction and the ambient. This total junction ambient resistance R_{TJA} can be obtained from the upper four partial thermal resistances, by
a) simply adding only the two thermal resistances (junction lead and lead ambient), or
b) by taking all four resistances into account, depending on the precision needed.