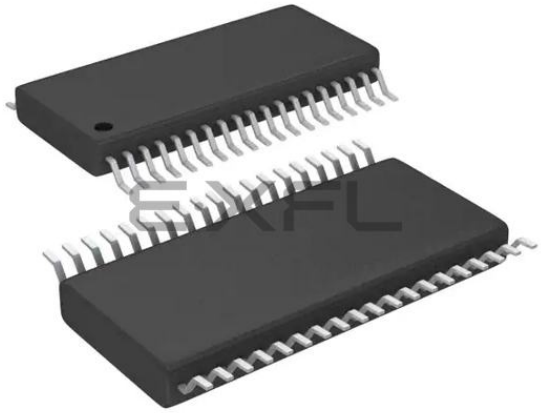

Welcome to E-XFL.COM

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	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/sak-xc866l-2fri-bc

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)

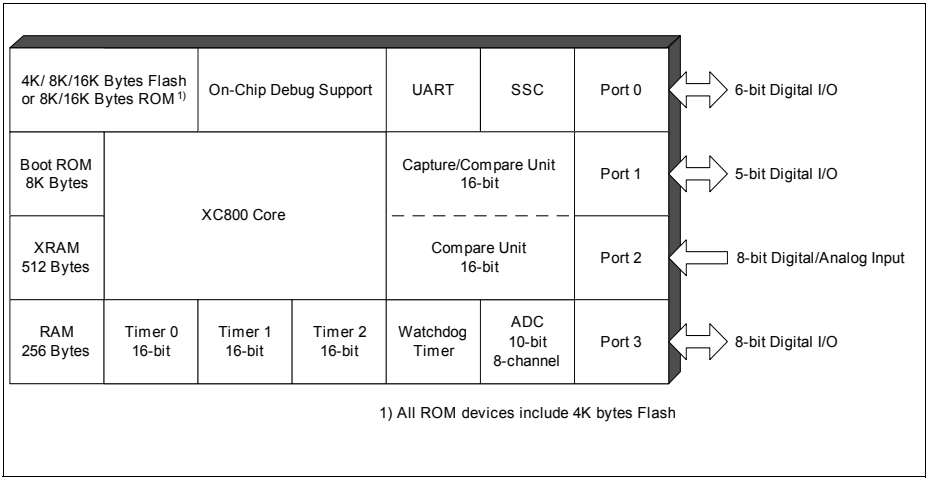

Figure 1 XC866 Functional Units

Table 3 Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Type	Reset State	Function
V_{DDP}	18	–	–	**I/O Port Supply (3.3 V/5.0 V)** Also used by EVR and analog modules.
V_{SSP}	19	–	–	**I/O Port Ground**
V_{DDC}	8	–	–	**Core Supply Monitor (2.5 V)**
V_{SSC}	7	–	–	**Core Supply Ground**
V_{AREF}	25	–	–	**ADC Reference Voltage**
V_{AGND}	24	–	–	**ADC Reference Ground**
XTAL1	6	I	Hi-Z	**External Oscillator Input** **(NC if not needed)**
XTAL2	5	O	Hi-Z	**External Oscillator Output** **(NC if not needed)**
TMS	11	I	PD	**Test Mode Select**
$\overline{\text{RESET}}$	38	I	PU	**Reset Input**
MBC[1)]	1	I	PU	**Monitor & BootStrap Loader Control**

1) An external pull-up device in the range of 4.7 kΩ to 100 kΩ is required to enter user mode. Alternatively MBC can be tied to high if alternate functions (for debugging) of the pin are not utilized.

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.

3.2.4 XC866 Register Overview

The SFRs of the XC866 are organized into groups according to their functional units. The contents (bits) of the SFRs are summarized in **Table 7** to **Table 15**, with the addresses of the bitaddressable SFRs appearing in bold typeface.

The CPU SFRs can be accessed in both the standard and mapped memory areas (RMAP = 0 or 1).

Table 7 CPU Register Overview

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
RMAP = 0 or 1										
81_H	**SP** **Reset: 07_H** Stack Pointer Register	Bit Field	SP							
		Type	rw							
82_H	**DPL** **Reset: 00_H** Data Pointer Register Low	Bit Field	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
83_H	**DPH** **Reset: 00_H** Data Pointer Register High	Bit Field	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
87_H	**PCON** **Reset: 00_H** Power Control Register	Bit Field	SMOD	0			GF1	GF0	0	IDLE
		Type	rw	r			rw	rw	r	rw
88_H	**TCON** **Reset: 00_H** Timer Control Register	Bit Field	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		Type	rwh	rw	rwh	rw	rwh	rw	rwh	rw
89_H	**TMOD** **Reset: 00_H** Timer Mode Register	Bit Field	GATE1	0	T1M		GATE0	0	T0M	
		Type	rw	r	rw		rw	r	rw	
$8A_H$	**TL0** **Reset: 00_H** Timer 0 Register Low	Bit Field	VAL							
		Type	rwh							
$8B_H$	**TL1** **Reset: 00_H** Timer 1 Register Low	Bit Field	VAL							
		Type	rwh							
$8C_H$	**TH0** **Reset: 00_H** Timer 0 Register High	Bit Field	VAL							
		Type	rwh							
$8D_H$	**TH1** **Reset: 00_H** Timer 1 Register High	Bit Field	VAL							
		Type	rwh							
98_H	**SCON** **Reset: 00_H** Serial Channel Control Register	Bit Field	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
		Type	rw	rw	rw	rw	rw	rwh	rwh	rwh
99_H	**SBUF** **Reset: 00_H** Serial Data Buffer Register	Bit Field	VAL							
		Type	rwh							
$A2_H$	**EO** **Reset: 00_H** Extended Operation Register	Bit Field	0			TRAP_EN	0			DPSEL0
		Type	r			rw	r			rw
$A8_H$	**IEN0** **Reset: 00_H** Interrupt Enable Register 0	Bit Field	EA	0	ET2	ES	ET1	EX1	ET0	EX0
		Type	rw	r	rw	rw	rw	rw	rw	rw
$B8_H$	**IP** **Reset: 00_H** Interrupt Priority Register	Bit Field	0		PT2	PS	PT1	PX1	PT0	PX0
		Type	r		rw	rw	rw	rw	rw	rw
$B9_H$	**IPH** **Reset: 00_H** Interrupt Priority Register High	Bit Field	0		PT2H	PSH	PT1H	PX1H	PT0H	PX0H
		Type	r		rw	rw	rw	rw	rw	rw
$D0_H$	**PSW** **Reset: 00_H** Program Status Word Register	Bit Field	CY	AC	F0	RS1	RS0	OV	F1	P
		Type	rw	rwh	rwh	rw	rw	rwh	rwh	rh
$E0_H$	**ACC** **Reset: 00_H** Accumulator Register	Bit Field	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
		Type	rw	rw	rw	rw	rw	rw	rw	rw
$E8_H$	**IEN1** **Reset: 00_H** Interrupt Enable Register 1	Bit Field	ECCIP3	ECCIP2	ECCIP1	ECCIP0	EXM	EX2	ESSC	EADC
		Type	rw	rw	rw	rw	rw	rw	rw	rw

Table 11 ADC Register Overview (cont'd)

Addr	Register Name	Bit	7	6	5	4	3	2	1	0
CA_H	**ADC_RESR0L** **Reset: 00_H** Result Register 0 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CB_H	**ADC_RESR0H** **Reset: 00_H** Result Register 0 High	Bit Field	RESULT[9:2]							
		Type	rh							
CC_H	**ADC_RESR1L** **Reset: 00_H** Result Register 1 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CD_H	**ADC_RESR1H** **Reset: 00_H** Result Register 1 High	Bit Field	RESULT[9:2]							
		Type	rh							
CE_H	**ADC_RESR2L** **Reset: 00_H** Result Register 2 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
CF_H	**ADC_RESR2H** **Reset: 00_H** Result Register 2 High	Bit Field	RESULT[9:2]							
		Type	rh							
$D2_H$	**ADC_RESR3L** **Reset: 00_H** Result Register 3 Low	Bit Field	RESULT[1:0]		0	VF	DRC	CHNR		
		Type	rh		r	rh	rh	rh		
$D3_H$	**ADC_RESR3H** **Reset: 00_H** Result Register 3 High	Bit Field	RESULT[9:2]							
		Type	rh							
RMAP = 0, Page 3										
CA_H	**ADC_RESRA0L** **Reset: 00_H** Result Register 0, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CB_H	**ADC_RESRA0H** **Reset: 00_H** Result Register 0, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
CC_H	**ADC_RESRA1L** **Reset: 00_H** Result Register 1, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CD_H	**ADC_RESRA1H** **Reset: 00_H** Result Register 1, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
CE_H	**ADC_RESRA2L** **Reset: 00_H** Result Register 2, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
CF_H	**ADC_RESRA2H** **Reset: 00_H** Result Register 2, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
$D2_H$	**ADC_RESRA3L** **Reset: 00_H** Result Register 3, View A Low	Bit Field	RESULT[2:0]			VF	DRC	CHNR		
		Type	rh			rh	rh	rh		
$D3_H$	**ADC_RESRA3H** **Reset: 00_H** Result Register 3, View A High	Bit Field	RESULT[10:3]							
		Type	rh							
RMAP = 0, Page 4										
CA_H	**ADC_RCR0** **Reset: 00_H** Result Control Register 0	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CB_H	**ADC_RCR1** **Reset: 00_H** Result Control Register 1	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CC_H	**ADC_RCR2** **Reset: 00_H** Result Control Register 2	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CD_H	**ADC_RCR3** **Reset: 00_H** Result Control Register 3	Bit Field	VFCTR	WFR	0	IEN	0			DRCTR
		Type	rw	rw	r	rw	r			rw
CE_H	**ADC_VFCR** **Reset: 00_H** Valid Flag Clear Register	Bit Field	0				VFC3	VFC2	VFC1	VFC0
		Type	r				w	w	w	w
RMAP = 0, Page 5										

Table 12 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 13 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							

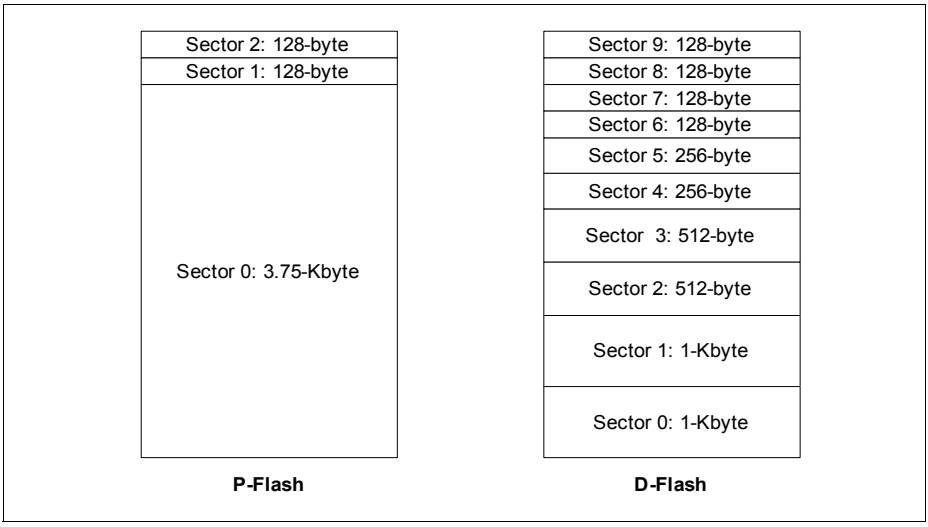

Figure 11 Flash Bank Sectorization

The internal structure of each Flash bank represents a sector architecture for flexible erase capability. The minimum erase width is always a complete sector, and sectors can be erased separately or in parallel. Contrary to standard EPROMs, erased Flash memory cells contain 0s.

The D-Flash bank is divided into more physical sectors for extended erasing and reprogramming capability; even numbers for each sector size are provided to allow greater flexibility and the ability to adapt to a wide range of application requirements.

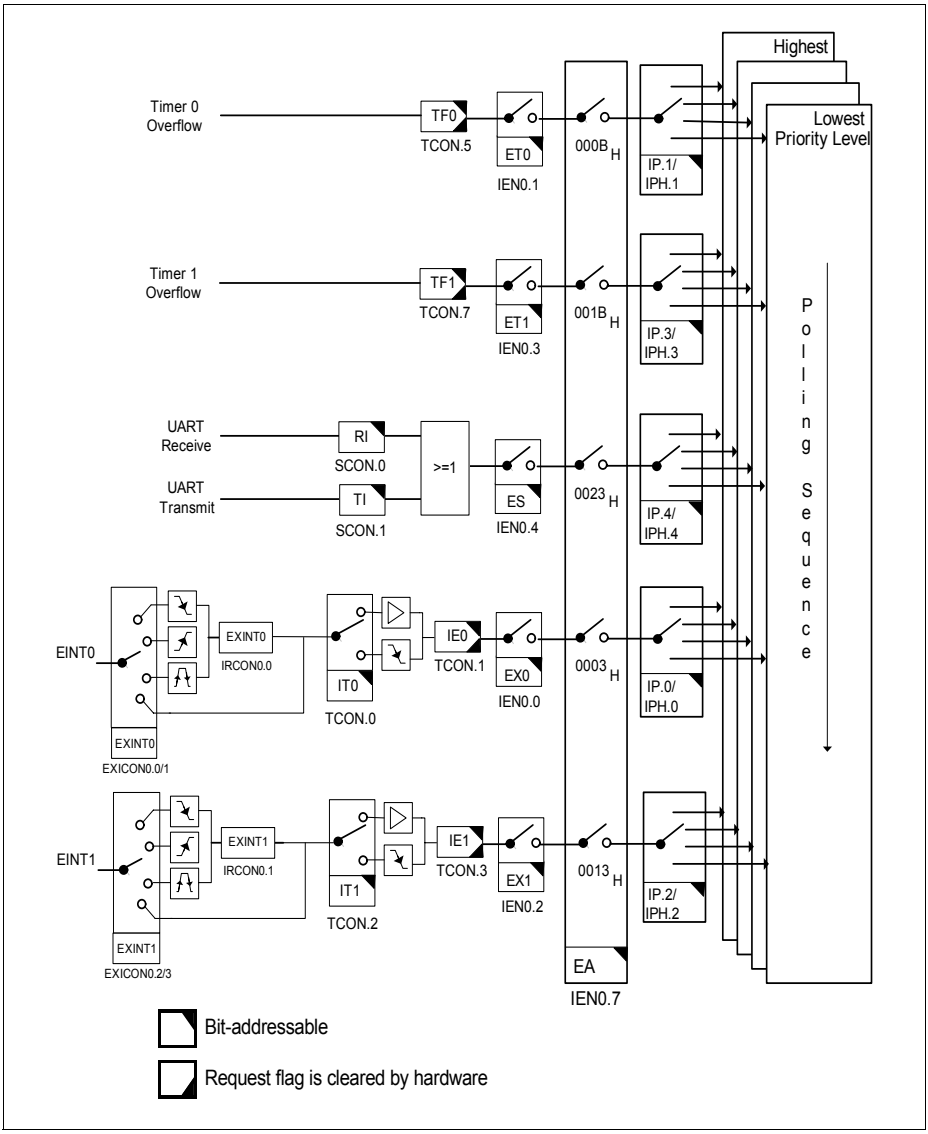

Figure 14 Interrupt Request Sources (Part 1)

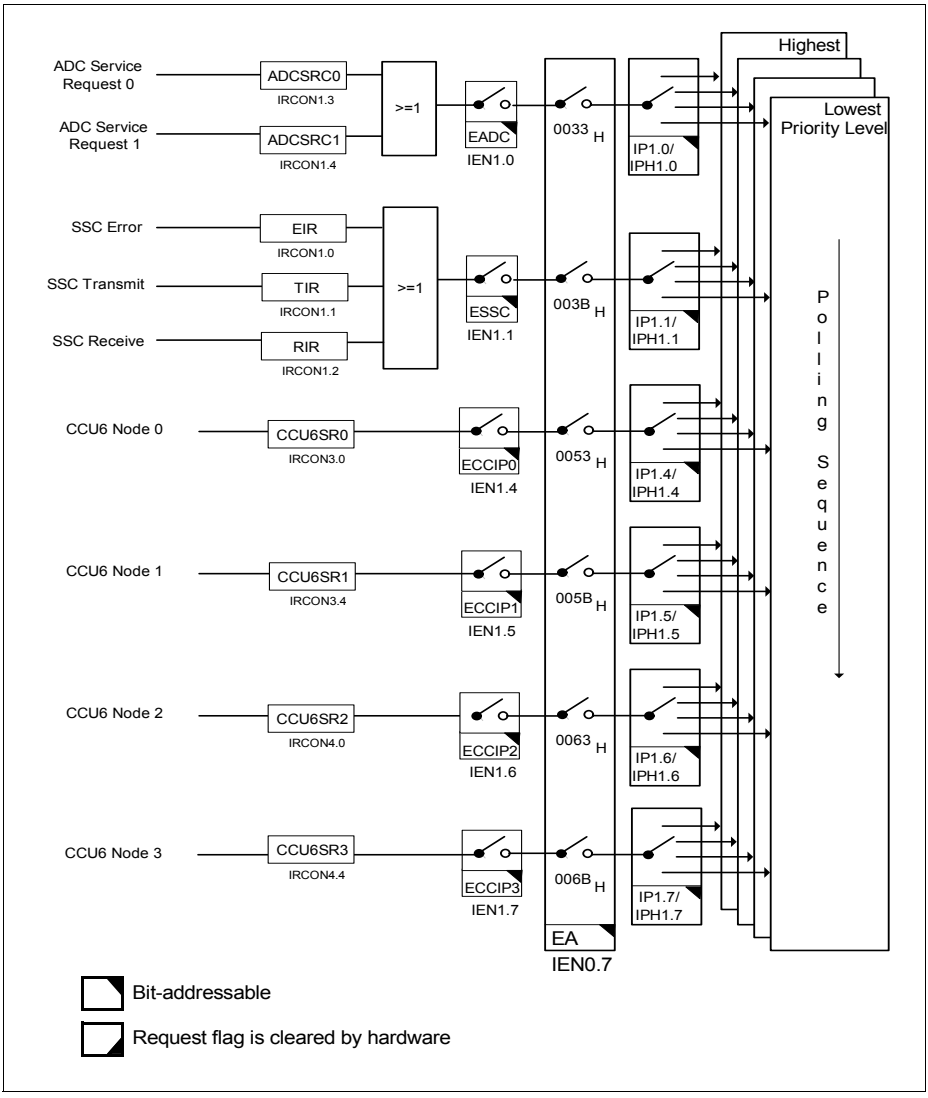

Figure 16 Interrupt Request Sources (Part 3)

3.4.3 Interrupt Priority

Each interrupt source, except for NMI, can be individually programmed to one of the four possible priority levels. The NMI has the highest priority and supersedes all other interrupts. Two pairs of interrupt priority registers (IP and IPH, IP1 and IPH1) are available to program the priority level of each non-NMI interrupt vector.

A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another interrupt of the same or lower priority. Further, an interrupt of the highest priority cannot be interrupted by any other interrupt source.

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 18**.

Table 18 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, Fractional Divider, LIN Interrupts	6
ADC Interrupt	7
SSC Interrupt	8
External Interrupt 2	9
External Interrupt [6:3]	10
CCU6 Interrupt Node Pointer 0	11
CCU6 Interrupt Node Pointer 1	12
CCU6 Interrupt Node Pointer 2	13
CCU6 Interrupt Node Pointer 3	14

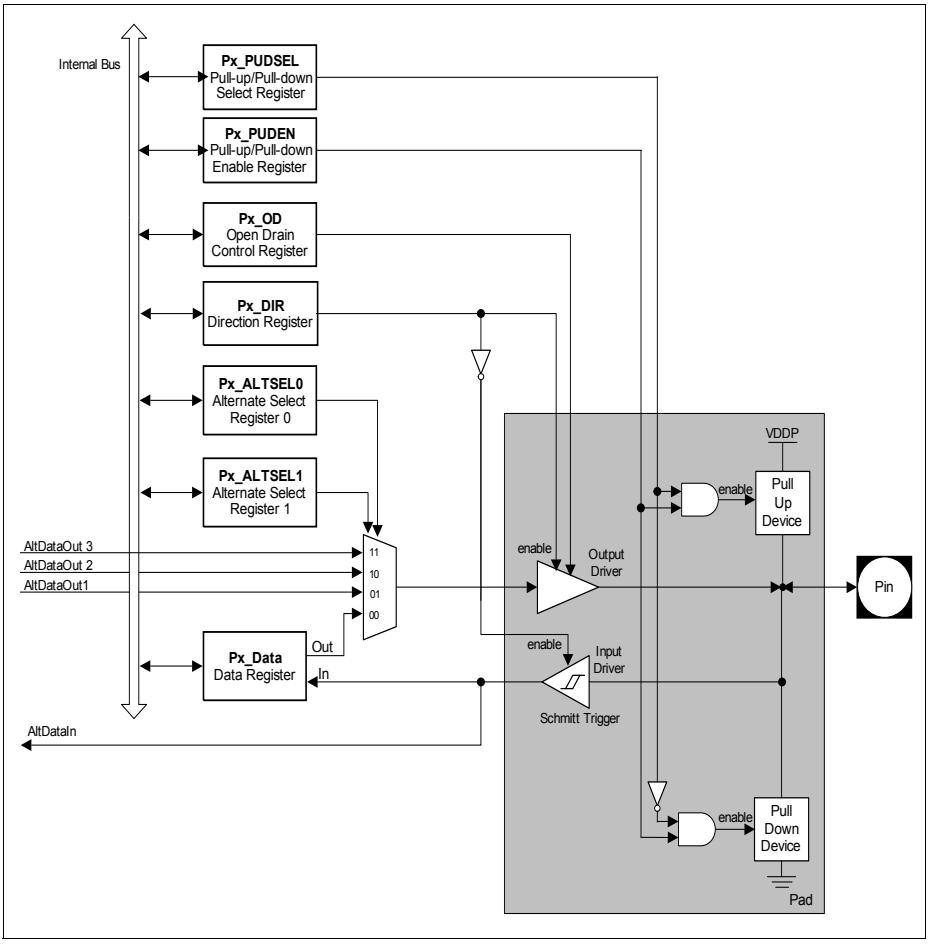

Figure 18 General Structure of Bidirectional Port

3.7 Reset Control

The XC866 has five types of reset: power-on reset, hardware reset, watchdog timer reset, power-down wake-up reset, and brownout reset.

When the XC866 is first powered up, the status of certain pins (see **Table 20**) must be defined to ensure proper start operation of the device. At the end of a reset sequence, the sampled values are latched to select the desired boot option, which cannot be modified until the next power-on reset or hardware reset. This guarantees stable conditions during the normal operation of the device.

In order to power up the system properly, the external reset pin $\overline{\text{RESET}}$ must be asserted until V_{DDC} reaches 0.9*V_{DDC}. The delay of external reset can be realized by an external capacitor at $\overline{\text{RESET}}$ pin. This capacitor value must be selected so that V_{RESET} reaches 0.4 V, but not before V_{DDC} reaches 0.9* V_{DDC}.

A typical application example is shown in **Figure 21**. V_{DDP} capacitor value is 300 nF. V_{DDC} capacitor value is 220 nF. The capacitor connected to $\overline{\text{RESET}}$ pin is 100 nF.

Typically, the time taken for V_{DDC} to reach 0.9*V_{DDC} is less than 50 µs once V_{DDP} reaches 2.3V. Hence, based on the condition that 10% to 90% V_{DDP} (slew rate) is less than 500 µs, the $\overline{\text{RESET}}$ pin should be held low for 500 µs typically. See **Figure 22**.

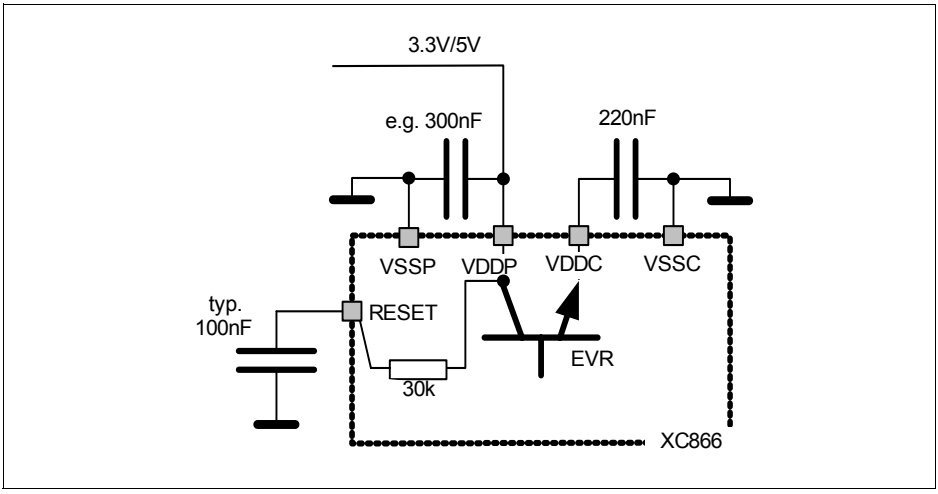

Figure 21 Reset Circuitry

For power saving purposes, the clocks may be disabled or slowed down according to **Table 23**.

Table 23 System frequency (f_{sys} = 80 MHz)

Power Saving Mode	Action
Idle	Clock to the CPU is disabled.
Slow-down	Clocks to the CPU and all the peripherals, including CCU6, are divided by a common programmable factor defined by bit field CMCON.CLKREL.
Power-down	Oscillator and PLL are switched off.

3.9 Power Saving Modes

The power saving modes of the XC866 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 26**) 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

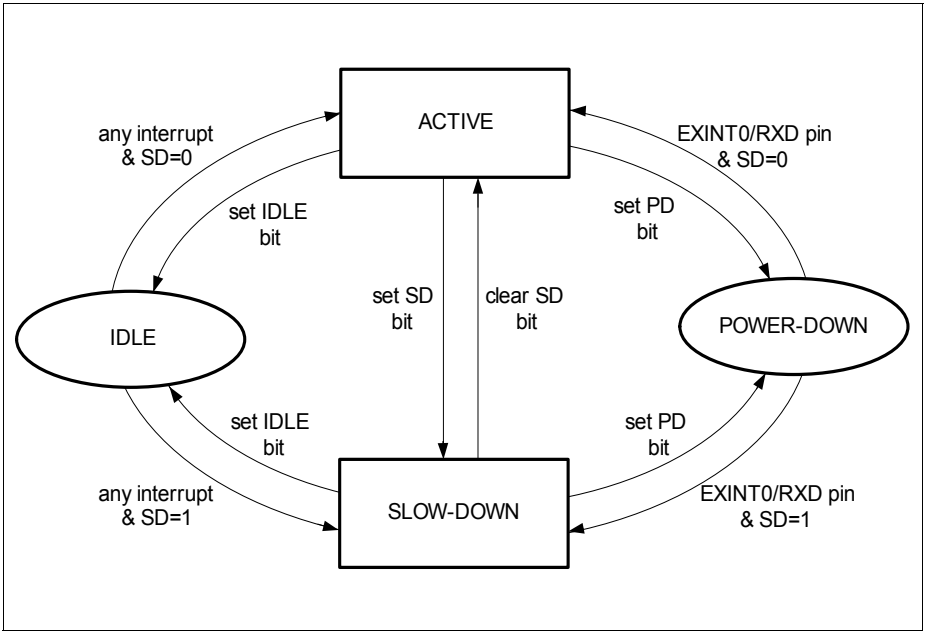

Figure 26 Transition between Power Saving Modes

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**.

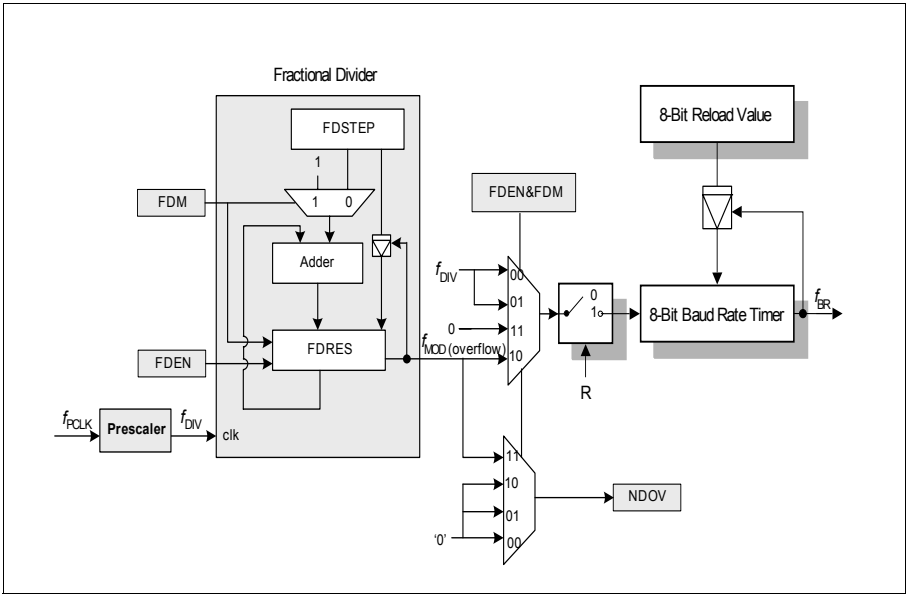

Figure 29 Baud-rate Generator Circuitry

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

3.13 LIN Protocol

The UART can be used to support the Local Interconnect Network (LIN) protocol for both master and slave operations. The LIN baud rate detection feature 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.

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 30**. 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

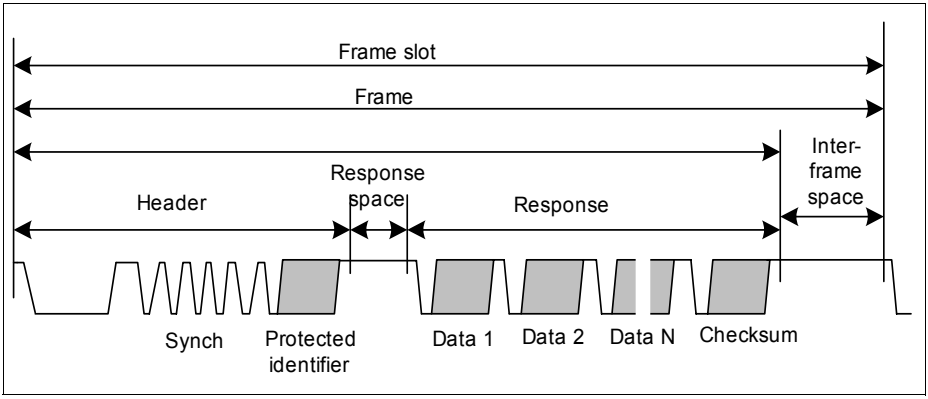

Figure 30 Structure of LIN Frame

3.13.1 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.

Table 38 Power Down Current (Operating Conditions apply; V_{DDP} = 5V range)

Parameter	Symbol	Limit Values		Unit	Test Condition Remarks
		typ.[1]	max.[2]		
V_{DDP} = 5V Range					
Power-Down Mode[3]	I_{PDP}	1	10	µA	T_A = + 25 °C.[4]
		-	30	µA	T_A = + 85 °C, XC866-4FR, XC866-2FR [4][5]
		-	35	µA	T_A = + 85 °C, XC866-1FR, ROM device[4][5]

1) The typical I_{PDP} values are measured at V_{DDP} = 5.0 V.

2) The maximum I_{PDP} values are measured at V_{DDP} = 5.5 V.

3) I_{PDP} (power-down mode) has a maximum value of 200 µA at T_A = + 125 °C.

4) I_{PDP} (power-down mode) is measured with: $\overline{\text{RESET}}$ = V_{DDP}, V_{AGND}= V_{SS}, RXD/INT0 = V_{DDP}; rest of the ports are programmed to be input with either internal pull devices enabled or driven externally to ensure no floating inputs.

5) Not subject to production test, verified by design/characterization.

4.3.3 Power-on Reset and PLL Timing

Table 42 Power-On Reset and PLL Timing (Operating Conditions apply)

Parameter	Symbol	Limit Values			Unit	Test Conditions
		min.	typ.	max.		
Pad operating voltage	V_{PAD} CC	2.3	–	–	V	
On-Chip Oscillator start-up time	t_{OSCST} CC	–	–	500	ns	
Flash initialization time	t_{FINIT} CC	–	160	–	µs	
$\overline{RESET}$ hold time[1]	t_{RST} SR	–	500	–	µs	V_{DDP} rise time (10% – 90%) ≤ 500µs
PLL lock-in in time	t_{LOCK} CC	–	–	200	µs	
PLL accumulated jitter	D_P	–	–	0.7	ns	2)

1) $\overline{RESET}$ signal has to be active (low) until V_{DDC} has reached 90% of its maximum value (typ. 2.5V).

2) PLL lock at 80 MHz using a 4 MHz external oscillator. The PLL Divider settings are K = 2, N = 40 and P = 1.

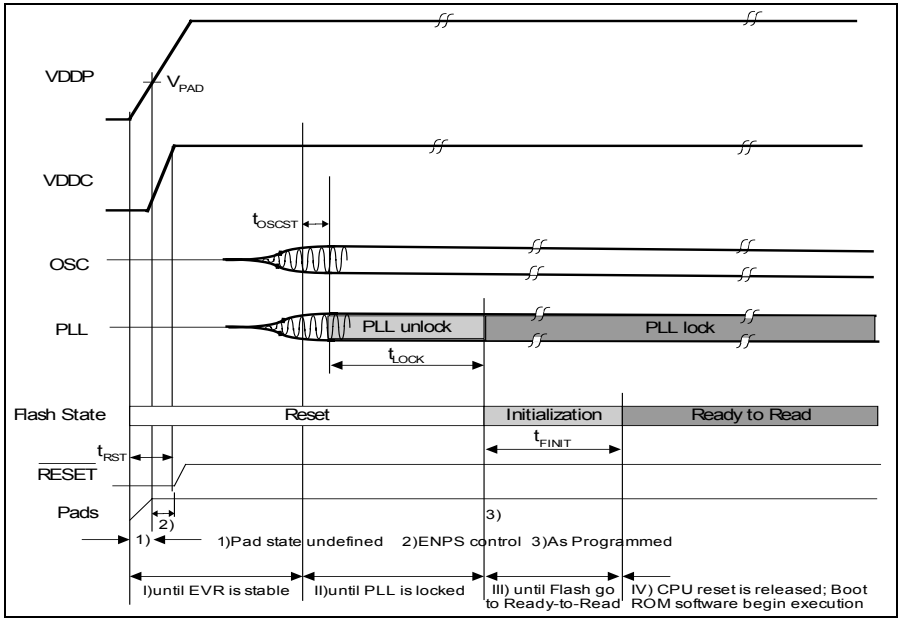

Figure 42 Power-on Reset Timing

Table 45 JTAG Timing (Operating Conditions apply; C_L = 50 pF)

Parameter	Symbol		Limits		Unit
			min	max	
TMS setup to TCK ↑	t_1	SR	8.0	–	ns
TMS hold to TCK ↑	t_2	SR	5.0	–	ns
TDI setup to TCK ↑	t_1	SR	11.0	–	ns
TDI hold to TCK ↑	t_2	SR	6.0	–	ns
TDO valid output from TCK ↓	t_3	CC	–	23	ns
TDO high impedance to valid output from TCK ↓	t_4	CC	–	26	ns
TDO valid output to high impedance from TCK ↓	t_5	CC	–	18	ns

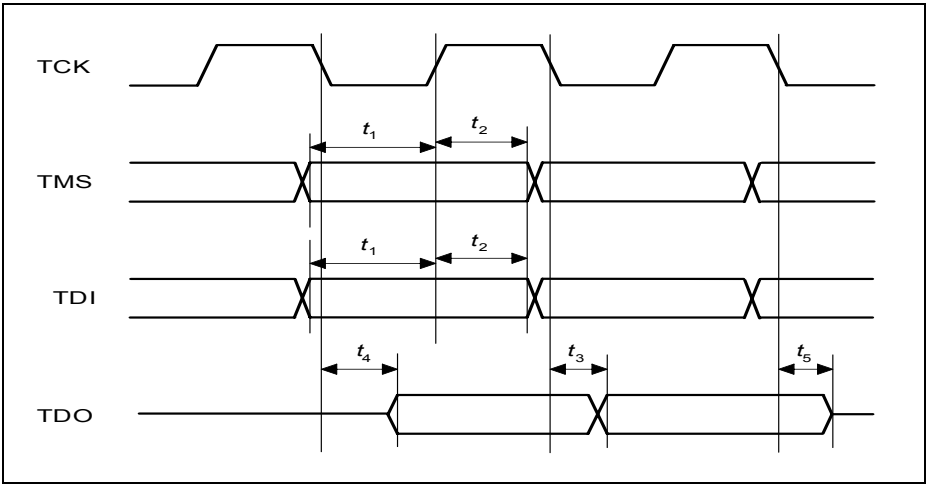

Figure 44 JTAG Timing