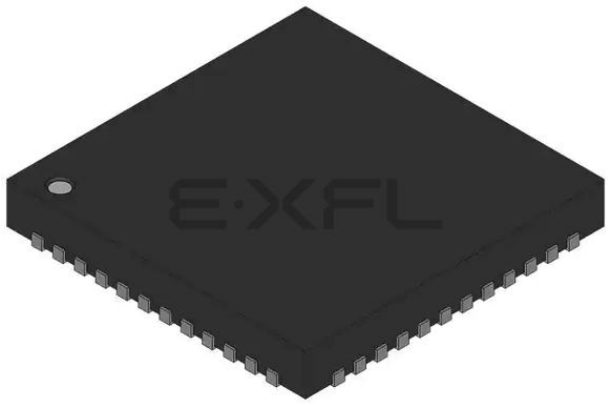
Atmel - ATUC128L4U-D3HT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART, USB
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-UFLGA Exposed Pad
Supplier Device Package	48-TLLGA (5.5x5.5)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atuc128l4u-d3ht

8.7.2.9 Device Frame Number Register

Register Name: UDFNUM

Access Type: Read-Only

Offset: 0x0020

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
FNCERR	-	FNUM[10:5]					

7	6	5	4	3	2	1	0
FNUM[4:0]					-	-	-

- **FNCERR: Frame Number CRC Error**
  This bit is cleared upon receiving a USB reset.
  This bit is set when a corrupted frame number is received. This bit and the SOF interrupt bit are updated at the same time.
- **FNUM: Frame Number**
  This field is cleared upon receiving a USB reset.
  This field contains the 11-bit frame number information, as provided from the last SOF packet.
  FNUM is updated even if a corrupted SOF is received.

Table 9-7. Semantic of PAGEN field in different commands

Command	PAGEN description
Program GP Fuse Byte	WriteData[7:0], ByteAddress[2:0]
Erase All GP Fuses	Not used
Quick Page Read	Page number
Write User Page	Not used
Erase User Page	Not used
Quick Page Read User Page	Not used
High Speed Mode Enable	Not used
High Speed Mode Disable	Not used

- **CMD: Command**
  This field defines the flash command. Issuing any unused command will cause the Programming Error bit in FSR to be set, and the corresponding interrupt to be requested if the PROGE bit in FCR is one.

Table 9-8. Set of commands

Command	Value	Mnemonic
No operation	0	NOP
Write Page	1	WP
Erase Page	2	EP
Clear Page Buffer	3	CPB
Lock region containing given Page	4	LP
Unlock region containing given Page	5	UP
Erase All	6	EA
Write General-Purpose Fuse Bit	7	WGPB
Erase General-Purpose Fuse Bit	8	EGPB
Set Security Bit	9	SSB
Program GP Fuse Byte	10	PGPFB
Erase All GPFuses	11	EAGPF
Quick Page Read	12	QPR
Write User Page	13	WUP
Erase User Page	14	EUP
Quick Page Read User Page	15	QPRUP
High Speed Mode Enable	16	HSEN
High Speed Mode Disable	17	HSDIS
RESERVED	16-31	

11. HSB Bus Matrix (HMATRIXB)

Rev: 1.3.0.3

11.1 Features

- **User Interface on peripheral bus**
- **Configurable number of masters (up to 16)**
- **Configurable number of slaves (up to 16)**
- **One decoder for each master**
- **Programmable arbitration for each slave**
  - **Round-Robin**
  - **Fixed priority**
- **Programmable default master for each slave**
  - **No default master**
  - **Last accessed default master**
  - **Fixed default master**
- **One cycle latency for the first access of a burst**
- **Zero cycle latency for default master**
- **One special function register for each slave (not dedicated)**

11.2 Overview

The Bus Matrix implements a multi-layer bus structure, that enables parallel access paths between multiple High Speed Bus (HSB) masters and slaves in a system, thus increasing the overall bandwidth. The Bus Matrix interconnects up to 16 HSB Masters to up to 16 HSB Slaves. The normal latency to connect a master to a slave is one cycle except for the default master of the accessed slave which is connected directly (zero cycle latency). The Bus Matrix provides 16 Special Function Registers (SFR) that allow the Bus Matrix to support application specific features.

11.3 Product Dependencies

In order to configure this module by accessing the user registers, other parts of the system must be configured correctly, as described below.

11.3.1 Clocks

The clock for the HMATRIX bus interface (CLK_HMATRIX) is generated by the Power Manager. This clock is enabled at reset, and can be disabled in the Power Manager.

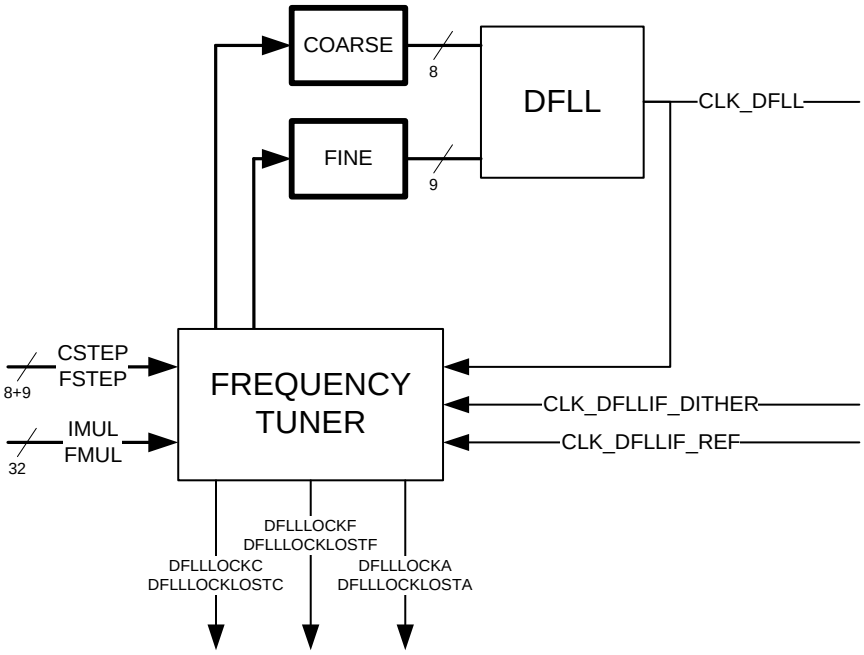
11.4 Functional Description

11.4.1 Special Bus Granting Mechanism

The Bus Matrix provides some speculative bus granting techniques in order to anticipate access requests from some masters. This mechanism reduces latency at first access of a burst or single transfer. This bus granting mechanism sets a different default master for every slave.

At the end of the current access, if no other request is pending, the slave remains connected to its associated default master. A slave can be associated with three kinds of default masters: no default master, last access master, and fixed default master.

Figure 14-2. DFLLIF Block Diagram



14.5.4.1 *Enabling the DFLL*

The DFLL is enabled by writing a one to the Enable bit (EN) in the DFLLn Configuration Register (DFLLnCONF). No other bits or fields in DFLLnCONF must be changed simultaneously, or before the DFLL is enabled.

14.5.4.2 *Internal synchronization*

Due to multiple clock domains in the DFLLIF, values in the DFLLIF configuration registers need to be synchronized to other clock domains. The status of this synchronization can be read from the Power and Clocks Status Register (PCLKSR). Before writing to a DFLLIF configuration register, the user must check that the DFLLn Synchronization Ready bit (DFLLnRDY) in PCLKSR is set. When this bit is set, the DFLL can be configured, and CLK_DFLL is ready to be used. Any write to a DFLLIF configuration register while DFLLnRDY is cleared will be ignored.

Before reading the value in any of the DFLL configuration registers a one must be written to the Synchronization bit (SYNC) in the DFLLn Synchronization Register (DFLLnSYNC). The DFLL configuration registers are ready to be read when PCLKSR.DFLLnRDY is set.

14.5.4.3 *Disabling the DFLL*

The DFLL is disabled by writing a zero to DFLLnCONF.EN. No other bits or fields in DFLLnCONF must be changed simultaneously.

After disabling the DFLL, PCLKSR.DFLLnRDY will not be set. It is not required to wait for PCLKSR.DFLLnRDY to be set before re-enabling the DFLL.

14.5.4.4 *Open loop operation*

After enabling the DFLL, open loop mode is selected by writing a zero to the Mode Selection bit (MODE) in DFLLnCONF. When operating in open loop mode the output frequency of the DFLL will be determined by the values written to the Coarse Calibration Value field (COARSE) and the Fine Calibration Value field (FINE) in the DFLLnCONF register. When writing to COARSE and

14.6.37 Voltage Regulator Version Register

Name: VREGIFBVERSION

Access Type: Read-only

Offset: 0x03D8

Reset Value: -

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

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

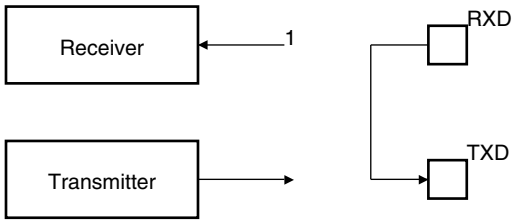
15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

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

- **VARIANT: Variant number**
  Reserved. No functionality associated.
- **VERSION: Version number**
  Version number of the module. No functionality associated.

- **DAR: WDT Disable After Reset**
  0: After a watchdog reset, the WDT will still be enabled.
  1: After a watchdog reset, the WDT will be disabled.
- **EN: WDT Enable**
  0: WDT is disabled.
  1: WDT is enabled.
  After writing to this bit the read back value will not change until the WDT is enabled/disabled. This due to internal synchronization.

Figure 20-39. Remote Loopback Mode Configuration



20.6.12 Write Protection Registers

To prevent single software errors from corrupting USART behavior, certain address spaces can be write-protected by writing the correct Write Protect KEY and a one to the Write Protect Enable bit in the Write Protect Mode Register (WPMR.WPKEY, and WPMR.WPEN). Disabling the write protection is done by writing the correct key, and a zero to WPEN.

Write attempts to a write protected register are detected and the Write Protect Violation Status bit in the Write Protect Status Register (WPSR.WPVS) is set, while the Write Protect Violation Source field (WPSR.WPVSRC) indicates the targeted register. Writing the correct key to the Write Protect KEY bit (WPMR.WPKEY) clears WPVSRC and WPVS.

The protected registers are:

- ”Mode Register” on page 466
- ”Baud Rate Generator Register” on page 476
- ”Receiver Time-out Register” on page 477
- ”Transmitter Timeguard Register” on page 478

Figure 21-8. Peripheral Deselection

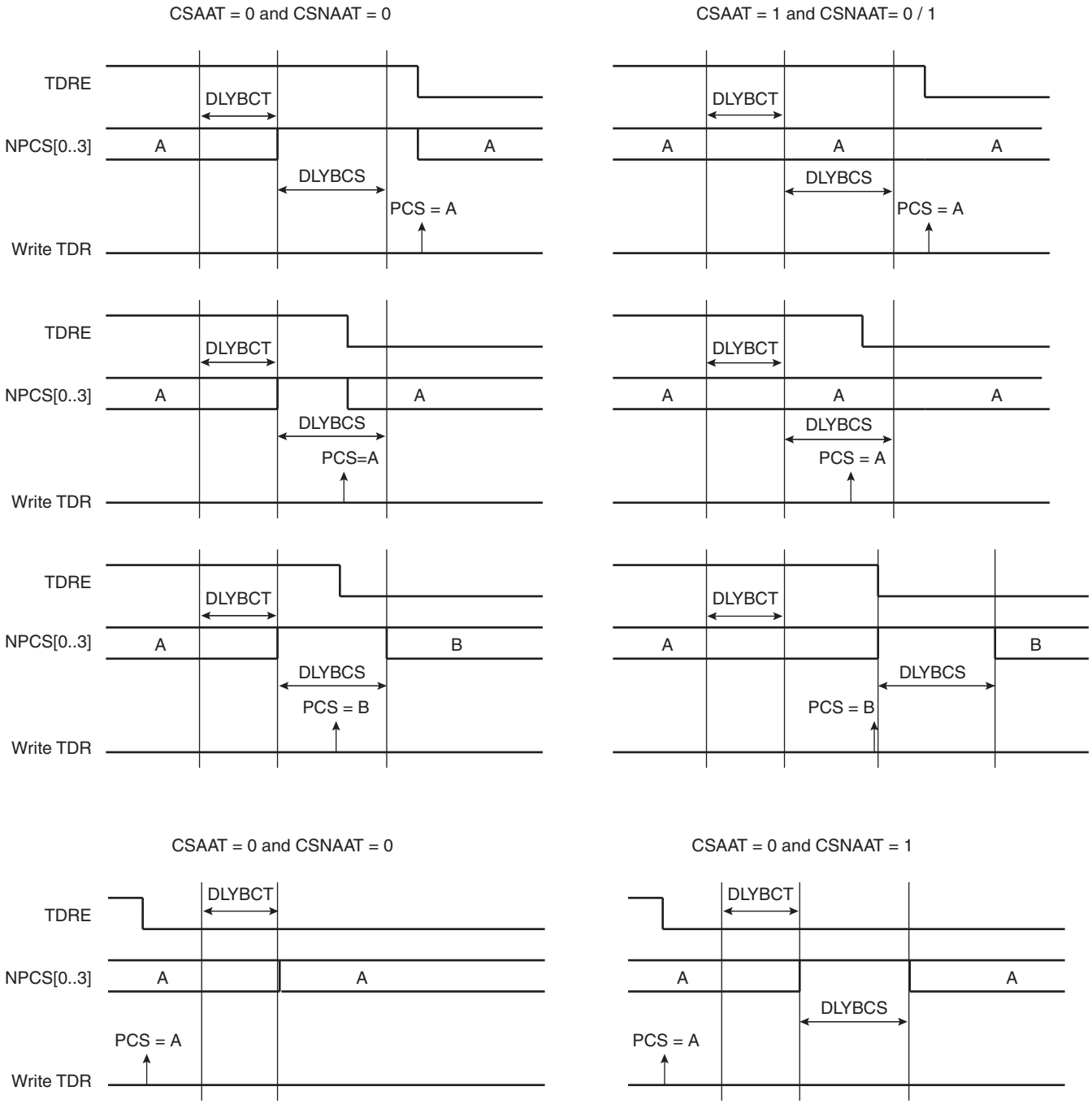


Figure 21-8 on page 496 shows different peripheral deselection cases and the effect of the CSRn.CSAAT and CSRn.CSNAAT bits.

21.7.3.9 Mode fault detection

The SPI is capable of detecting a mode fault when it is configured in master mode and NPCS0, MOSI, MISO, and SPCK are configured as open drain through the I/O Controller with either internal or external pullup resistors. If the I/O Controller does not have open-drain capability, mode fault detection **must** be disabled by writing a one to the Mode Fault Detection bit in the MR

22.8.8 Ten Bit Addressing

Writing a one to CMDR.TENBIT enables 10-bit addressing in hardware. Performing transfers with 10-bit addressing is similar to transfers with 7-bit addresses, except that bits 9:7 of CMDR.SADR must be written appropriately.

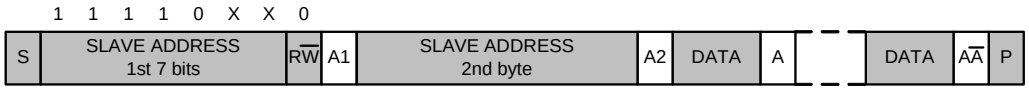
In Figure 22-14 and Figure 22-15, the grey boxes represent signals driven by the master, the white boxes are driven by the slave.

22.8.8.1 Master Transmitter

To perform a master transmitter transfer:

1. Write CMDR with TENBIT=1, REPSAME=0, READ=0, START=1, STOP=1 and the desired address and NBYTES value.

Figure 22-14. A Write Transfer with 10-bit Addressing



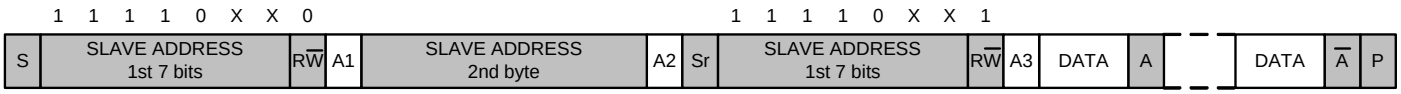
22.8.8.2 Master Receiver

When using master receiver mode with 10-bit addressing, CMDR.REPSAME must also be controlled. CMDR.REPSAME must be written to one when the address phase of the transfer should consist of only 1 address byte (the 11110xx byte) and not 2 address bytes. The I²C standard specifies that such addressing is required when addressing a slave for reads using 10-bit addressing.

To perform a master receiver transfer:

1. Write CMDR with TENBIT=1, REPSAME=0, READ=0, START=1, STOP=0, NBYTES=0 and the desired address.
2. Write NCMDR with TENBIT=1, REPSAME=1, READ=1, START=1, STOP=1 and the desired address and NBYTES value.

Figure 22-15. A Read Transfer with 10-bit Addressing



22.8.9 SMBus Mode

SMBus mode is enabled and disabled by writing to the SMEN and SMDIS bits 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 into SMBTR.
- 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.

22.10 Module Configuration

The specific configuration for each TWIM 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 22-7. Module Clock Name

Module Name	Clock Name	Description
TWIM0	CLK_TWIM0	Clock for the TWIM0 bus interface
TWIM1	CLK_TWIM1	Clock for the TWIM1 bus interface

Table 22-8. Register Reset Values

Register	Reset Value
VERSION	0x00000110
PARAMETER	0x00000000

23. Two-wire Slave Interface (TWIS)

Rev.: 1.2.0.1

23.1 Features

- **Compatible with I²C standard**
  - **Transfer speeds of 100 and 400 kbit/s**
  - **7 and 10-bit and General Call addressing**
- **Compatible with SMBus standard**
  - **Hardware Packet Error Checking (CRC) generation and verification with ACK response**
  - **SMBALERT interface**
  - **25 ms clock low timeout delay**
  - **25 ms slave cumulative clock low extend time**
- **Compatible with PMBus**
- **DMA interface for reducing CPU load**
- **Arbitrary transfer lengths, including 0 data bytes**
- **Optional clock stretching if transmit or receive buffers not ready for data transfer**
- **32-bit Peripheral Bus interface for configuration of the interface**

23.2 Overview

The Atmel Two-wire Slave Interface (TWIS) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 kbit/s, based on a byte-oriented transfer format. It can be used with any Atmel Two-wire Interface bus, I²C, or SMBus-compatible master. The TWIS is always a bus slave and can transfer sequential or single bytes.

Below, Table 23-1 lists the compatibility level of the Atmel Two-wire Slave Interface and a full I²C compatible device.

Table 23-1. Atmel TWIS Compatibility with I²C Standard

I²C Standard	Atmel TWIS
Standard-mode (100 kbit/s)	Supported
Fast-mode (400 kbit/s)	Supported
7 or 10 bits Slave Addressing	Supported
START BYTE(1)	Not Supported
Repeated Start (Sr) Condition	Supported
ACK and NAK Management	Supported
Slope control and input filtering (Fast mode)	Supported
Clock stretching	Supported

Note: 1. START + b000000001 + Ack + Sr

- **SWRST: Software Reset**
  This bit will always read as 0.
  Writing a zero to this bit has no effect.
  Writing a one to this bit resets the TWIS.
- **STREN: Clock Stretch Enable**
  0: Disables clock stretching if RHR/THR buffer full/empty. May cause over/underrun.
  1: Enables clock stretching if RHR/THR buffer full/empty.
- **GCMATCH: General Call Address Match**
  0: Causes the TWIS not to acknowledge the General Call Address.
  1: Causes the TWIS to acknowledge the General Call Address.
- **SMATCH: Slave Address Match**
  0: Causes the TWIS not to acknowledge the Slave Address.
  1: Causes the TWIS to acknowledge the Slave Address.
- **SMEN: SMBus Mode Enable**
  0: Disables SMBus mode.
  1: Enables SMBus mode.
- **SEN: Slave Enable**
  0: Disables the slave interface.
  1: Enables the slave interface.

Figure 28-2. Output signals with CMOC=0

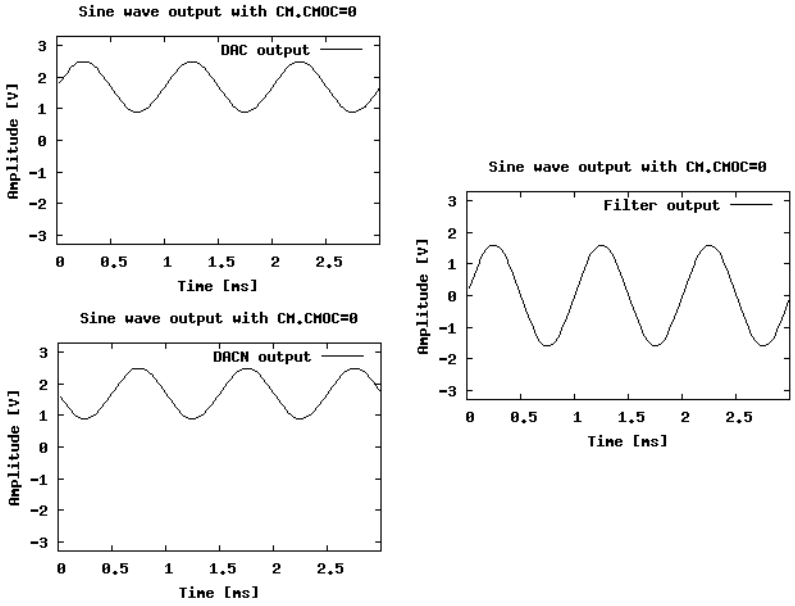
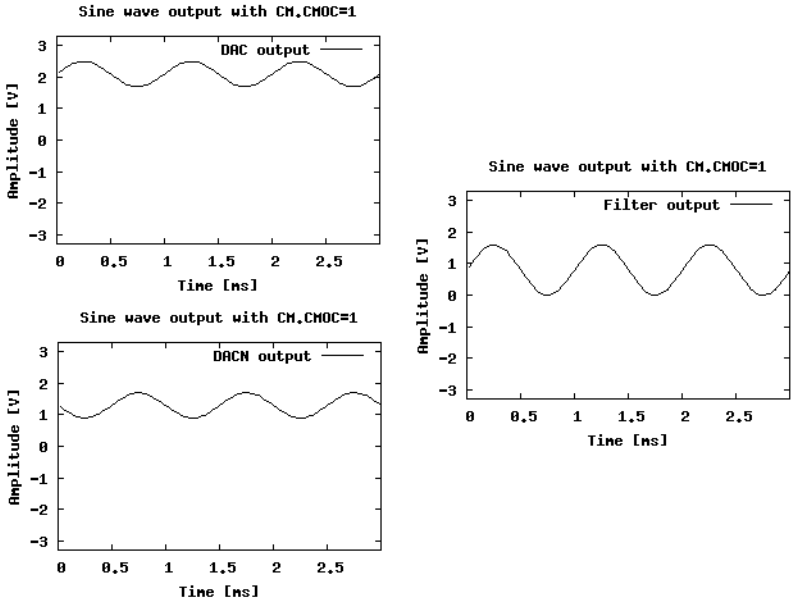


Figure 28-3. Output signals with CMOC=1



28.6.7 Volume Control

The Audio Bitstream DAC have two volume control registers, Volume Control Register 0 (VCR0) and Volume Control Register 1 (VCR1), that can be used to adjust the volume for the corresponding channel. The volume control is linear and will only scale each sample according to the value in the Volume Control (VOLUME) field in the volume control registers. The register also has a Mute bit (MUTE) which can be used to mute the corresponding channel. The filtered out-

29.9.17 Channel Status Register

Name: CHSR

Access Type: Read-only

Offset: 0x48

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
23	22	21	20	19	18	17	16
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
15	14	13	12	11	10	9	8
CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
7	6	5	4	3	2	1	0
CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

- **CHn: Channel N Status**

  0: The corresponding channel is disabled.

  1: The corresponding channel is enabled.

  A bit in this register is cleared by writing a one to the corresponding bit in Channel Disable Register (CHDR).

  A bit in this register is set by writing a one to the corresponding bit in Channel Enable Register (CHER).

  The number of available channels is device dependent. Please refer to the Module Configuration section at the end of this chapter for information regarding how many channels are implemented.

30.9.3 Interrupt Enable Register

Name: IER

Access Type: Write-only

Offset: 0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	WFINT3	WFINT2	WFINT1	WFINT0
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
SUTINT7	ACINT7	SUTINT6	ACINT6	SUTINT5	ACINT5	SUTINT4	ACINT4
7	6	5	4	3	2	1	0
SUTINT3	ACINT3	SUTINT2	ACINT2	SUTINT1	ACINT1	SUTINT0	ACINT0

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.

32.8 Module Configuration

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

Table 32-4. GLOC Configuration

Feature	GLOC
Number of LUT units	2

Table 32-5. GLOC Clocks

Clock Name	Description
CLK_GLOC	Clock for the GLOC bus interface
GCLK	The generic clock used for the GLOC is GCLK5

Table 32-6. Register Reset Values

Register	Reset Value
VERSION	0x00000100
PARAMETER	0x00000002

33.7.3 Status Clear Register

Name: SCR

Access Type: Write-only

Offset: 0x08

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
-	-	TRMIS	-	-	OVERRUN	DREADYINT	READYINT
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in SR and the corresponding interrupt request.

33.7.9 Baud Rate Register

Name: BRR

Access Type: Read/Write

Offset: 0x20

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
BR[15:8]							

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

- **BR: Baud Rate**

  The baud rate (f_{br}) of the transmission, calculated using the following formula (f_{aw} is the RC120M frequency):

$$f_{br} = \frac{8 f_{aw}}{BR}$$

  BR should not be set to a value smaller than 32.
  Writing a value to this field will update the baud rate of the transmission.
  Reading this field will give the current baud rate of the transmission.

33.8 Module Configuration

The specific configuration for each aWire 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 33-4. AW Clocks

Clock Name	Description
CLK_AW	Clock for the AW bus interface

Table 33-5. Register Reset Values

Register	Reset Value
VERSION	0x00000230

38.5.7 TWI

1. **TWIS may not wake the device from sleep mode**
   If the CPU is put to a sleep mode (except Idle and Frozen) directly after a TWI Start condition, the CPU may not wake upon a TWIS address match. The request is NACKed.
   Fix/Workaround
   When using the TWI address match to wake the device from sleep, do not switch to sleep modes deeper than Frozen. Another solution is to enable asynchronous EIC wake on the TWIS clock (TWCK) or TWIS data (TWD) pins, in order to wake the system up on bus events.

2. **SMBALERT bit may be set after reset**
   The SMBus Alert (SMBALERT) bit in the Status Register (SR) might be erroneously set after system reset.
   Fix/Workaround
   After system reset, clear the SR.SMBALERT bit before commencing any TWI transfer.

3. **Clearing the NAK bit before the BTF bit is set locks up the TWI bus**
   When the TWIS is in transmit mode, clearing the NAK Received (NAK) bit of the Status Register (SR) before the end of the Acknowledge/Not Acknowledge cycle will cause the TWIS to attempt to continue transmitting data, thus locking up the bus.
   Fix/Workaround
   Clear SR.NAK only after the Byte Transfer Finished (BTF) bit of the same register has been set.

4. **TWIS stretch on Address match error**
   When the TWIS stretches TWCK due to a slave address match, it also holds TWD low for the same duration if it is to be receiving data. When TWIS releases TWCK, it releases TWD at the same time. This can cause a TWI timing violation.
   Fix/Workaround
   None.

5. **TWIM TWALM polarity is wrong**
   The TWALM signal in the TWIM is active high instead of active low.
   Fix/Workaround
   Use an external inverter to invert the signal going into the TWIM. When using both TWIM and TWIS on the same pins, the TWALM cannot be used.

38.5.8 PWMA

1. **The SR.READY bit cannot be cleared by writing to SCR.READY**
   The Ready bit in the Status Register will not be cleared when writing a one to the corresponding bit in the Status Clear register. The Ready bit will be cleared when the Busy bit is set.
   Fix/Workaround
   Disable the Ready interrupt in the interrupt handler when receiving the interrupt. When an operation that triggers the Busy/Ready bit is started, wait until the ready bit is low in the Status Register before enabling the interrupt.

38.5.9 TC

1. **Channel chaining skips first pulse for upper channel**
   When chaining two channels using the Block Mode Register, the first pulse of the clock between the channels is skipped.