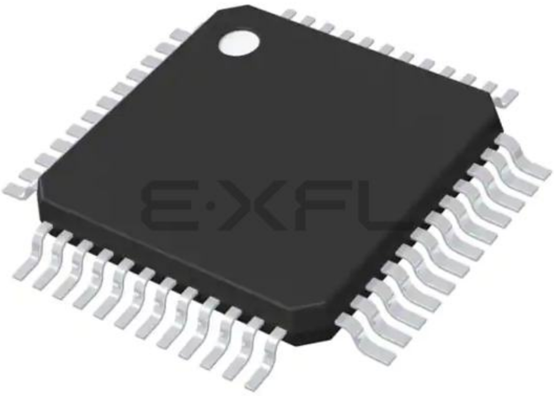

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	XC800
Core Size	8-Bit
Speed	24MHz
Connectivity	CANbus, LINbus, SSI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.75K 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	48-LQFP
Supplier Device Package	PG-TQFP-48
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/saf-xc886clm-6ffa-5v-ac

XC886/888 Variant Devices

The XC886/888 product family features devices with different configurations, program memory sizes, package options, power supply voltage, temperature and quality profiles (Automotive or Industrial), to offer cost-effective solutions for different application requirements.

The list of XC886/888 device configurations are summarized in **Table 1**. For each configuration, 2 types of packages are available:

- PG-TQFP-48, which is denoted by XC886 and;
- PG-TQFP-64, which is denoted by XC888.

Table 1 Device Configuration

Device Name	CAN Module	LIN BSL Support	MDU Module
XC886/888	No	No	No
XC886/888C	Yes	No	No
XC886/888CM	Yes	No	Yes
XC886/888LM	No	Yes	Yes
XC886/888CLM	Yes	Yes	Yes

Note: For variants with LIN BSL support, only LIN BSL is available regardless of the availability of the CAN module.

From these 10 different combinations of configuration and package type, each are further made available in many sales types, which are grouped according to device type, program memory sizes, power supply voltage, temperature and quality profile (Automotive or Industrial), as shown in **Table 2**.

Table 2 Device Profile

Sales Type	Device Type	Program Memory (Kbytes)	Power Supply (V)	Temp-erature (°C)	Quality Profile
SAK-XC886*/888*-8FFA 5V	Flash	32	5.0	-40 to 125	Automotive
SAK-XC886*/888*-6FFA 5V	Flash	24	5.0	-40 to 125	Automotive
SAF-XC886*/888*-8FFA 5V	Flash	32	5.0	-40 to 85	Automotive
SAF-XC886*/888*-6FFA 5V	Flash	24	5.0	-40 to 85	Automotive
SAF-XC886*/888*-8FFI 5V	Flash	32	5.0	-40 to 85	Industrial
SAF-XC886*/888*-6FFI 5V	Flash	24	5.0	-40 to 85	Industrial

2 General Device Information

Chapter 2 contains the block diagram, pin configurations, definitions and functions of the XC886/888.

2.1 Block Diagram

The block diagram of the XC886/888 is shown in **Figure 2**.

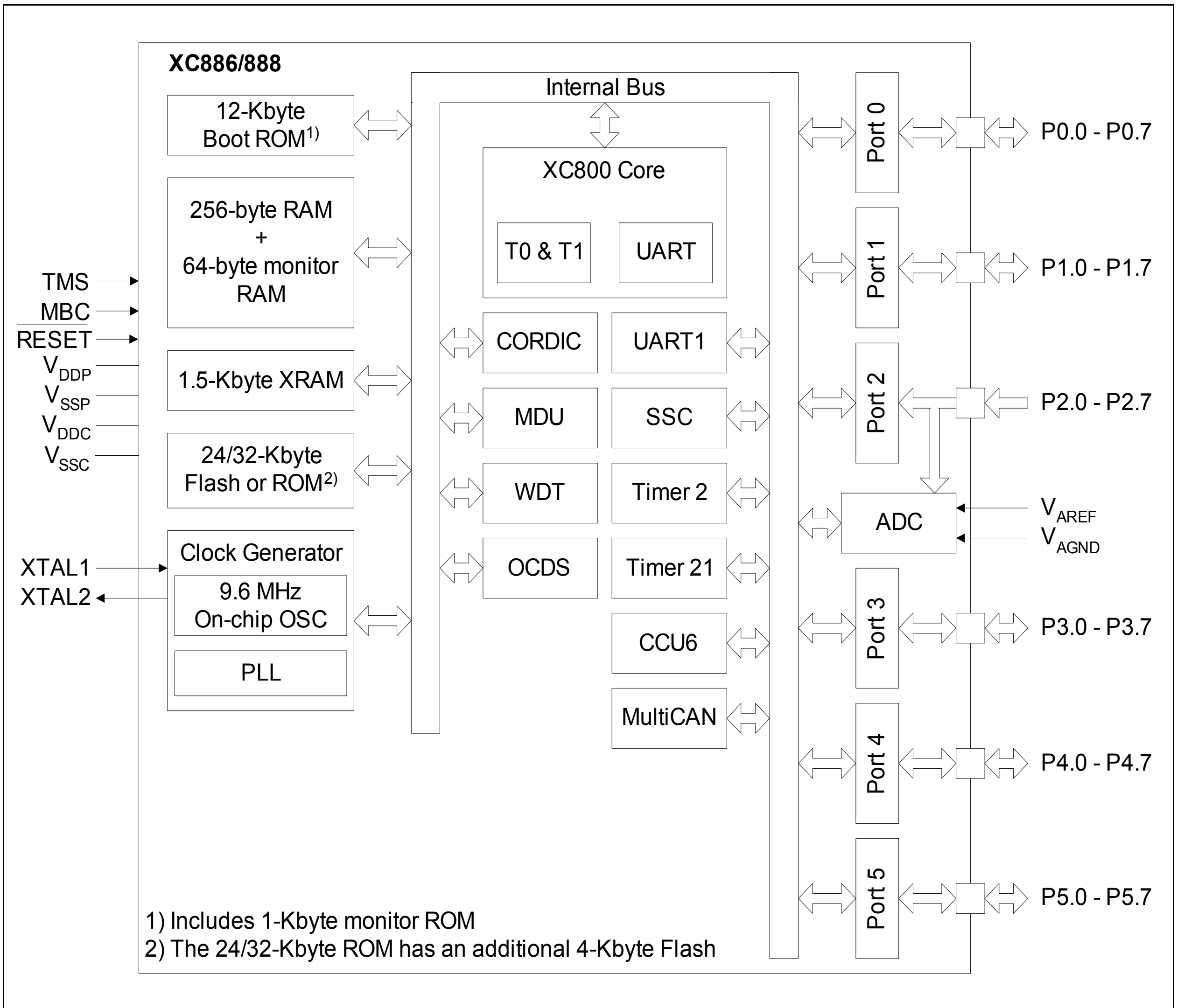

Figure 2 XC886/888 Block Diagram

2.4 Pin Definitions and Functions

The functions and default states of the XC886/888 external pins are provided in Table 3.

Table 3 Pin Definitions and Functions

Symbol	Pin Number (TQFP-48/64)	Type	Reset State	Function	
P0		I/O		**Port 0** Port 0 is an 8-bit bidirectional general purpose I/O port. It can be used as alternate functions for the JTAG, CCU6, UART, UART1, Timer 2, Timer 21, MultiCAN and SSC.	
P0.0	11/17		Hi-Z	TCK_0	JTAG Clock Input
				T12HR_1	CCU6 Timer 12 Hardware Run Input
				CC61_1	Input/Output of Capture/Compare channel 1
				CLKOUT_0	Clock Output
				RXDO_1	UART Transmit Data Output
P0.1	13/21		Hi-Z	TDI_0	JTAG Serial Data Input
				T13HR_1	CCU6 Timer 13 Hardware Run Input
				RXD_1	UART Receive Data Input
				RXDC1_0	MultiCAN Node 1 Receiver Input
				COUT61_1	Output of Capture/Compare channel 1
				EXF2_1	Timer 2 External Flag Output
P0.2	12/18		PU	$\overline{\mathrm{CTRAP\_2}}$	CCU6 Trap Input
				TDO_0	JTAG Serial Data Output
				TXD_1	UART Transmit Data Output/Clock Output
				TXDC1_0	MultiCAN Node 1 Transmitter Output
P0.3	48/63		Hi-Z	SCK_1	SSC Clock Input/Output
				COUT63_1	Output of Capture/Compare channel 3
				RXDO1_0	UART1 Transmit Data Output

Table 3 Pin Definitions and Functions (cont'd)

Symbol	Pin Number (TQFP-48/64)	Type	Reset State	Function	
P0.4	1/64		Hi-Z	MTSR_1	SSC Master Transmit Output/ Slave Receive Input
				CC62_1	Input/Output of Capture/Compare channel 2
				TXD1_0	UART1 Transmit Data Output/Clock Output
P0.5	2/1		Hi-Z	MRST_1	SSC Master Receive Input/Slave Transmit Output
				EXINT0_0	External Interrupt Input 0
				T2EX1_1	Timer 21 External Trigger Input
				RXD1_0	UART1 Receive Data Input
				COUT62_1	Output of Capture/Compare channel 2
P0.6	–/2		PU	GPIO	
P0.7	47/62		PU	CLKOUT_1	Clock Output

Table 3 Pin Definitions and Functions (cont'd)

Symbol	Pin Number (TQFP-48/64)	Type	Reset State	Function	
P1.6	8/10		PU	CCPOS1_1	CCU6 Hall Input 1
				T12HR_0	CCU6 Timer 12 Hardware Run Input
				EXINT6_0	External Interrupt Input 6
				RXDC0_2	MultiCAN Node 0 Receiver Input
				T21_1	Timer 21 Input
P1.7	9/11		PU	CCPOS2_1	CCU6 Hall Input 2
				T13HR_0	CCU6 Timer 13 Hardware Run Input
				T2_1	Timer 2 Input
				TXDC0_2	MultiCAN Node 0 Transmitter Output
				P1.5 and P1.6 can be used as a software chip select output for the SSC.	

code or data. Therefore, even though the ROM device contains either a 24-Kbyte or 32-Kbyte ROM, the maximum size of code that can be placed in the ROM is the given size less four bytes.

3.2.1 Memory Protection Strategy

The XC886/888 memory protection strategy includes:

- Read-out protection: The user is able to protect the contents in the Flash (for Flash devices) and ROM (for ROM devices) memory from being read
  – Flash protection is enabled by programming a valid password (8-bit non-zero value) via BSL mode 6.
  – ROM protection is fixed with the ROM mask and is always enabled.
- Flash program and erase protection: This feature is available only for Flash devices.

3.2.1.1 Flash Memory Protection

As long as a valid password is available, all external access to the device, including the Flash, will be blocked.

For additional security, the Flash hardware protection can be enabled to implement a second layer of read-out protection, as well as to enable program and erase protection.

Flash hardware protection is available only for Flash devices and comes in two modes:

- Mode 0: Only the P-Flash is protected; the D-Flash is unprotected
- Mode 1: Both the P-Flash and D-Flash are protected

The selection of each protection mode and the restrictions imposed are summarized in Table 4.

Table 4 Flash Protection Modes

Flash Protection	Without hardware protection	With hardware protection	
Hardware Protection Mode	-	0	1
Activation	Program a valid password via BSL mode 6		
Selection	Bit 4 of password = 0	Bit 4 of password = 1 MSB of password = 0	Bit 4 of password = 1 MSB of password = 1
P-Flash contents can be read by	Read instructions in any program memory	Read instructions in the P-Flash	Read instructions in the P-Flash or D-Flash
External access to P-Flash	Not possible	Not possible	Not possible

Field	Bits	Type	Description
OP	[7:6]	w	**Operation** 0X Manual page mode. The value of STNR is ignored and PAGE is directly written. 10 New page programming with automatic page saving. The value written to the bit positions of PAGE is stored. In parallel, the previous contents of PAGE are saved in the storage bit field STx indicated by STNR. 11 Automatic restore page action. The value written to the bit positions PAGE is ignored and instead, PAGE is overwritten by the contents of the storage bit field STx indicated by STNR.
0	3	r	**Reserved** Returns 0 if read; should be written with 0.

3.2.3 Bit Protection Scheme

The bit protection scheme prevents direct software writing of selected bits (i.e., protected bits) using the PASSWD register. When the bit field MODE is 11_B, writing 10011_B to the bit field PASS opens access to writing of all protected bits, and writing 10101_B to the bit field PASS closes access to writing of all protected bits. In both cases, the value of the bit field MODE is not changed even if PASSWD register is written with 98_H or $A8_H$. It can only be changed when bit field PASS is written with 11000_B, for example, writing $D0_H$ to PASSWD register disables the bit protection scheme.

Note that access is opened for maximum 32 CCLKs if the “close access” password is not written. If “open access” password is written again before the end of 32 CCLK cycles, there will be a recount of 32 CCLK cycles. The protected bits include the N- and K-Divider bits, NDIV and KDIV; the Watchdog Timer enable bit, WDTEN; and the power-down and slow-down enable bits, PD and SD.

Table 11 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
CC_H	**ADC_CHINSR** **Reset: 00_H** Channel Interrupt Set Register	Bit Field	CHINS 7	CHINS 6	CHINS 5	CHINS 4	CHINS 3	CHINS 2	CHINS 1	CHINS 0
		Type	w	w	w	w	w	w	w	w
CD_H	**ADC_CHINPR** **Reset: 00_H** Channel Interrupt Node Pointer Register	Bit Field	CHINP 7	CHINP 6	CHINP 5	CHINP 4	CHINP 3	CHINP 2	CHINP 1	CHINP 0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
CE_H	**ADC_EVINFR** **Reset: 00_H** Event Interrupt Flag Register	Bit Field	EVINF 7	EVINF 6	EVINF 5	EVINF 4	0		EVINF 1	EVINF 0
		Type	rh	rh	rh	rh	r		rh	rh
CF_H	**ADC_EVINCR** **Reset: 00_H** Event Interrupt Clear Flag Register	Bit Field	EVINC 7	EVINC 6	EVINC 5	EVINC 4	0		EVINC 1	EVINC 0
		Type	w	w	w	w	r		w	w
$D2_H$	**ADC_EVINSR** **Reset: 00_H** Event Interrupt Set Flag Register	Bit Field	EVINS 7	EVINS 6	EVINS 5	EVINS 4	0		EVINS 1	EVINS 0
		Type	w	w	w	w	r		w	w
$D3_H$	**ADC_EVINPR** **Reset: 00_H** Event Interrupt Node Pointer Register	Bit Field	EVINP 7	EVINP 6	EVINP 5	EVINP 4	0		EVINP 1	EVINP 0
		Type	rw	rw	rw	rw	r		rw	rw
RMAP = 0, PAGE 6										
CA_H	**ADC_CRCR1** **Reset: 00_H** Conversion Request Control Register 1	Bit Field	CH7	CH6	CH5	CH4	0			
		Type	rwh	rwh	rwh	rwh	r			
CB_H	**ADC_CRPR1** **Reset: 00_H** Conversion Request Pending Register 1	Bit Field	CHP7	CHP6	CHP5	CHP4	0			
		Type	rwh	rwh	rwh	rwh	r			
CC_H	**ADC_CRMR1** **Reset: 00_H** Conversion Request Mode Register 1	Bit Field	Rsv	LDEV	CLRP ND	SCAN	ENSI	ENTR	0	ENGT
		Type	r	w	w	rw	rw	rw	r	rw
CD_H	**ADC_QMR0** **Reset: 00_H** Queue Mode Register 0	Bit Field	CEV	TREV	FLUS H	CLRV	0	ENTR	0	ENGT
		Type	w	w	w	w	r	rw	r	rw
CE_H	**ADC_QSR0** **Reset: 20_H** Queue Status Register 0	Bit Field	Rsv	0	EMPT Y	EV	0		FILL	
		Type	r	r	rh	rh	r		rh	
CF_H	**ADC_Q0R0** **Reset: 00_H** Queue 0 Register 0	Bit Field	EXTR	ENSI	RF	V	0	REQCHNR		
		Type	rh	rh	rh	rh	r	rh		
$D2_H$	**ADC_QBUR0** **Reset: 00_H** Queue Backup Register 0	Bit Field	EXTR	ENSI	RF	V	0	REQCHNR		
		Type	rh	rh	rh	rh	r	rh		
$D2_H$	**ADC_QINR0** **Reset: 00_H** Queue Input Register 0	Bit Field	EXTR	ENSI	RF	0		REQCHNR		
		Type	w	w	w	r		w		

3.2.4.12 SSC Registers

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

Table 16 SSC Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$A9_H$	**SSC_PISEL** **Reset: 00_H** Port Input Select Register	Bit Field	0					CIS	SIS	MIS
		Type	r					rw	rw	rw
AA_H	**SSC_CONL** **Reset: 00_H** Control Register Low Programming Mode	Bit Field	LB	PO	PH	HB	BM			
		Type	rw	rw	rw	rw	rw			
AA_H	**SSC_CONL** **Reset: 00_H** Control Register Low Operating Mode	Bit Field	0				BC			
		Type	r				rh			
AB_H	**SSC_CONH** **Reset: 00_H** Control Register High Programming Mode	Bit Field	EN	MS	0	AREN	BEN	PEN	REN	TEN
		Type	rw	rw	r	rw	rw	rw	rw	rw
AB_H	**SSC_CONH** **Reset: 00_H** Control Register High Operating Mode	Bit Field	EN	MS	0	BSY	BE	PE	RE	TE
		Type	rw	rw	r	rh	rwh	rwh	rwh	rwh
AC_H	**SSC_TBL** **Reset: 00_H** Transmitter Buffer Register Low	Bit Field	TB_VALUE							
		Type	rw							
AD_H	**SSC_RBL** **Reset: 00_H** Receiver Buffer Register Low	Bit Field	RB_VALUE							
		Type	rh							
AE_H	**SSC_BRL** **Reset: 00_H** Baud Rate Timer Reload Register Low	Bit Field	BR_VALUE							
		Type	rw							
AF_H	**SSC_BRH** **Reset: 00_H** Baud Rate Timer Reload Register High	Bit Field	BR_VALUE							
		Type	rw							

3.2.4.13 MultiCAN Registers

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

Table 17 CAN Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$D8_H$	**ADCON** **Reset: 00_H** CAN Address/Data Control Register	Bit Field	V3	V2	V1	V0	AUAD		BSY	RWEN
		Type	rw	rw	rw	rw	rw		rh	rw
$D9_H$	**ADL** **Reset: 00_H** CAN Address Register Low	Bit Field	CA9	CA8	CA7	CA6	CA5	CA4	CA3	CA2
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
DA_H	**ADH** **Reset: 00_H** CAN Address Register High	Bit Field	0				CA13	CA12	CA11	CA10
		Type	r				rwh	rwh	rwh	rwh

3.4.3 Interrupt Priority

An interrupt that is currently being serviced can only be interrupted by a higher-priority interrupt, but not by another interrupt of the same or lower priority. Hence, an interrupt of the highest priority cannot be interrupted by any other interrupt request.

If two or more requests of different priority levels are received simultaneously, the request of the highest priority is serviced first. If requests of the same priority are received simultaneously, then an internal polling sequence determines which request is serviced first. Thus, within each priority level, there is a second priority structure determined by the polling sequence shown in **Table 21**.

Table 21 Priority Structure within Interrupt Level

Source	Level
Non-Maskable Interrupt (NMI)	(highest)
External Interrupt 0	1
Timer 0 Interrupt	2
External Interrupt 1	3
Timer 1 Interrupt	4
UART Interrupt	5
Timer 2,UART Normal Divider Overflow, MultiCAN, LIN Interrupt	6
ADC, MultiCAN Interrupt	7
SSC Interrupt	8
External Interrupt 2, Timer 21, UART1, UART1 Normal Divider Overflow, MDU, CORDIC Interrupt	9
External Interrupt [6:3], MultiCAN Interrupt	10
CCU6 Interrupt Node Pointer 0, MultiCAN interrupt	11
CCU6 Interrupt Node Pointer 1, MultiCAN Interrupt	12
CCU6 Interrupt Node Pointer 2, MultiCAN Interrupt	13
CCU6 Interrupt Node Pointer 3, MultiCAN Interrupt	14

Figure 20 shows the structure of an input-only port pin.

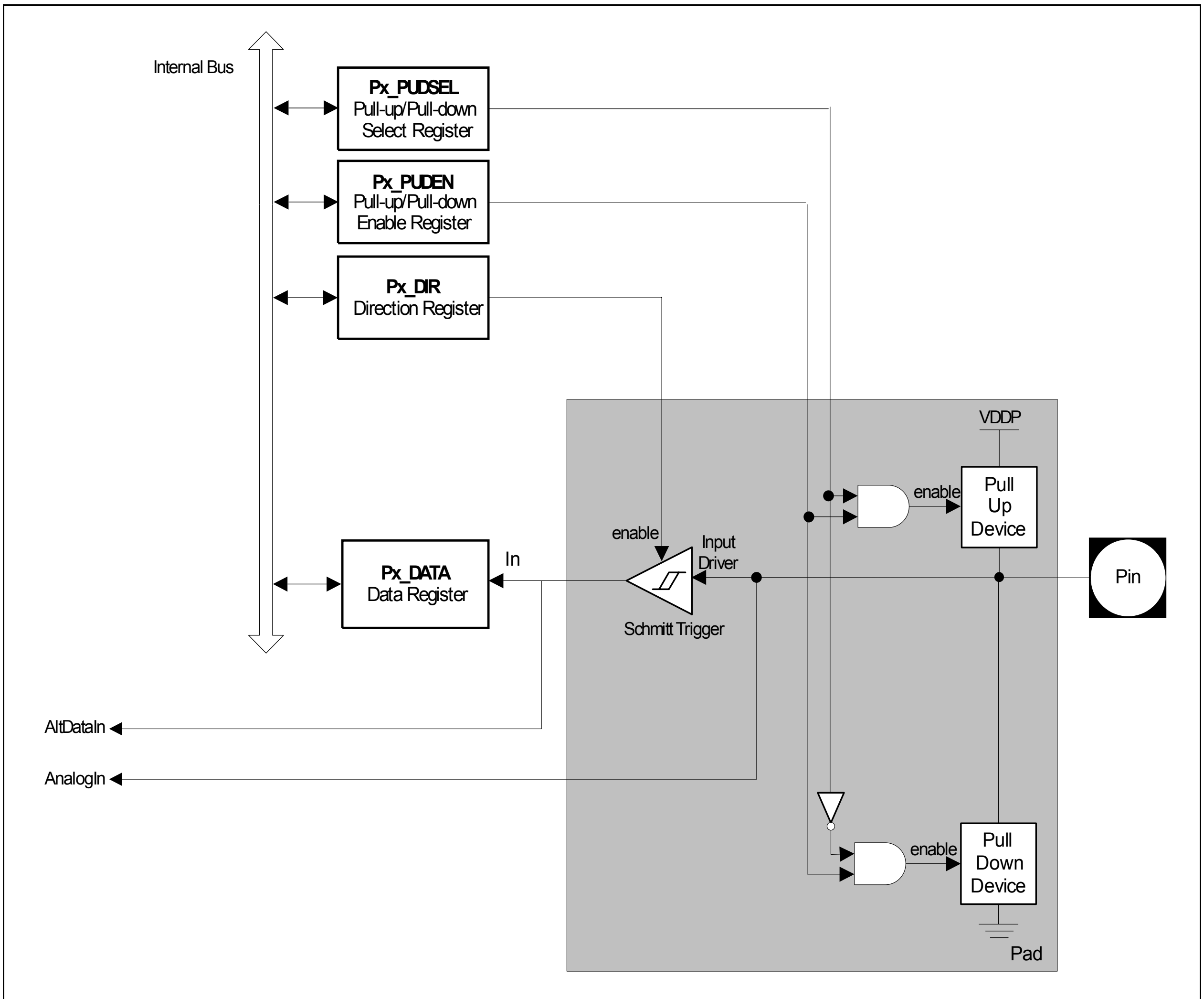

Figure 20 General Structure of Input Port

1) BSL mode is automatically entered if no valid password is installed and data at memory address 0000H equals zero.
2) OSC is bypassed in MultiCAN BSL mode
3) 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.

3.8 Clock Generation Unit

The Clock Generation Unit (CGU) allows great flexibility in the clock generation for the XC886/888. The power consumption is indirectly proportional to the frequency, whereas the performance of the microcontroller is directly proportional to the frequency. During user program execution, the frequency can be programmed for an optimal ratio between performance and power consumption. Therefore the power consumption can be adapted to the actual application state.

Features

- Phase-Locked Loop (PLL) for multiplying clock source by different factors
- PLL Base Mode
- Prescaler Mode
- PLL Mode
- Power-down mode support

The CGU consists of an oscillator circuit and a PLL. In the XC886/888, the oscillator can be from either of these two sources: the on-chip oscillator (9.6 MHz) or the external oscillator (4 MHz to 12 MHz). The term “oscillator” is used to refer to both on-chip oscillator and external oscillator, unless otherwise stated. After the reset, the on-chip oscillator will be used by default.The external oscillator can be selected via software. In addition, the PLL provides a fail-safe logic to perform oscillator run and loss-of-lock detection. This allows emergency routines to be executed for system recovery or to perform system shut down.

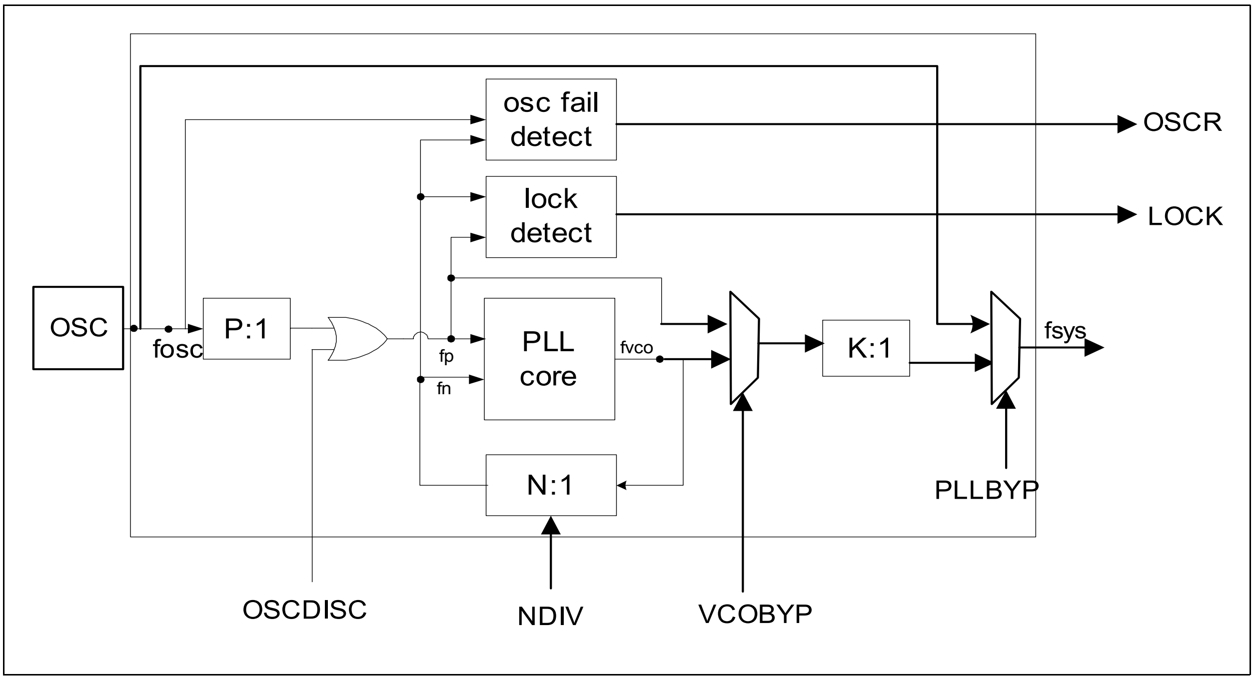

Figure 24 CGU Block Diagram

PLL Base Mode

When the oscillator is disconnected from the PLL, the system clock is derived from the VCO base (free running) frequency clock (**Table 25**) divided by the K factor.

$$f_{SYS} = f_{VCObase} \times \frac{1}{K} \tag{3.1}$$

Prescaler Mode (VCO Bypass Operation)

In VCO bypass operation, the system clock is derived from the oscillator clock, divided by the P and K factors.

$$f_{SYS} = f_{OSC} \times \frac{1}{P \times K} \tag{3.2}$$

3.15 LIN Protocol

The UART module can be used to support the Local Interconnect Network (LIN) protocol for both master and slave operations. The LIN baud rate detection feature, which consists of the hardware logic for Break and Synch Byte detection, provides the capability to detect the baud rate within LIN protocol using Timer 2. This allows the UART to be synchronized to the LIN baud rate for data transmission and reception.

Note: The LIN baud rate detection feature is available for use only with UART. To use UART1 for LIN communication, software has to be implemented to detect the Break and Synch Byte.

LIN is a holistic communication concept for local interconnected networks in vehicles. The communication is based on the SCI (UART) data format, a single-master/multiple-slave concept, a clock synchronization for nodes without stabilized time base. An attractive feature of LIN is self-synchronization of the slave nodes without a crystal or ceramic resonator, which significantly reduces the cost of hardware platform. Hence, the baud rate must be calculated and returned with every message frame.

The structure of a LIN frame is shown in **Figure 31**. The frame consists of the:

- Header, which comprises a Break (13-bit time low), Synch Byte (55_H), and ID field
- Response time
- Data bytes (according to UART protocol)
- Checksum

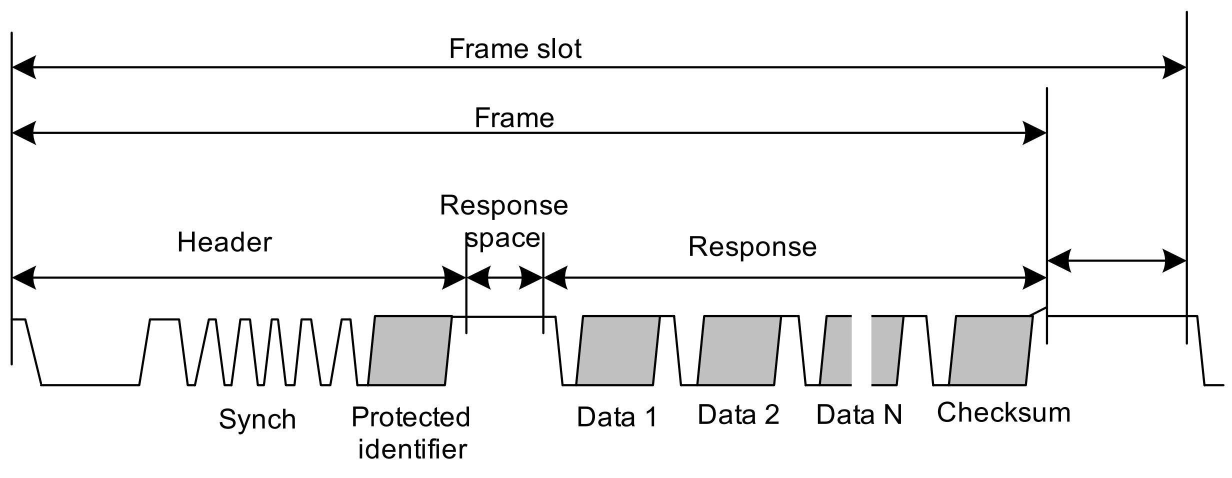

Figure 31 Structure of LIN Frame

3.15.1 LIN Header Transmission

LIN header transmission is only applicable in master mode. In the LIN communication, a master task decides when and which frame is to be transferred on the bus. It also identifies a slave task to provide the data transported by each frame. The information 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.*

GLOBCTR. A prescaling ratio of 32 can be selected when the maximum performance of the ADC is not required.

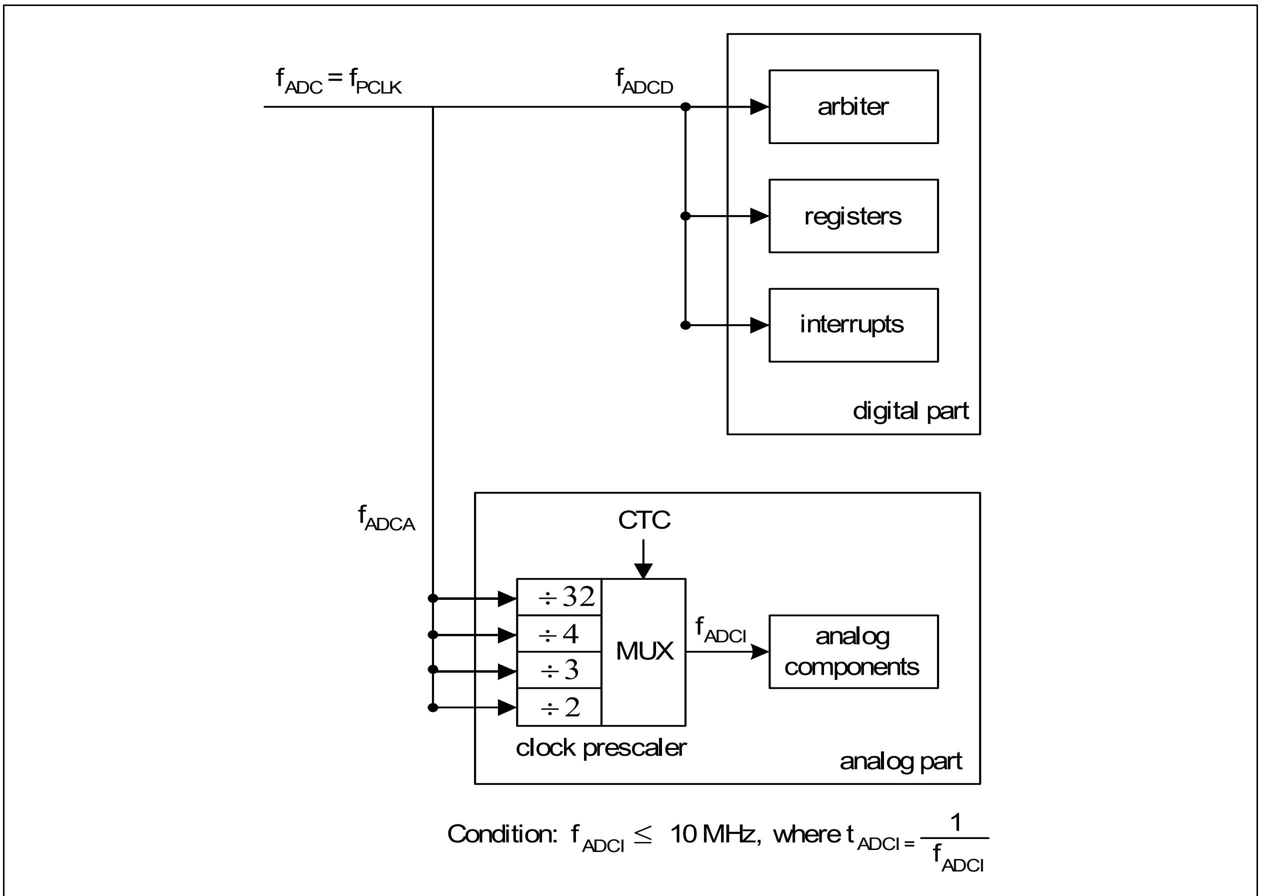

Figure 35 ADC Clocking Scheme

For module clock f_{ADC} = 24 MHz, the analog clock f_{ADCI} frequency can be selected as shown in **Table 34**.

Table 34 f_{ADCI} Frequency Selection

Module Clock f_{ADC}	CTC	Prescaling Ratio	Analog Clock f_{ADCI}
24 MHz	00_B	÷ 2	12 MHz (N.A)
	01_B	÷ 3	8 MHz
	10_B	÷ 4	6 MHz
	11_B (default)	÷ 32	750 kHz

As f_{ADCI} cannot exceed 10 MHz, bit field CTC should not be set to 00_B when f_{ADC} is 24 MHz. During slow-down mode where f_{ADC} may be reduced to 12 MHz, 6 MHz etc., CTC can be set to 00_B as long as the divided analog clock f_{ADCI} does not exceed 10 MHz.

However, it is important to note that the conversion error could increase due to loss of charges on the capacitors, if f_{ADC} becomes too low during slow-down mode.

3.21.2 ADC Conversion Sequence

The analog-to-digital conversion procedure consists of the following phases:

- Synchronization phase (t_{SYN})
- Sample phase (t_S)
- Conversion phase
- Write result phase (t_{WR})

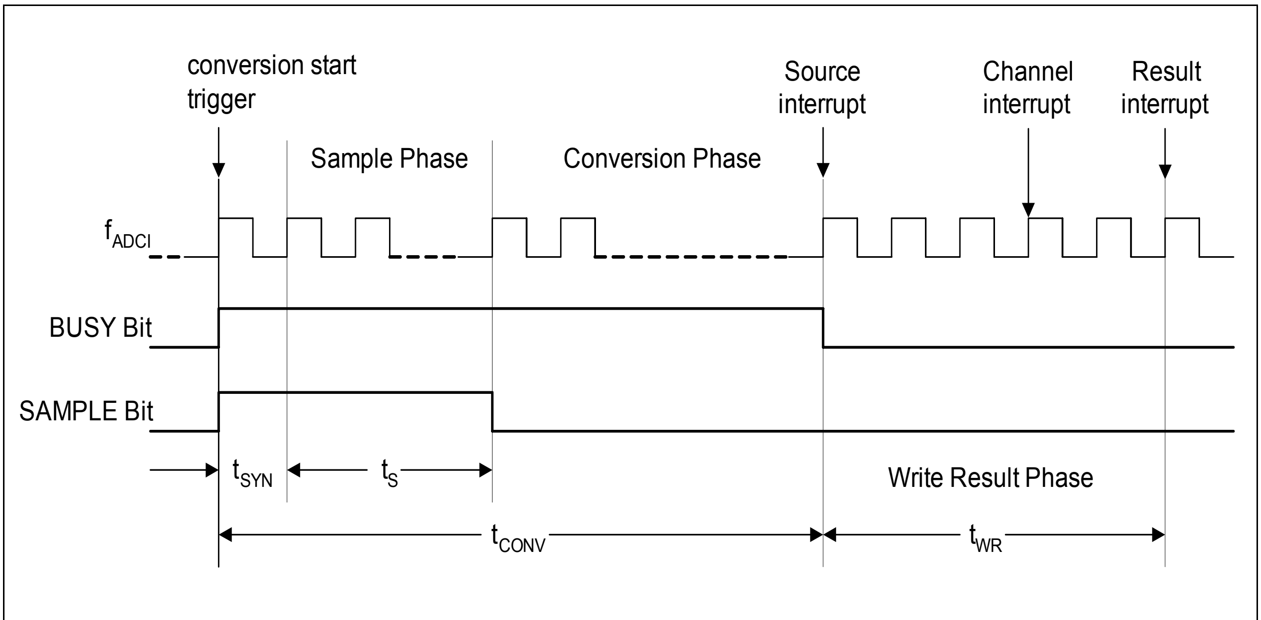

Figure 36 ADC Conversion Timing

Table 36 Chip Identification Number (cont'd)

Product Variant	Chip Identification Number		
	AA-Step	**AB-Step**	**AC-Step**
XC886-6FFA 3V3	-	$095D1562_H$	$0B5D1562_H$
XC888-6FFA 3V3	-	$095D1563_H$	$0B5D1563_H$
XC886CLM-8FFA 5V	-	09900102_H	$0B900102_H$
XC888CLM-8FFA 5V	-	09900103_H	$0B900103_H$
XC886LM-8FFA 5V	-	09900122_H	$0B900122_H$
XC888LM-8FFA 5V	-	09900123_H	$0B900123_H$
XC886CLM-6FFA 5V	-	09951502_H	$0B951502_H$
XC888CLM-6FFA 5V	-	09951503_H	$0B951503_H$
XC886LM-6FFA 5V	-	09951522_H	$0B951522_H$
XC888LM-6FFA 5V	-	09951523_H	$0B951523_H$
XC886CM-8FFA 5V	-	09980102_H	$0B980102_H$
XC888CM-8FFA 5V	-	09980103_H	$0B980103_H$
XC886C-8FFA 5V	-	09980142_H	$0B980142_H$
XC888C-8FFA 5V	-	09980143_H	$0B980143_H$
XC886-8FFA 5V	-	09980162_H	$0B980162_H$
XC888-8FFA 5V	-	09980163_H	$0B980163_H$
XC886CM-6FFA 5V	-	$099D1502_H$	$0B9D1502_H$
XC888CM-6FFA 5V	-	$099D1503_H$	$0B9D1503_H$
XC886C-6FFA 5V	-	$099D1542_H$	$0B9D1542_H$
XC888C-6FFA 5V	-	$099D1543_H$	$0B9D1543_H$
XC886-6FFA 5V	-	$099D1562_H$	$0B9D1562_H$
XC888-6FFA 5V	-	$099D1563_H$	$0B9D1563_H$
ROM Devices			
XC886CLM-8RFA 3V3	22400502_H	-	-
XC888CLM-8RFA 3V3	22400503_H	-	-
XC886LM-8RFA 3V3	22400522_H	-	-
XC888LM-8RFA 3V3	22400523_H	-	-
XC886CLM-6RFA 3V3	22411502_H	-	-
XC888CLM-6RFA 3V3	22411503_H	-	-

Table 36 Chip Identification Number (cont'd)

Product Variant	Chip Identification Number		
	AA-Step	**AB-Step**	**AC-Step**
XC888CM-6RFA 5V	22891503_H	-	-
XC886C-6RFA 5V	22891542_H	-	-
XC888C-6RFA 5V	22891543_H	-	-
XC886-6RFA 5V	22891562_H	-	-
XC888-6RFA 5V	22891563_H	-	-

4 Electrical Parameters

Chapter 4 provides the characteristics of the electrical parameters which are implementation-specific for the XC886/888.

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 XC886/888 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 XC886/888 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 XC886/888 is designed in.