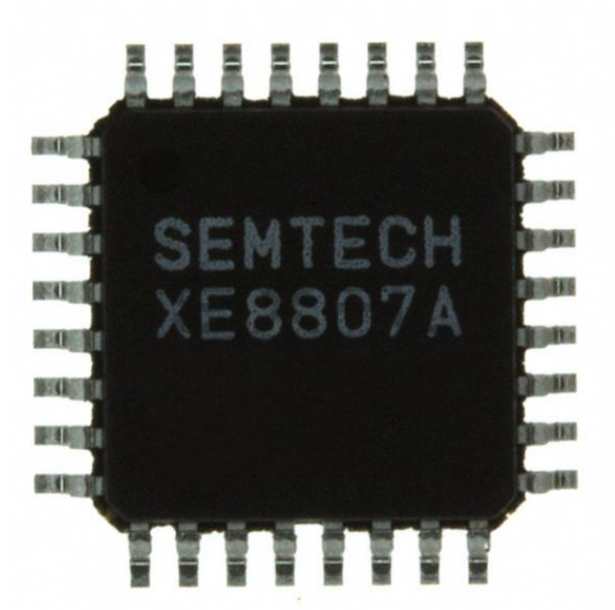

Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	Sensing Machine
Core Processor	Coolrisc816®
Program Memory Type	FLASH (11kB)
Controller Series	XE8000
RAM Size	512 x 8
Interface	UART, USRT
Number of I/O	24
Voltage - Supply	2.4V ~ 5.5V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/semtech/xe8807ami026tlf

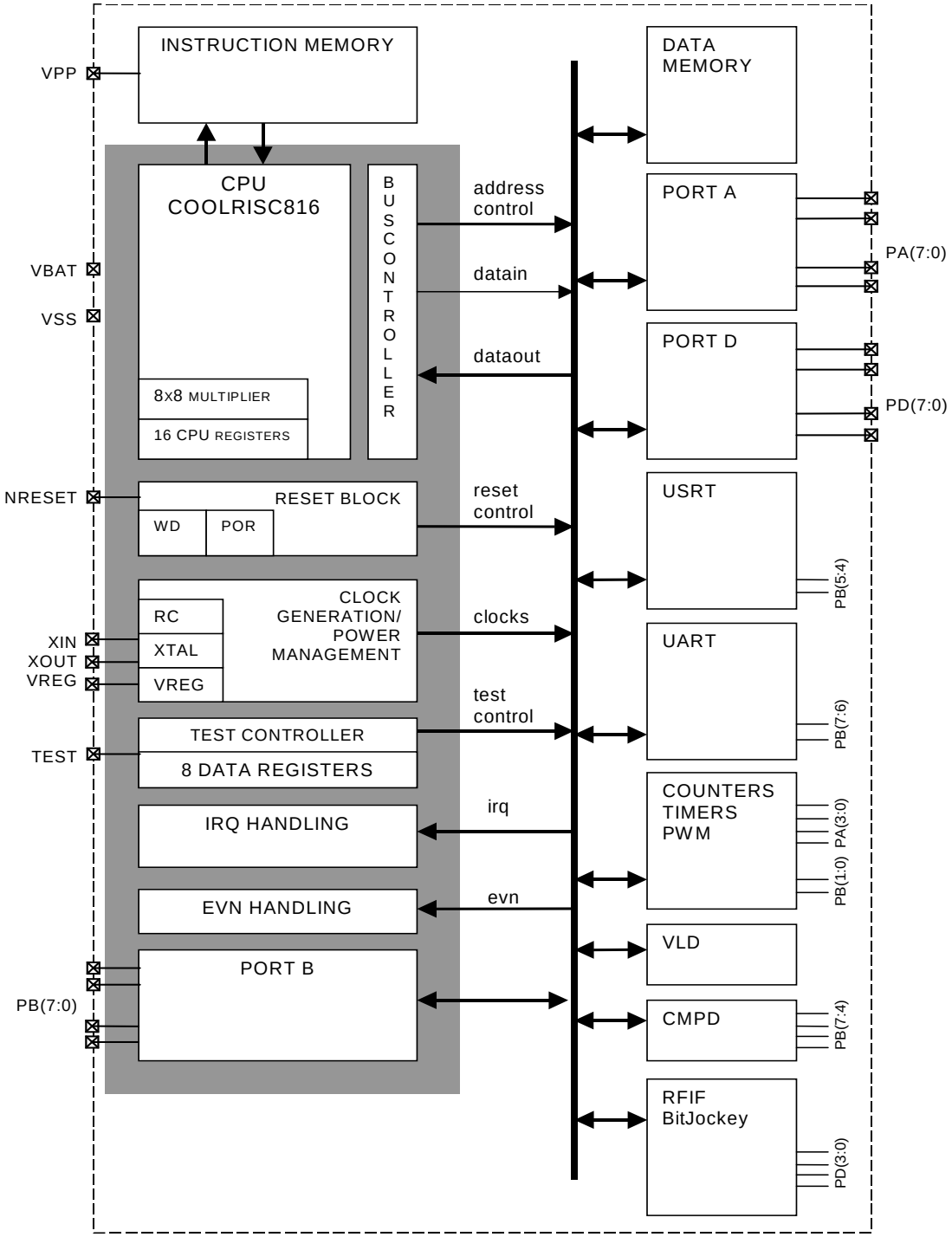

Figure 1-1. Block schematic of the XE8806A and XE8807A circuits.

1.2 Pin map

The XE8806A and XE8807A can be delivered in different packages. The pin maps for the different packages are given below.

1.2.1 TQFP-32

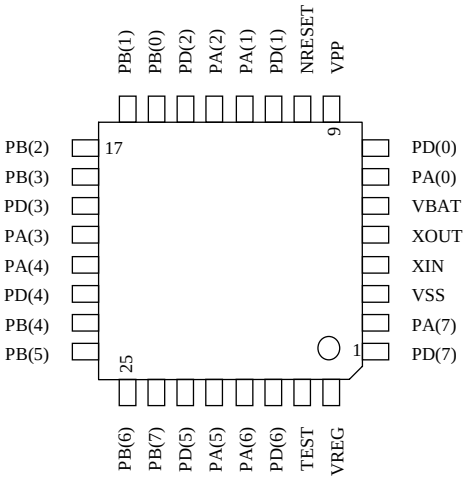

Figure 1-2. TQFP-32 pin map

1.2.2 SO-28

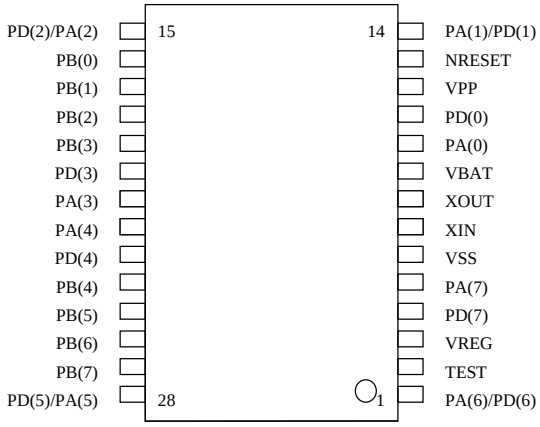

Figure 1-3. SO28 pin map

In the SO-28 package, 4 pins of Port A and Port D are connected together. It is up to the user to choose between the functionality of Port A or Port D for these pins.

Note: if one of the pins PD(1), PD(2), PD(5), PD(6) is used as output, the pull up of the corresponding pin of Port A should be disabled in order to have low power consumption.

1.2.5 Bare die XE8807A

The circuit is also available in bare die for chip on board assembly. All VBAT pins and all VSS pins should be connected together. The substrate of the circuit is connected to VSS.

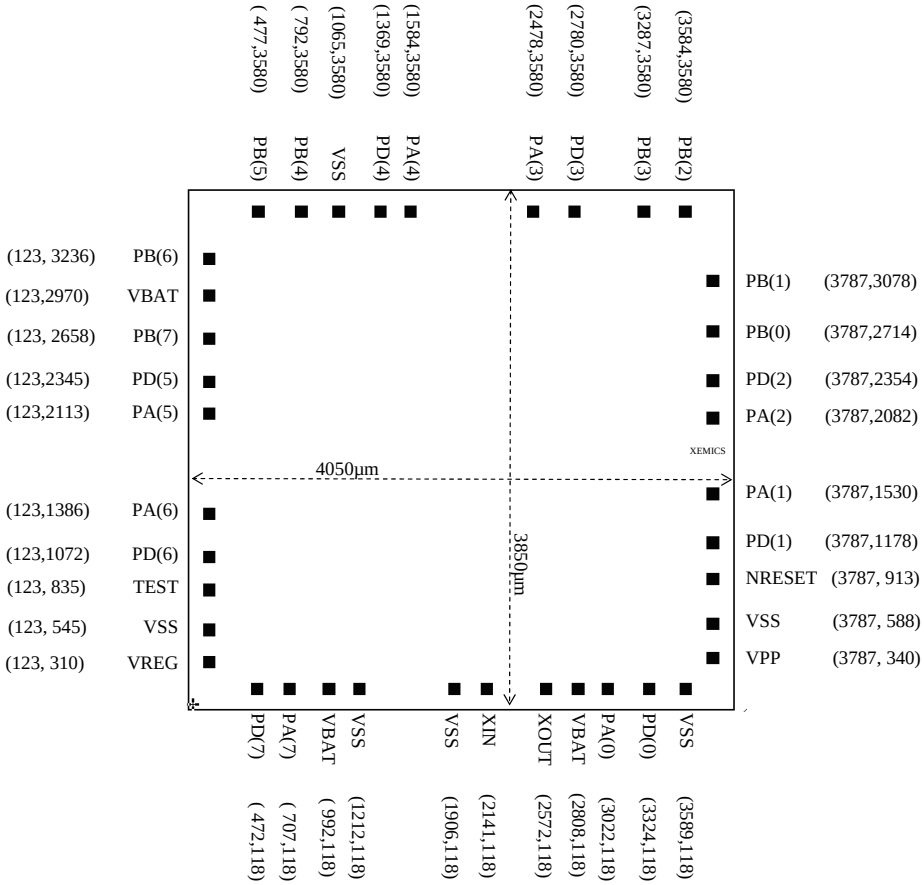

Figure 1-6. Die dimension and pin location of the XE8807A

1.3 Pin assignment

The table below gives a short description of the different pin assignments.

Pin	Assignment
VBAT	Positive power supply
VSS	Negative power supply
VREG	Connection for the mandatory external capacitor of the voltage regulator
VPP	High voltage supply for flash memory programming (NC in ROM versions)
NRESET	Resets the circuit when the voltage is low
TEST	Sets the pin to flash programming mode
XIN/XOUT	Quartz crystal connections, also used for flash memory programming
PA(7:0)	Parallel input port A pins
PB(7:0)	Parallel I/O port B pins
PD(7:0)	Parallel I/O port D pins

Table 1-1. Pin assignment

2 XE8806A and XE8807A Performance

Sflag	-,-,-, a	a[7] := C; a[6] := C xor V; a[5] := ST full; a[4] := ST empty
Rflag reg	C, V, Z, a	a := reg << 1; ; a[0] := 0; C := reg[7]
Rflag eaddr	C, V, Z, a	a := DM(eaddr)<<1; a[0] :=0; C := DM(eaddr)[7]
Freq divn	-,-,-, -	reduces the CPU frequency (divn=nodiv, div2, div4, div8, div16)
Halt	-,-,-, -	halts the CPU
Nop	-,-,-, -	no operation

- = unchanged, u = undefined, *MSHR reg,# 1 doesn't shift by 1

Table 3-4. Instruction short reference

The Coolrisc816 has 8 different addressing modes. These modes are described in Table 3-5. In this table, the notation ix refers to one of the data memory index registers i0, i1, i2 or i3. Using eaddr in an instruction of Table 3-4 will access the data memory at the address DM(eaddr) and will simultaneously execute the index operation.

extended address eaddr	accessed data memory location DM(eaddr)	index operation	
addr[7:0]	DM(h00&addr[7:0])	-	direct addressing
(ix)	DM(ix)	-	indexed addressing
(ix, offset[7:0])	DM(ix+offset)	-	indexed addressing with immediate offset
(ix,r3)	DM(ix+r3)	-	indexed addressing with register offset
(ix)+	DM(ix)	ix := ix+1	indexed addressing with index post-increment
(ix,offset[7:0])+	DM(ix+offset)	ix := ix+offset	indexed addressing with index post-increment by the offset
-(ix)	DM(ix-1)	ix := ix-1	indexed addressing with index pre-decrement
-(ix,offset[7:0])	DM(ix-offset)	ix := ix -offset	indexed addressing with index pre-decrement by the offset

Table 3-5. Extended address mode description

Eleven different jump conditions are implemented as shown in Table 3-6. The contents of the column CC in this table should replace the CC notation in the instruction description of Table 3-4.

CC	condition
CS	C=1
CC	C=0
ZS	Z=1
ZC	Z=0
VS	V=1
VC	V=0
EV	(EV1 or EV0)=1
After CMP op1,op2	
EQ	op1=op2
NE	op1≠op2
GT	op1>op2
GE	op1≥op2
LT	op1<op2
LE	op1≤op2

Table 3-6. Jump condition description

4.2.3 Port A (h0020-h0027)

Name / Address	7	6	5	4	3	2	1	0
RegPAIn h0020	PAIn[7:0] r							
RegPADebounce h0021	PADebounce[7:0] rw,00000000,pconf							
RegPAEdge h0022	PAEdge[7:0] rw,00000000,glob							
RegPAPullup h0023	PAPullup[7:0] rw,11111111,pconf							
RegPARes0 h0024	PARes0[7:0] rw, 00000000, glob							
RegPARes1 h0025	PARes1[7:0] rw,00000000,glob							
RegPACtrl h0026	r0	r0	r0	r0	r0	r0	r0	DebFast rw,0,pconf
RegPASnaptorail h0027	PASnaptorail[7:0] rw,00000000,pconf							

Table 4-5. Port A registers

4.2.4 Port B (h0028-h002F)

Name / Address	7	6	5	4	3	2	1	0
RegPBOut h0028	PBOut[7:0] rw,00000000,pconf							
RegPBIn h0029	PBIn[7:0] r							
RegPBDir h002A	PBDir[7:0] rw,00000000,pconf							
RegPBOpen h002B	PBOpen[7:0] rw,00000000,pconf							
RegPBPullup h002C	PBPullup[7:0] rw,11111111,pconf							
RegPBAna h002D	PBAna[7:0] rw,00000000,pconf							

Table 4-6. Port B registers

4.2.5 Port D (h0030-h0033)

Name / Address	7	6	5	4	3	2	1	0
RegPDOut h0030	PDOut[7:0] rw,00000000,pconf							
RegPDIn h0031	PDIn[7:0] r							
RegPDDir h0032	PDDir[7:0] rw,00000000,pconf							
RegPDPullup h0033	PDSnapToRail[3:0] rw,0000,pconf				PDPullup[3:0] rw,1111,pconf			

Table 4-7. Port D registers

4.2.6 Flash programming (h0038-003B)

These four registers are used during flash programming only. Refer to the flash programming algorithm documentation for more details.

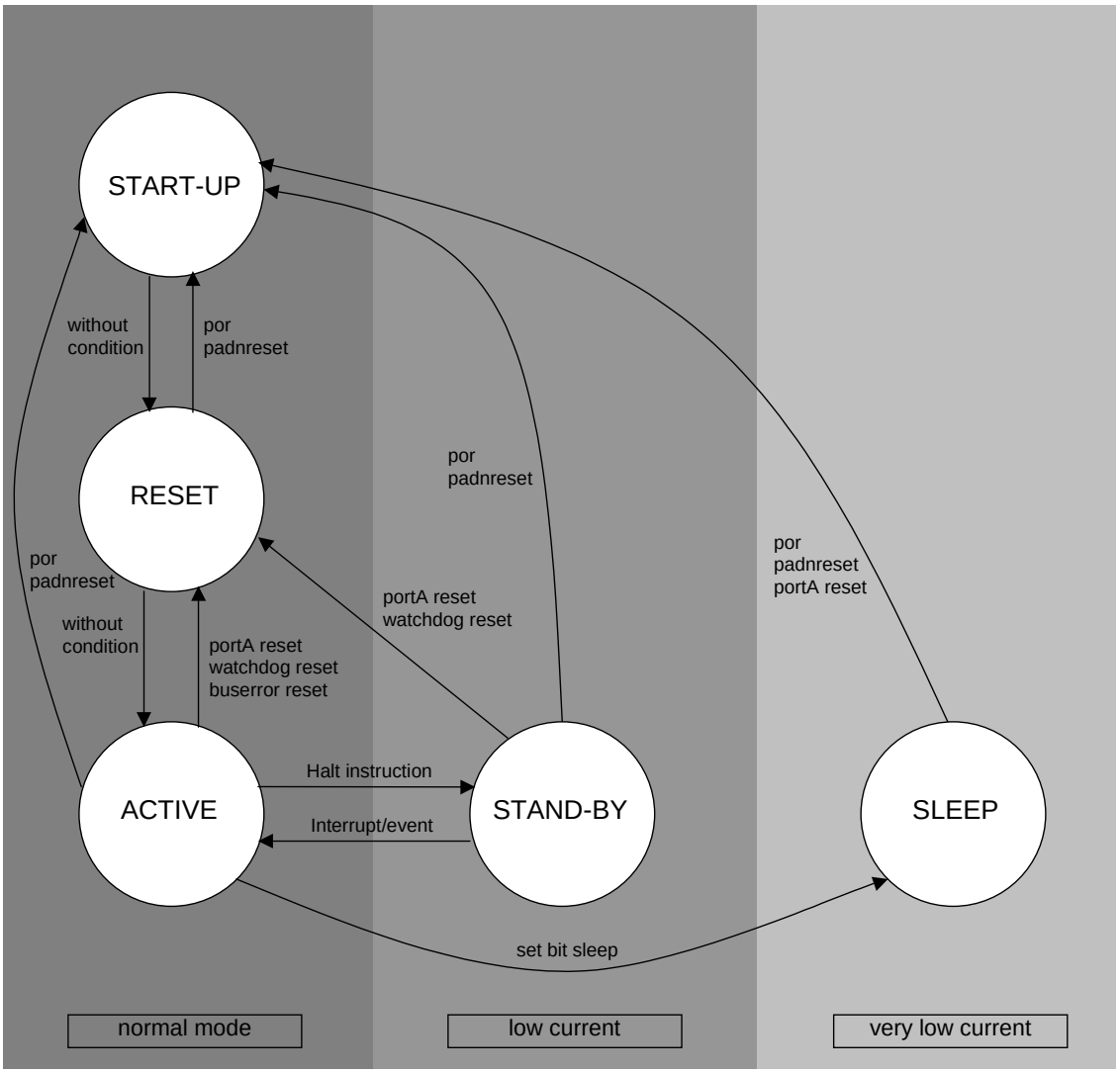

Figure 5-1. XE8000 operating modes.

7.1 Features

3 available clock sources (RC oscillator, quartz oscillator and external clock).
- 2 divider chains: high-prescaler (8 bits) and low-prescaler (15 bits).
- CPU clock disabling in halt mode.

7.2 Overview

The XE88LCxx chips can work on different clock sources (RC oscillator, quartz oscillator and external clock). The clock generator block is in charge of distributing the necessary clock frequencies to the circuit.

Figure 7-1 represents the functionality of the clock block.

The internal RC oscillator or an external clock source can be selected to drive the high prescaler. This prescaler generates frequency divisions down to 1/256 of its input frequency. A 32kHz clock is generated by enabling the quartz oscillator (if present in the product) or by selecting the appropriate tap on the high prescaler. The low prescaler generates clock signals from 32kHz down to 1Hz. The clock source for the CPU can be selected from the RC oscillator, the external clock or the 32kHz clock.

7.3 Register map

pos.	RegSysClock	rw	reset	function
7	CpuSel	r/w	0 nresetsleep	Select speed for cpuck
6	-	r	0	Unused
5	EnExtClock	r/w	0 nresetcold	Enable for external clock
4	BiasRc	r/w	1 nresetcold	Enable Rcbias (reduces start-up time of RC).
3	ColdXtal	r	1 nresetsleep	Xtal in start phase
2	-	r	0	Unused
1	EnableXtal	r/w	0 nresetsleep	Enable Xtal oscillator
0	EnableRc	r/w	1 nresetsleep	Enable RC oscillator

Table 7-1: **RegSysClock** register

pos.	RegSysMisc	rw	reset	function
7-2	--	r	000000	Unused
1	Output16k	r/w	0 nresetsleep	Output 16 kHz signal on PB[3]
0	OutputCpuCk	r/w	0 nresetsleep	Output CPU clock on PB[2]

Table 7-2: **RegSysMisc** register

pos.	RegSysPre0	rw	reset	function
7-1	--	r	0000000	Unused
0	ClearLowPrescal	w1 r0	0	Write 1 to reset low prescaler, but always reads 0

Table 7-3: **RegSysPre0** register

8.1 Features

The XE8000 chips support 24 interrupt sources, divided into 3 levels of priority.

8.2 Overview

The interrupt handler allows 24 interrupt sources to be managed individually.

The 24 interrupt sources are divided into 3 levels of priority: High (8 interrupt sources), Mid (8 interrupt sources), and Low (8 interrupt sources). Those 3 levels of priority are directly mapped to those supported by the CoolRisc® (IN0, IN1 and IN2; see CoolRisc documentation for more information).

Additional functions are given that allow fast detection of the highest priority interrupt that has been activated.

8.3 Register map

Register name
RegIrqHig
RegIrqMid
RegIrqLow
RegIrqEnHig
RegIrqEnMid
RegIrqEnLow
RegIrqPriority
RegIrqIrq

Table 8-1: IRQ handler registers

pos.	RegIrqHig	rw	reset	function
7	RegIrqHig[7]	r c1	0 nresetglobal	interrupt #23 (high priority) clear interrupt #23 when 1 is written
6	RegIrqHig[6]	r c1	0 nresetglobal	interrupt #22 (high priority) clear interrupt #22 when 1 is written
5	RegIrqHig[5]	r c1	0 nresetglobal	interrupt #21 (high priority) clear interrupt #21 when 1 is written
4	RegIrqHig[4]	r c1	0 nresetglobal	interrupt #20 (high priority) clear interrupt #20 when 1 is written
3	RegIrqHig[3]	r c1	0 nresetglobal	interrupt #19 (high priority) clear interrupt #19 when 1 is written
2	RegIrqHig[2]	r c1	0 nresetglobal	interrupt #18 (high priority) clear interrupt #18 when 1 is written
1	RegIrqHig[1]	r c1	0 nresetglobal	interrupt #17 (high priority) clear interrupt #17 when 1 is written
0	RegIrqHig[0]	r c1	0 nresetglobal	interrupt #16 (high priority) clear interrupt #16 when 1 is written

Table 8-2: **RegIrqHig**

10. Low Power RAM

pos.	RegPACtrl	rw	reset	description
7:1	[7:1]	r	0000000	Unused
0	DebFast	r w	0 nresetpconf	0 = slow debounce, 1 = fastdebounce

Table 11-8: RegPACtrl

pos.	RegPASnaptorail	rw	reset	description
7:0	PASnaptorail[7:0]	rw	0 nresetpconf	set snap-to-rail input on

Table 11-9: RegPASnaptorail

Note*:* Depending on the status of the **EnResetPConf** bit in **RegSysCtrl**, **RegPAEdge, RegPADebounce** and **RegPACtrl** can be reset by any of the possible system resets or only with power-on reset and NRESET pad.

11.4 Interrupts and events map

Interrupt source	Default mapping in the interrupt manager	Default mapping in the event manager
pa_irqbus[5]	RegIrqMid[5]	
pa_irqbus[4]	RegIrqMid[4]	
pa_irqbus[1]	RegIrqMid[1]	RegEvn[4]
pa_irqbus[0]	RegIrqMid[0]	RegEvn[0]
pa_irqbus[7]	RegIrqLow[7]	
pa_irqbus[6]	RegIrqLow[6]	
pa_irqbus[3]	RegIrqLow[3]	
pa_irqbus[2]	RegIrqLow[2]	

11.5 Port A (PA) Operation

The Port A input status (debounced or not) can be read from **RegPAin**.

Debounce mode:
Each bit in Port A can be individually debounced by setting the corresponding bit in **RegPADebounce**. After reset, the debounce function is disabled. After enabling the debouncer, the change of the input value is accepted only if height consecutive samples are identical. Selection of the clock is done by bit **DebFast** in Register **RegPACtrl**.

DebFast	Clock filter
0	1kHz
1	32kHz

Table 11-10**: debounce frequency selection**

Note: The tolerance on the debounce frequency depends on the selected clock source. When the external clock is used, the pulse width will be correct if the input of the low prescaler is set to a frequency close to 32kHz (see clock block documentation).

Pullups/Snap-to-rail:
Different functions are possible depending on the value of the registers **RegPAPullup** and **RegPASnaptorail**. When the corresponding bit in **RegPAPullup** is set to 0, the inputs are floating (pullup and pulldown resistors are disconnected). When the corresponding bit in **RegPAPullup** is 1 and in **RegPASnaptorail** is 0, a pullup resistor is connected to the input pin. Finally, when the corresponding bit in **RegPAPullup** is 1 and in **RegPASnaptorail** is 1, the snap-to-rail function is active.

Note: depending of the value of PA[0] to PA[7], changes to **RegPARes0** and **RegPARes1** can cause a reset. Therefore it is safe to have always one (RegPARes0[x], RegPARes1[x]) equal to '00' during the setting operations.

11.6 Port A electrical specification

sym	description	min	typ	max	unit	Comments
V_{INH}	Input high voltage	0.7*VBAT		VBAT	V	VBAT≥2.4V
V_{INL}	Input low voltage	VSS		0.2*VBAT	V	VBAT≥2.4V
R_{PU}	Pull-up resistance	20	50	80	kΩ	
Cin	Input capacitance		2.5		pF	Note 1

Note 1: this value is indicative only since it depends on the package.

Table 11-13. Electrical specification

Pos.	RegPDIn	Rw	Reset	Description
7:0	PDIn[7:0]	r	-	pad PD[7:0] input value

Table 13-2: RegPDIn

Pos.	RegPDOut	Rw	Reset	Description
7:0	PDOut[7:0]	r w	0 nresetpconf	pad PD[7:0] output value

Table 13-3: RegPDOut

Pos.	RegPDDir	Rw	Reset	Description
7:0	PDDir[7:0]	r w	0 nresetpconf	pad PD[7:0] direction (0=input)

Table 13-4: RegPDDir

Pos.	RegPDPullup	Rw	Reset	Description
7	PDSnapToRail[3]	r w	1 nresetpconf	snap-to-rail for pad PD[7] and PD[6] (1=active)
6	PDSnapToRail[2]	r w	1 nresetpconf	snap-to-rail for pad PD[5] and PD[4] (1=active)
5	PDSnapToRail[1]	r w	1 nresetpconf	snap-to-rail for pad PD[3] and PD[2] (1=active)
4	PDSnapToRail[0]	r w	1 nresetpconf	snap-to-rail for pad PD[1] and PD[0] (1=active)
3	PDPullup[3]	r w	1 nresetpconf	pullup for pad PD[7] and PD[6] (1=active)
2	PDPullup[2]	r w	1 nresetpconf	pullup for pad PD[5] and PD[4] (1=active)
1	PDPullup[1]	r w	1 nresetpconf	pullup for pad PD[3] and PD[2] (1=active)
0	PDPullup[0]	r w	1 nresetpconf	pullup for pad PD[1] and PD[0] (1=active)

Table 13-5: RegPDPullup

13.4 Port D (PD) Operation

The direction of each pin of Port D (input or input/output) can be individually set by using the **RegPDDir** register. If **PDDir[x]** = 1, the output buffer on the corresponding Port D pin is enabled. After reset, Port D is in input only mode (**PDDir[x]** are reset to 0). The input buffer is always enabled independently from the **RegPDDir** contents.

Output data:
Data are stored in **RegPDOut** prior to output at Port D.

Input data:
The status of Port D is available in **RegPDIn** (read only). Reading is always direct - there is no digital debounce function associated with Port D. In case of possible noise on input signals, a software debouncer or an external filter must be realised.

Pull-up/Snap to Rail:
When configured as an input (**PDDir[x]**=0), pull-ups are available on every pin. The pull-up function of the pins is controlled two by two by the **PDPullup** and **PDSnapToRail** bits in the register **RegPDPullup**. When a bit

pos.	RegRfifCmd3	rw	reset	description
7-5	RfifRxIrqEn	r/w	000 nresetglobal	'1' enable the Rx Irq sources (see page 14-11)
4-2	RfifRxIrqMem	r/c1	000 nresetglobal	Rx Irq Status (see page 14-11)
1	RfifEnRx	r/w	0 nresetpconf	'1' enable Rfif reception mode
0	RfifEnTx	r/w	0 nresetpconf	'1' enable Rfif transmission mode

Table 14-4: RegRfifCmd3

pos.	RegRfifTx	rw	reset	description
7-0	RfifTx	w	00000000 nresetglobal	Data to be sent FIFO, depth = 4

Table 14-5: RegRfifTx

pos.	RegRfifTxSta	rw	reset	description
7-4	-	-	-	-
3	RfifTxFifoOverrun	r/c1	0 nresetglobal	Tx Fifo overrun error. Write: clear FIFO
2	RfifTxFifoFull	r	0 nresetglobal	Tx Fifo full
1	RfifTxFifoEmpty	r	1 nresetglobal	Tx Fifo empty [IRQ]
0	RfifTxStopped	r	0 nresetglobal	Tx Fifo + Tx shift register empty: stopped

Table 14-6: RegRfifTxSta

pos.	RegRfifRx	rw	reset	description
7-0	RfifRx	r	00000000 nresetglobal	Received data FIFO, depth = 4

Table 14-7: RegRfifRx

pos.	RegRfifRxSta	rw	reset	description
7-5	-	-	-	-
4	RfifRxFifoOverrun	r/c1	0 nresetglobal	Rx Fifo overrun error. Write: clear FIFO
3	RfifRxFifoFull	r	0 nresetglobal	Rx Fifo full [IRQ]
2	RfifRxStartDet	r/c1	0 nresetglobal	Start Stream Detected [IRQ]
1	RfifRxBusy	r	0 nresetglobal	Rfif busy receiving
0	RfifRxReady = RfifRxNotEmpty	r	0 nresetglobal	**RegRfifRx** contains at least one byte to read. Cleared when reading **RegRfifRx**.[IRQ]

Table 14-8: RegRfifRxSta

pos.	RegRfifRxSPat	rw	reset	description
7-0	RfifRxSPat	r/w	00000000 nresetglobal	Programmable start pattern

Table 14-9: RegRfifRxSPat

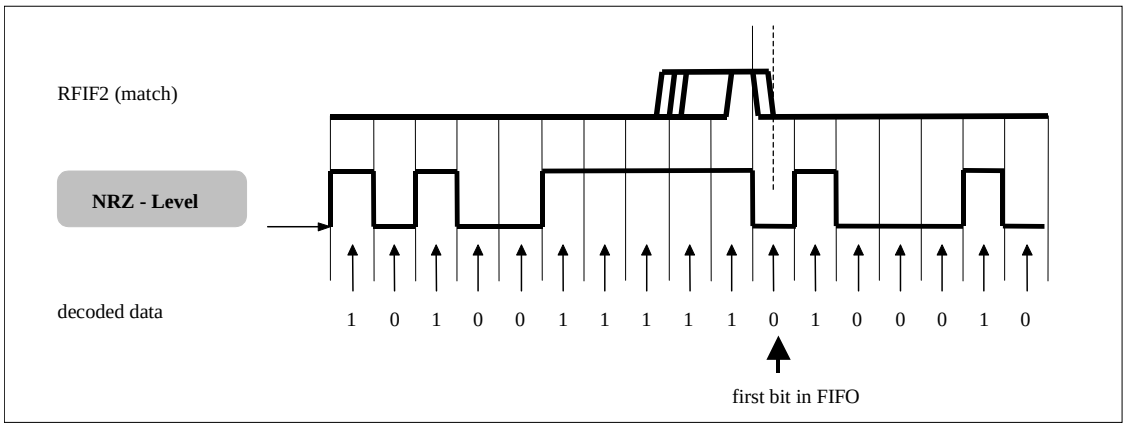

Figure 14-4. Timing of the external word synchronization signal

14.8.2.4 "Pattern" start detection

The start sequence is defined by the pattern **RfifRxSPat[7:0]** written in the register **RegFifRxSPat**. This pattern is compared to the incoming bit stream before decoding. This is shown in Figure 14-5 and Figure 14-6 where an identical input data bit stream and an identical pattern give the same start sequence detection independent from the decoded data.

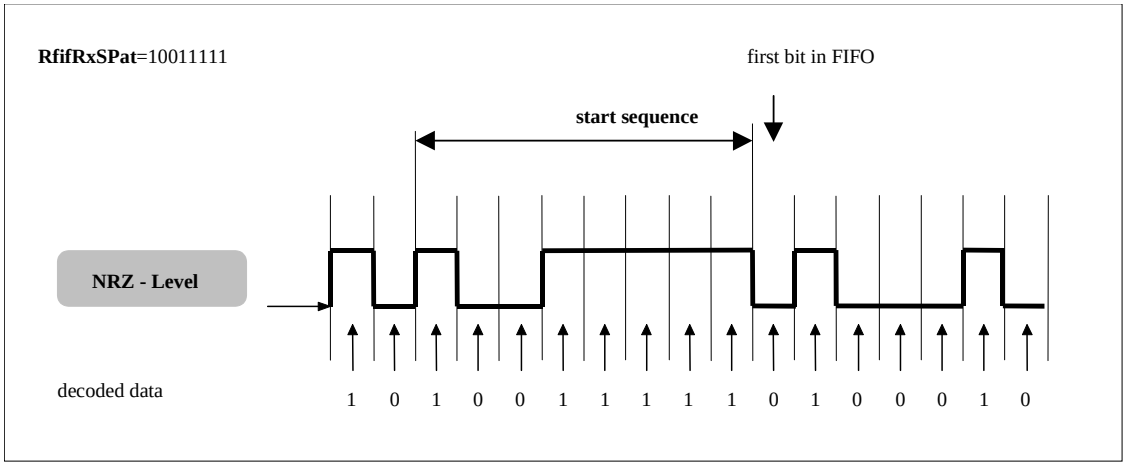

Figure 14-5. Pattern start detection in NRZ-Level for RfifRxSPat[7:0]=10011111

If **RfifEnCod** in register **RegRfifCmd2** is set to 1, the bit stream coming from the shift register is encoded first before sending it to the RFIF3 pin. The type of protocol is selected using the **RfifPCM[2:0]** control word in the register **RegRfifCmd2**. The selection control bits are given in Table 14-14.

PCM Codes	NRZ			Bi-Phase/Manchester			Miller
	Level	Mark	Space	Level	Mark	Space	
RfifPCM[2:0]	000	001	010	011	100	101	11X

Table 14-16: PCM code selection

When the bit **RfifEnCod** is modified while a transmission is active, the modification will take effect only when a new byte is loaded from the FIFO into the shift register.

While the encoder is enabled, it is not possible to send a protocol violation as a start pattern. If a protocol violation is used as a start sequence in the Manchester or Miller protocol, the following sequence should be used:

1) wait until the transmission FIFO is empty (which means the last byte of the preceding message is being coded and sent),
2) clear **RfifEnCod**,
3) write the uncoded start pattern to the transmission FIFO
4) wait until the transmission FIFO is empty (which means the last byte of the uncoded start pattern is being sent)
5) set **RfifEnCod**
6) write the message bytes to the FIFO
7) back to 1.

Beware that in the case a protocol violation is used as a start sequence in Manchester-level or Miller coding, the used violation will depend on the value of the first bit of the message. As an example: if the Manchester level coding is used and the first bit of the message is a 0 (encoded 01), the start sequence needs to be a series of 1's. If a series of 0's would be used, the receiver would be unable to distinguish the leading 0 of the message from the start sequence.

When the encoder is bypassed (**RfifEnCod** = 0), bits **RfifPCM[2:0]** still have an influence on the output bit stream : when the NRZ coding is chosen, the bits are shifted out at the selected bit rate. If the Manchester or Miller codes are selected however, the bits are shifted out at twice the data rate. This allows the use of two bits in the FIFO to encode the bits by software.

14.9.4 Transmission synchronization clock

If required, a bit synchronization clock can be generated on the pin RFIF2 if the data is uncoded or if the NRZ coding is used (bits **RfifEnCod** and **RfifPCM[2:0]** in **RegRfifCmd2**). To generate this clock, the bit **RfifTxClock** in **RegRfifCmd2** has to be set. The timing of the generated clock is shown in Figure 14-7.

14.12.2.5 Transmission mode using NRZ coding

This RF link uses a wide band transmission in the 915MHz frequency band. The protocol of the messages to be sent is NRZ Level. The start sequence of the message is a 4 byte pattern (10101010 10101010 10101010 11010100). The messages have a data rate of 76.8 kbit/s with a frequency deviation of 100 kHz. The output power is 0 dBm. The following paragraphs will show how to set-up the XE1202A, how to set-up the RF interface and how to handle the received data.

14.12.2.5.1 XE1202A set-up

To set the XE1202A in transmission mode, set the pins MODE(2:0) = 111.

To select the 915MHz band, the bits FSParam_band = 11. The frequency deviation is set to 100kHz by writing RTParam_Dev = 11. The baud rate is set to 76.8 kbit/s by writing FSParam_BR = 100. The output power is set to 0 dBm by writing RTParam_Tpow = 00. The setup of the XE1202a is shown in Table 14-31.

Register	Register Address	7	6	5	4	3	2	1	0
RTParam1	00000	0	0	0	0	0	0	0	0
RTParam2	00001	0	1	0	0	0	0	0	0
FSParam1	00010	1	1	1	0	0	1	0	0
FSParam2	00011	0	0	0	0	0	0	0	0
FSParam3	00100	0	0	0	0	0	0	0	0
DataOut	00101	0	0	0	0	0	0	0	0
ADParam1	00110	0	0	0	0	0	0	0	0
ADParam2	00111	0	0	0	0	0	0	0	0
Pattern1	01000	0	0	0	0	0	0	0	0
Pattern2	01001	0	0	0	0	0	0	0	0
Pattern3	01010	0	0	0	0	0	0	0	0
Pattern4	01011	0	0	0	0	0	0	0	0

Table 14-31. XE1202A register set-up (see XE1202A datasheet for bit explanation)

14.12.2.5.2 RF interface set-up

We then set-up the RF interface of the microcontroller circuit as a transmitter (**RfifEnRx** = 0 and **RfifEnTx** = 1).

Assume that the RC clock frequency used in the microcontroller is 1.2 MHz. To select the correct baud rate of 76.8 kbit/s according to the equation in chapter 14.10, *fine*coarse*=1.2e06/(16*76.8e3)=0.98. We can approximate this at 1 (see specification in Table 14-19). This can be done by setting **RfifBRCoarse** = 00 and **RfifBRFine** = 0000.

The external bit synchronization clock is switched off by clearing the bit **RfifTxClock** = 0.

The encoder is enabled and set to NRZ Level encoding by setting **RfifEnCod** = 1 and **RfifPCM** = 000.

The set-up of the interface is summarized in the Table 14-32.

Register	contents
RegRfifCmd1	00000000
RegRfifCmd2	00100010
RegRfifCmd3	00000001

Table 14-32. RF interface set-up

15.1 Features

- Full duplex operation with buffered receiver and transmitter.
- Internal baudrate generator with 10 programmable baudrates (300 - 153600).
- 7 or 8 bits word length.
- Even, odd, or no-parity bit generation and detection
- 1 stop bit
- Error receive detection: Start, Parity, Frame and Overrun
- Receiver echo mode
- 2 interrupts (receive full and transmit empty)
- Enable receive and/or transmit
- Invert pad Rx and/or Tx

15.2 Overview

The UART pins are PB[7], which is used as Rx - receive and PB[6] as Tx - transmit.

15.3 Registers map

register name
RegUartCtrl
RegUartCmd
RegUartTx
RegUartTxSta
RegUartRx
RegUartRxSta

Table 15-1: Uart register default addresses

pos.	RegUartCmd	rw	reset	description
7	SelXtal	r/w	0 nresetglobal	Select input clock: 0 = RC/external, 1 = xtal
6	-	r	0	Unused
5-3	UartRcSel(2:0)	r/w	000 nresetglobal	RC prescaler selection
2	UartPM	r/w	0 nresetglobal	Select parity mode: 1 = odd, 0 = even
1	UartPE	r/w	0 nresetglobal	Enable parity: 1 = with parity, 0 = no parity
0	UartWL	r/w	1 nresetglobal	Select word length: 1 = 8 bits, 0 = 7 bits

Table 15-2: RegUartCmd

pos.	RegUartCtrl	rw	reset	description
7	UartEcho	r/w	0 nresetglobal	Enable echo mode: 1 = echo Rx->Tx, 0 = no echo
6	UartEnRx	r/w	0 nresetglobal	Enable uart reception
5	UartEnTx	r/w	0 nresetglobal	Enable uart transmission
4	UartXRx	r/w	0 nresetglobal	Invert pad Rx
3	UartXTx	r/w	0 nresetglobal	Invert pad Tx
2-0	UartBR(2:0)	r/w	101 nresetglobal	Select baud rate

Table 15-3: RegUartCtrl

The cmpd peripheral is a 4-channel low power comparator. It is intended to compare analog input signals with an internally set threshold voltage. The comparator maintains low current consumption even if the input signal is very close to the threshold. The comparison result of each channel can be used to generate an interrupt and/or is available for polling.

The comparator can be enabled or disabled by programming the **Enable** bit in the **RegCmpdCtrl** register. When disabled, the block consumes no current.

The peripheral has a single interrupt output which is a combination of the four channels. The combination can be chosen by programming the **RegCmpdCtrl** register. The **EnIrqCh[3:0]** bits select the channel that can activate the interrupt. The **IrqOnRisingCh[2:0]** bits indicate if the interrupt is generated on detection of the rising or falling edge of the channel.

The comparison results of the peripheral can be read in the **RegCmpdStat** register. The bits **CmpdOut[3:0]** are the value of the comparisons at the moment the register is read. The **CmpdStat[3:0]** indicates which channel generated an interrupt since the register was last read.

Comparator specifications:

Sym	description	min	typ	max	unit	comments
t_{pulse}	Required input pulse width	500			ns	VBAT≥1.2V
IDD_q	Quiescent current		0.8	1.5	µA	1
IDD_{stat}	Maximal static current		1.5		µA	2
V_{th}	Threshold voltage	0.7		1.1	V	3
$\Delta V_{th}/\Delta T$	Threshold temperature drift		-0.9		mV/°C	
V_{hyst}	Threshold hysteresis		13		mV	

Table 19-1: Comparator specifications

Comments:
1. The quiescent current is defined for a static input voltage <0.5V or >1.3V. The specified consumption is the sum for all 4 channels.
2. The maximal static current is defined for any static input voltage between VDD and VSS. The specified consumption is the sum for all 4 channels.
3. Defined with respect to VSS.

How to start the cmpd:
To avoid unwanted irqs one has first to configure the rising / falling edge of the detection (bit IrqOnRisingCh[2:0]) and to enable the comparator (bit Enable). Only after that may the user enable the channel interrupts with bit EnIrqCh[3:0].

19.3 Register map

There are two registers in the Cmpd, namely **RegCmpdStat** and **RegCmpdCtrl**. Table 19-3 and Table 19-4 show the mapping of the control bits and the functionality of these registers.

register name
RegCmpdStat
RegCmpdCtrl

Table 19-2: Cmpd registers

Contact Information

Semtech Corporation
Wireless and Sensing Products Division
200 Flynn Road, Camarillo, CA 93012
Phone (805) 498-2111 Fax : (805) 498-3804