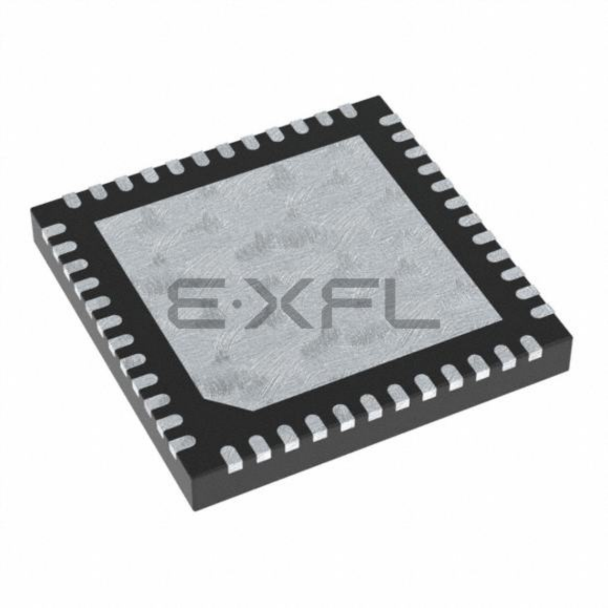
Microchip Technology - AT32UC3L0128-ZAUR Datasheet



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	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3l0128-zaur

5.3 Peripheral Address Map

Table 5-3. Peripheral Address Mapping

Address		Peripheral Name
0xFFFE0000	FLASHCDW	Flash Controller - FLASHCDW
0xFFFE0400	HMATRIX	HSB Matrix - HMATRIX
0xFFFE0800	SAU	Secure Access Unit - SAU
0xFFFF0000	PDCA	Peripheral DMA Controller - PDCA
0xFFFF1000	INTC	Interrupt controller - INTC
0xFFFF1400	PM	Power Manager - PM
0xFFFF1800	SCIF	System Control Interface - SCIF
0xFFFF1C00	AST	Asynchronous Timer - AST
0xFFFF2000	WDT	Watchdog Timer - WDT
0xFFFF2400	EIC	External Interrupt Controller - EIC
0xFFFF2800	FREQM	Frequency Meter - FREQM
0xFFFF2C00	GPIO	General-Purpose Input/Output Controller - GPIO
0xFFFF3000	USART0	Universal Synchronous Asynchronous Receiver Transmitter - USART0
0xFFFF3400	USART1	Universal Synchronous Asynchronous Receiver Transmitter - USART1
0xFFFF3800	USART2	Universal Synchronous Asynchronous Receiver Transmitter - USART2
0xFFFF3C00	USART3	Universal Synchronous Asynchronous Receiver Transmitter - USART3
0xFFFF4000	SPI	Serial Peripheral Interface - SPI
0xFFFF4400	TWIM0	Two-wire Master Interface - TWIM0

7.8 Module Configuration

The specific configuration for each PDCA instance is listed in the following tables.The module bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 7-6. PDCA Configuration

Feature	PDCA
Number of channels	12
Number of performance monitors	1

Table 7-7. PDCA Clocks

Clock Name	Description
CLK_PDCA_HSB	Clock for the PDCA HSB interface
CLK_PDCA_PB	Clock for the PDCA PB interface

Table 7-8. Register Reset Values

Register	Reset Value
PSR CH 0	0
PSR CH 1	1
PSR CH 2	2
PSR CH 3	3
PSR CH 4	4
PSR CH 5	5
PSR CH 6	6
PSR CH 7	7
PSR CH 8	8
PSR CH 9	9
PSR CH 10	10
PSR CH 11	11
VERSION	123

The PDCA and the peripheral modules communicate through a set of handshake signals. The following table defines the valid settings for the Peripheral Identifier (PID) in the PDCA Peripheral Select Register (PSR). The direction is specified as observed from the memory, so RX means transfers from peripheral to memory, and TX means from memory to peripheral.

Table 7-9. Peripheral Identity Values

PID	Direction	Peripheral Instance	Peripheral Register
0	RX	USART0	RHR
1	RX	USART1	RHR
2	RX	USART2	RHR

8.3.4 Debug Operation

When an external debugger forces the CPU into debug mode, the FLASHCDW continues normal operation. If the FLASHCDW is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

8.4 Functional Description

8.4.1 Bus Interfaces

The FLASHCDW has two bus interfaces, one High Speed Bus (HSB) interface for reads from the flash memory and writes to the page buffer, and one Peripheral Bus (PB) interface for issuing commands and reading status from the controller.

8.4.2 Memory Organization

The flash memory is divided into a set of pages. A page is the basic unit addressed when programming the flash. A page consists of several words. The pages are grouped into 16 regions of equal size. Each of these regions can be locked by a dedicated fuse bit, protecting it from accidental modification.

- *p* pages (*FLASH_P*)
- *w* bytes in each page and in the page buffer (*FLASH_W*)
- *pw* bytes in total (*FLASH_PW*)
- *f* general-purpose fuse bits (*FLASH_F*), used as region lock bits and for other device-specific purposes
- 1 security fuse bit
- 1 User page

8.4.3 User Page

The User page is an additional page, outside the regular flash array, that can be used to store various data, such as calibration data and serial numbers. This page is not erased by regular chip erase. The User page can only be written and erased by a special set of commands. Read accesses to the User page are performed just as any other read accesses to the flash. The address map of the User page is given in Figure 8-1 on page 80.

8.4.4 Read Operations

The on-chip flash memory is typically used for storing instructions to be executed by the CPU. The CPU will address instructions using the HSB bus, and the FLASHCDW will access the flash memory and return the addressed 32-bit word.

In systems where the HSB clock period is slower than the access time of the flash memory, the FLASHCDW can operate in 0 wait state mode, and output one 32-bit word on the bus per clock cycle. If the clock frequency allows, the user should use 0 wait state mode, because this gives the highest performance as no stall cycles are encountered.

The FLASHCDW can also operate in systems where the HSB bus clock period is faster than the access speed of the flash memory. Wait state support and a read granularity of 64 bits ensure efficiency in such systems.

Performance for systems with high clock frequency is increased since the internal read word width of the flash memory is 64 bits. When a 32-bit word is to be addressed, the word itself and

8.8.3 Flash Status Register

Name:	FSR
Access Type:	Read-only
Offset:	0x08
Reset Value:	0x00000000

31	30	29	28	27	26	25	24
LOCK15	LOCK14	LOCK13	LOCK12	LOCK11	LOCK10	LOCK9	LOCK8

23	22	21	20	19	18	17	16
LOCK7	LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
-	HSMODE	QPRR	SECURITY	PROGE	LOCKE	-	FRDY

- **LOCKx: Lock Region x Lock Status**

  0: The corresponding lock region is not locked.

  1: The corresponding lock region is locked.

- **HSMODE: High-Speed Mode**

  0: High-speed mode disabled.

  1: High-speed mode enabled.

- **QPRR: Quick Page Read Result**

  0: The result is zero, i.e. the page is not erased.

  1: The result is one, i.e. the page is erased.

- **SECURITY: Security Bit Status**

  0: The security bit is inactive.

  1: The security bit is active.

- **PROGE: Programming Error Status**

  Automatically cleared when FSR is read.

  0: No invalid commands and no bad keywords were written in the Flash Command Register FCMD.

  1: An invalid command and/or a bad keyword was/were written in the Flash Command Register FCMD.

- **LOCKE: Lock Error Status**

  Automatically cleared when FSR is read.

  0: No programming of at least one locked lock region has happened since the last read of FSR.

  1: Programming of at least one locked lock region has happened since the last read of FSR.

- **FRDY: Flash Ready Status**

  0: The Flash Controller is busy and the application must wait before running a new command.

  1: The Flash Controller is ready to run a new command.

8.9.1 Flash General Purpose Fuse Register Low (FGPFRLO)

31	30	29	28	27	26	25	24
BODEN		BODHYST	BODLEVEL[5:1]				

23	22	21	20	19	18	17	16
BODLEVEL[0]	UPROT	SECURE		BOOTPROT			EPFL

15	14	13	12	11	10	9	8
LOCK[15:8]							

7	6	5	4	3	2	1	0
LOCK[7:0]							

- **BODEN: Brown Out Detector Enable**

BODEN	Description
00	BOD disabled
01	BOD enabled, BOD reset enabled
10	BOD enabled, BOD reset disabled
11	BOD disabled

- **BODHYST: Brown Out Detector Hysteresis**

  0: The Brown out detector hysteresis is disabled

  1: The Brown out detector hysteresis is enabled

- **BODLEVEL: Brown Out Detector Trigger Level**

  This controls the voltage trigger level for the Brown out detector. Refer to ”Electrical Characteristics” on page 791.

- **UPROT, SECURE, BOOTPROT, EPFL, LOCK**

  These are Flash Controller fuses and are described in the FLASHCDW section.

8.9.1.1 Default Fuse Value

The devices are shipped with the FGPFRLO register value:0xE07FFFFF:

- BODEN fuses set to 11. BOD is disabled.
- BODHYST fuse set to 1. The BOD hysteresis is enabled.
- BODLEVEL fuses set to 000000. This is the minimum voltage trigger level for BOD. This level is lower than the POR level, so when BOD is enabled, it will never trigger with this default value.
- UPROT fuse set to 1.
- SECURE fuse set to 11.
- BOOTPROT fuses set to 111. The bootloader protection is disabled.
- EPFL fuse set to 1. External privileged fetch is not locked.
- LOCK fuses set to 1111111111111111. No region locked.

After the JTAG or aWire chip erase command, the FGPFR register value is 0xFFFFFFFF.

10.6 Module Configuration

The specific configuration for each HMATRIX instance is listed in the following tables.The module bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 10-3. HMATRIX Clocks

Clock Name	Description
CLK_HMATRIX	Clock for the HMATRIX bus interface

10.6.1 Bus Matrix Connections

The bus matrix has the several masters and slaves. Each master has its own bus and its own decoder, thus allowing a different memory mapping per master. The master number in the table below can be used to index the HMATRIX control registers. For example, HMATRIX MCFG0 register is associated with the CPU Data master interface.

Table 10-4. High Speed Bus Masters

Master 0	CPU Data
Master 1	CPU Instruction
Master 2	CPU SAB
Master 3	SAU
Master 4	PDCA

Each slave has its own arbiter, thus allowing a different arbitration per slave. The slave number in the table below can be used to index the HMATRIX control registers. For example, SCFG3 is associated with the Internal SRAM Slave Interface.

Accesses to unused areas returns an error result to the master requesting such an access.

Table 10-5. High Speed Bus Slaves

Slave 0	Internal Flash
Slave 1	HSB-PB Bridge A
Slave 2	HSB-PB Bridge B
Slave 3	Internal SRAM
Slave 4	SAU

13.4.1 I/O Lines

The SCIF provides a number of generic clock outputs, which can be connected to output pins, multiplexed with GPIO lines. The programmer must first program the GPIO controller to assign these pins to their peripheral function. If the I/O pins of the SCIF are not used by the application, they can be used for other purposes by the GPIO controller. Oscillator pins are also multiplexed with GPIO. When oscillators are used, the related pins are controlled directly by the SCIF, overriding GPIO settings.

RC32OUT will be output after reset, and the GPIO controller can assign this pin to other peripheral function after start-up.

13.4.2 Power Management

The BODs and all the oscillators, except the 32KHz oscillator (OSC32K) are turned off in some sleep modes and turned automatically on when the device wakes up. The Voltage Regulator is set in low power mode in some sleep modes and automatically set back in normal mode when the device wakes up. Please refer to the Power Manager chapter for details.

The BOD control registers will not be reset by the Power Manager on a BOD reset.

13.4.3 Clocks

The SCIF controls all oscillators in the device. The oscillators can be used as source for the CPU and peripherals. Selection of source is done in the Power Manager. The oscillators can also be used as source for generic clocks.

13.4.4 Interrupts

The SCIF interrupt request line is connected to the interrupt controller. Using the SCIF interrupt requires the interrupt controller to be programmed first.

13.4.5 Debug Operation

The SCIF does not interact with debug operations.

13.5 Functional Description

13.5.1 Oscillator (OSC) Operation

Rev: 1.1.1.0

The main oscillator (OSCn) is designed to be used with an external 0.450 to 16MHz crystal and two biasing capacitors, as shown in the Electrical Characteristics chapter, or with an external clock connected to the XIN. The oscillator can be used as source for the main clock in the device, as described in the Power Manager chapter. The oscillator can be used as source for the generic clocks, as described in the Generic Clocks section.

The oscillator is disabled by default after reset. When the oscillator is disabled, the XIN and XOUT pins can be used as general purpose I/Os. When the oscillator is enabled, the XIN and XOUT pins are controlled directly by the SCIF, overriding GPIO settings. When the oscillator is configured to use an external clock, the clock must be applied to the XIN pin while the XOUT pin can be used as general purpose I/O.

The oscillator is enabled by writing a one to the Oscillator Enable bit in the Oscillator Control register (OSCCTRLn.OSCEN). Operation mode (external clock or crystal) is selected by writing to the Oscillator Mode bit in OSCCTRLn (OSCCTRLn.MODE). The oscillator is automatically dis-

18.4 I/O Lines Description

Pin Name	Description	Type
GPIOn	GPIO pin n	Digital

18.5 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

18.5.1 Power Management

If the CPU enters a sleep mode that disables clocks used by the GPIO, the GPIO will stop functioning and resume operation after the system wakes up from sleep mode.

If a peripheral function is configured for a GPIO pin, the peripheral will be able to control the GPIO pin even if the GPIO clock is stopped.

18.5.2 Clocks

The GPIO is connected to a Peripheral Bus clock (CLK_GPIO). This clock is generated by the Power Manager. CLK_GPIO is enabled at reset, and can be disabled by writing to the Power Manager. CLK_GPIO must be enabled in order to access the configuration registers of the GPIO or to use the GPIO interrupts. After configuring the GPIO, the CLK_GPIO can be disabled by writing to the Power Manager if interrupts are not used.

If the CPU Local Bus is used to access the configuration interface of the GPIO, the CLK_GPIO must be equal to the CPU clock to avoid data loss.

18.5.3 Interrupts

The GPIO interrupt request lines are connected to the interrupt controller. Using the GPIO interrupts requires the interrupt controller to be programmed first.

18.5.4 Peripheral Events

The GPIO peripheral events are connected via the Peripheral Event System. Refer to the Peripheral Event System chapter for details.

18.5.5 Debug Operation

When an external debugger forces the CPU into debug mode, the GPIO continues normal operation. If the GPIO is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

- TFrame_Maximum = 1.4 x (THeader_Nominal + TResponse_Nominal + 1)(Note:)

Note: The term “+1” leads to an integer result for TFrame_Max (LIN Specification 1.3)

If the Checksum is sent (CHKDIS=0):

- TResponse_Nominal = 10 x (NData + 1)
- TFrame_Maximum = 1.4 x (34 + 10 x (DLC + 1 + 1) + 1)
- TFrame_Maximum = 77 + 14 x DLC

If the Checksum is not sent (CHKDIS=1):

- TResponse_Nominal = 10 x NData
- TFrame_Maximum = 1.4 x (34 + 10 x (DLC + 1) + 1)
- TFrame_Maximum = 63 + 14 x DLC

19.6.6 LIN Errors

These error bits are cleared by writing a one to CSR.RSTSTA.

19.6.6.1 Slave Not Responding Error (CSR.LINSNRE)

This error is generated if no valid message appears within the TFrame_Maximum time frame slot, while the USART is expecting a response from another node (NACT=SUBSCRIBE).

19.6.6.2 Checksum Error (CSR.LINCE)

This error is generated if the received checksum is wrong. This error can only be generated if the checksum feature is enabled (CHKDIS=0).

19.6.6.3 Identifier Parity Error (CSR.LINIPE)

This error is generated if the identifier parity is wrong. This error can only be generated if parity is enabled (PARDIS=0).

19.6.6.4 Inconsistent Sync Field Error (CSR.LINISFE)

This error is generated in slave mode if the Sync Field character received is not 0x55. Synchronization procedure is aborted.

19.6.6.5 Bit Error (CSR.LINBE)

This error is generated if the value transmitted by the USART on Tx differs from the value sampled on Rx. If a bit error is detected, the transmission is aborted at the next byte border.

19.6.7 LIN Frame Handling

19.6.7.1 Master Node Configuration

- Write a one to CR.TXEN and CR.RXEN to enable both transmitter and receiver
- Select LIN mode and master node by writing to MR.MODE
- Configure the baud rate by writing to CD and FP in BRGR
- Configure the frame transfer by writing to NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS, and DLC in LINMR
- Check that CSR.TXRDY is one
- Send the header by writing to LINIR.IDCHR

The following procedure depends on the NACT setting:

19.7.16 Version Register

Name: VERSION

Access Type: Read-only

Offset: 0xFC

Reset Value: -

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	MFN			

15	14	13	12	11	10	9	8
–	–	–	–	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **MFN**

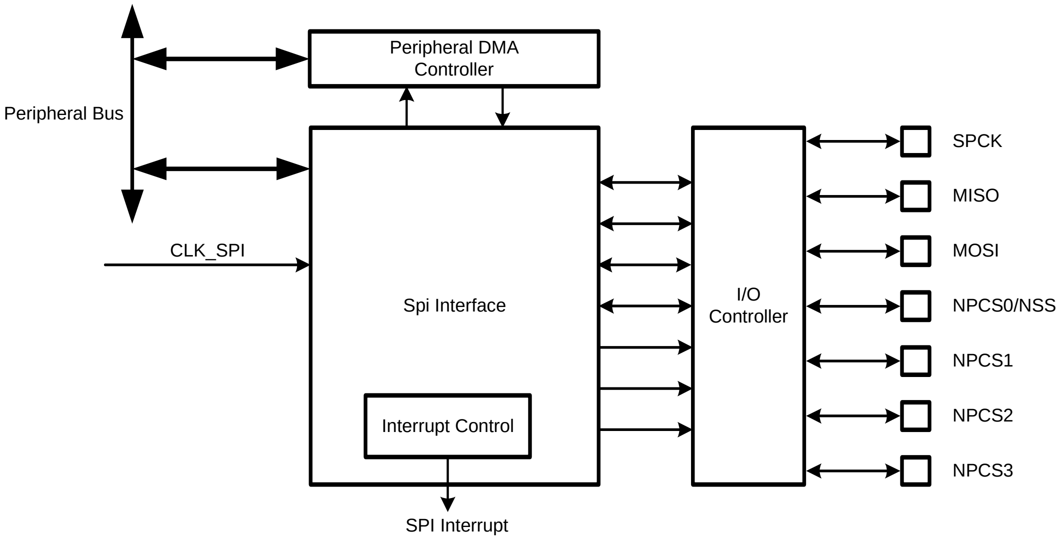
  Reserved. No functionality associated.

- **VERSION**

  Version of the module. No functionality associated.

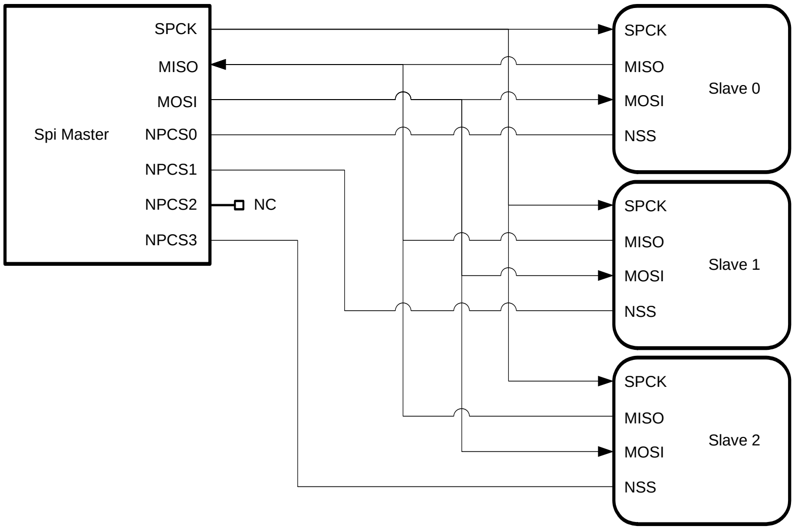
20.3 Block Diagram

Figure 20-1. SPI Block Diagram



20.4 Application Block Diagram

Figure 20-2. Application Block Diagram: Single Master/Multiple Slave Implementation



20.8.6 Interrupt Enable Register

Name: IER

Access Type: Write-only

Offset: 0x14

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	-	UNDES	TXEMPTY	NSSR

7	6	5	4	3	2	1	0
-	-	-	-	OVRES	MODF	TDRE	RDRF

Writing a zero to a bit in this register has no effect.

Writing a one to a bit in this register will set the corresponding bit in IMR.

21.9.13 Parameter Register (PR)

Name: PR

Access Type: Read-only

Offset: 0x30

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

To assure correct behavior, respect the following programming sequences:

22.8.6.1 Data Transmit with the Peripheral DMA Controller

1. Initialize the transmit Peripheral DMA Controller (memory pointers, size, etc.).
2. Configure the TWIS (ADR, NBYTES, etc.).
3. Start the transfer by enabling the Peripheral DMA Controller to transmit.
4. Wait for the Peripheral DMA Controller end-of-transmit flag.
5. Disable the Peripheral DMA Controller.

22.8.6.2 Data Receive with the Peripheral DMA Controller

1. Initialize the receive Peripheral DMA Controller (memory pointers, size - 1, etc.).
2. Configure the TWIS (ADR, NBYTES, etc.).
3. Start the transfer by enabling the Peripheral DMA Controller to receive.
4. Wait for the Peripheral DMA Controller end-of-receive flag.
5. Disable the Peripheral DMA Controller.

22.8.7 SMBus Mode

SMBus mode is enabled by writing a one to the SMBus Mode Enable (SMEN) bit in CR. SMBus mode operation is similar to I²C operation with the following exceptions:

- Only 7-bit addressing can be used.
- The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be written to TR.
- Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
- A dedicated bus line, SMBALERT, allows a slave to get a master's attention.
- A set of addresses have been reserved for protocol handling, such as Alert Response Address (ARA) and Host Header (HH) Address. Address matching on these addresses can be enabled by configuring CR appropriately.

22.8.7.1 Packet Error Checking (PEC)

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing a one to the Packet Error Checking Enable (PECEN) bit in CR enables automatic PEC handling in the current transfer. The PEC generator is always updated on every bit transmitted or received, so that PEC handling on following linked transfers will be correct.

In slave receiver mode, the master calculates a PEC value and transmits it to the slave after all data bytes have been transmitted. Upon reception of this PEC byte, the slave will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the slave will return an ACK to the master. If the PEC values differ, data was corrupted, and the slave will return a NAK value. The SR.SMBPECERR bit is set automatically if a PEC error occurred.

In slave transmitter mode, the slave calculates a PEC value and transmits it to the master after all data bytes have been transmitted. Upon reception of this PEC byte, the master will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the master must take appropriate action.

The PEC byte is automatically inserted in a slave transmitter transmission if PEC enabled when NBYTES reaches zero. The PEC byte is identified in a slave receiver transmission if PEC

26.9.5 Compare Value Register

Name: CVR

Access Type: Read/Write

Offset: 0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	HV[11:8]			

23	22	21	20	19	18	17	16
HV[7:0]							

15	14	13	12	11	10	9	8
-	-	-	-	LV[11:8]			

7	6	5	4	3	2	1	0
LV[7:0]							

- **HV: High Value**

  Defines the high value used when comparing analog input.

- **LV: Low Value**

  Defines the low value used when comparing analog input.

This bit is set when pen contact is detected and pen detect is enabled.

- **OVRE: Overrun Error Status**

  This bit is cleared when no Overrun Error has occurred since the start of a conversion sequence.

  This bit is set when one or more Overrun Error has occurred since the start of a conversion sequence.

- **DRDY: Data Ready Status**

  0: No data has been converted since the last reset.

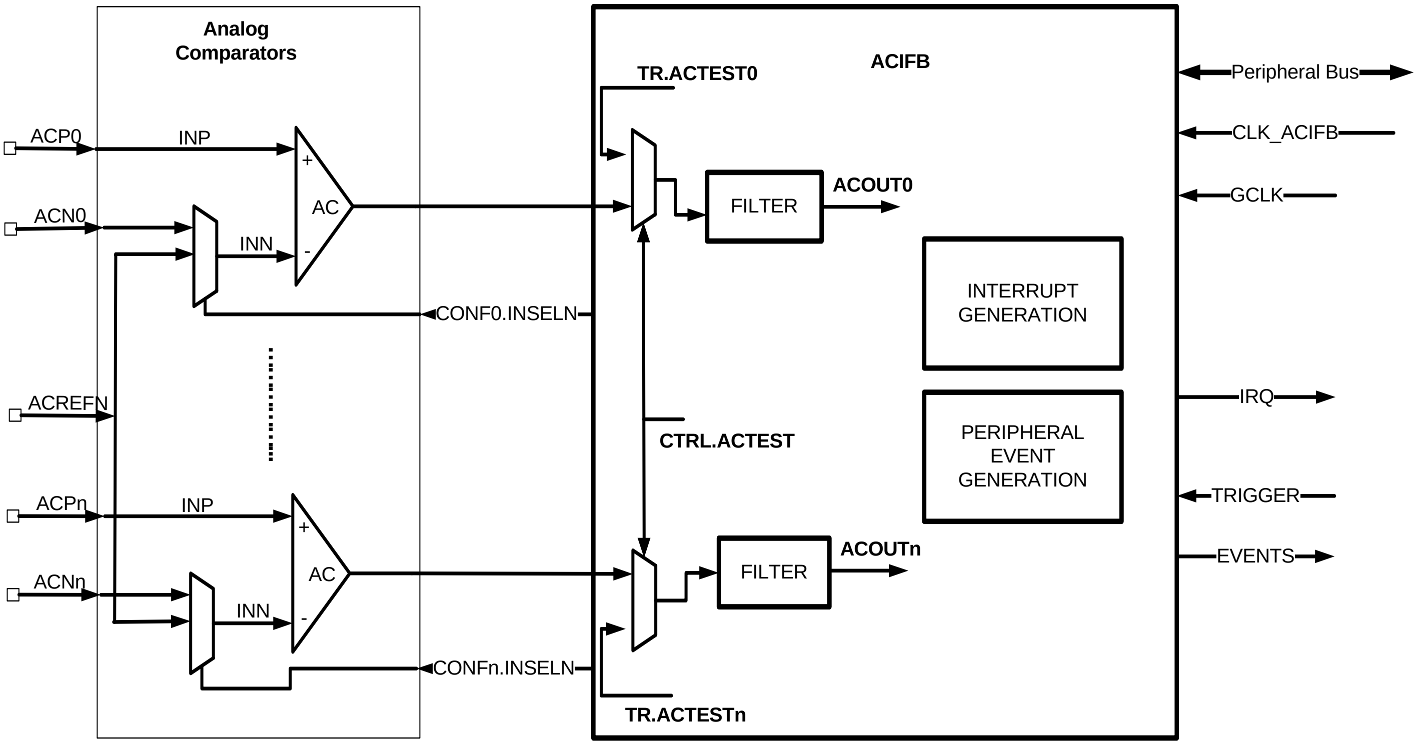
  1: One or more conversions have completed since the last reset and data is available in LCDR.

  This bit is cleared when CR.SWRST is written to one.

  This bit is set when one or more conversions have completed and data is available in LCDR.

27.3 Block Diagram

Figure 27-1. ACIFB Block Diagram



27.4 I/O Lines Description

There are two groups of analog comparators, A and B, as shown in Table 27-1. In normal mode, this grouping does not have any meaning. In window mode, two analog comparators, one from group A and the corresponding comparator from group B, are paired.

Table 27-1. Analog Comparator Groups for Window Mode

Group A	Group B	Pair Number
AC0	AC1	0
AC2	AC3	1
AC4	AC5	2
AC6	AC7	3

Table 27-2. I/O Line Description

Pin Name	Pin Description	Type
ACAPn	Positive reference pin for Analog Comparator A n	Analog
ACANn	Negative reference pin for Analog Comparator A n	Analog

28.7.12 Matrix Group Configuration Register 2

Name: MGCFG2

Access Type: Read/Write

Offset: 0x38

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
ACCTRL	CONSEN			-			
23	22	21	20	19	18	17	16
CXDILEN							
15	14	13	12	11	10	9	8
-				SYNCTIM[11:8]			
7	6	5	4	3	2	1	0
SYNCTIM[7:0]							

- **ACCTRL: Analog Comparator Control**
  When written to one, allows the CAT to disable the analog comparators when they are not needed. When written to zero, the analog comparators are always enabled.
- **CONSEN: Consensus Filter Length**
  For QMatrix sensors, specifies that discharge will be terminated when CONSEN out of the most recent 5 comparator samples are positive. For example, a value of 3 in the CONSEN field will terminate discharge when 3 out of the most recent 5 comparator samples are positive. When CONSEN has the default value of 0, discharge will be terminated immediately when the comparator output goes positive.
- **CXDILEN: Cx Capacitor Discharge Length**
  For QMatrix sensors, specifies how many burst prescaler clock cycles the CAT should use to discharge the Cx capacitor at the end of each burst cycle.
- **SYNCTIM: Sync Time Interval**
  When non-zero, determines the number of prescaled clock cycles between the start of the acquisition on each X line for QMatrix acquisition.

32.6.4 Digital Frequency Locked Loop (DFLL) Characteristics

Table 32-14. Digital Frequency Locked Loop Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OUT}	Output frequency[2]		20		150	MHz
f_{REF}	Reference frequency[2]		8		150	kHz
	FINE resolution step	FINE > 100, all COARSE values [3]		0.38		%
	Frequency drift over voltage and temperature	Open loop mode		See Figure 32-4		
	Accuracy[2]	FINE lock, f_{REF} = 32kHz, SSG disabled		0.1	0.5	%
		ACCURATE lock, f_{REF} = 32kHz, dither clk RCSYS/2, SSG disabled		0.06	0.5	
		FINE lock, f_{REF} = 8-150kHz, SSG disabled		0.2	1	
		ACCURATE lock, f_{REF} = 8-150kHz, dither clk RCSYS/2, SSG disabled		0.1	1	
I_{DFLL}	Power consumption			25		µA/MHz
$t_{STARTUP}$	Startup time[2]	Within 90% of final values			100	µs
t_{LOCK}	Lock time	f_{REF} = 32kHz, FINE lock, SSG disabled		8		ms
		f_{REF} = 32kHz, ACCURATE lock, dithering clock = RCSYS/2, SSG disabled		28		

Notes: 1. Spread Spectrum Generator (SSG) is disabled by writing a zero to the EN bit in the DFLL0SSG register.
2. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.
3. The FINE and COARSE values are selected by wrirting to the DFLL0VAL.FINE and DFLL0VAL.COARSE field respectively.

Table 32-20. Flash Endurance and Data Retention

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N_{FARRAY}	Array endurance (write/page)		100k			cycles
N_{FFUSE}	General Purpose fuses endurance (write/bit)		10k			
t_{RET}	Data retention		15			years

32.8 Analog Characteristics

32.8.1 Voltage Regulator Characteristics

Table 32-21. VREG Electrical Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{VDDIN}	Input voltage range		1.98	3.3	3.6	V
$V_{VDDCORE}$	Output voltage, calibrated value	V_{VDDIN} >= 1.98V		1.8		
	Output voltage accuracy[1]	I_{OUT} = 0.1mA to 60mA, V_{VDDIN} > 1.98V		2		%
		I_{OUT} = 0.1mA to 60mA, V_{VDDIN} < 1.98V		4		
I_{OUT}	DC output current[1]	Normal mode			60	mA
		Low power mode			1	
I_{VREG}	Static current of internal regulator	Normal mode		13		µA
		Low power mode		4		

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Table 32-22. Decoupling Requirements

Symbol	Parameter	Condition	Typ	Techno.	Units
C_{IN1}	Input regulator capacitor 1		33		nF
C_{IN2}	Input regulator capacitor 2		100		
C_{IN3}	Input regulator capacitor 3		10		µF
C_{OUT1}	Output regulator capacitor 1		100		nF
C_{OUT2}	Output regulator capacitor 2		2.2	Tantalum 0.5<ESR<10Ohm	µF

Note: 1. Refer to Section 6.1.2 on page 36.