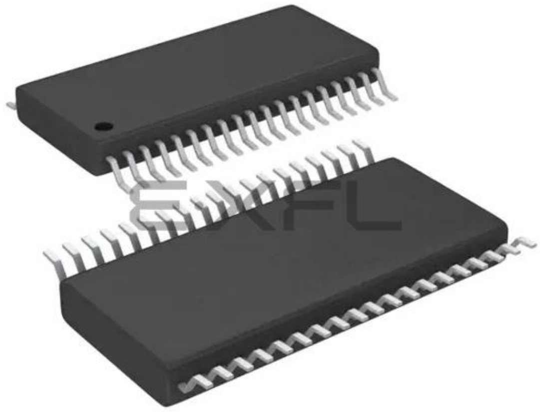
Infineon Technologies - SAA-XC866L-4FRA 5V BE Datasheet



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	Obsolete
Core Processor	XC800
Core Size	8-Bit
Speed	86MHz
Connectivity	LINbus, SSI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	27
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 140°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/saa-xc866l-4fra-5v-be

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)

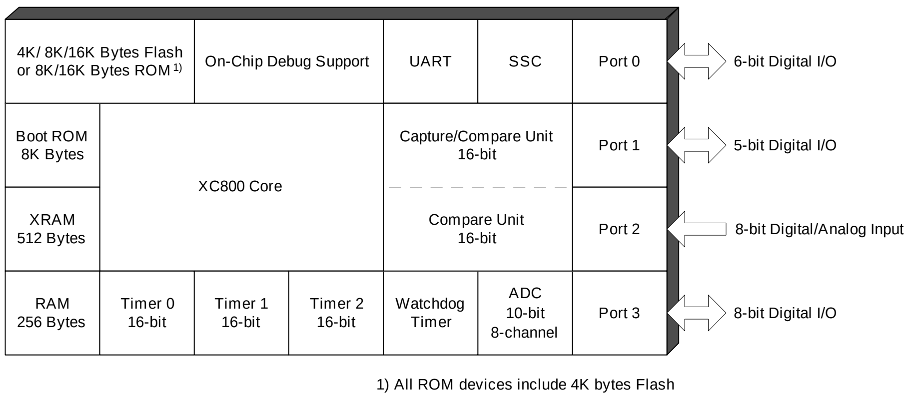


Figure 1 SAA-XC866 Functional Units

2 General Device Information

2.1 Block Diagram

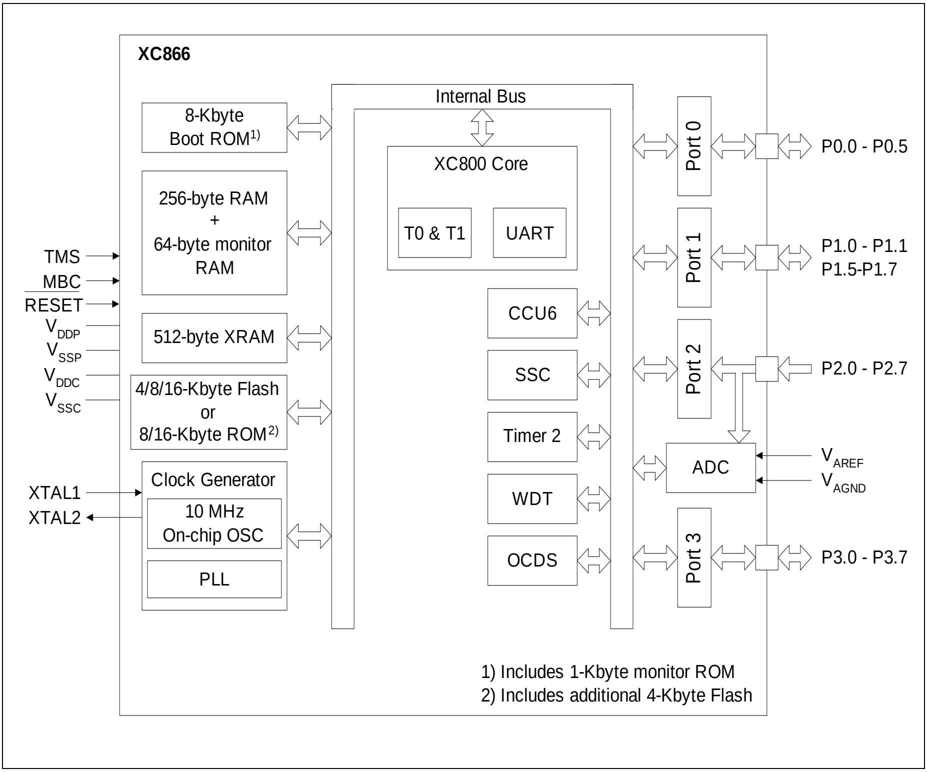


Figure 2 SAA-XC866 Block Diagram

Table 2 Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Type	Reset State	Function	
P2		I		**Port 2** Port 2 is an 8-bit general purpose input-only port. It can be used as alternate functions for the digital inputs of the JTAG and CCU6. It is also used as the analog inputs for the ADC.	
P2.0	15		Hi-Z	CCPOS0_0 EXINT1 T12HR_2 TCK_1 CC61_3 AN0	CCU6 Hall Input 0 External Interrupt Input 1 CCU6 Timer 12 Hardware Run Input JTAG Clock Input Input of Capture/Compare channel 1 Analog Input 0
P2.1	16		Hi-Z	CCPOS1_0 EXINT2 T13HR_2 TDI_1 CC62_3 AN1	CCU6 Hall Input 1 External Interrupt Input 2 CCU6 Timer 13 Hardware Run Input JTAG Serial Data Input Input of Capture/Compare channel 2 Analog Input 1
P2.2	17		Hi-Z	CCPOS2_0 $\overline{\text{CTRAP}}$_1 CC60_3 AN2	CCU6 Hall Input 2 CCU6 Trap Input Input of Capture/Compare channel 0 Analog Input 2
P2.3	20		Hi-Z	AN3	Analog Input 3
P2.4	21		Hi-Z	AN4	Analog Input 4
P2.5	22		Hi-Z	AN5	Analog Input 5
P2.6	23		Hi-Z	AN6	Analog Input 6
P2.7	26		Hi-Z	AN7	Analog Input 7

3 Functional Description

3.1 Processor Architecture

The SAA-XC866 is based on a high-performance 8-bit Central Processing Unit (CPU) that is compatible with the standard 8051 processor. While the standard 8051 processor is designed around a 12-clock machine cycle, the SAA-XC866 CPU uses a 2-clock machine cycle. This allows fast access to ROM or RAM memories without wait state. Access to the Flash memory, however, requires an additional wait state (one machine cycle). The instruction set consists of 45% one-byte, 41% two-byte and 14% three-byte instructions.

The SAA-XC866 CPU provides a range of debugging features, including basic stop/start, single-step execution, breakpoint support and read/write access to the data memory, program memory and SFRs. **Figure 5** shows the CPU functional blocks.

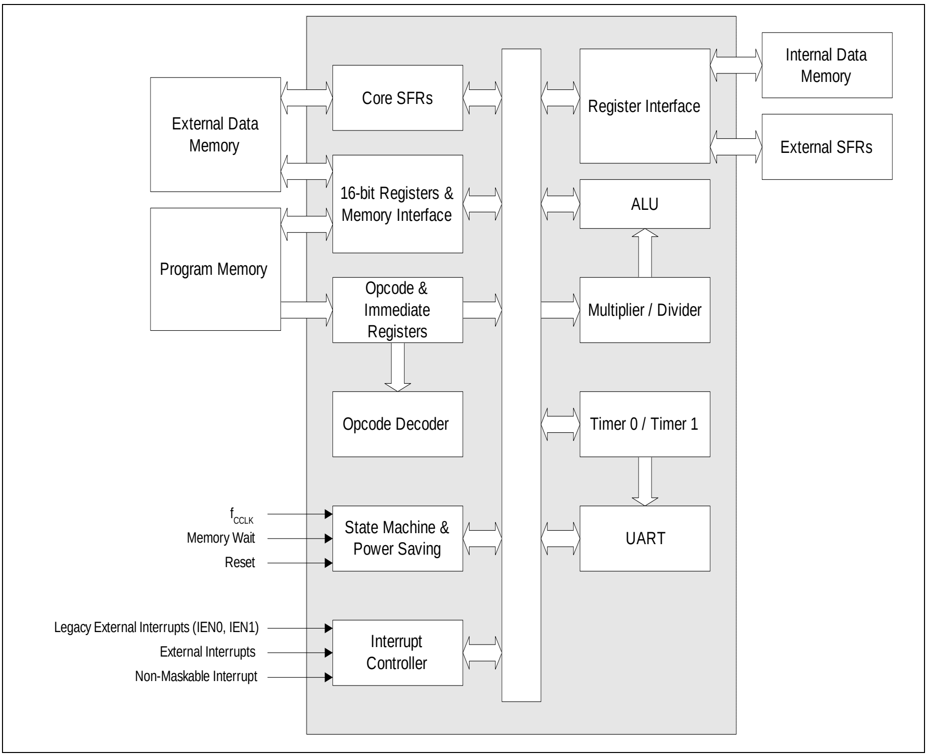


Figure 5 CPU Block Diagram

Figure 7 illustrates the memory address spaces of the SAA-XC866-4RR device.

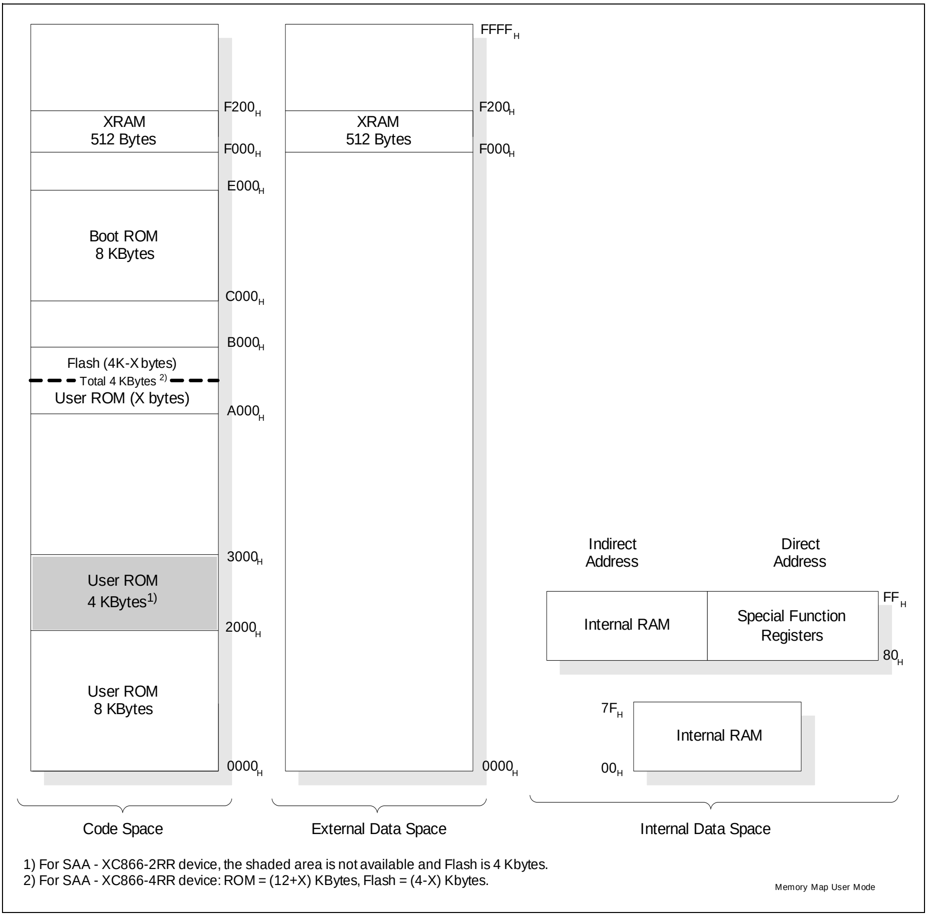


Figure 7 Memory Map of SAA-XC866 ROM Devices

Note: The RMAP bit must be cleared/set by ANL or ORL instructions. The rest bits of SYSCON0 should not be modified.

As long as bit RMAP is set, the mapped SFR area can be accessed. This bit is not cleared automatically by hardware. Thus, before standard/mapped registers are accessed, bit RMAP must be cleared/set, respectively, by software.

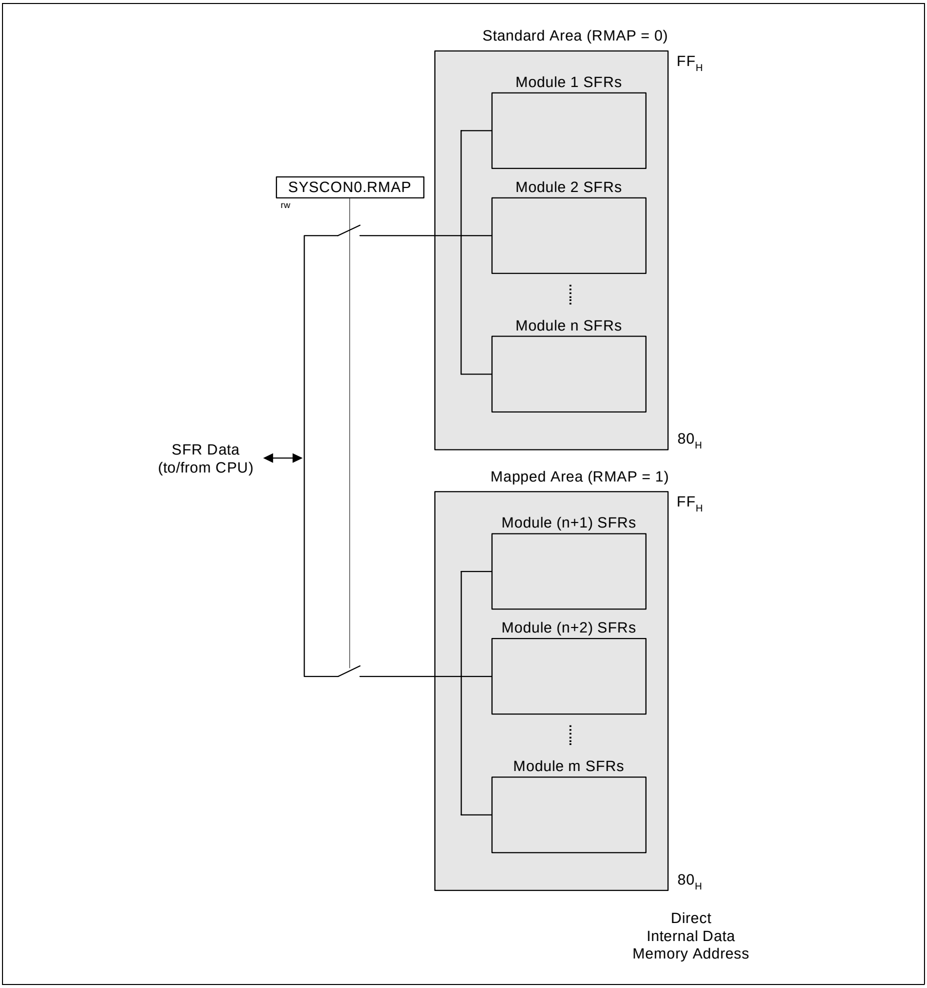


Figure 8 Address Extension by Mapping

Table 6 CPU Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$F0_H$	**B** **Reset: 00_H** B Register	Bit Field	B7	B6	B5	B4	B3	B2	B1	B0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$F8_H$	**IP1** **Reset: 00_H** Interrupt Priority Register 1	Bit Field	PCCIP 3	PCCIP 2	PCCIP 1	PCCIP 0	PXM	PX2	PSSC	PADC
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$F9_H$	**IPH1** **Reset: 00_H** Interrupt Priority Register 1 High	Bit Field	PCCIP 3H	PCCIP 2H	PCCIP 1H	PCCIP 0H	PXMH	PX2H	PSSCH	PADC H
		Type	rw	rw	rw	rw	rw	rw	rw	rw

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

Table 7 System Control Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0 or 1										
$8F_H$	**SYSCON0** **Reset: 00_H** System Control Register 0	Bit Field	0							RMAP
		Type	r							rw
RMAP = 0										
BF_H	**SCU_PAGE** **Reset: 00_H** Page Register for System Control	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		JTAG TDIS	JTAG TCKS	0		EXINT 0IS	URRIS
		Type	r		rw	rw	r		rw	rw
$B4_H$	**IRCON0** **Reset: 00_H** Interrupt Request Register 0	Bit Field	0	EXINT 6	EXINT 5	EXINT 4	EXINT 3	EXINT 2	EXINT 1	EXINT 0
		Type	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$B5_H$	**IRCON1** **Reset: 00_H** Interrupt Request Register 1	Bit Field	0			ADCS RC1	ADCS RC0	RIR	TIR	EIR
		Type	r			rwh	rwh	rwh	rwh	rwh
$B7_H$	**EXICON0** **Reset: 00_H** External Interrupt Control Register 0	Bit Field	EXINT3		EXINT2		EXINT1		EXINT0	
		Type	rw		rw		rw		rw	
BA_H	**EXICON1** **Reset: 00_H** External Interrupt Control Register 1	Bit Field	0		EXINT6		EXINT5		EXINT4	
		Type	r		rw		rw		rw	
BB_H	**NMICON** **Reset: 00_H** NMI Control Register	Bit Field	0	NMI ECC	NMI VDDP	NMI VDD	NMI OCDS	NMI FLASH	NMI PLL	NMI WDT
		Type	r	rw	rw	rw	rw	rw	rw	rw
BC_H	**NMISR** **Reset: 00_H** NMI Status Register	Bit Field	0	FNMI ECC	FNMI VDDP	FNMI VDD	FNMI OCDS	FNMI FLASH	FNMI PLL	FNMI WDT
		Type	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh
BD_H	**BCON** **Reset: 00_H** Baud Rate Control Register	Bit Field	BGSEL		0	BREN	BRPRE			R
		Type	rw		r	rw	rw			rw
BE_H	**BG** **Reset: 00_H** Baud Rate Timer/Reload Register	Bit Field	BR_VALUE							
		Type	rw							
$E9_H$	**FDCON** **Reset: 00_H** Fractional Divider Control Register	Bit Field	BGS	SYNEN	ERRSY N	EOFSY N	BRK	NDOV	FDM	FDEN
		Type	rw	rw	rwh	rwh	rwh	rwh	rw	rw
EA_H	**FDSTEP** **Reset: 00_H** Fractional Divider Reload Register	Bit Field	STEP							
		Type	rw							
EB_H	**FDRES** **Reset: 00_H** Fractional Divider Result Register	Bit Field	RESULT							
		Type	rh							
RMAP = 0, Page 1										

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

Table 9 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 for PORT	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	0		P5	P4	P3	P2	P1	P0
		Type	r		rwh	rwh	rwh	rwh	rwh	rwh
86_H	**P0_DIR** **Reset:** 00_H P0 Direction Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_DATA** **Reset:** 00_H P1 Data Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rwh	rwh	rwh	r			rwh	rwh
91_H	**P1_DIR** **Reset:** 00_H P1 Direction Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rw	rw	rw	r			rw	rw
$A0_H$	**P2_DATA** **Reset:** 00_H P2 Data Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$A1_H$	**P2_DIR** **Reset:** 00_H P2 Direction Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$B0_H$	**P3_DATA** **Reset:** 00_H P3 Data Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh
$B1_H$	**P3_DIR** **Reset:** 00_H P3 Direction Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
RMAP = 0, Page 1										
80_H	**P0_PUDSEL** **Reset:** FF_H P0 Pull-Up/Pull-Down Select Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
86_H	**P0_PUDEN** **Reset:** $C4_H$ P0 Pull-Up/Pull-Down Enable Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_PUDSEL** **Reset:** FF_H P1 Pull-Up/Pull-Down Select Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rw	rw	rw	r			rw	rw
91_H	**P1_PUDEN** **Reset:** FF_H P1 Pull-Up/Pull-Down Enable Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rw	rw	rw	r			rw	rw
$A0_H$	**P2_PUDSEL** **Reset:** FF_H P2 Pull-Up/Pull-Down Select Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$A1_H$	**P2_PUDEN** **Reset:** 00_H P2 Pull-Up/Pull-Down Enable Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$B0_H$	**P3_PUDSEL** **Reset:** BF_H P3 Pull-Up/Pull-Down Select Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$B1_H$	**P3_PUDEN** **Reset:** 40_H P3 Pull-Up/Pull-Down Enable Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
RMAP = 0, Page 2										
80_H	**P0_ALTSEL0** **Reset:** 00_H P0 Alternate Select 0 Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
86_H	**P0_ALTSEL1** **Reset:** 00_H P0 Alternate Select 1 Register	Bit Field	0		P5	P4	P3	P2	P1	P0
		Type	r		rw	rw	rw	rw	rw	rw
90_H	**P1_ALTSEL0** **Reset:** 00_H P1 Alternate Select 0 Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rw	rw	rw	r			rw	rw
91_H	**P1_ALTSEL1** **Reset:** 00_H P1 Alternate Select 1 Register	Bit Field	P7	P6	P5	0			P1	P0
		Type	rw	rw	rw	r			rw	rw
$B0_H$	**P3_ALTSEL0** **Reset:** 00_H P3 Alternate Select 0 Register	Bit Field	P7	P6	P5	P4	P3	P2	P1	P0
		Type	rw	rw	rw	rw	rw	rw	rw	rw

Table 10 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
CA_H	**ADC_CHINFR** **Reset: 00_H** Channel Interrupt Flag Register	Bit Field	CHINF 7	CHINF 6	CHINF 5	CHINF 4	CHINF 3	CHINF 2	CHINF 1	CHINF 0
		Type	rh	rh	rh	rh	rh	rh	rh	rh
CB_H	**ADC_CHINCR** **Reset: 00_H** Channel Interrupt Clear Register	Bit Field	CHINC 7	CHINC 6	CHINC 5	CHINC 4	CHINC 3	CHINC 2	CHINC 1	CHINC 0
		Type	w	w	w	w	w	w	w	w
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	CLR PND	SCAN	ENSI	ENTR	ENGT	
		Type	r	w	w	rw	rw	rw	rw	
CD_H	**ADC_QMR0** **Reset: 00_H** Queue Mode Register 0	Bit Field	CEV	TREV	FLUSH	CLRV	TRMD	ENTR	ENGT	
		Type	w	w	w	w	rw	rw	rw	
CE_H	**ADC_QSR0** **Reset: 20_H** Queue Status Register 0	Bit Field	Rsv	0	EMPTY	EV	0			
		Type	r	r	rh	rh	r			
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		

The Timer 2 SFRs can be accessed in the standard memory area (RMAP = 0).

Table 11 Timer 2 Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
$C0_H$	**T2_T2CON** **Reset: 00_H** Timer 2 Control Register	Bit Field	TF2	EXF2	0		EXEN2	TR2	0	$\overline{CP}$/ RL2
		Type	rwh	rwh	r		rw	rwh	r	rw

Table 11 Timer 2 Register Overview (cont'd)

$C1_H$	**T2_T2MOD** **Reset: 00_H** Timer 2 Mode Register	Bit Field	T2 REGS	T2 RHEN	EDGE SEL	PREN	T2PRE			DCEN
		Type	rw	rw	rw	rw	rw			rw
$C2_H$	**T2_RC2L** **Reset: 00_H** Timer 2 Reload/Capture Register Low	Bit Field	RC2[7:0]							
		Type	rwh							
$C3_H$	**T2_RC2H** **Reset: 00_H** Timer 2 Reload/Capture Register High	Bit Field	RC2[15:8]							
		Type	rwh							
$C4_H$	**T2_T2L** **Reset: 00_H** Timer 2 Register Low	Bit Field	THL2[7:0]							
		Type	rwh							
$C5_H$	**T2_T2H** **Reset: 00_H** Timer 2 Register High	Bit Field	THL2[15:8]							
		Type	rwh							

The CCU6 SFRs can be accessed in the standard memory area (RMAP = 0).

Table 12 CCU6 Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0										
$A3_H$	**CCU6_PAGE** **Reset: 00_H** Page Register for CCU6	Bit Field	OP		STNR		0	PAGE		
		Type	w		w		r	rwh		
RMAP = 0**,** Page 0										
$9A_H$	**CCU6_CC63SRL** **Reset: 00_H** Capture/Compare Shadow Register for Channel CC63 Low	Bit Field	CC63SL							
		Type	rw							
$9B_H$	**CCU6_CC63SRH** **Reset: 00_H** Capture/Compare Shadow Register for Channel CC63 High	Bit Field	CC63SH							
		Type	rw							
$9C_H$	**CCU6_TCTR4L** **Reset: 00_H** Timer Control Register 4 Low	Bit Field	T12 STD	T12 STR	0		DTRES	T12 RES	T12RS	T12RR
		Type	w	w	r		w	w	w	w
$9D_H$	**CCU6_TCTR4H** **Reset: 00_H** Timer Control Register 4 High	Bit Field	T13 STD	T13 STR	0			T13 RES	T13RS	T13RR
		Type	w	w	r			w	w	w
$9E_H$	**CCU6_MCMOUTSL** **Reset: 00_H** Multi-Channel Mode Output Shadow Register Low	Bit Field	STRM CM	0	MCMPS					
		Type	w	r	rw					
$9F_H$	**CCU6_MCMOUTSH** **Reset: 00_H** Multi-Channel Mode Output Shadow Register High	Bit Field	STRHP	0	CURHS			EXPHS		
		Type	w	r	rw			rw		
$A4_H$	**CCU6_ISRL** **Reset: 00_H** Capture/Compare Interrupt Status Reset Register Low	Bit Field	RT12P M	RT12O M	RCC62 F	RCC62 R	RCC61 F	RCC61 R	RCC60 F	RCC60 R
		Type	w	w	w	w	w	w	w	w
$A5_H$	**CCU6_ISRH** **Reset: 00_H** Capture/Compare Interrupt Status Reset Register High	Bit Field	RSTR	RIDLE	RWHE	RCHE	0	RTRPF	RT13 PM	RT13 CM
		Type	w	w	w	w	r	w	w	w
$A6_H$	**CCU6_CMPMODIFL** **Reset: 00_H** Compare State Modification Register Low	Bit Field	0	MCC63 S	0			MCC62 S	MCC61 S	MCC60 S
		Type	r	w	r			w	w	w
$A7_H$	**CCU6_CMPMODIFH** **Reset: 00_H** Compare State Modification Register High	Bit Field	0	MCC63 R	0			MCC62 R	MCC61 R	MCC60 R
		Type	r	w	r			w	w	w
FA_H	**CCU6_CC60SRL** **Reset: 00_H** Capture/Compare Shadow Register for Channel CC60 Low	Bit Field	CC60SL							
		Type	rwh							

3.3 Flash Memory

The Flash memory provides an embedded user-programmable non-volatile memory, allowing fast and reliable storage of user code and data. It is operated from a single 2.5 V supply from the Embedded Voltage Regulator (EVR) and does not require additional programming or erasing voltage. The sectorization of the Flash memory allows each sector to be erased independently.

Features

- In-System Programming (ISP) via UART
- In-Application Programming (IAP)
- Error Correction Code (ECC) for dynamic correction of single-bit errors
- Background program and erase operations for CPU load minimization
- Support for aborting erase operation
- Minimum program width[1] of 32-byte for D-Flash and 32-byte for P-Flash
- 1-sector minimum erase width
- 1-byte read access
- Flash is delivered in erased state (read all zeros)
- Operating supply voltage: 2.5 V ± 7.5 %
- Read access time: 3 × t_{CCLK} = 112.5 ns[2]
- Program time: 209440 / f_{SYS} = 2.6 ms[3]
- Erase time: 8175360 / f_{SYS} = 102 ms[3]

1) P-Flash: 32-byte wordline can only be programmed once, i.e., one gate disturb allowed.
D-Flash: 32-byte wordline can be programmed twice, i.e., two gate disturbs allowed.

2) f_{sys} = 80 MHz ± 7.5% (f_{CCLK} = 26.7 MHz ± 7.5 %) is the maximum frequency range for Flash read access.

3) f_{sys} = 80 MHz ± 7.5% is the only frequency range for Flash programming and erasing. f_{sysmin} is used for obtaining the worst case timing.

3.4 Interrupt System

The XC800 Core supports one non-maskable interrupt (NMI) and 14 maskable interrupt requests. In addition to the standard interrupt functions supported by the core, e.g., configurable interrupt priority and interrupt masking, the XC866 interrupt system provides extended interrupt support capabilities such as the mapping of each interrupt vector to several interrupt sources to increase the number of interrupt sources supported, and additional status registers for detecting and determining the interrupt source.

3.4.1 Interrupt Source

Figure 13 to **Figure 17** give a general overview of the interrupt sources and illustrates the request and control flags.

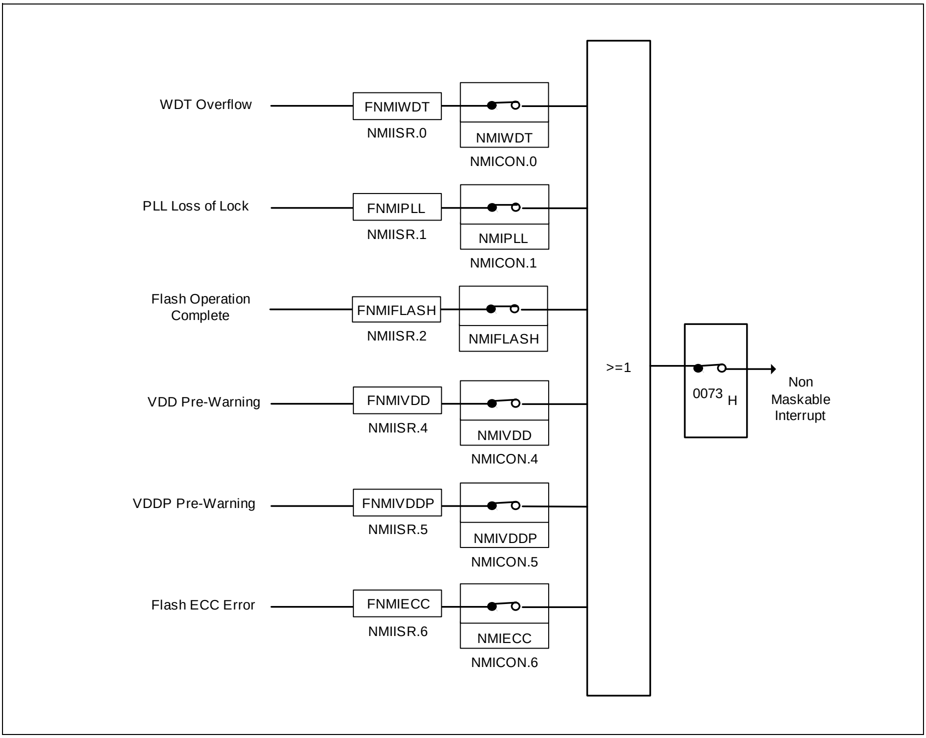


Figure 13 Non-Maskable Interrupt Request Sources

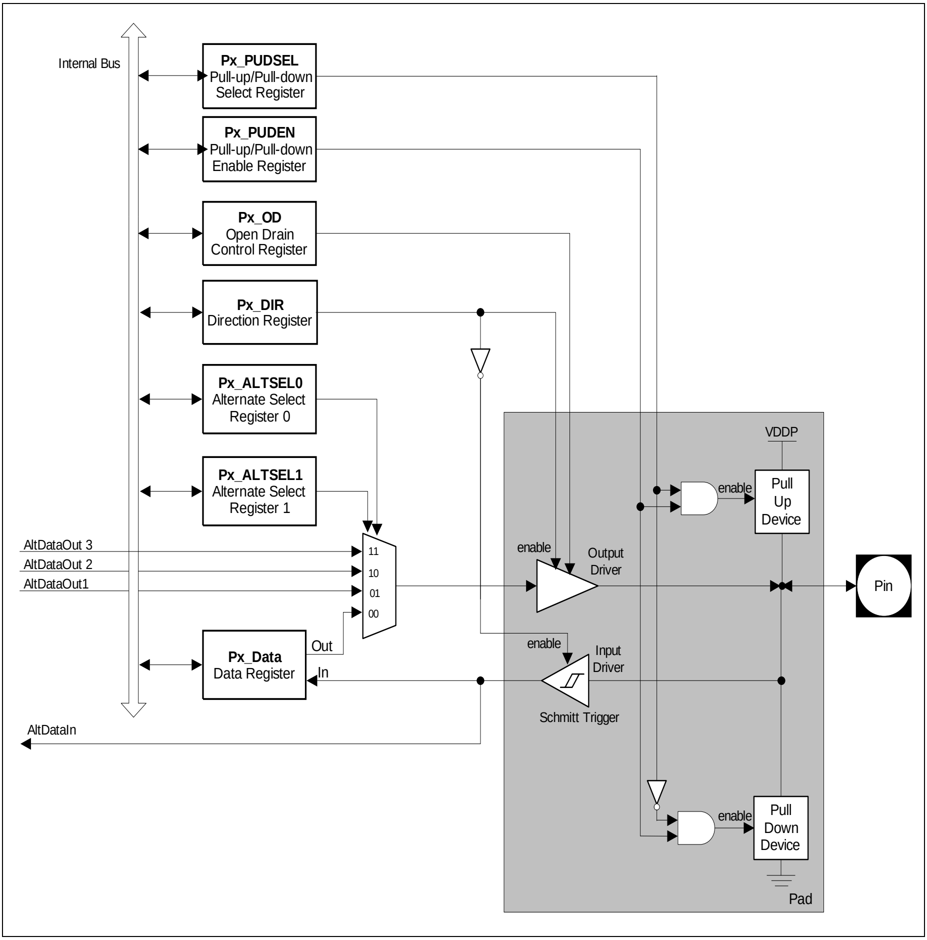


Figure 18 General Structure of Bidirectional Port

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 20 shows the settings of bits OSCDISC and VCOBYP for different clock mode selection.

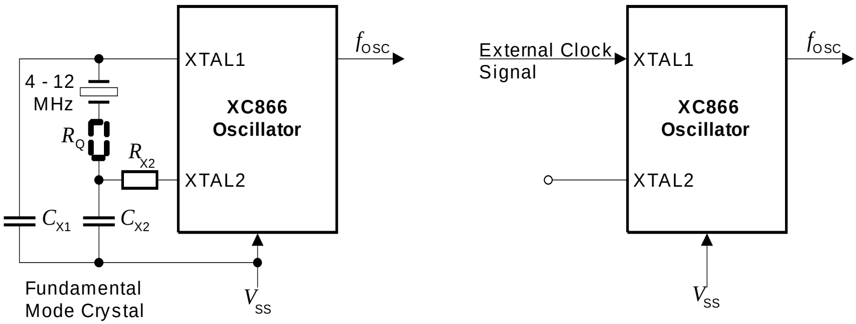
Table 20 Clock Mode Selection

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 SAA-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.



Crystal Frequency	C_{X1}, C_{X2} [1)]	R_{X2} [1)]
4 MHz	33 pF	0
8 MHz	18 pF	0
10 MHz	15 pF	0
12 MHz	12 pF	0

1) Note that these are evaluation start values!

Clock_EXOSC

Figure 24 External Oscillator Circuitries

Note: For crystal operation, it is strongly recommended to measure the negative resistance in the final target system (layout) to determine the optimum parameters for the oscillator operation. Please refer to the minimum and maximum values of the negative resistance specified by the crystal supplier.

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]

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

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]

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

3.17 Capture/Compare Unit 6

The Capture/Compare Unit 6 (CCU6) provides two independent timers (T12, T13), which can be used for Pulse Width Modulation (PWM) generation, especially for AC-motor control. The CCU6 also supports special control modes for block commutation and multi-phase machines.

The timer T12 can function in capture and/or compare mode for its three channels. The timer T13 can work in compare mode only.

The multi-channel control unit generates output patterns, which can be modulated by T12 and/or T13. The modulation sources can be selected and combined for the signal modulation.

Timer T12 Features:

- Three capture/compare channels, each channel can be used either as a capture or as a compare channel
- Supports generation of a three-phase PWM (six outputs, individual signals for highside and lowside switches)
- 16-bit resolution, maximum count frequency = peripheral clock frequency
- Dead-time control for each channel to avoid short-circuits in the power stage
- Concurrent update of the required T12/13 registers
- Generation of center-aligned and edge-aligned PWM
- Supports single-shot mode
- Supports many interrupt request sources
- Hysteresis-like control mode

Timer T13 Features:

- One independent compare channel with one output
- 16-bit resolution, maximum count frequency = peripheral clock frequency
- Can be synchronized to T12
- Interrupt generation at period-match and compare-match
- Supports single-shot mode

Additional Features:

- Implements block commutation for Brushless DC-drives
- Position detection via Hall-sensor pattern
- Automatic rotational speed measurement for block commutation
- Integrated error handling
- Fast emergency stop without CPU load via external signal ($\overline{\text{CTRAP}}$)
- Control modes for multi-channel AC-drives
- Output levels can be selected and adapted to the power stage

4.1.2 Absolute Maximum Rating

Maximum ratings are the extreme limits to which the SAA-XC866 can be subjected to without permanent damage.

Table 32 Absolute Maximum Rating Parameters

Parameter	Symbol	Limit Values		Unit	Notes
		min.	max.		
Ambient temperature	T_A	-40	140	°C	under bias
Storage temperature	T_{ST}	-65	150	°C	1)
Junction temperature	T_J	-40	150	°C	under bias1)
Voltage on power supply pin with respect to V_{SS}	V_{DDP}	-0.5	6	V	1)
Voltage on core supply pin with respect to V_{SS}	V_{DDC}	-0.5	3.25	V	1)
Voltage on any pin with respect to V_{SS}	V_{IN}	-0.5	V_{DDP} + 0.5 or max. 6	V	Whichever is lower1)
Input current on any pin during overload condition	I_{IN}	-10	10	mA	1)
Absolute sum of all input currents during overload condition	$\Sigma\|I_{IN}\|$	–	50	mA	1)

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

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During absolute maximum rating overload conditions ($V_{IN} > V_{DDP}$ or $V_{IN} < V_{SS}$) the voltage on V_{DDP} pin with respect to ground (V_{SS}) must not exceed the values defined by the absolute maximum ratings.

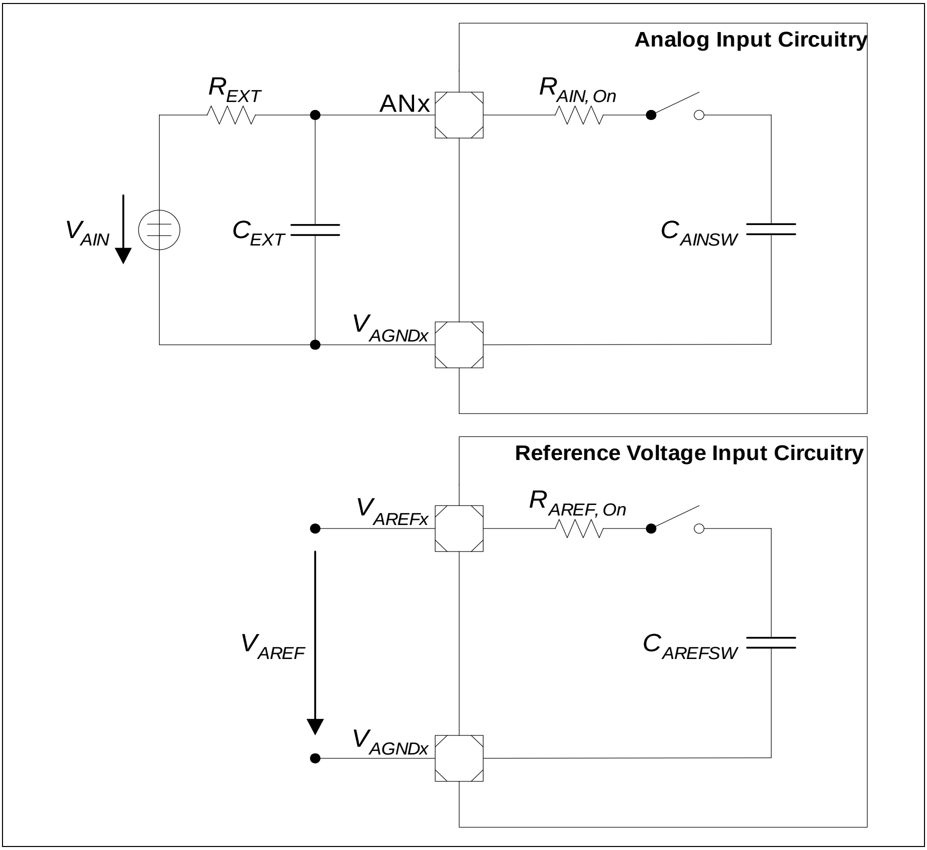


Figure 37 ADC Input Circuits

4.2.3.1 ADC Conversion Timing

Conversion time, $t_C = t_{ADC} \times (1 + r \times (3 + n + STC))$, where

$r = CTC + 2$ for $CTC = 00_B$, 01_B or 10_B,

$r = 32$ for $CTC = 11_B$,

CTC = Conversion Time Control (GLOBCTR.CTC),

STC = Sample Time Control (INPCR0.STC),

n = 8 or 10 (for 8-bit and 10-bit conversion respectively),

$t_{ADC} = 1 / f_{ADC}$