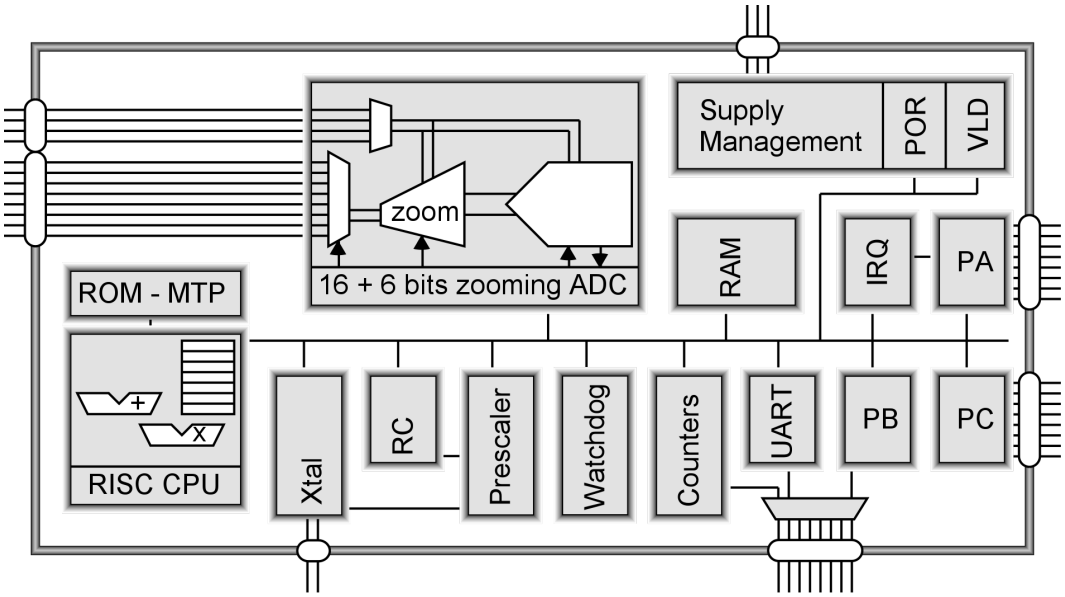


XE88LC01

Ultra Low-Power Microcontroller with ZoomingADC

General Description

The XE88LC01 is an ultra low-power low-voltage microcontroller unit (MCU) with extremely high efficiency, allowing for 1 MIPS at 300uA and 2.4 V, and 8 x 8 bits multiplying in one clock cycle.

XE88LC01 is available with on chip Multiple-Time-Programmable (MTP) program memory.

Applications

- Internet connected appliances
- Portable, battery operated instruments
- RF system supervisor
- Remote control
- Piezoresistive bridge sensors
- HVAC control
- Motor control

Key product Features

- Ultra low-power MCU (300 µA at 1 MIPS)
- Low-voltage operation (2.4 - 5.5 V supply voltage)
- 22 kB (8 kW) MTP, 512 B RAM
- 4 counters
- PWM, UART
- Low-power, high resolution zooming
  - up to 10 bits zoom
  - up to 16 bits ADC
- 4 x 2 or 7 x 1 PGA-ADC input multiplexer
- Analog matrix switching
- RC and crystal oscillators
- 5 reset, 16 interrupt, 8 event sources

Ordering Information

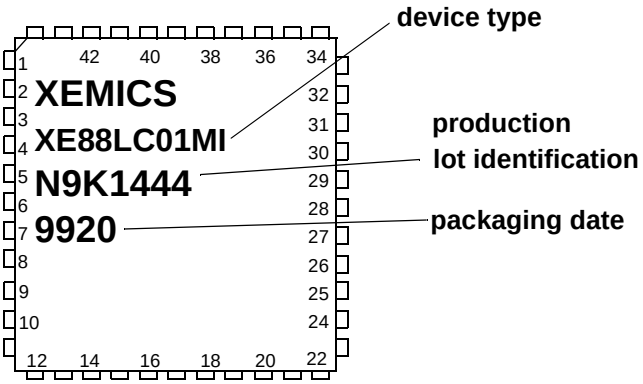
Nomenclature: (XX stands for pre-production samples)

XX-XE88LC01ME027

- program memory: M: MTP
- temperature: I: -40 to 85 °C
- package: TQFP44

XEMICS SA, Switzerland. Tel: +41 32 720 51 70 Fax: +41 32 720 57 70 e-mail: info@xemics.com web: www.xemics.com

Detailed Pin Description



Pinout of the XE88LC01 in TQFP44 package

Pin				Description
Position in TQFP44	Function name	Second function name	Type	
1	PA(5)		Input	Input of Port A
2	PA(6)		Input	Input of Port A
3	PA(7)		Input	Input of Port A
4	PC(0)		Input/Output	Input-Output of Port C
5	PC(1)		Input/Output	Input-Output of Port C
6	PC(2)		Input/Output	Input-Output of Port C
7	PC(3)		Input/Output	Input-Output of Port C
8	PC(4)		Input/Output	Input-Output of Port C
9	PC(5)		Input/Output	Input-Output of Port C
10	PC(6)		Input/Output	Input-Output of Port C
11	PC(7)		Input/Output	Input-Output of Port C
12	PB(0)	testout	Input/Output/Analog	Input-Output-Analog of Port B/ Data output for test and MTP programing/ PWM output
13	PB(1)		Input/Output/Analog	Input-Output-Analog of Port B/ PWM output
14	PB(2)		Input/Output/Analog	Input-Output-Analog of Port B
15	PB(3)	SOU	Input/Output/Analog	Input-Output-Analog of Port B, Output pin of USRT
16	PB(4)	SCL	Input/Output/Analog	Input-Output-Analog of Port B/ Clock pin of USRT
17	PB(5)	SIN	Input/Output/Analog	Input-Output-Analog of Port B/ Data input or input-output pin of USRT
18	PB(6)	Tx	Input/Output/Analog	Input-Output-Analog of Port B/ Emission pin of UART
19	PB(7)	Rx	Input/Output/Analog	Input-Output-Analog of Port B/ Reception pin of UART
20	VPP/TEST	Vhigh	Special	Test mode/High voltage for MTP programing
21	AC_R(3)		Analog	Highest potential node for 2nd reference of ADC
22	AC_R(2)		Analog	Lowest potential node for 2nd reference of ADC
23	AC_A(7)		Analog	ADC input node
24	AC_A(6)		Analog	ADC input node

Pin-out of the XX-XE88LC01 in TQFP44

Pin				Description
Position in TQFP44	Function name	Second function name	Type	
25	AC_A(5)		Analog	ADC input node
26	AC_A(4)		Analog	ADC input node
27	AC_A(3)		Analog	ADC input node
28	AC_A(2)		Analog	ADC input node
29	AC_A(1)		Analog	ADC input node
30	AC_A(0)		Analog	ADC input node
31	AC_R(1)		Analog	Highest potential node for 1st reference of ADC
32	AC_R(0)		Analog	Lowest potential node for 1st reference of ADC
33	VSS		Power	Negative power supply, connected to substrate
34	Vbat		Power	Positive power supply
35	Vreg		Analog	Regulated supply
36	RESET		Input	Reset pin (active high)
37	Vmult		Analog	Pad for optional voltage multiplier capacitor
38	OscIn	ck_cr	Analog/Input	Connection to Xtal/ CoolRISC clock for test and MTP programing
39	OscOut	ptck	Analog/Input	Connection to Xtal/ Peripheral clock for test and MTP programing
40	PA(0)	testin	Input	Input of Port A/ Data input for test and MTP programing/ Counter A input
41	PA(1)	testck	Input	Input of Port A/ Data clock for test and MTP programing/ Counter B input
42	PA(2)		Input	Input of Port A/ Counter C input/ Counter capture input
43	PA(3)		Input	Input of Port A/ Counter D input/ Counter capture input
44	PA(4)		Input	Input of Port A

Pin-out of the XX-XE88LC01 in TQFP44

XE88LC01xI Electrical Characteristics

Operation conditions		min	typ	max	Unit	Remarks
Power supply	ROM version	2.4		5.5	V	
	MTP version	2.4		5.5	V	
Current requirement	CPU running at 1 MIPS			310	uA	1
	CPU running at 32 kHz on Xtal, RC off			10	uA	
	CPU halt, timer on Xtal, RC off			1	uA	
	CPU halt, timer on Xtal, RC ready			1.7	uA	
	CPU halt, Xtal off timer on RC at 100 kHz			1.4	uA	
	CPU halt, ADC 12 bits at 4 kHz			200	uA	1
	CPU halt, ADC 12 bits at 4 kHz, PGA gain 100			250	uA	1
	Voltage level detection			15	uA	
MTP	Prog. voltage	10.3		10.8	V	
	Erase time		3	30	s	
	Write/Erase cycles	10	100			
	Data retention	10			year	

Current requirement of the XE88LC01

Note: 1)Power supply: 2.4 V - 5.5 V, at 27°C; min voltage of XX version may be higher.

CPU

The XE88LC01 CPU is a low power RISC core. It has 16 internal registers for efficient implementation of the C compiler. Its instruction set is made of 35 generic instructions, all coded on 22 bits, with 8 addressing modes. All instructions are executed in one clock cycle, including conditional jumps and 8x8 multiplication, therefore the XE88LC01 runs at 1 MIPS on a 1 MHz clock.

A complete tool suite for development is available from XEMICS, including programmer, C-compiler, assembler, simulator, linker, all integrated in a modern and efficient graphical user interface.

NAME	Parameters	res	op1	op2	FUNCTION	MODIF.
JUMP	addr:16				PC0 <- addr	- , - , - , -
	ip				PC0 <- ip	
Jcc	addr:16				if cc then PC0 <- addr	
	ip				if cc then PC0 <- ip	
CALL	addr:16				PCn <- PCn-1 (n>1), PC1 <- PC0+1, PC0 <- addr	
	ip				PCn <- PCn-1 (n>1), PC1 <- PC0+1, PC0 <- ip	
CALLS	addr:16				ip <- PC0+1, PC0 <- addr:16	
	ip				ip <- PC0+1, PC0 <- ip	
RET					PCn-1 (n>0) <- PCn	- , - , - , -
RETS					PC0 <- ip	
RETI					PCn-1 (n>0) <- PCn, GIE <- 1	
PUSH					PCn <- PCn-1 (n>1), PC1 <- ip, PC0 <- PC0+1	
POP					ip <- PC1, PCn-1 (n>1) <- PCn, PC0 <- PC0+1	
MOVE	reg, data:8	reg	data		res <- op1	- , - , Z , a
	reg1, reg2	reg1	reg2			
	reg, eaddr	reg	eaddr			
	eaddr, reg	eaddr	reg			- , - , - , -
	addr:8, data:8	addr	data			
CMVD	reg1, reg2	reg1	reg2		if C=0 then res <- op1	- , - , Z , a
CMVS	reg, eaddr	reg	eaddr		if C=1 then res <- op1	
SHL	reg1, reg2	reg1	reg2		res(bitn) <- op1(bitn-1) (0<n<8), res(0) <- 0, C <- op1(7)	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
SHLC	reg1, reg2	reg1	reg2		res(bitn) <- op1(bitn-1) (0<n<8), res(0) <- C, C <- op1(7)	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
SHR	reg1, reg2	reg1	reg2		res(bitn-1) <- op1(bitn) (0<n<8), res(7) <- 0, C <- op1(0)	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
SHRC	reg1, reg2	reg1	reg2		res(bitn-1) <- op1(bitn) (0<n<8), res(7) <- C, C <- op1(0)	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
SHRA	reg1, reg2	reg1	reg2		res(bitn-1) <- op1(bitn) (0<n<8), res(7) <- op1(7), C <- op1(0)	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
CPL1	reg1, reg2	reg1	reg2		res <- NOT (op1)	-, -, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
CPL2	reg1, reg2	reg1	reg2		res <- NOT (op1) +1, if op1 = 0 then C = 1	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
CPL2C	reg1, reg2	reg1	reg2		res <- NOT (op1) +C, if op1 = 0 then C = 1	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
INC	reg1, reg2	reg1	reg2		res <- op1 +1, if overflow then C = 1	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
INCC	reg1, reg2	reg1	reg2		res <- op1 +C, if overflow then C = 1	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
DEC	reg1, reg2	reg1	reg2		res <- op1 -1, if underflow then C = 0	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			
DECC	reg1, reg2	reg1	reg2		res <- op1 -(1 -C), if underflow then C = 0	C, V, Z, a
	reg	reg	reg			
	reg, eaddr	reg	eaddr			

Table 1.2: XE8000 Instruction Set

NAME	Parameters	res	op1	op2	FUNCTION	MODIF.
AND	reg, data:8	reg	reg	data	res <- op1 AND op2	-, -, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
OR	reg, data:8	reg	reg	data	res <- op1 OR op2	-, -, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
XOR	reg, data:8	reg	reg	data	res <- op1 XOR op2	-, -, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
ADD	reg, data:8	reg	reg	data	res <- op1 + op2, if overflow then C=1	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
ADDC	reg, data:8	reg	reg	data	res <- op1 + op2 + C, if overflow then C=1	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
SUBD	reg, data:8	reg	reg	data	res <- op1 -op2, if underflow then C=0	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
SUBDC	reg, data:8	reg	reg	data	res <- op1 -op2 - (1-C), if underflow then C=0	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
SUBS	reg, data:8	reg	reg	data	res <- op2 -op1, if underflow then C=0	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
SUBSC	reg, data:8	reg	reg	data	res <- op2 -op1 - (1-C), if underflow then C=0	C, V, Z, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
MUL	reg, data:8	reg	reg	data	res <- op1 * op2 (15:8), a <- op1 * op2 (7:0), unsigned	-, -, -, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
MULA	reg, data:8	reg	reg	data	res <- op1 * op2 (15:8), a <- op1 * op2 (7:0), signed (2 complement)	-, -, -, a
	reg1, reg2, reg3	reg1	reg2	reg3		
	reg1, reg2	reg1	reg2	reg1		
	reg	reg	reg	eaddr		
MSHL	reg, shift:3				a(bitn) <- reg(bitn-shift) for (bitn >= shift), reg(bitn) <- reg (bitn+8-shift) for (bitn < shift)	-, -, -, a
MSHR	reg, shift:3				reg(bitn) <- reg(bitn+shift) for (bitn + shift < 8), a(bitn) <- reg (bitn-8+shift) for (bitn + shift >= 8)	-, -, -, a
MSHRA	reg, shift:3				a <- SHRA(shift,reg), a <- SHL(8-shift,reg), SHRA propagates sign, do not use with shift=0x01	-, -, -, a
CMP	reg, data:8		reg	data	if op2 > op1 then C <- 0, V = C AND NOT(Z), unsigned	C, V, Z, a
	reg1, reg2		reg1	reg2		
	reg, eaddr		reg	eaddr		
CMPA	reg, data:8		reg	data	if op2 > op1 then C <- 0, V = C AND NOT(Z), signed	C, V, Z, a
	reg1, reg2		reg1	reg2		
	reg, eaddr		reg	eaddr		
TSTB	reg, bit:3				Z <- NOT(reg(bit))	-, -, Z, a
SETB	reg, bit:3				reg(bit) <- 1	-, -, Z, a
CLRB	reg, bit:3				reg(bit) <- 0	-, -, Z, a

Table 1.2: XE8000 Instruction Set

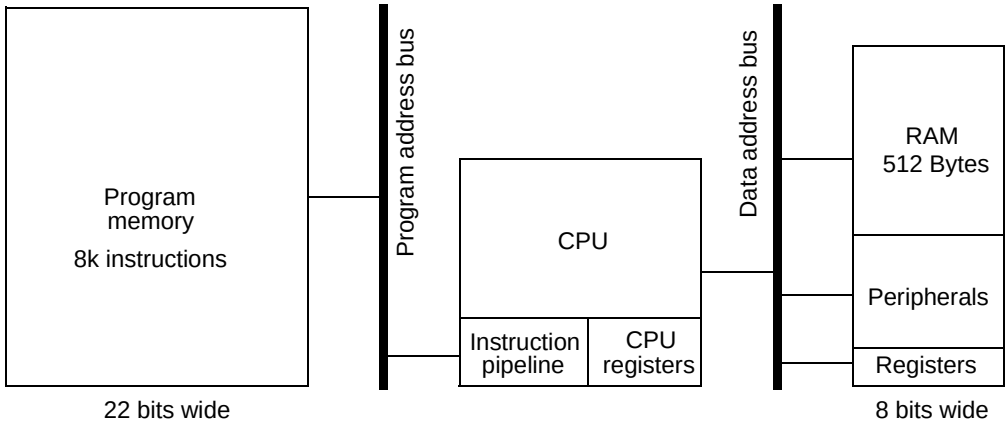
NAME	Parameters	res	op1	op2	FUNCTION	MODIF.
INVB	reg, bit:3				reg(bit) <- NOT(reg(bit))	-, -, Z, a
SFLAG					a(7) <- C, a(6) <- C XOR V	-, -, -, a
RFLAG	reg		reg		flags <- op1, SHL op1, SHL a	C, V, Z, a
	eaddr		eaddr			
FREQ	divn:4				set cpu frequency divider	-, -, -, -
HALT					stops CPU	-, -, -, -
NOP					no operation	-, -, -, -
PMD	s:1				if s=1 then starts program dump, if s=0 stops program dump	-, -, -, -

Table 1.2: XE8000 Instruction Set

Memory organisation

The CPU uses a Harvard architecture, so that memory is organised in two separated fields: program memory and data memory. As both memory are separated, the central processing unit can read/write data at the same time it loads an instruction. Peripherals and system control registers are mapped on data memory space.

Program memory is made in one page. Data is made of several 256 bytes pages.



Memory organization

Program memory

The program memory is implemented as Multiple Time Programmable (MTP) Flash memory .

The power consumption of MTP is linear with the access frequency (no significant static current).
Memory sizes:

- Flash MTP: 8192 x 22 bits (= 22 kBytes)

block	size	address
MTP	8192 x 22	H0000 - H1FFF

Program addresses

Data memory

The data memory is implemented as static Random-Access Memory (RAM). The size is 512 x 8 bits plus 8 low power RAM bytes that require very low current when addressed, programs using this low power RAM instead of regular RAM will spare even more current.

Note: The registers in Data memory are not related to the CPU registers.

block	size	address
LP RAM	8 x 8	H0000 - H0007
RAM	512 x 8	H0080 - H027F

RAM addresses

Peripherals mapping

block	size	address	Page
LP RAM	8x8	H0000-H0007	Page 0
System control	16x8	H0010-H001F	
Port A	8x8	H0020-H0027	
Port B	8x8	H0028-H002F	
Port C	4x8	H0030-H0033	
Port D	4x8	H0034-H0037	
MTP	4x8	H0038-H003B	
Event	4x8	H003C-H003F	
Interrupts control	8x8	H0040-H0047	
reserved	8x8	H0048-H004F	
UART	8x8	H0050-H0057	
Counters	8x8	H0058-H005F	
reserved	8x8	H0060-H0067	
reserved	12x8	H0068-H0073	
reserved	8x8	H0074-H007B	
Other (VLD)	4x8	H007C-H007F	
RAM1	128x8	H0080 - H00FF	
RAM2	256x8	H0100 - H01FF	Page 1
RAM3	128x8	H0200 - H027F	Page 2

Peripherals addresses

Peripherals

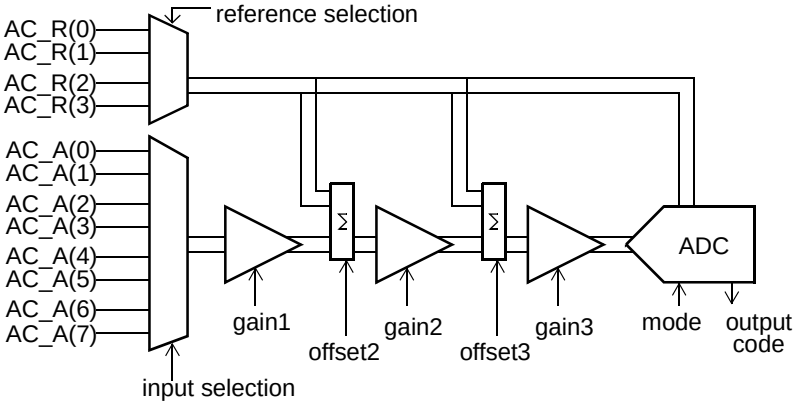
The XE88LC01 includes usual microcontroller peripherals and some other blocks more specific to low-voltage or mixed-signal operation. They are 3 parallel ports, one input port (A), one IO and analog port (B) with analog switching capabilities and one general purpose IO port (C). A watchdog is available, connected to a prescaler. Four 8-bit counters, with capture, PWM and chaining capabilities are available. The UART can handle transmission speeds as high as 38kbaud.

Low-power low-voltage blocks include a voltage level detector, two oscillators (one internal 0.1 - 2 MHz RC oscillator and a 32 kHz crystal oscillator) and a specific regulation scheme that largely uncouples current requirement from external power supply (usual CMOS ASICs require much more current at 5.5 V than they need at 2.4 V. This is not the case for the XE88LC01).

Zooming ADC

Principle

The fully differential acquisition chain is formed of a programmable gain (0.5 - 1000) and offset amplifier and a programmable speed and resolution ADC (example: 12 bits at 4 kHz, 16 bits at 1 kHz). It can handle inputs with very low full scale signal and large offsets.



Acquisition channel block diagram

Input selection is made from 1 of 4 differential pair or 1 of seven single signal versus AC_A(0). Reference is chosen from the 2 differential references. Acquisition path offset can be suppressed by inverting input polarity.

The gain of each amplifier is programmed individually. Each amplifier is powered on and off on command to minimize the total current requirement. All blocks can be set to low frequency operation and lower their current requirement by a factor 2 or 4.

The ADC can run continuously (end of conversion signalled by an interrupt, event or by pooling the ready bit), or it can be started on request.

Input signal multiplexing

There are 8 inputs named AC_A[0] to AC_A[7]. Inputs can be used either as four differential channels (Vin1=AC_A[1]-AC_A[0], …, Vin4=AC_A[7]-AC_A[6]) or AC_A[0] can be used as a common reference, providing 7 signal paths (AC_A[1]-AC_A[0], …, AC_A[7]-AC_A[0]), all referred to AC_A[0]. Default input is Vin1.

On top of these settings, inputs can be crossed or not. All multiplexing combinations are summarised in the following table (see Table 1.3) :

uni/bi-polar	sign	channel selection			selected differential input	
AMUX(4)	AMUX(3)	AMUX(2)	AMUX(1)	AMUX(0)	VIN-	VIN+
0	0	unused	0	0	A(0)	A(1)
			0	1	A(2)	A(3)
			1	0	A(4)	A(5)
			1	1	A(6)	A(7)
	1	unused	0	0	A(1)	A(0)
			0	1	A(3)	A(2)
			1	0	A(5)	A(4)
			1	1	A(7)	A(6)

Table 1.3: AMUX selection

uni/bi-polar	sign	channel selection			selected differential input	
AMUX(4)	AMUX(3)	AMUX(2)	AMUX(1)	AMUX(0)	VIN-	VIN+
1	0	0	0	0	A(0)	A(0)
		0	0	1	A(0)	A(1)
		0	1	0	A(0)	A(2)
		0	1	1	A(0)	A(3)
		1	0	0	A(0)	A(4)
		1	0	1	A(0)	A(5)
		1	1	0	A(0)	A(6)
		1	1	1	A(0)	A(7)
	1	0	0	0	A(0)	A(0)
		0	0	1	A(1)	A(0)
		0	1	0	A(2)	A(0)
		0	1	1	A(3)	A(0)
		1	0	0	A(4)	A(0)
		1	0	1	A(5)	A(0)
		1	1	0	A(6)	A(0)
		1	1	1	A(7)	A(0)

Table 1.3: AMUX selection

Input reference multiplexing

One must select one of two differential signal as reference signal (Vref1=AC_R[1]-AC_R[0], Vref2=AC_R[3]-AC_R[2]). Default is Vref1.

Amplifier chain

The 3 stages transfer functions are:

$$VD3 = GD3 \cdot VD2 - GDoff3 \cdot Vref$$
$$VD2 = GD2 \cdot VD1 - GDoff2 \cdot Vref$$
$$VD1 = GD1 \cdot Vin$$

where:
Vin=Selected input voltage
Vref=Selected reference voltage
VD1=Differential voltage at the output of first amplifier
VD2=Differential voltage at the output of second amplifier
VD3=Differential voltage at the output of third amplifier
GD1=Differential gain of stage 1
GD2=Differential gain of stage 2
GD3=Differential gain of stage 3
GDoff2= Offset gain of stage 2
GDoff3=Offset gain of stage 3

and therefore the whole transfer function is:

$$Vout\ of\ PGA = VD3 = GD3 \cdot GD2 \cdot GD1 \cdot Vin - (GDoff3 + GDoff2 \cdot GD3) \cdot Vref$$

Note: As the offset compensation is realized together with the amplification on the same summing node, the only voltages that have to stay within the supplies are Vref and the VD_i. $GD_i \cdot VD_{i-1}$ and $GDoff_i \cdot Vref$ can be larger without any saturation.

Note: All stages use a fully differential architecture and all gain and offset settings are realized with ratios of capacitors.

Note: As the ADC also provides a gain (2 nominal), the total chain transfer function is:

$$Data\_out = 2 \cdot GD3 \cdot GD2 \cdot GD1 \cdot \frac{Vin}{Vref} - 2 \cdot (GDoff3 + GDoff2 \cdot GD3)$$

Each stage is called PGAi. Features of these stages are:
- Gain can be chosen between 1 and 10 (between 0 and 10 for PGA3)
- Offset can be compensated for in PGA2 (a little) and in PGA3 (to a large extent)
- Granularity of settings is rough for PGA1, medium for PGA2, fine for PGA3
- Zero, one or two or three of the PGA stages can be used.

A functional example of one of the stages is given on Figure 1.1.

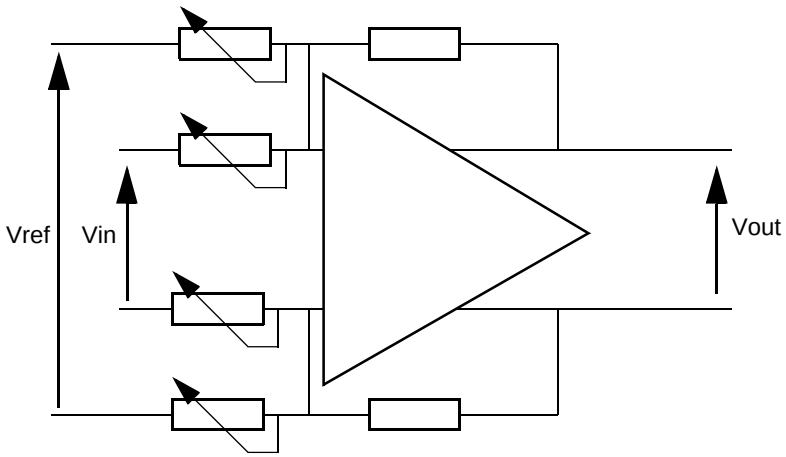


Figure 1.1: PGA stage principle implementation

1.1.1 PGA 1

symbol	description	min	typ	max	unit	Comments
GD1	PGA1 Signal Gain	1		10	-	GD1 = 1 or 10
GD_preci	Precision on gain settings	-5		+5	%	
GD_TC	Temperature dependency of gain settings	-5		+5	ppm/°C	
fs	input sampling frequency			512	kHz	
Zin1	Input impedance	150			kW	1
Zin1p	Input impedance for gain 1	1500			kW	1
VN1	Input referred noise			18	nV/ sqrt(Hz)	2

Table 1.4: PGA1 Performances

Note: 1) Measured with block connected to inputs through AMUX block. Normalized input sampling frequency for input impedance is 512 kHz. This figure has to be multiplied by 2 for fs = 256 kHz and 4 for fs = 128 kHz.
Note: 2) Input referred rms noise is 10 uV per input sample. This corresponds to 18 nV/sqrt(Hz) for fs = 512 kHz.

1.1.2 PGA2

sym	description	min	typ	max	unit	Comments
GD2	PGA2 Signal Gain	1		10	-	GD2 = 1, 2, 5 or 10
GDoff2	PGA2 Offset Gain	-1		1	FS	

Table 1.5: PGA2 Performances

sym	description	min	typ	max	unit	Comments
GDoff2_step	GDoff2(code+1) – GDoff2(code)	0.18	0.2	0.22	-	
GD_preci	Precision on gain settings	-5		+5	%	valid for GD2 and GDoff2
GD_TC	Temperature dependency of gain settings	-5		+5	ppm/°C	
fs	Input sampling frequency			512	kHz	
Zin2	Input impedance	150			kW	1
VN2	Input referred noise			36	nV/ sqrt(Hz)	2

Table 1.5: PGA2 Performances

Note: 1) Measured with block connected to inputs through AMUX block. Normalized input sampling frequency for input impedance is 512 kHz. This figure has to be multiplied by 2 for fs = 256 kHz and 4 for fs = 128 kHz.
Note: 2) Input referred rms noise is 26uV per sample.This corresponds to 36 nV/sqrt(Hz) max for fs = 512 kHz.

1.1.3 PGA3

sym	description	min	typ	max	unit	Comments
GD3	PGA3 Signal Gain	0		10	-	
GDoff3	PGA3 Offset Gain	-5		5	FS	
GD3_step	GD3(code+1) - GD3(code)	0.075	0.08	0.085	-	
GDoff3_step	GDoff2(code+1) – GDoff2(code)	0.075	0.08	0.085	-	
GD_preci	Precision on gain settings	-5		+5	%	valid for GD3 and GDoff3
GD_TC	Temperature dependency of gain settings	-5		+5	ppm/°C	
fs	Input sampling frequency			512	kHz	
Zin3	Input impedance	150			kW	1
VN3	Input referred noise			36	nV/ sqrt(Hz)	2

Table 1.6: PGA3 Performances

Note: 1) Measured with block connected to inputs through AMUX block. Normalized input sampling frequency for input impedance is 512 kHz. This figure has to be multiplied by 2 for fs = 256 kHz and 4 for fs = 128 kHz.
Note: 2) Input referred rms noise is 26uV per sample. This corresponds to 36 nV/sqrt(Hz) max for fs = 512 kHz.

ADC

1.1.4 Input-output relation

The ADC block is used to convert the differential input signal into a 16 bits 2's complement output format. The output code corresponds to the ratio:

$$\text{Output code} = \frac{\text{Vin}}{\text{Vref}} \cdot \frac{\text{smax} + 1}{\text{smax}} \cdot \text{h7FFF}$$

smax being the number of samples used to generate one output sample per elementary conversion. smax is set by **OSR** on **RegACCfg0**.

Vref can be selected up to the power supply rails and must be positive. The 2's complement output code corresponding is given in hexadecimal notation by 8000 (negative full scale) and 7FFF (positive full scale). Code outside the range are saturated to the closest full scale value. The output code is normalized into a 16 bits format. First non significant bit is forced to 1, further non signifant bits are forced to 0.

register	data							
	MSB							LSB
RegACOutLSB	bit 5	bit 4	bit 3	bit 2	bit 1	1	0	0
RegACOutMSB	sign	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6

Table 1.7: Output code exemple, for 13 bits resolution (OSR: h11, NELCONV: h01)

1.1.5 Operation mode

The mode can be either “on request” or “continuously running”.

In the “on request” mode, after a request, an initialization sequence is performed, then an algorithm is applied and an output code is produced. The converter is idle until the next request.

In the “continuously running“ mode, an internal conversion request is generated each time a conversion is finished, so that the converter is never idle. The output code is updated at a fixed rate corresponding to 1/Tout, with Tout being the conversion time.

1.1.6 Conversion sequence

The whole conversion sequence is basically made of an initialisation, a set of Nelconv elementary incremental conversions and finally a termination phase(**N_{umCONV}** is set by 2 bits on **RegACCfg0**). The result is a mean of the results of the elementary conversions.

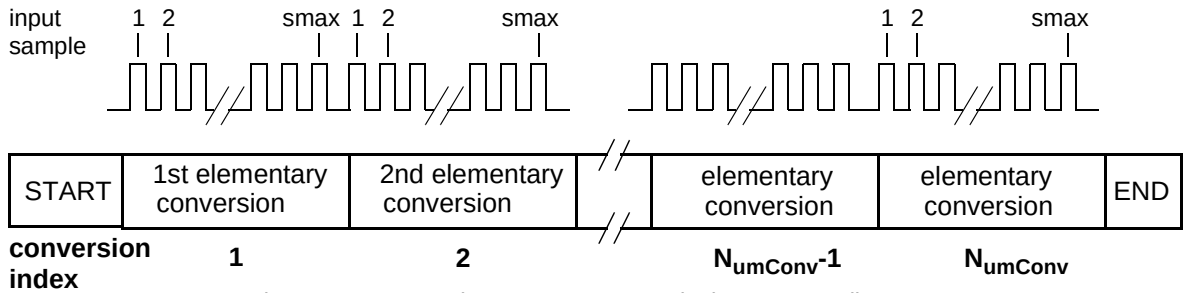


Figure 1.2: Conversion sequence. smax is the oversampling rate.

Note: **N_{umCONV}** elementary conversions are performed, each elementary conversion being made of smax input samples.

$N_{umCONV} = 2^{NELCONV}$

$smax = 8*2^{OSR}$

During the elementary conversions, the operation of the converter is the same as in a sigma delta modulator. During one conversion sequence, the elementary conversions are alternatively performed with direct and crossed PGA-ADC differential inputs, so that when two elementary conversions or more are performed, the offset of the converter is cancelled.

Note: The sizing of the decimation filter puts some limits on the total number of conversions and it is not possible to combine the maximum number of elementary conversions with the maximum oversampling (see the Nelconv*smax specification).

Some additional clock cycles (N_{INIT}+N_{END}) clock cycles are required to initiate and terminate the conversion properly.

1.1.7 Conversion duration

The conversion time is given by :

$$T_{OUT} = (2^{NELCONV} *(8*2^{OSR} +1) + (N_{INIT} + N_{END})) / fs$$

1.1.8 Resolution

As far as it is not limited by thermal noise and internal registers width, the resolution is given by :

Resolution (in bits) = 6 + 2*OSR + NELCONV

1.1.9 ADC performances

sym	description	min	typ	max	unit	Comments
VINR	Input range	-0.5		0.5	Vref	
Resol	Resolution	12			bits	1
NResol	Numerical resolution			16	bits	4
INL	Integral non-linearity			4	LSB	1,3, LSB at 12bits
fs	sampling frequency	10		512	kHz	
smax	Oversampling Ratio	8		1024	-	2
N_{UMCONV}	Number of elementary conversions in incremental mode	1		8	-	2
Ninit	Number of periods for incremental conversion initialization			5	-	
Nend	Number of periods for incremental conversion termination			5	-	

Table 1.8: ADC Performances

Note: 1) Resolution specification also includes thermal noise and differential non-linearity (DNL) for a reference signal of 2.4 V. It is defined for default operating mode (See “Default operation mode (not yet implemented)” on page 15.)

Note: 2) Only powers of 2

Note: 3) INL is defined as the deviation of the DC transfer curve from the best fit straight line. This specification holds over 100% of the full scale.

Note: 4) NResol is the maximal readable resolution of the digital filter. Input noise may be higher than NResol.

Control part

Starting a convertion

A conversion is started each time **START** or **DEF** is set. PGAs are reset after each writing operation to registers **RegACCfg1** to **RegACCfg5**. When using the PGAs, one has to start the ADCs after a PGA common-mode stabilisation delay. This is done by writing bit **START** several cycles after PGA settings modification. Delay between PGA start and ADC start should be equivalent to smax number of cycles.

End of a conversion

The end of the conversion is marked by the return to zero of busy bit, and, if set, by the generation of ADC interrupt.

Busy bit = 0 is not sufficient to denote the end of the conversion, as the ADC needs some clock cycles to set it to one at the conversion beginning. Only the transition from 1 to 0 denotes the end of conversion.

For low power or low noise applications, one should prefer to use the interruption as the processor can go to HALT between conversion start and conversion end.

Clocks generation

Peripheral clock (psck) can be chosen among four prescaler clocks (bit **FIN** of **RegACCfg2**), see Table 1.10, derived from the XE8000 RC oscillator.
The clock of the acquisition path (fs) is derived from the peripheral clock.

$$fs = psck / 4$$

Acquisition of a sample

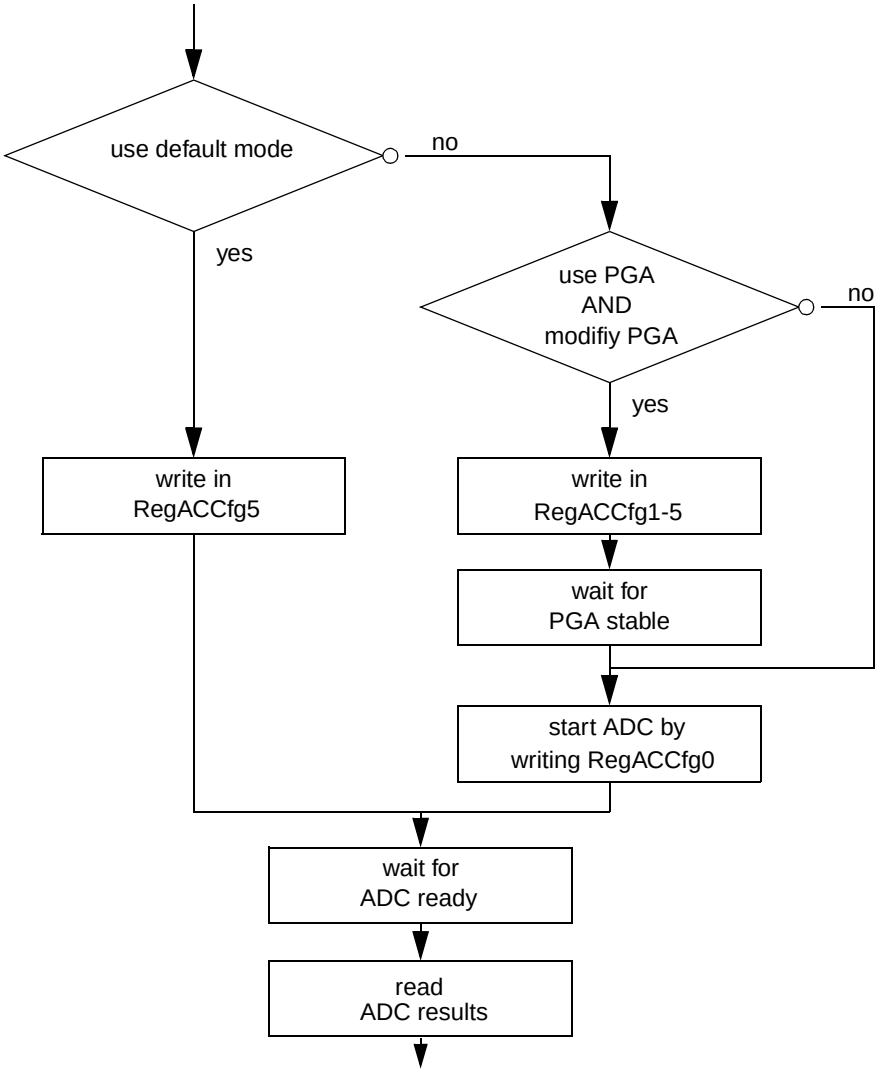


Figure 1.3: Acquisition flow

Default operation mode (not yet implemented)

The **DEF** bit (**RegACCfg5**) allows the use of the ADC in a default mode without any gain nor offset adjustment (see values in the right column of Table 1.10). This default mode is used in specifications to define resolution and INL. The only action to launch the operation of the peripheral is in this case to write a 'x1xx xxxx' at address 111. **VMUX** and **AMUX** are written at the same time and are not reset to default value. **BUSY** is not affected. The only way to stop a running conversion before completion is to shut the ADC down writing '0000' in **ENABLE** (**RegACCfg1**).

Registers

Eight registers control this peripheral. Two registers are for the data output, six for peripheral general set-up. Registers are defined in Table 1.9 and Table 1.10.

register	data
RegACOutLSB	ADC_OUT_L
RegACOutMSB	ADC_OUT_H

Table 1.9: Peripheral register memory map

register	data				
RegACCfg0	START	NELCONV	OSR	CONT	reserved
RegACCfg1	IB_AMP_ADC	IB_AMP_PGA	ENABLE		
RegACCfg2	FIN	PGA2_GAIN	PGA2_OFF		
RegACCfg3	PGA1_ GAIN	PGA3_GAIN			
RegACCfg4	reserved	PGA3_OFF			
RegACCfg5	BUSY	DEF	AMUX		VMUX

Table 1.9: Peripheral register memory map

Name	Register	rm	description	Default (reset and DEF mode)
ADC_OUT(15:0)	**RegACOutLSB RegACOutMSB**	r	data output	0000h
AMUX(4:0)	**RegACCfg5**	wr	Selection of PGA inputs	00000 (reset only)
BUSY	**RegACCfg5**	r	'1' : conversion is in progress '0' : data is available	0
CONT	**RegACCfg0**	wr	'1' : continuous operation. '0' : one shot mode	0
DEF	**RegACCfg5**	wr0	Default Operation bit '1': All registers but VMUX and AMUX are reset and default values are used '0': Normal operation	N/A
ENABLE(3:0)	**RegACCfg1**	wr	bit3 : PGA3, bit2 : PGA2, bit1 : PGA1, bit 0 : ADC If a bit is '1', the block is powered. If not, the block is switched off and all internal digital signals are reset. Concerning the PGAs, ENABLE=0 means also that inputs and outputs are wired together and that the acquisition chain is not perturbed by the block.	0000
FIN(1:0)	**RegACCfg2**	wr	'00' : RC = psck = 4 fs '01' : RC / 2 = psck = 4 fs '10' : RC / 8 = psck = 4 fs '11' : RC / 32 = psck = 4 fs Rem: do not select an fs clock that is faster than 512 kHz.	00
IB_AMP_PGA(1:0)	**RegACCfg1**	wr	PGA amplifiers biasing current reduction factor '00' : current magnification factor = 0.25 '01' : current magnification factor = 0.5 '10' : current magnification factor = 0.75 '11' : current magnification factor = 1	11
IB_AMP_ADC(1:0)	**RegACCfg1**	wr	ADC amplifiers biasing current reduction factor. Tuning identical to IB_AMP_PGA	11
NELCONV(1:0)	**RegACCfg0**	wr	Number of elementary conversions '00' : 1 conversion, '01' : 2 conversions '10' : 4 conversions, '11' : 8 conversions	01
OSR(2:0)	**RegACCfg0**	wr	OverSampling Ratio. Defined as fs/fout. $O_{SR} = 8*2^{OSR(2:0)}$. '000' : oversampling = 8, ..., '111' : oversampling = 1024	010
PGA1_GAIN	**RegACCfg3**	wr	signal gain of first PGA stage (GD1) '1' : nominal gain is 10. '0' : nominal gain is 1	0
PGA2_GAIN(1:0)	**RegACCfg2**	wr	signal gain of second PGA stage (GD2) '11' : nominal gain is 10 '10' : nominal gain is 5 '01' : nominal gain is 2 '00' : nominal gain is 1	00
PGA2_OFF(3:0)	**RegACCfg2**	wr	offset gain of second PGA stage (GDoff2) bit 3: offset sign ('0' : GDoff2 > 0, '1' : GDoff2 < 0) bits (2:0) : offset amplitude '01x1' : GDoff2 = 1.0 nominal, '0100' : GDoff2 = 0.8 nominal, '0011' : GDoff2 = 0.6 nominal, ..., '0000' : GDoff2 = 0.0 nominal, '1001'' : GDoff2 = -0.2 nominal, ..., '11x1' : GDoff2 = -1.0 nominal	0000
PGA3_GAIN(6:0)	**RegACCfg3**	wr	signal gain of third stage (GD3) GD3 = 0.08*PGA3_GAIN(6:0) Nominal values : 0 ('000 0000'), ..., 10 ('111 1000')	000 1100
PGA3_OFF(6:0)	**RegACCfg4**	wr	offset gain of third stage (GDoff3) bit 6: offset sign ('0' : GDoff3 > 0, '1' : GDoff3 < 0) GDoff3 = 0.08*PGA3_OFF(5:0), maximum = 5.04 Nominal values : -5.04 ('111 1111'), 0 ('x00 0000'), +5.04 ('011 1111')	000 0000
START	**RegACCfg0**	wr0	writing a "1" in START bit restarts the ADC. It does not affect the PGAs.	0

Table 1.10: Peripheral register memory map, bits description

Name	Register	rm	description	Default (reset and DEF mode)
TEST	**RegACCfg0**		reserved	0
VMUX	**RegACCfg5**	wr	VREF selection multiplexer '0' : VREF0 is used, '1' : VREF1 is used	0 (reset only)

Table 1.10: Peripheral register memory map, bits description

Low-Power Microcontroller
XX-XE88LC01

XE8000 Family

Features

The main characteristics of the XE8000 MCU family is

- Ultra low power operation
- Low voltage operation (1.2 V for the XE88LC04, XE88LC06 and XE88LC07, 2.4 V for the others)
- High efficiency CPU
  - 1 instruction per clock cycle, for all instructions
  - 22 bits wide instructions
  - Integrated 8x8 -> 16 bits multiplier
  - All instructions on one page
  - 8 bits data bus
  - 8 addressing modes
- MTP (multiple time programmable) memory available
- Dual clock (X-tal and/or RC)
- Each peripheral can be set on/off individually for minimal power consumption
- UART
- Watch dog
- 4x8 bits timers with PWM ability
- Advanced acquisition path
  - Fully differential analog signal path on signal and reference
  - 4x2 or 7x1 + 1 signal input
  - 2x2 reference input
  - 0.5 - 1000 programmable gain amplifier
  - Offset programmed over +- 10 full scale
  - 5 - 16 bits resolution ADC
  - Low speed modes with reduced bias current for minimal power consumption
- Bias and signal DACs for resistive bridge sensing and analog output
- Complete development tools using Windows95 or NT graphical interface
  - Assembler
  - ANSI-C compiler
  - Source level debugger
  - Current and memory usage monitoring (Profiler)
  - CPU Simulator
  - CPU Emulator XE8000HaCE
  - Programmer and starter kit (XE88LC01ProStart)
  - Hardware emulators (works with XE8000HaCE, in preparation)

Family

The XE8000 Family ultra low-power microcontroller is made of several members, all using the same microprocessor core and differing by the peripherals available.

The XE88LC01 is a low power sensing microcontroller, based on the XE88LC03, with an advanced acquisition path including differential programmable gain amplifiers and a high resolution analog to digital converter. Its main applications are dataloggers and process control.

The XE88LC02 is a low power sensing microcontroller, based on the XE88LC06 with the analog part of the XE88LC01, with an additional LCD driver. Its main applications are metering and dataloggers.

TThe XE88LC03 is a low power, low voltage, general purpose microcontroller. Its main points are the very efficient CoolRISC core, the low voltage function and the real time clock. Its main applications are low voltage control and supervision.

Low-Power Microcontroller

XX-XE88LC01

The XE88LC04 is a low power, low voltage, general purpose microcontroller, based on the XE88LC06, with an additional LCD driver. Its main points are the very efficient CoolRISC core, the low voltage function and the real time clock. Its main applications are low voltage control and supervision.

The XE88LC05 is a low power sensing microcontroller, based on the XE88LC01, with analog outputs. Its main applications are piezoresistive sensors and 4 - 20 mA loops systems.

The XE88LC06 is an improved XE88LC03, with 4 low power analog comparators. Its main applications are low voltage control and supervision.

The XE88LC07 is a smaller and even lower power microcontroller, based on the XE88LC06, with less memory.

	XE88LC01	XE88LC02	XE88LC03	XE88LC04	XE88LC05
Supply voltage	2.4 - 5.5 V	2.4 - 5.5 V	2.4 - 5.5 V	1.2- 5.5 V for ROM 2.4 - 5.5 V for MTP	2.4 - 5.5 V
Max speed	2 MIPS	4 MIPS	2 MIPS	4 MIPS at 2.4 V	2 MIPS
Operating temperature	-40 - 85 °C	-40 - 85 °C -40 - 125 °C	-40 - 85 °C	-40 - 85 °C -40 - 125 °C	-40 - 85 °C
CPU	CoolRISC 816, 22 bits instructions 8 bits data HW multiplier	CoolRISC 816, 22 bits instructions 8 bits data HW multiplier	CoolRISC 816, 22 bits instructions 8 bits data HW multiplier	CoolRISC 816, 22 bits instructions 8 bits data HW multiplier	CoolRISC 816, 22 bits instructions 8 bits data HW multiplier
Program memory	8k Instructions = 22 kB ROM or MTP	8k Instruction = 22 kB ROM or MTP	8k Instructions = 22 kB ROM or MTP	8k Instructions = 22 kB ROM or MTP	8k Instructions = 22 kB ROM or MTP
Data memory	512 + 8 Bytes	768 + 8 Bytes	512 + 8 Bytes	768 + 8 Bytes	512 + 8 Bytes
Port A	8 input and external interrupt	8 input and external interrupt	8 input and external interrupt	8 input and external interrupt	8 input and external interrupt
Port B	8 input/output and analog	8 input/output and analog	8 input/output and analog	8 input/output and analog	8 input/output and analog
Port C	8 input/output	8 input/output	4 to 8 input/output	4 to 8 input/output	8 input/output
Watchdog timer	yes	yes	yes	yes	yes
General purpose timers with PWM	4 x 8 bits	4 x 8 bits	4 x 8 bits	4 x 8 bits	4 x 8 bits
UART	yes	yes	yes	yes	yes
2-3 wires serial interface	transition detection + software	transition detection + software	transition detection + software	transition detection + software	transition detection + software
Voltage level detector	yes	yes	yes	yes	yes
Oscillators	32 kHz quartz, internal RC	32 kHz quartz, internal RC	32 kHz quartz, internal RC	32 kHz quartz, internal RC	32 kHz quartz, internal RC
LCD drivers		120 segments		120 segments	
Analog mux	Port B and 4x2 or 7x1+1	Port B and 4x2 or 7x1+1	Port B	Port B	Port B and 4x2 or 7x1+1
LP comparators		4		4	
PGA	gain 0.5 - 1000	gain 0.5 - 1000			gain 0.5 - 1000
ADC	5 - 16 bits resolution	5 - 16 bits resolution			5 - 16 bits resolution
DAC	PWM	PWM	PWM	PWM	PWM 8 bit bias DAC, 4 - 16 bits signal DAC
Package	TQFP44, die		SO28, TQFP32, die		TQFP64, die
Availability	yes	samples Q2/01	yes	samples Q2/01	yes

Table 1.11: List of the XE8000 family members functions

Contacting XEMICS

You can contact XEMICS at
XEMICS SA
Tel: +41 32 720 5170
Fax: +41 32 720 5770
E-mail: info@xemics.com

You will find more information about the XE88LC01 and other XEMICS products, as well as addresses of our representatives and distributors for your region on www.xemics.com.