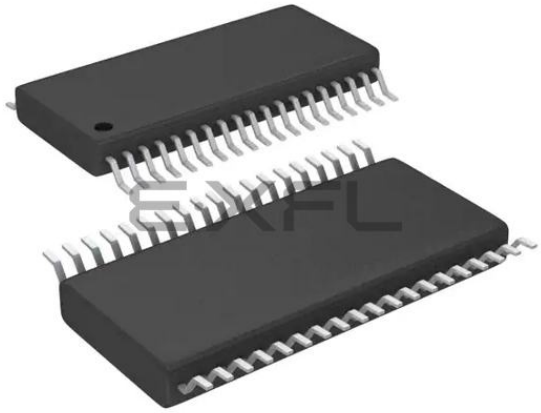


Welcome to E-XFL.COM

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	Last Time Buy
Core Processor	XC800
Core Size	8-Bit
Speed	86MHz
Connectivity	SSI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	19
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	38-TFSOP (0.173", 4.40mm Width)
Supplier Device Package	PG-TSSOP-38
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/xc8662fribekxuma1

1 Summary of Features

- High-performance XC800 Core
  - compatible with standard 8051 processor
  - two clocks per machine cycle architecture (for memory access without wait state)
  - two data pointers
- On-chip memory
  - 8 Kbytes of Boot ROM
  - 256 bytes of RAM
  - 512 bytes of XRAM
  - 4/8/16 Kbytes of Flash; or
    8/16 Kbytes of ROM, with additional 4 Kbytes of Flash
    (includes memory protection strategy)
- I/O port supply at 3.3 V/5.0 V and core logic supply at 2.5 V (generated by embedded voltage regulator)

(further features are on next page)

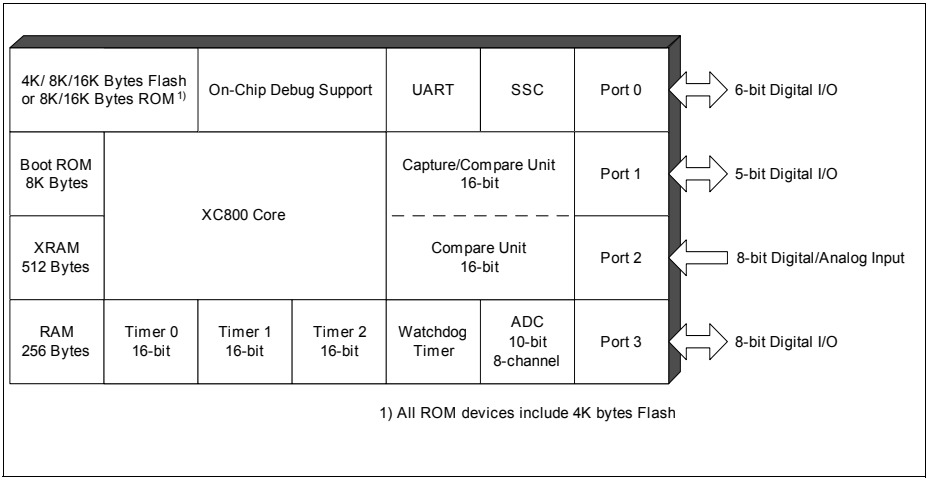


Figure 1 XC866 Functional Units

2.4 Pin Definitions and Functions

Table 3 Pin Definitions and Functions

Symbol	Pin Number	Type	Reset State	Function	
P0		I/O		**Port 0** Port 0 is a 6-bit bidirectional general purpose I/O port. It can be used as alternate functions for the JTAG, CCU6, UART, and the SSC.	
P0.0	12		Hi-Z	TCK_0 T12HR_1 CC61_1 CLKOUT RXDO_1	JTAG Clock Input CCU6 Timer 12 Hardware Run Input Input/Output of Capture/Compare channel 1 Clock Output UART Transmit Data Output
P0.1	14		Hi-Z	TDI_0 T13HR_1 RXD_1 COUT61_1 EXF2_1	JTAG Serial Data Input CCU6 Timer 13 Hardware Run Input UART Receive Data Input Output of Capture/Compare channel 1 Timer 2 External Flag Output
P0.2	13		PU	$\overline{\text{CTRAP}}$_2 TDO_0 TXD_1	CCU6 Trap Input JTAG Serial Data Output UART Transmit Data Output/ Clock Output
P0.3	2		Hi-Z	SCK_1 COUT63_1	SSC Clock Input/Output Output of Capture/Compare channel 3
P0.4	3		Hi-Z	MTSR_1 CC62_1	SSC Master Transmit Output/ Slave Receive Input Input/Output of Capture/Compare channel 2
P0.5	4		Hi-Z	MRST_1 EXINT0_0 COUT62_1	SSC Master Receive Input/ Slave Transmit Output External Interrupt Input 0 Output of Capture/Compare channel 2

3.2.2 Special Function Register

The Special Function Registers (SFRs) occupy direct internal data memory space in the range 80_H to FF_H. All registers, except the program counter, reside in the SFR area. The SFRs include pointers and registers that provide an interface between the CPU and the on-chip peripherals. As the 128-SFR range is less than the total number of registers required, address extension mechanisms are required to increase the number of addressable SFRs. The address extension mechanisms include:

- Mapping
- Paging

3.2.2.1 Address Extension by Mapping

Address extension is performed at the system level by mapping. The SFR area is extended into two portions: the standard (non-mapped) SFR area and the mapped SFR area. Each portion supports the same address range 80_H to FF_H, bringing the number of addressable SFRs to 256. The extended address range is not directly controlled by the CPU instruction itself, but is derived from bit RMAP in the system control register SYSCON0 at address $8F_H$. To access SFRs in the mapped area, bit RMAP in SFR SYSCON0 must be set. Alternatively, the SFRs in the standard area can be accessed by clearing bit RMAP. The SFR area can be selected as shown in **Figure 8**.

SYSCON0
System Control Register 0 **Reset Value: 00_H**

7	6	5	4	3	2	1	0
0					**1**	**0**	**RMAP**
r					rw	r	rw

Field	Bits	Type	Description
RMAP	0	rw	**Special Function Register Map Control** 0 The access to the standard SFR area is enabled. 1 The access to the mapped SFR area is enabled.
1	2	rw	**Reserved** Returns the last value if read; should be written with 1.
0	1,[7:3]	r	**Reserved** Returns 0 if read; should be written with 0.

Table 13 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	EN IDLE	EN WHE	EN CHE	0	EN TRPF	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					
FD_H	**CCU6_MODCTRH Reset: 00_H** Modulation Control Register High	Bit Field	ECT13O	0	T13MODEN					
		Type	rw	r	rw					
FE_H	**CCU6_TRPCTRL Reset: 00_H** Trap Control Register Low	Bit Field	0					TRPM2	TRPM1	TRPM0
		Type	r					rw	rw	rw

Table 13 CCU6 Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
FF_H	**CCU6_TRPCTRH** **Reset: 00_H** Trap Control Register High	Bit Field	TRPPEN	TRPEN13	TRPEN					
		Type	rw	rw	rw					
RMAP = 0, Page 3										
$9A_H$	**CCU6_MCMOUTL** **Reset: 00_H** Multi-Channel Mode Output Register Low	Bit Field	0	R	MCMP					
		Type	r	rh	rh					
$9B_H$	**CCU6_MCMOUTH** **Reset: 00_H** Multi-Channel Mode Output Register High	Bit Field	0		CURH			EXPH		
		Type	r		rh			rh		
$9C_H$	**CCU6_ISL** **Reset: 00_H** Capture/Compare Interrupt Status Register Low	Bit Field	T12PM	T12OM	ICC62F	ICC62R	ICC61F	ICC61R	ICC60F	ICC60R
		Type	rh	rh	rh	rh	rh	rh	rh	rh
$9D_H$	**CCU6_ISH** **Reset: 00_H** Capture/Compare Interrupt Status Register High	Bit Field	STR	IDLE	WHE	CHE	TRPS	TRPF	T13PM	T13CM
		Type	rh	rh	rh	rh	rh	rh	rh	rh
$9E_H$	**CCU6_PISEL0L** **Reset: 00_H** Port Input Select Register 0 Low	Bit Field	ISTRP		ISCC62		ISCC61		ISCC60	
		Type	rw		rw		rw		rw	
$9F_H$	**CCU6_PISEL0H** **Reset: 00_H** Port Input Select Register 0 High	Bit Field	IST12HR		ISPOS2		ISPOS1		ISPOS0	
		Type	rw		rw		rw		rw	
$A4_H$	**CCU6_PISEL2** **Reset: 00_H** Port Input Select Register 2	Bit Field	0						IST13HR	
		Type	r						rw	
FA_H	**CCU6_T12L** **Reset: 00_H** Timer T12 Counter Register Low	Bit Field	T12CVL							
		Type	rwh							
FB_H	**CCU6_T12H** **Reset: 00_H** Timer T12 Counter Register High	Bit Field	T12CVH							
		Type	rwh							
FC_H	**CCU6_T13L** **Reset: 00_H** Timer T13 Counter Register Low	Bit Field	T13CVL							
		Type	rwh							
FD_H	**CCU6_T13H** **Reset: 00_H** Timer T13 Counter Register High	Bit Field	T13CVH							
		Type	rwh							
FE_H	**CCU6_CMPSTATL** **Reset: 00_H** Compare State Register Low	Bit Field	0	CC63ST	CCPOS2	CCPOS1	CCPOS0	CC62ST	CC61ST	CC60ST
		Type	r	rh	rh	rh	rh	rh	rh	rh
FF_H	**CCU6_CMPSTATH** **Reset: 00_H** Compare State Register High	Bit Field	T13IM	COUT63PS	COUT62PS	CC62PS	COUT61PS	CC61PS	COUT60PS	CC60PS
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

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

Table 14 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			
	Operating Mode	Bit Field	0				BC			
		Type	r				rh			

3.4.2 Interrupt Source and Vector

Each interrupt source has an associated interrupt vector address. This vector is accessed to service the corresponding interrupt source request. The interrupt service of each interrupt source can be individually enabled or disabled via an enable bit. The assignment of the XC866 interrupt sources to the interrupt vector addresses and the corresponding interrupt source enable bits are summarized in **Table 17**.

Table 17 Interrupt Vector Addresses

Interrupt Source	Vector Address	Assignment for XC866	Enable Bit	SFR
NMI	0073_H	Watchdog Timer NMI	NMIWDT	NMICON
		PLL NMI	NMIPLL	
		Flash NMI	NMIFLASH	
		VDDC Prewarning NMI	NMIVDD	
		VDDP Prewarning NMI	NMIVDDP	
		Flash ECC NMI	NMIECC	
XINTR0	0003_H	External Interrupt 0	EX0	IEN0
XINTR1	$000B_H$	Timer 0	ET0	
XINTR2	0013_H	External Interrupt 1	EX1	
XINTR3	$001B_H$	Timer 1	ET1	
XINTR4	0023_H	UART	ES	
XINTR5	$002B_H$	T2	ET2	
		Fractional Divider (Normal Divider Overflow)		
		LIN		

Table 17 Interrupt Vector Addresses (cont'd)

XINTR6	0033_H	ADC	EADC	IEN1
XINTR7	$003B_H$	SSC	ESSC	
XINTR8	0043_H	External Interrupt 2	EX2	
XINTR9	$004B_H$	External Interrupt 3	EXM	
		External Interrupt 4		
		External Interrupt 5		
		External Interrupt 6		
XINTR10	0053_H	CCU6 INP0	ECCIP0	
XINTR11	$005B_H$	CCU6 INP1	ECCIP1	
XINTR12	0063_H	CCU6 INP2	ECCIP2	
XINTR13	$006B_H$	CCU6 INP3	ECCIP3	

3.6 Power Supply System with Embedded Voltage Regulator

The XC866 microcontroller requires two different levels of power supply:

- 3.3 V or 5.0 V for the Embedded Voltage Regulator (EVR) and Ports
- 2.5 V for the core, memory, on-chip oscillator, and peripherals

Figure 20 shows the XC866 power supply system. A power supply of 3.3 V or 5.0 V must be provided from the external power supply pin. The 2.5 V power supply for the logic is generated by the EVR. The EVR helps to reduce the power consumption of the whole chip and the complexity of the application board design.

The EVR consists of a main voltage regulator and a low power voltage regulator. In active mode, both voltage regulators are enabled. In power-down mode, the main voltage regulator is switched off, while the low power voltage regulator continues to function and provide power supply to the system with low power consumption.

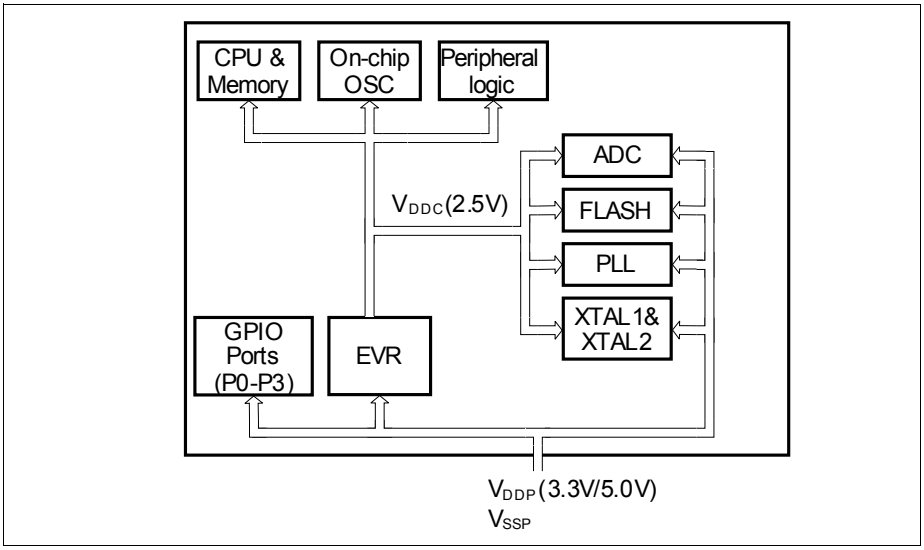


Figure 20 XC866 Power Supply System

EVR Features:

- Input voltage (V_{DDP}): 3.3 V/5.0 V
- Output voltage (V_{DDC}): 2.5 V ± 7.5%
- Low power voltage regulator provided in power-down mode
- V_{DDC} and V_{DDP} prewarning detection
- V_{DDC} brownout detection

The clock system provides three ways to generate the system clock:

PLL Base Mode

The system clock is derived from the VCO base (free running) frequency clock divided by the K factor.

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

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.

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

PLL Mode

The system clock is derived from the oscillator clock, multiplied by the N factor, and divided by the P and K factors. Both VCO bypass and PLL bypass must be inactive for this PLL mode. The PLL mode is used during normal system operation. .

$$f_{SYS} = f_{OSC} \times \frac{N}{P \times K}$$

Table 3-1 shows the settings of bits OSCDISC and VCOBYP for different clock mode selection.

Table 3-1 Clock Mode Selection

OSCDISC	VCOBYP	Clock Working Modes
0	0	PLL Mode
0	1	Prescaler Mode
1	0	PLL Base Mode
1	1	PLL Base Mode

Note: When oscillator clock is disconnected from PLL, the clock mode is PLL Base mode regardless of the setting of VCOBYP bit.

System Frequency Selection

For the XC866, the values of P and K are fixed to “1” and “2”, respectively. In order to obtain the required system frequency, f_{sys}, the value of N can be selected by bit NDIV for different oscillator inputs. **Table 21** provides examples on how f_{sys} = 80 MHz can be obtained for the different oscillator sources.

3.8.2 Clock Management

The CGU generates all clock signals required within the microcontroller from a single clock, f_{sys}. During normal system operation, the typical frequencies of the different modules are as follow:

- CPU clock: CCLK, SCLK = 26.7 MHz
- CCU6 clock: FCLK = 26.7 MHz
- Other peripherals: PCLK = 26.7 MHz
- Flash Interface clock: CCLK3 = 80 MHz and CCLK = 26.7 MHz

In addition, different clock frequency can output to pin CLKOUT(P0.0). The clock output frequency can further be divided by 2 using toggle latch (bit TLEN is set to 1), the resulting output frequency has 50% duty cycle. **Figure 25** shows the clock distribution of the XC866.

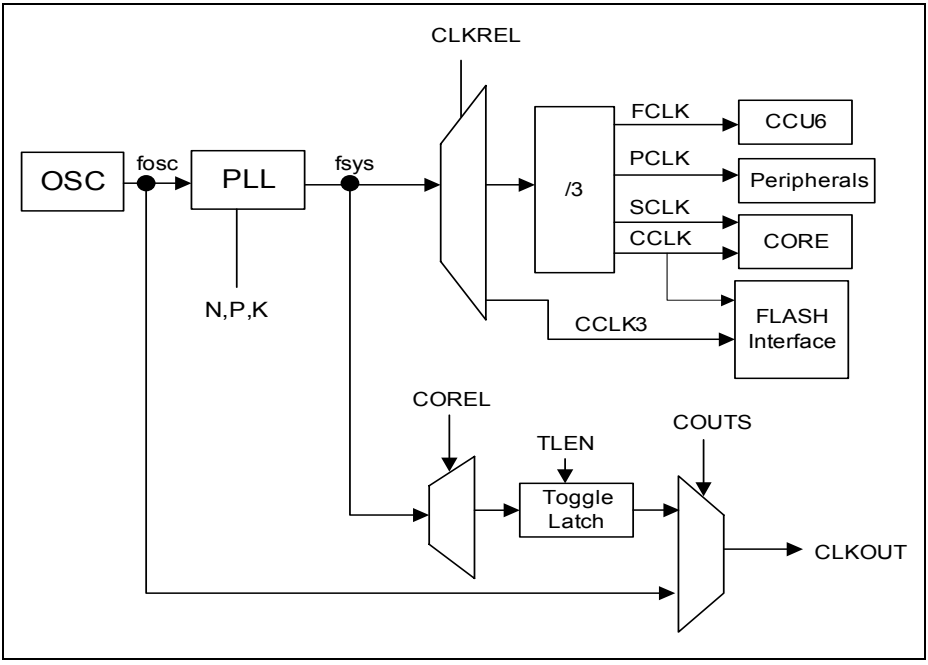


Figure 25 Clock Generation from f_{sys}

3.11.1 Baud-Rate Generator

The baud-rate generator is based on a programmable 8-bit reload value, and includes divider stages (i.e., prescaler and fractional divider) for generating a wide range of baud rates based on its input clock f_{PCLK}, see **Figure 29**.

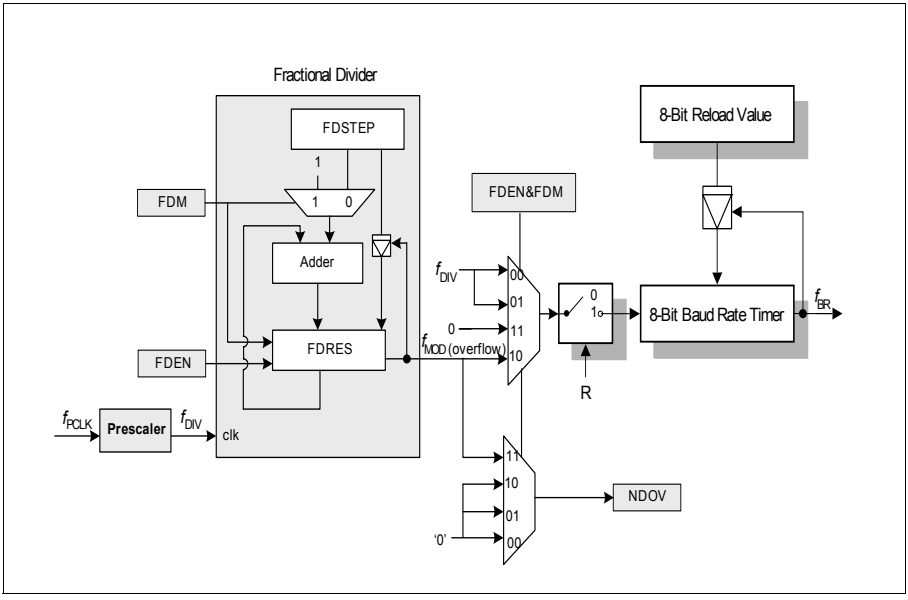


Figure 29 Baud-rate Generator Circuitry

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 3.12**.

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:

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

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

The maximum baud rate that can be generated is limited to $f_{PCLK}/32$. Hence, for a module clock of 26.7 MHz, the maximum achievable baud rate is 0.83 MBaud.

Standard LIN protocal can support a maximum baud rate of 20kHz, 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 20kHz to 115.2kHz, the higher baud rates require the use of the fractional divider for greater accuracy.

Table 26 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 26.7 MHz is used.

Table 26 Typical Baud rates for UART with Fractional Divider disabled

Baud rate	Prescaling Factor (2^{BRPRE})	Reload Value (BR_VALUE + 1)	Deviation Error
19.2 kBaud	1 (BRPRE=000_B)	87 (57_H)	-0.22 %
9600 Baud	1 (BRPRE=000_B)	174 (AE_H)	-0.22 %
4800 Baud	2 (BRPRE=001_B)	174 (AE_H)	-0.22 %
2400 Baud	4 (BRPRE=010_B)	174 (AE_H)	-0.22 %

The fractional divider allows baud rates of higher accuracy (lower deviation error) to be generated. **Table 27** lists the resulting deviation errors from generating a baud rate of

3.11.2 Baud Rate Generation using Timer 1

In UART modes 1 and 3, Timer 1 can be used for generating the variable baud rates. In theory, this timer could be used in any of its modes. But in practice, it should be set into auto-reload mode (Timer 1 mode 2), with its high byte set to the appropriate value for the required baud rate. The baud rate is determined by the Timer 1 overflow rate and the value of SMOD as follows:

[3.1]

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \times f_{PCLK}}{32 \times 2 \times (256 - TH1)}$$

3.12 Normal Divider Mode (8-bit Auto-reload Timer)

Setting bit FDM in register FDCON to 1 configures the fractional divider to normal divider mode, while at the same time disables baud rate generation (see **Figure 29**). Once the fractional divider is enabled (FDEN = 1), it functions as an 8-bit auto-reload timer (with no relation to baud rate generation) and counts up from the reload value with each input clock pulse. Bit field RESULT in register FDRES represents the timer value, while bit field STEP in register FDSTEP defines the reload value. At each timer overflow, an overflow flag (FDCON.NDOV) will be set and an interrupt request generated. This gives an output clock f_{MOD} that is 1/n of the input clock f_{DIV}, where n is defined by 256 - STEP.

The output frequency in normal divider mode is derived as follows:

[3.2]

$$f_{MOD} = f_{DIV} \times \frac{1}{256 - STEP}$$

3.16 Timer 2

Timer 2 is a 16-bit general purpose timer (THL2) that has two modes of operation, a 16-bit auto-reload mode and a 16-bit one channel capture mode. If the prescalar is disabled, Timer 2 counts with an input clock of PCLK/12. Timer 2 continues counting as long as it is enabled.

Table 29 Timer 2 Modes

Mode	Description
Auto-reload	**Up/Down Count Disabled** • Count up only • Start counting from 16-bit reload value, overflow at $FFFF_H$ • Reload event configurable for trigger by overflow condition only, or by negative/positive edge at input pin T2EX as well • Programmble reload value in register RC2 • Interrupt is generated with reload event
	Up/Down Count Enabled • Count up or down, direction determined by level at input pin T2EX • No interrupt is generated • Count up – Start counting from 16-bit reload value, overflow at $FFFF_H$ – Reload event triggered by overflow condition – Programmble reload value in register RC2 • Count down – Start counting from $FFFF_H$, underflow at value defined in register RC2 – Reload event triggered by underflow condition – Reload value fixed at $FFFF_H$
Channel capture	• Count up only • Start counting from 0000_H, overflow at $FFFF_H$ • Reload event triggered by overflow condition • Reload value fixed at 0000_H • Capture event triggered by falling/rising edge at pin T2EX • Captured timer value stored in register RC2 • Interrupt is generated with reload or capture event

3.18.1 ADC Clocking Scheme

A common module clock f_{ADC} generates the various clock signals used by the analog and digital parts of the ADC module:

- f_{ADCA} is input clock for the analog part.
- f_{ADCI} is internal clock for the analog part (defines the time base for conversion length and the sample time). This clock is generated internally in the analog part, based on the input clock f_{ADCA} to generate a correct duty cycle for the analog components.
- f_{ADCD} is input clock for the digital part.

The internal clock for the analog part f_{ADCI} is limited to a maximum frequency of 10 MHz. Therefore, the ADC clock prescaler must be programmed to a value that ensures f_{ADCI} does not exceed 10 MHz. The prescaler ratio is selected by bit field CTC in register GLOBCTR. A prescaling ratio of 32 can be selected when the maximum performance of the ADC is not required.

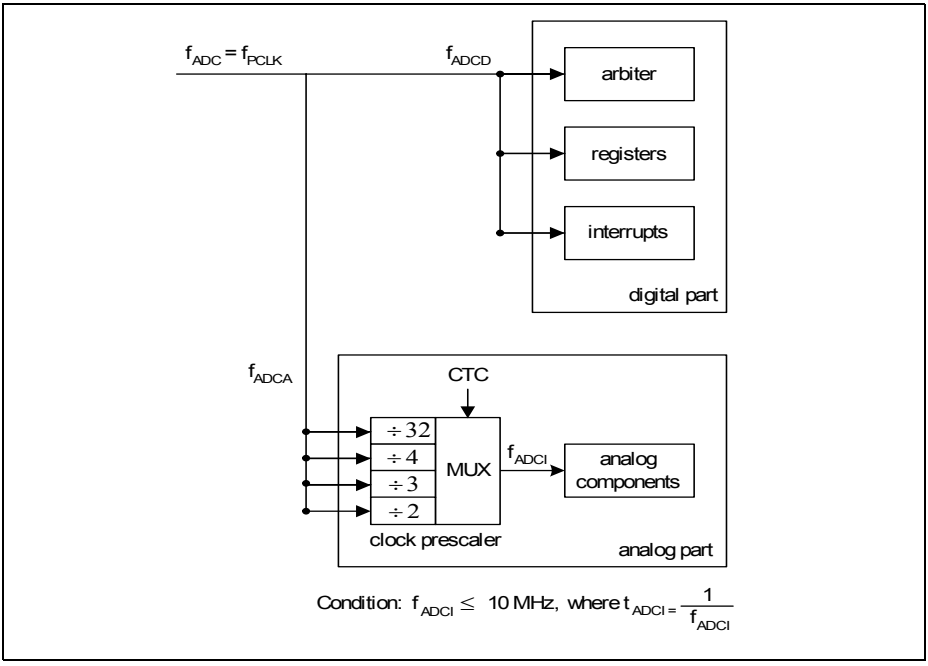


Figure 33 ADC Clocking Scheme

3.19 On-Chip Debug Support

The On-Chip Debug Support (OCDS) provides the basic functionality required for the software development and debugging of XC800-based systems.

The OCDS design is based on these principles:

- use the built-in debug functionality of the XC800 Core
- add a minimum of hardware overhead
- provide support for most of the operations by a Monitor Program
- use standard interfaces to communicate with the Host (a Debugger)

Features:

- Set breakpoints on instruction address and within a specified address range
- Set breakpoints on internal RAM address
- Support unlimited software breakpoints in Flash/RAM code region
- Process external breaks
- Step through the program code

The OCDS functional blocks are shown in **Figure 35**. The Monitor Mode Control (MMC) block at the center of OCDS system brings together control signals and supports the overall functionality. The MMC communicates with the XC800 Core, primarily via the Debug Interface, and also receives reset and clock signals. After processing memory address and control signals from the core, the MMC provides proper access to the dedicated extra-memories: a Monitor ROM (holding the code) and a Monitor RAM (for work-data and Monitor-stack). The OCDS system is accessed through the JTAG[1)], which is an interface dedicated exclusively for testing and debugging activities and is not normally used in an application. The dedicated MBC pin is used for external configuration and debugging control.

Note: All the debug functionality described here can normally be used only after XC866 has been started in OCDS mode.

1) The pins of the JTAG port can be assigned to either Port 0 (primary) or Ports 1 and 2 (secondary). User must set the JTAG pins (TCK and TDI) as input during connection with the OCDS system.

Table 34 Input/Output Characteristics (Operating Conditions apply)

Parameter	Symbol		Limit Values		Unit	Test Conditions Remarks
			min.	max.		
Pull-up current	I_{PU}	SR	–	-5	µA	$V_{IH,min}$
			-50	–	µA	$V_{IL,max}$
Pull-down current	I_{PD}	SR	–	5	µA	$V_{IL,max}$
			50	–	µA	$V_{IH,min}$
Input leakage current[2]	I_{OZ1}	CC	-1	1	µA	$0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$, XC866-4FR and XC866-2FR
			-2.5	1	µA	$0 < V_{IN} < V_{DDP}$, $T_A \leq 125°C$, XC866-1FR and ROM device
Input current at XTAL1	I_{ILX}	CC	- 10	10	µA	
Overload current on any pin	I_{OV}	SR	-5	5	mA	
Absolute sum of overload currents	$\Sigma\|I_{OV}\|$	SR	–	25	mA	[3]
Voltage on any pin during V_{DDP} power off	V_{PO}	SR	–	0.3	V	[4]
Maximum current per pin (excluding V_{DDP} and V_{SS})	I_M	SR	–	15	mA	
Maximum current for all pins (excluding V_{DDP} and V_{SS})	$\Sigma\|I_M\|$	SR	–	60	mA	
Maximum current into V_{DDP}	I_{MVDDP}	SR	–	80	mA	
Maximum current out of V_{SS}	I_{MVSS}	SR	–	80	mA	

1) Not subjected to production test, verified by design/characterization. Hysteresis is implemented to avoid meta stable states and switching due to internal ground bounce. It cannot be guaranteed that it suppresses switching due to external system noise.

2) An additional error current (I_{INJ}) will flow if an overload current flows through an adjacent pin. TMS pin and $\overline{\text{RESET}}$ pin have internal pull devices and are not included in the input leakage current characteristic.

3) Not subjected to production test, verified by design/characterization.

4.3.5 JTAG Timing

Table 44 TCK Clock Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limits		Unit
			min	max	
TCK clock period	t_{TCK}	SR	50	–	ns
TCK high time	t_1	SR	20	–	ns
TCK low time	t_2	SR	20	–	ns
TCK clock rise time	t_3	SR	–	4	ns
TCK clock fall time	t_4	SR	–	4	ns

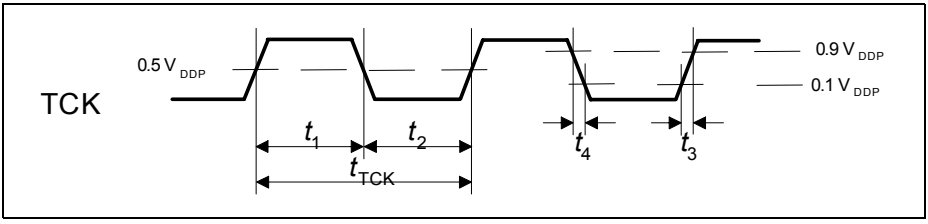


Figure 43 TCK Clock Timing

4.3.6 SSC Master Mode Timing

Table 46 SSC Master Mode Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limit Values		Unit
			min.	max.	
SCLK clock period	t_0	CC	2*T_{SSC} 1)	–	ns
MTSR delay from SCLK ↓	t_1	CC	0	8	ns
MRST setup to SCLK ↑	t_2	SR	22	–	ns
MRST hold from SCLK ↑	t_3	SR	0	–	ns

1) T_{SSCmin} = T_{CPU} = 1/f_{CPU}. When f_{CPU} = 26.7MHz, t_0 = 74.9ns. T_{CPU} is the CPU clock period.

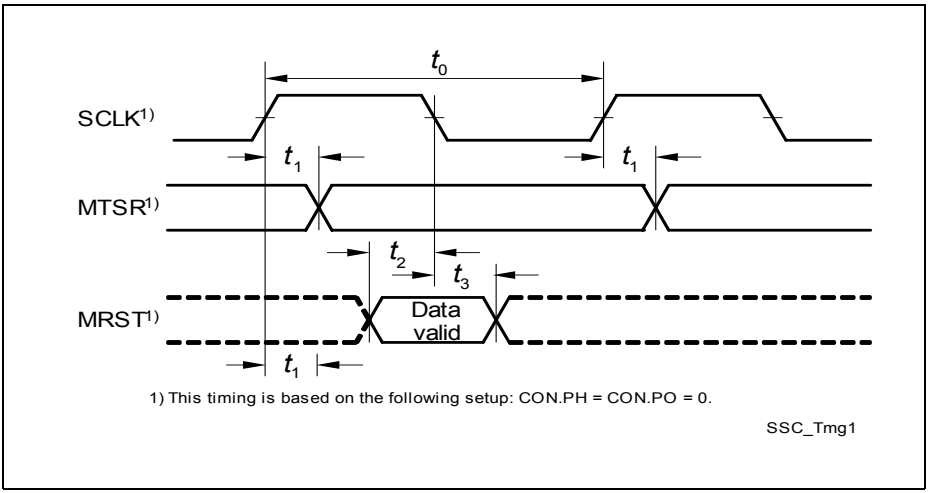
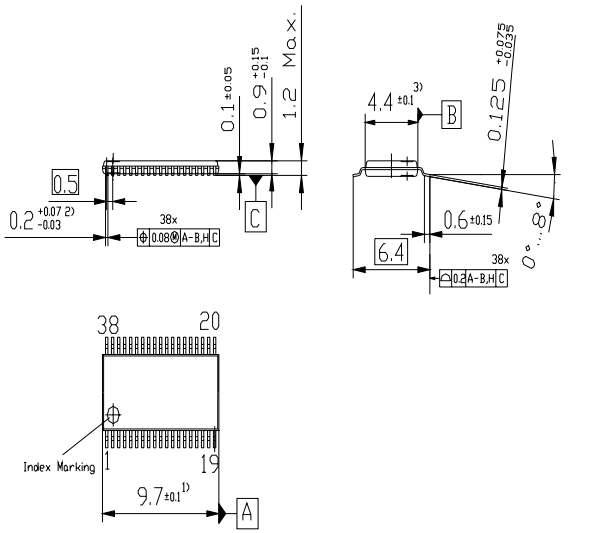


Figure 45 SSC Master Mode Timing

5.2 Package Outline



1) DOES NOT INCLUDE PLASTIC OR METAL PROTRUSION OF 0.15 MAX. PER SIDE

2) DOES NOT INCLUDE DAMBAR PROTRUSION OF 0.08 MAX. PER SIDE

3) DOES NOT INCLUDE PLASTIC OR METAL PROTRUSION OF 0.25 MAX. PER SIDE

Figure 46 PG-TSSOP-38-4 Package Outline