TCD2210/2220(-E)

Digital Interface Communications Engine

User Guide

Revision 1.02

July 29, 2010

IMPORTANT NOTICE

TC Applied Technologies Ltd. ("TCAT") believes that the information contained herein was accurate and reliable at time of writing. However, the information is subject to change without notice and is provided "AS IS" without warranty of any kind (express or implied), and TCAT reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time, and to discontinue any product or service without notice. Customers are advised to obtain the latest version of any and all relevant information to verify, before placing orders or beginning development of products based on TCAT technologies, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, indemnification, and limitation of liability.

No responsibility is assumed by TCAT for the use of this information, including use of this information as the basis for development, manufacture or sale of any items, or for infringement of patents or other rights of third parties. This document is the property of TCAT; by furnishing this information, TCAT grants no license, express or implied, under any patents, mask work rights, copyrights, trademarks, trade secrets or other intellectual property rights. TCAT owns the copyrights associated with the information contained herein and gives consent for copies to be made of the information only for use within your organization with respect to TCAT integrated circuits, software, design files and any other products of TCAT. This consent does not extend to other copying such as copying for general distribution, advertising or promotional purposes, or for creating any work for resale. Resale of TCAT products or services with statements different from or beyond the parameters stated by TCAT for that product or service voids all express and any implied warranties for the associated TCAT product or service and is an unfair and deceptive business practice. TCAT is not responsible or liable for any such statements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (“CRITICAL APPLICATIONS”). TCAT PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED FOR USE IN AIRCRAFT SYSTEMS, MILITARY APPLICATIONS, PRODUCTS SURGICALLY IMPLANTED INTO THE BODY, AUTOMOTIVE SAFETY OR SECURITY DEVICES, LIFE SUPPORT PRODUCTS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TCAT PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER’S RISK AND TCAT DISCLAIMS AND MAKES NO WARRANTY, EXPRESS, STATUTORY OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE, WITH REGARD TO ANY TCAT PRODUCT THAT IS USED IN SUCH A MANNER. IF THE CUSTOMER OR CUSTOMER’S CUSTOMER USES OR PERMITS THE USE OF TCAT PRODUCTS IN CRITICAL APPLICATIONS, CUSTOMER AGREES, BY SUCH USE, TO FULLY INDEMNIFY TCAT, ITS OFFICERS, DIRECTORS, EMPLOYEES, DISTRIBUTORS AND OTHER AGENTS FROM ANY AND ALL LIABILITY, INCLUDING ATTORNEYS’ FEES AND COSTS, THAT MAY RESULT FROM OR ARISE IN CONNECTION WITH THESE USES.

TC Applied Technologies, TCAT and the TC Applied Technologies, DICE^{TM} and JetPLL^{TM} logo designs are trademarks of TC Applied Technologies Ltd. All other brand and product names in this document may be trademarks or service marks of their respective owners.

Table of Contents

List of Figures

List of Tables

Chapter 1 About DICE TCD22xx

DICE 22xx family of chips contains the following members:

- **TCD2220** Full version with dual audio port, this chip is in a LQFP 144 package. This chip is also known as **DICE JR**

- **TCD2210** Reduced version with one audio port, this chip is in a QFP 128 package. This chip is also known as **DICE Mini.**

Both devices are in a **LEAD FREE** package and are **RHOS** compliant.

Ordering information

ID	RHOS	Temp.	Package
TCD2210	√	0 **°C** to 70 **°C**	QFP 128
TCD2210-E	√	-40 **°C** to 85 **°C**	QFP 128
TCD2220	√	-0 **°C** to 70 **°C**	LQFP 144
TCD2220-E	√	-40 **°C** to 85 **°C**	LQFP 144

Table 1: Ordering Information

1.1 Introduction

The TCD22xx chip family covers a wide range of audio applications, professional as well as consumer. Apart from its IEEE1394 audio streaming capability the chip features all common digital audio interfaces and a 50MHz 32 bit RISC processor including a wide range of peripherals. The DICE cross bar router allows any audio sink to connect to any audio source on a per channel basis. The IEEE1394 streaming engine can handle a total of 32 input channels and 32 output channels distributed on several isochronous channels.

1.2 Block Diagram

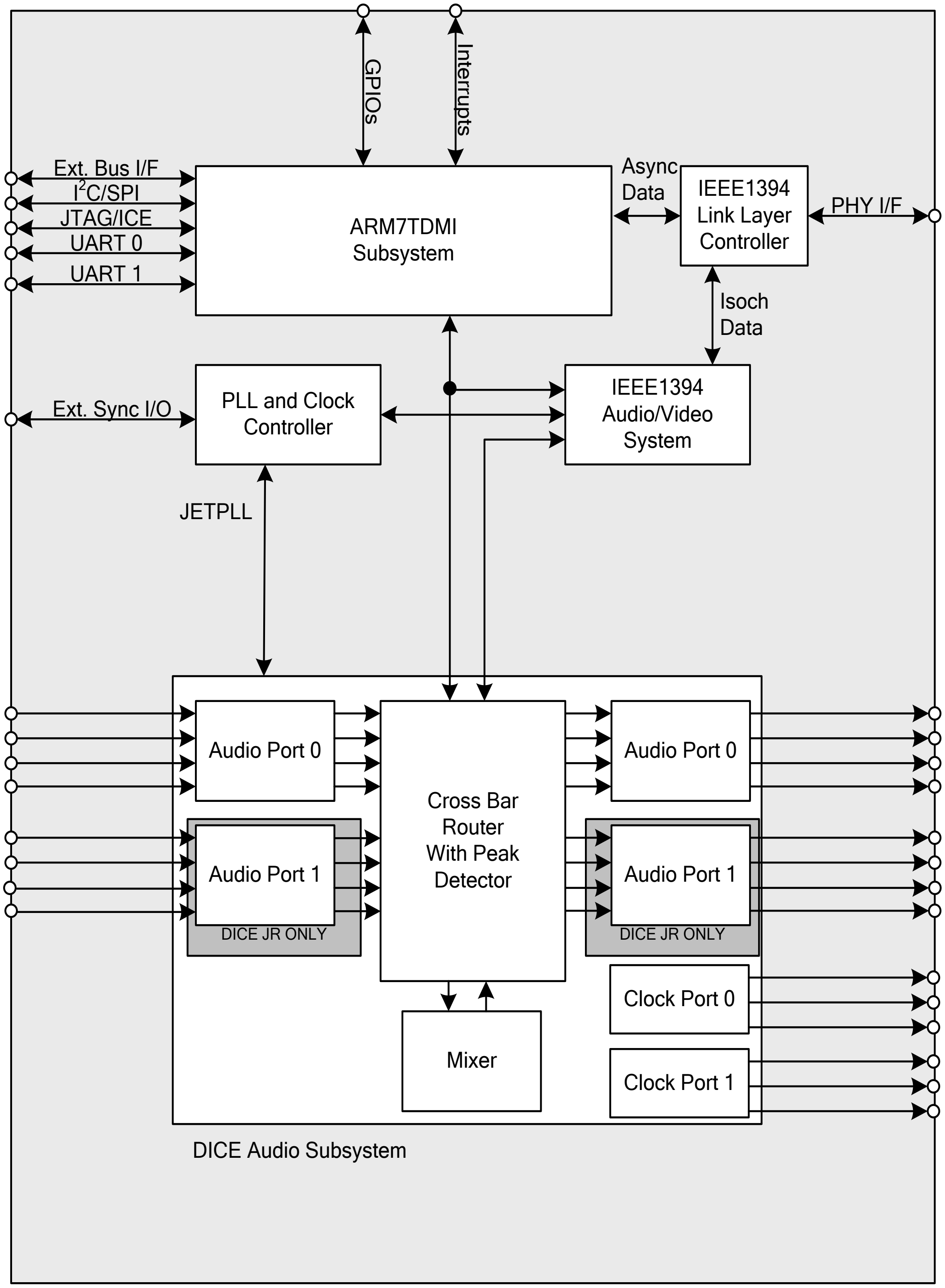


Figure 1: DICE 22xx Block Diagram

1.3 Chip Features

CPU core

- Full 32-bit ARM7TDMI RISC processor
- 32-bit internal bus
- 16-bit Thumb mode
- 16 Kb 0 wait state RAM
- 15 general-purpose 32-bit registers
- 32-bit program counter and status register
- 5 supervisor modes, 1 user mode
- External Bus Interface (EBI)
- Remap of Internal RAM during boot.

I2C Interface

- Standard and Full Speed support
- Slave mode with address match logic
- Master Mode
- 10 bit and 7 bit addressing mode
- 16 deep FIFO buffer

SPI Interface

- Master and Slave mode
- GPIO used for Slave Select
- Interrupt on Byte transfer complete

Dual Timer Unit

- 32 bit counter
- Free running and user-defined count
- Interrupt on counter wrap
- Clocked by CPU clock

Watch Dog

Dual Universal Asynchronous Receiver Transmitter (UART)

- Industry standard 16550 Compliant
- 16 deep receive and transmit FIFOs
- Supports all standard RS232 Rates
- Supports MIDI rate

General Purpose Input Output (GPIO)

- 15 individual ports
- Each port configurable as input or output

- Each port configurable for level or edge sensitive interrupts
- Configurable deglitching logic for each port

Dual Rotary Encoder Interface (Gray Decoder)

- individual rotary encoder counters
- 8 bit signed counter per port
- Configurable interrupt on value change

IEEE 1394 Link Layer Controller (LLC)

- IEEE 1394a compliant LLC
- Compliant PHY interface
- Support for isolation barrier
- 512x32 FIFO for asynchronous communication

Digital Interface Communication Engine (DICE)

- **Jet^{TM} PLL**
- Cross-bar router with peak detector.
- 2 (1) generic audio port.
  - 4 x 2 ch. of I2S Per port (32KHz to 192KHz)
  - 4 x 4 ch. of I4S per port (32KHz to 192KHz)
  - 2 x 8 ch. of I8S per port (32KHz to 96KHz)
  - 4 x 2 ch. of AES, port 1 or port 2 (32KHz to 192KHz)
  - 2 x 8 ch. of ADAT, port 1 only (8 ch. @96KHz, 4 ch @ 192KHz)
- ARM Audio Receiver/Transmitter, 8 channels (4 ch @ 192KHz)
- IEC 61883-6 Isoc. Receiver, 32 channels (16 ch @ 192KHz)
- IEC 61883-6 Isoc. Transmitter, 32 channels (16 ch @ 192KHz)

Power and operating voltage

- 950 mW maximum, 500 mW typical (TBD)
- 3.3 volts - I/O
- 1.8 volts – core

1.4 Package

Product TCD2220 is only available in LQFP144 package. Product TCD2210 is only available in QFP128 package. Both packages are LEAD FREE.

Please refer to the TCD22xxHardwareGuide for details on the packages.

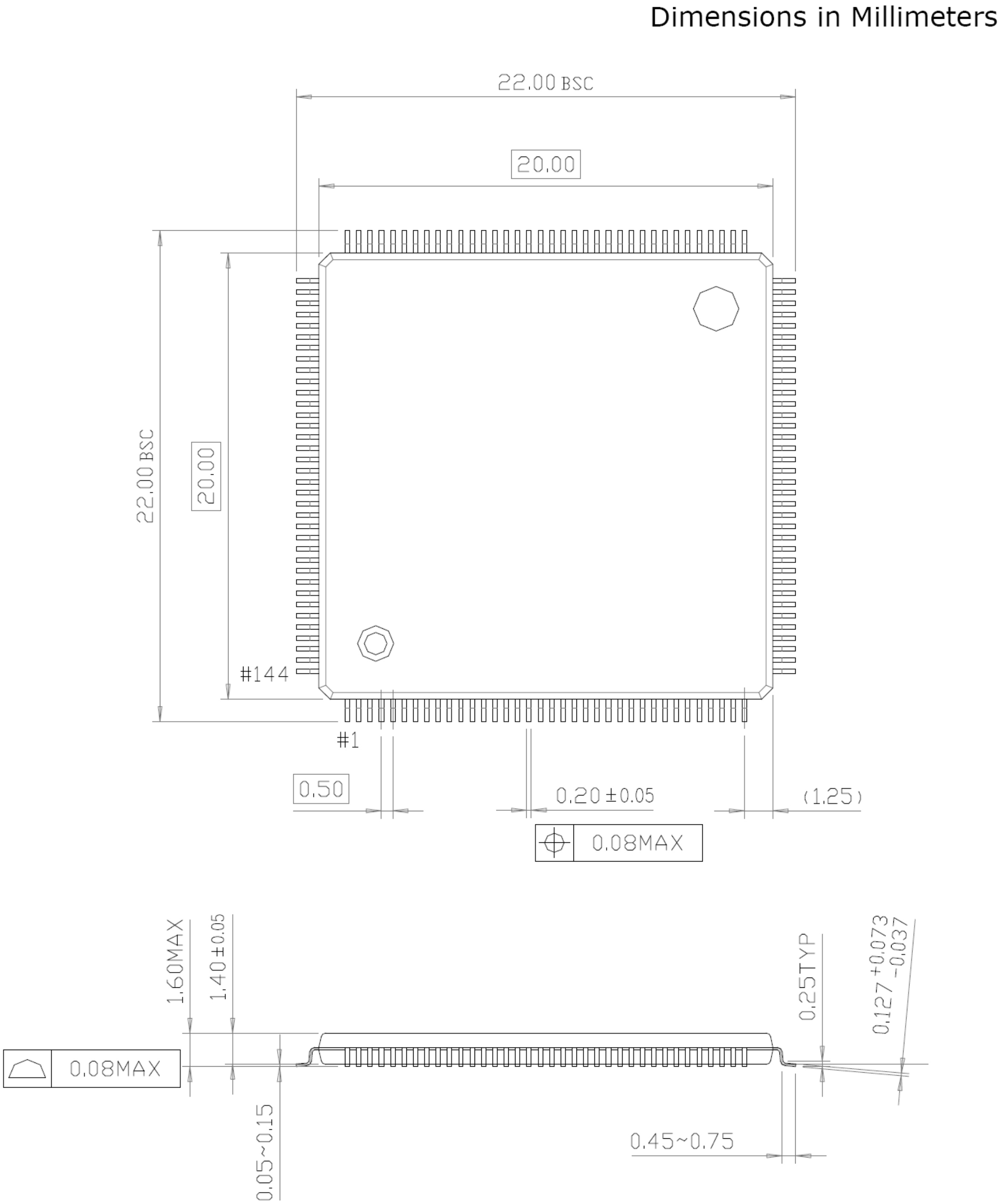


Figure 2: TCD2220, LQFP144

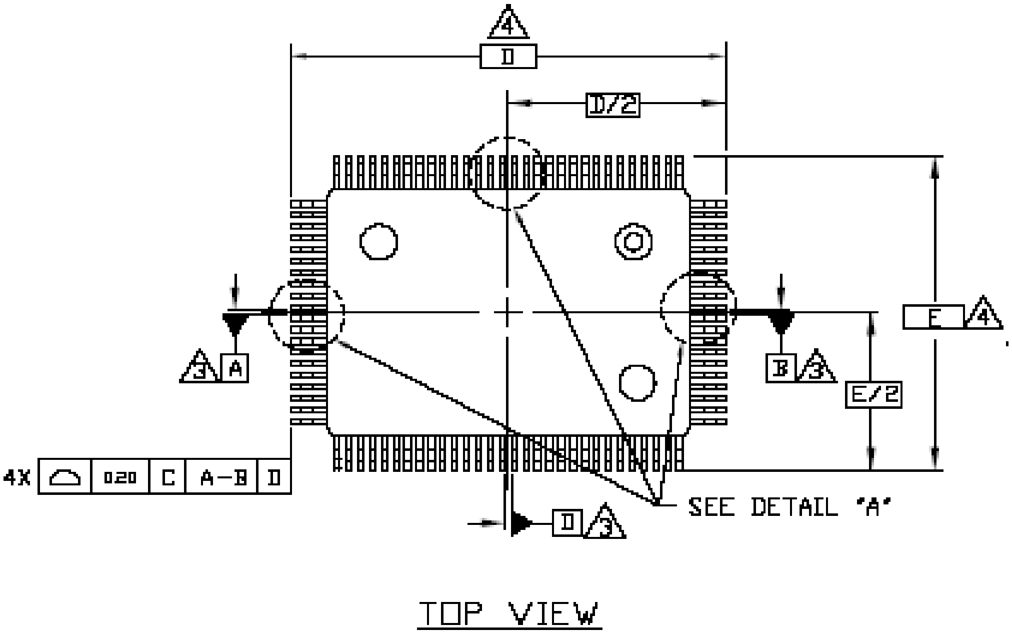


Figure 3: TCD2210, QFP128

1.5 Signal Description

The following table lists each I/O signal for the TCD2210 and TCD2220. Note that the chips use a number of shared pins, whereby each shared pin can be configured to contain different signals. The GPCSR module is used to configure the shared pins for a particular signal. The shared pins and their multiple functions are also listed in a table that follows the table below. Note that all the shared pins are bi-directional.

Pins that not available in TCD2210 (QFP128) are marked as N/A in the following table.

1.5.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
Data Bus					
D0	144	1	I/O (S[1])	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU[2],5V[3])
D1	1	2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D2	2	3	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D3	3	4	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D4	4	5	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D5	5	6	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D6	6	7	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D7	7	8	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D8	8	9	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D9	9	10	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)

[1] S indicates Schmitt Trigger Input

[2] PU indicates that internal Pull-Up resistor is present on PAD

[3] 5V indicates that the input is 5V tolerant

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
D10	10	11	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D11	13	14	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D12	14	15	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D13	15	16	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D14	16	17	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D15	17	18	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
Address Bus					
A0	18	19	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A1	19	20	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A2	20	21	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A3	21	22	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A4	22	23	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A5	23	24	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A6	26	27	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A7	27	28	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A8	28	29	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A9	29	30	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A10	30	31	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A11	31	32	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A12	32	33	O	8	Address Bus. Shared address pins

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
					for SDRAM and Static memory.
A13	33	34	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A14	34	35	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A15	35	36	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A16	36	37	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A17	37	38	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A18	38	39	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A19	39	40	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
Chip Selects					
CS0*	136	121	O	4	Shared SDRAM and Static Memory Chip Selects
CS1*	46	45	O	4	Shared SDRAM and Static Memory Chip Selects
CS2*	138 (shared)	123 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
CS3*	139 (shared)	124 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
Grey Code Rotary Encoder					
EN1_A	138 (shared)	123 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN1_B	139 (shared)	124 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_A	65 (shared)	N/A	I (S)	6	Rotary Encoder Input (5V)
EN2_B	66 (shared)	N/A	I (S)	6	Rotary Encoder Input (5V)
General Purpose I/O					
GPIO0	42 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO1	138 (shared)	123 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO2	139 (shared)	124 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO3	137 (shared)	122 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO4	85 (shared)	78 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO5	86 (shared)	79 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO6	117 (shared)	106 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO7	118 (shared)	107 (shared)	I/O (S)	6	General Purpose I/O (5V)

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
GPIO8	119 (shared)	108 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO9	55	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO10	56	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO11	57 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO12	65 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO13	66 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO14	67 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
RAM Clock					
CLKO	40	41	O	8	SDRAM Interface AHB Bus Clock (Z[4])
SDRAM Dedicated Signals					
CLKE	42 (shared)	N/A	O	6	SDRAM Interface Clock Enable
RAS*	41	42	O	8	SDRAM Interface Row Address Strobe
CAS*	44	43	O	8	SDRAM Interface Column Address Strobe
SDRAM_WE	45	44	O	8	SDRAM Interface Write Enable
SDRAM_DQM0	47	46	O	8	SDRAM Interface Lower byte mask
SDRAM_DQM1	48	47	O	8	SDRAM Interface Upper byte mask
SDRAM_BNK0	52	51	O	8	SDRAM Interface Bank Address
SDRAM_BNK1	53	52	O	8	SDRAM Interface Bank Address
SDRAM_A10	43	N/A	O	8	SDRAM Precharge A10
SRAM Interface					
SRAM_READY	137(shared)	122 (shared)	I	6	SRAM ready
SRAM_BS[0]	140	125	O	4	SRAM lower byte select
SRAM_BS[1]	141	126	O	4	SRAM upper byte select
SRAM_WE*	142	127	O	8	SRAM write enable
SRAM_OE*	143	128	O	8	SRAM output enable
Phy Interface					
SCLK	54	53	I (S)	-	49.152MHz PHY Clock input
PHD0	58	54	I/O (S)	8	PHY tristatable data line bit 0
PHD1	59	55	I/O (S)	8	PHY tristatable data line bit 1

[4] Z indicates that the output is Z-stateable

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
PHD2	60	56	I/O (S)	8	PHY tristatable data line bit 2
PHD3	63	59	I/O (S)	8	PHY tristatable data line bit 3
PHD4	64	60	I/O (S)	8	PHY tristatable data line bit 4
PHD5	68	61	I/O (S)	8	PHY tristatable data line bit 5
PHD6	69	62	I/O (S)	8	PHY tristatable data line bit 6
PHD7	70	63	I/O (S)	8	PHY tristatable data line bit 7
PHCT0	71	64	I/O (S)	8	PHY tristatable control line bit 0
PHCT1	72	65	I/O (S)	8	PHY tristatable control line bit 1
PHDI	73	66	I (S)	-	A high indicates isolation barrier is not present (PU, 5V)
PHLR	74	67	O	8	Serial request output from S-LINK (Z)
PHLP	75	68	O	4	Link power status. Pulsing if isolation barrier present
PHLO	76	69	I (S)	-	Link on indication from PHY. Pulsing when asserted (PU, 5V)
Word Clock					
WCLK_IN0	85 (shared)	77 (shared)	I (S)		Word Clock In (5V)
WCLK_IN1	65 (shared)	N/A	I (S)		Word Clock In (5V)
WCLK_OUT0	86 (shared)	79 (shared)	O	6	Word Clock Out
WCLK_OUT1	66 (shared)	N/A	O	6	Word Clock Out
External Sample Clocks					
EXT_FBR	85 (shared)	78 (shared)	I/O (S)	6	External 1fs base rate clock (5V)
EXT_512BR	86 (shared)	79(shared)	I/O (S)	6	External 512 x base rate clock (5V)
Audio Port 0					
DO0_0	122	111	O	2	Audio port 0 data out 0
DO0_1	123	112	O	2	Audio port 0 data out 1
DO0_2	124	113	O	2	Audio port 0 data out 2
DO0_3	125	114	O	2	Audio port 0 data out 3
DI0_0	109	102	I	-	Audio port 0 data in 0
DI0_1	110	103	I	-	Audio port 0 data in 1
DI0_2	111	104	I	-	Audio port 0 data in 2
DI0_3	112	105	I	-	Audio port 0 data in 3

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
Audio Port 1					
DO1_0	126	N/A	O	2	Audio port 1 data out 0
DO1_1	127	N/A	O	2	Audio port 1 data out 1
DO1_2	128	N/A	O	2	Audio port 1 data out 2
DO1_3	129	N/A	O	2	Audio port 1 data out 3
DI1_0	113	N/A	I	-	Audio port 1 data in 0
DI1_1	114	N/A	I	-	Audio port 1 data in 1
DI1_2	115	N/A	I	-	Audio port 1 data in 2
DI1_3	116	N/A	I	-	Audio port 1 data in 3
Audio Clock Ports					
MCK0	106	99	O	8	Master Clock 0
FCK0	107	100	O	8	Frame Clock 0 (LR Clock)
BCK0	108	101	O	8	Bit Clock 0
MCK1	117 (shared)	106 (shared)	O	6	Master Clock 1
FCK1	118 (shared)	107 (shared)	O	6	Frame Clock 1 (LR Clock)
BCK1	119 (shared)	108 (shared)	O	6	Bit Clock 1
XTAL					
XTAL2	102	95	O	-	XTAL for clock doubler/power manager/LLC
XTAL1	103	96	I	-	XTAL for clock doubler/power manager/LLC
Reset					
RESET*	79	72	I (S)	-	Reset – active low (PU, 5V)
PLL					
PLLE	51	50	I	-	PLL Enable (5V)
Test					
TEMO	77	70	I	-	Test mode pin (PD[5], 5V)
SCMO	78	71	I	-	Scan mode select: LO – boundary scan, HI - debug (PD, 5V)
JTAG Interface					
TMS	80	73	I	-	JTAG - Test mode select (PU, 5V)
TCK	81	74	I	-	JTAG - Test clock (5V)

[5] PD indicates that internal Pull-Down resistor is present on pad.

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
TDI	82	75	I	-	JTAG - Test Data In (PU, 5V)
TDO	83	76	O	4	JTAG - Test Data Out (Z, 5V)
TRST*	84	77	I	-	JTAG – Test Reset (active low) (PD, 5V)
I2C Interface					
I2C_CLK	104 (shared)	97 (shared)	I/O (S)	6	I2C Clock (OD[6], 5V)
I2C_DATA	105 (shared)	98 (shared)	I/O (S)	6	I2C Data (OD, 5V)
SPI Interface					
SPIA_SS	138 (shared)	123 (shared)	I/O (S)	6	SPI Slave Select
SPIA_MISO	139 (shared)	124 (shared)	I/O (S)	6	SPI Master In, Slave Out
SPIA_MOSI	104 (shared)	97 (shared)	I/O (S)	6	SPI Master Out, Slave In
SPIA_CK	105 (shared)	98 (shared)	I/O (S)	6	SPI Clock
SPIB_SS	57 (shared)	N/A	I/O (S)	6	Alt. SPI Slave Select
SPIB_MISO	65 (shared)	N/A	I/O (S)	6	Alt. SPI Master. In, Slave Out
SPIB_MOSI	66 (shared)	N/A	I/O (S)	6	Alt. SPI Master. Out, Slave In
SPIB_CK	67 (shared)	N/A	I/O (S)	6	Alt. SPI Clock
UART Signals					
UART0_TX	130	115	O	4	Serial output; active-high
UART0_RX	131	116	I	-	Serial input; active-high (5V)
UART1_TX	134	119	O	4	Serial output; active-high
UART1_RX	135	120	I	-	Serial input; active-high (5V)
Filters					
FILTER_AES	90	83	A	-	AES Receiver filter component connection
FILTER_CLK_DBL	95	88	A	-	Clock Doubler VCO filter component connection
FILTER_HPLL	98	91	A	-	Jet^{TM} PLL filter component connection
PLL 1.8V					
PLL_1V8 (AES)	89	82	P	-	PLL 1.8 V
PLL_1V8 (CLK_DBL)	94	87	P	-	PLL 1.8 V
PLL_1V8 (HPLL)	99	92	P	-	PLL 1.8 V

[6] OD indicates Open Drain pad type. External Pull-Up resistor required.

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
PLL Bulk Bias					
PLL_BULK (CLK_DBL)	93	86	P	-	PLL Bulk Bias
PLL_BULK (HPLL)	97	90	P	-	PLL Bulk Bias
PLL Ground					
PLL_GND (AES)	91	84	P	-	PLL Ground
PLL_GND (CLK_DBL)	92	85	P	-	PLL Ground
PLL_GND (HPLL)	96	89	P	-	PLL Ground
Core 1.8V					
VDD1IH	24	25	P	-	Core 1.8 V
VDD1IH	61	57	P	-	Core 1.8 V
VDD1IH	100	93	P	-	Core 1.8 V
VDD1IH	132	117	P	-	Core 1.8 V
I/O 3.3V					
VDD3OP	11	12	P	-	I/O 3.3 V
VDD3OP	49	48	P	-	I/O 3.3 V
VDD3OP	87	80	P	-	I/O 3.3 V
VDD3OP	120	109	P	-	I/O 3.3 V
Core Ground					
VSS3I	25	26	P	-	Core Ground
VSS3I	62	58	P	-	Core Ground
VSS3I	101	94	P	-	Core Ground
VSS3I	133	118	P	-	Core Ground
I/O Ground					
VSS3OP	12	13	P	-	I/O Ground
VSS3OP	50	49	P	-	I/O Ground
VSS3OP	88	81	P	-	I/O Ground
VSS3OP	121	110	P	-	I/O Ground

Table 2: Signal Descriptions

1.5.2 Multi-function Pins

The following table lists all the multiple signal (shared) pins, along with the various signals assigned to each pin.

LQFP	QFP	Function 1		Function 2		Function 3		Function 4	
144	128								
85	78	EXT_FBR	(I/O)	GPIO4	(I/O)	WCLK_IN0	(I)		
86	79	EXT_512BR	(I/O)	GPIO5	(I/O)	WCLK_OUT0	(O)		
104	97	I2C_CLK	(I/O)	SPIA_MOSI	(I/O)				
105	98	I2C_DATA	(I/O)	SPIA_CK	(I/O)				
117	106	MCK1	(I/O)	GPIO6	(I/O)				
118	107	FCK1	(I/O)	GPIO7	(I/O)				
119	108	BCK1	(O)	GPIO8	(I/O)				
137	122	SRAM_READY	(I)	GPIO3	(I/O)				
138	123	CS2*	(O)	GPIO1	(I/O)	EN1_A	(I)	SPIA_SS	(I/O)
139	124	CS3*	(O)	GPIO2	(I/O)	EN1_B	(I)	SPIA_MISO	(I/O)
42	N/A	CLKE	(O)	GPIO0	(I/O)				
57	N/A	SPIB_SS	(I/O)	GPIO11	(I/O)				
65	N/A	SPIB_MISO	(I/O)	GPIO12	(I/O)	EN2_A	(I)	WCLK_IN1	(I)
66	N/A	SPIB_MOSI	(I/O)	GPIO13	(I/O)	EN2_B	(I)	WCLK_OUT1	(O)
67	N/A	SPIB_CK	(I/O)	GPIO14	(I/O)				

Table 3: Shared Pins

1.5.3 TCD2220 Pins (not available on TCD2210)

Signal	TCD2220	I/O	Drive (mA)	Description
Data Bus				
EN2_A	65 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_B	66 (shared)	I (S)	6	Rotary Encoder Input (5V)
General Purpose I/O				
GPIO0	42 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO9	55	I/O (S)	6	General Purpose I/O (5V)
GPIO10	56	I/O (S)	6	General Purpose I/O (5V)
GPIO11	57 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO12	65 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO13	66 (shared)	I/O (S)	6	General Purpose I/O (5V)

GPIO14	67 (shared)	I/O (S)	6	General Purpose I/O (5V)
SDRAM Dedicated Signals				
CLKE	42 (shared)	O	6	SDRAM Interface Clock Enable
SMPCHG	43	O	8	SDRAM Precharge A10
Word Clock				
WCLK_IN1	65 (shared)	I (S)	6	Word Clock In (5V), **Suggest using WCLK_IN0 as that is the default.**
WCLK_OUT1	66 (shared)	O	6	Word Clock Out, **Suggest using WCLK_OUT0 as that is the default.**
Audio Port 1				
DO1_0	126	O	2	Audio port 1 data out 0
DO1_1	127	O	2	Audio port 1 data out 1
DO1_2	128	O	2	Audio port 1 data out 2
DO1_3	129	O	2	Audio port 1 data out 3
DI1_0	113	I	-	Audio port 1 data in 0
DI1_1	114	I	-	Audio port 1 data in 1
DI1_2	115	I	-	Audio port 1 data in 2
DI1_3	116	I	-	Audio port 1 data in 3
SPI Interface				**TCD2210 has alternative SPI Pins.**
SPIB_SS	57 (shared)	I/O (S)	6	Alt. SPI Slave Select
SPIB_MISO	65 (shared)	I/O (S)	6	Alt. SPI Master. In, Slave Out
SPIB_MOSI	66 (shared)	I/O (S)	6	Alt. SPI Master. Out, Slave In
SPIB_CK	67 (shared)	I/O (S)	6	Alt. SPI Clock

Table 4: TCD2220 Only

Chapter 2 The ARM7TDMI

The ARM7TDMI is a member of the Advanced RISC Machines (ARM) family of general purpose 32-bit microprocessors, which offer high performance for very low power consumption and price.

The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers. This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective chip. Pipelining is employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The ARM memory interface has been designed to allow the performance potential to be realized without incurring high costs in the memory system. Speed-critical control signals are pipelined to allow system control functions to be implemented in standard low-power logic, and these control signals facilitate the exploitation of the fast local access modes offered by industry standard dynamic RAMs.

2.1 Architecture

The ARM7TDMI processor employs a unique architectural strategy known as THUMB, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

2.1.1 The THUMB Concept

The key idea behind THUMB is that of a super-reduced instruction set. Essentially, the ARM7TDMI processor has two instruction sets:

- the standard 32-bit ARM set
- a 16-bit THUMB set

The THUMB set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because THUMB code operates on the same 32-bit register set as ARM code.

THUMB code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

2.1.2 THUMB's Advantages

THUMB instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and THUMB states. Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model.

The major advantage of a 32-bit (ARM) architecture over a 16-bit architecture is its ability to manipulate 32-bit integers with single instructions, and to address a large address space efficiently. When processing 32-bit data, a 16-bit architecture will take at least two instructions to perform the same task as a single ARM instruction.

However, not all the code in a program will process 32-bit data (for example, code that performs character string handling), and some instructions, like Branches, do not process any data at all.

If a 16-bit architecture only has 16-bit instructions, and a 32-bit architecture only has 32-bit instructions, then overall the 16-bit architecture will have better code density, and better than one half the performance of the 32-bit architecture. Clearly 32-bit performance comes at the cost of code density.

THUMB breaks this constraint by implementing a 16-bit instruction length on a 32-bit architecture, making the processing of 32-bit data efficient with a compact instruction coding. This provides far better performance than a 16-bit architecture, with better code density than a 32-bit architecture.

THUMB also has a major advantage over other 32-bit architectures with 16-bit instructions. This is the ability to switch back to full ARM code and execute at full speed. Thus critical loops for applications such as

- fast interrupts
- DSP algorithms

can be coded using the full ARM instruction set, and linked with THUMB code. The overhead of switching from THUMB code to ARM code is folded into sub-routine entry time. Various portions of a system can be optimized for speed or for code density by switching between THUMB and ARM execution as appropriate.

Chapter 3 Memory Map and Interrupts

This section explains how the various modules in the system are addressed from the ARM. There are two memory maps, one in the case where Remap is active and one where it is inactive. The remap functionality is used during boot. The ARM processor assumes that the exception vectors are placed from address 0x0000 0000 after reset and therefore it is essential that the external program memory (typically a flash) is mapped to this address after reset. In most applications it is necessary to be able to change exception vectors at runtime, and for that purpose the internal RAM can be mapped into the low address space. A reset will always force CS0 to be mapped to address 0x0000 0000. By writing a "1" to register 0xc0000008 in the Remap module the low portion of the address space can be replaced with the internal RAM. A shadow of the internal RAM will always be present at address 0x8000 0000 enabling the application to write to it before the remapping is done. The size of internal SRAM is 16KB; however, 16MB of address space is allocated to it. The address bits 14-24 are ignored when internal SRAM is accessed.

Note, that in regular DICE JR/Mini applications only boot mode is used. All applications run from SDRAM. ARM core remap function is only used for testing purposes and not recommended for the users.

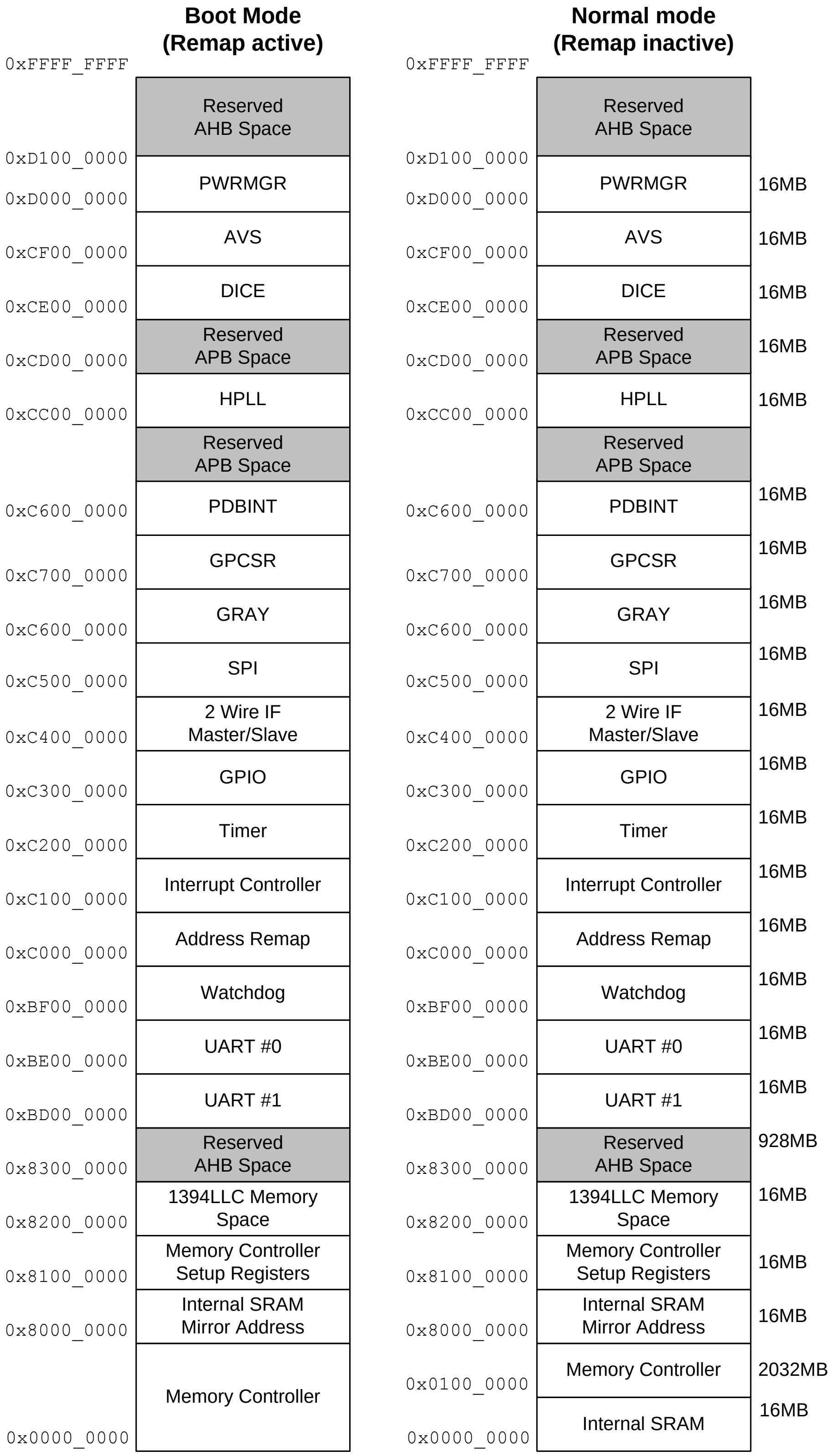


Figure 4: Global Memory Map (allocated address space)

Chapter 4 ARM Peripherals

Base Address	Functional Block
0x8100 0000	Memory Controller
0x8200 0000	1394LLC
0xBD00 0000	UART#1
0xBE00 0000	UART#0
0xBF00 0000	Watch Dog
0xC000 0000	Address Remap Register
0xC100 0000	Interrupt Controller
0xC200 0000	Dual Timer/Counter
0xC300 0000	GPIO
0xC400 0000	I2C
0xC500 0000	SPI
0xC600 0000	Gray
0xC700 0000	GPCSR
0xC800 0000	PDB Interface (AVS Global Enable)
0xCC00 0000	Jet^{TM} PLL
0xCE00 0000	DICE Sub System
0xCF00 0000	AVS Sub System (1394 Audio Interface)
0xD000 0000	Power Manager

Table 5: ARM Peripheral base addresses

4.1 General Purpose Control and Status Registers

The GPCSR controls various modes and pin mappings in the TCD2210/20. Due to the high integration some pins share several functions. The GPCSR selects the mapping of those pins. GPCSR Registers control the function of all multipurpose pins including GPIOs as well as the function of AUDIO PORTS

4.1.1 Audio Input ports

The DICE JR (2220) has 2 audio input ports and DICE Mini (2210) only has Port 0. Each port has 4 data lines. The 4 data lines in each port are routed to the corresponding InS interface. The lines can also be routed to the AES receivers and ADAT receivers depending on the configuration setting. See AUDIO PORT register description.

The following configuration signals exist for the configuration of the Audio Input Ports

Name	bits	Values
CFG_AES0_RX	2	00: AES Off, logic 0 routed to receiver 01: AES received from Audio port 0 signal 10:AES received from Audio port 1 signal (TCD2220 only) 11: Reserved
CFG_AES1_RX	2	
CFG_AES2_RX	2	
CFG_AES3_RX	2	

Table 6: Audio In-port Configuration Bits

4.1.2 Audio Output ports

The DICE JR (2220) has 2 audio input ports and DICE Mini (2210) only has Port 0. Each port has 4 data lines. The 4 data lines in each port can source from the corresponding InS, AES or ADAT interface. Only Port 0 can be configured for ADAT.

Name	bits	Values
CFG_A0_L0	1	0: InS0 Tx0/1 1: AES
CFG_A0_L1	1	
CFG_A0_L2	2	00: InS0 Tx2/3 01: AES 10: ADAT 11: Reserved
CFG_A0_L3	2	
CFG_A1_L0	1	0: InS1 Tx0/1/2/3 1: AES
CFG_A1_L1	1	
CFG_A1_L2	1	
CFG_A1_L3	1	

Table 7: Audio Out-port Configuration Bits

4.1.3 Module Configuration

Address	Register
0xc700 0000	GPCSR_SYSTEM
0xC700 0004	GPCSR_AUDIO_SELECT
0xC700 0008	GPCSR_GPIO_SELECT
0xC700 0014	GPCSR_CHIP_ID
0xC700 0024	GPCSR_IRQ_SEL0_5
0xC700 0028	GPCSR_IRQ_SEL6_11
0xC700 002c	GPCSR_IRQ_SEL12_17
0xC700 0030	GPCSR_IRQ_SEL18
0xC700 0034	GPCSR_FIQ_SEL0_5
0xC700 0038	GPCSR_FIQ_SEL6_7

Table 8: GPCSR Memory Map

4.1.4 GRCSR_SYSTEM

0xc700 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													RMAP	LPIE	LLCM
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Name	Bit	Reset	Dir	Description
RMAP	2	0	RW	Remap signal to Memory controller. This is not related to the boot time remap functionality.
LPIE	1	1	RW	Enable LPI during startup. 0: Ask PHY to remove SCLK. 1: Keep SCLK running.
LLCM	0	1	RW	Select 1394 LLC Mode. 0: 1394-1995 1: 1394-2000a

4.1.5 GPCSR_AUDIO_SELECT

The audio ports share the InS, AES and ADAT interfaces. The figures below show how the signals are multiplexed onto the audio port.

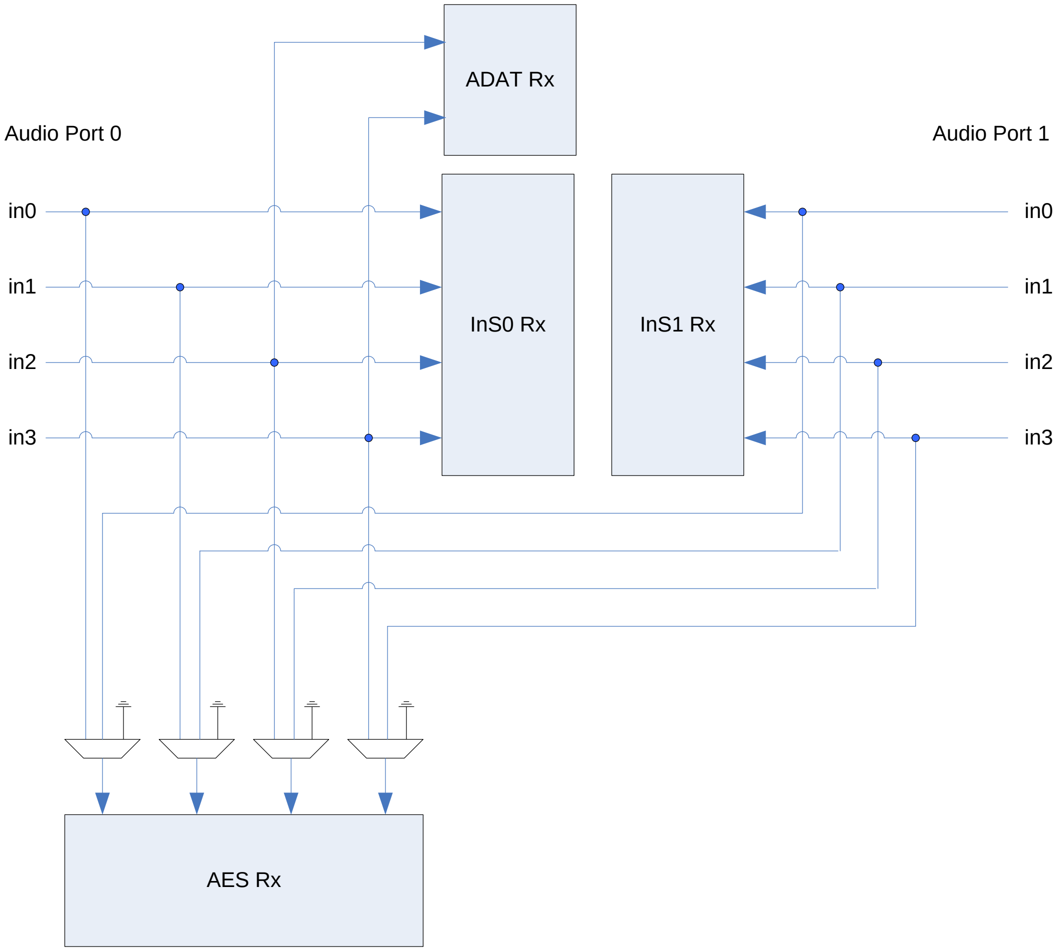


Figure 5: Audio In-port Routing

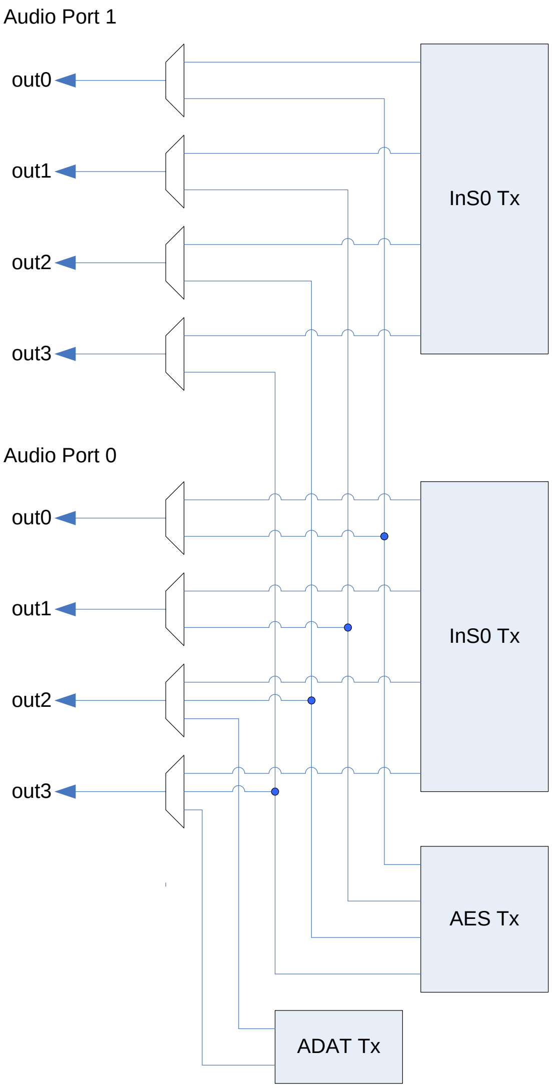


Figure 6: Audio Out-port Routing

Address - 0xc700 0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved														AO1_3	AO1_2
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AO1_1	AO1_0	AO0_3		AO0_2		AO0_1	AO0_0	AES3_RX		AES2_RX		AES1_RX		AES0_RX	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
AO1_3	17	0	RW	Audio Out Port 1, Line 3. Selects between I^{n}S or AES functionality. 0: I^{n}S1_3 1: AES3 Note: This only applies to TCD2220.
AO1_2	16	0	RW	Audio Out Port 1, Line 2. Selects between I^{n}S or AES functionality. 0: I^{n}S1_2 1: AES2 Note: This only applies to TCD2220.
AO1_1	15	0	RW	Audio Out Port 1, Line 1. Selects between I^{n}S or AES functionality. 0: I^{n}S1_1 1: AES1 Note: This only applies to TCD2220.
AO1_0	14	0	RW	Audio Out Port 1, Line 0. Selects between I^{n}S or AES functionality. 0: I^{n}S1_0 1: AES0 Note: This only applies to TCD2220.
AO0_3	13:12	00	RW	Audio Out Port 0, Line 3. Selects between I^{n}S, AES or ADAT functionality. 00: I^{n}S0_3 01: AES3 10: ADAT1 11: Reserved
AO0_2	11:10	00	RW	Audio Out Port 0, Line 2. Selects between I^{n}S, AES or ADAT functionality. 00: I^{n}S0_2 01: AES2 10: ADAT0 11: Reserved
AO0_1	9	0	RW	Audio Out Port 0, Line 1. Selects between I^{n}S or AES functionality. 0: I^{n}S0_1 1: AES1
AO0_0	8	0	RW	Audio Out Port 0, Line 0. Selects between I^{n}S or AES functionality. 0: I^{n}S0_0 1: AES0
AES3_RX	7:6	00	RW	Selects the source pin for AES3 Receiver 00: Off 01: Audio Port 0, line 3 10: Audio Port 1, line 3 (only valid for TCD2220) 11: Reserved
AES2_RX	5:4	00	RW	Selects the source pin for AES2 Receiver 00: Off 01: Audio Port 0, line 2 10: Audio Port 1, line 2 (only valid for TCD2220) 11: Reserved
AES1_RX	3:2	00	RW	Selects the source pin for AES1 Receiver 00: Off 01: Audio Port 0, line 1 10: Audio Port 1, line 1 (only valid for TCD2220) 11: Reserved
AES0_RX	1:0	00	RW	Selects the source pin for AES0 Receiver 00: Off 01: Audio Port 0, line 0 10: Audio Port 1, line 0 (only valid for TCD2220) 11: Reserved

4.1.6 GPCSR_GPIO_SELECT

Address - 0xc700 0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															ENC1
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO13	GPIO12	ENC2	GPIO8	GPIO7	GPIO6	GPIO5		GPIO4		GPIO3	GPIO2	GPIO1	SPI		GPIO0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
ENC1	16	0	RW	Selects between Encoder 1 and other functionality, this setting is ignored if pins configured for SPI. 0: Encoder 1 (Overrules setting for GPIO1/2) 1: Other function (See GPIO1/2)
GPIO13	15	00	RW	Selects between GPIO13 and WCLK_OUT1. 0: GPIO13 (I/O) 1: WCLK_OUT (O)
GPIO12	14	0	RW	Selects between GPIO12 and WCLK_IN1. 0: GPIO12 (I/O) 1: WCLK_IN1 (I)
ENC2	13	0	RW	Selects between Encoder 2 and other functionality, this setting is ignored if pins configured for SPI. 0: Encoder 2 (Overrules setting for GPIO12/13) 1: Other function (See GPIO12/13)
GPIO8	12	0	RW	Selects between GPIO8 and BCK1. 0: GPIO8 (I/O) 1: BCK1 (O)
GPIO7	11	0	RW	Selects between GPIO7 and FCK1. 0: GPIO7 (I/O) 1: FCK1 (O)
GPIO6	10	0	RW	Selects between GPIO6 and MCK1. 0: GPIO6 (I/O) 1: MCK1 (O)
GPIO5	9:8	0	RW	Selects between GPIO5, EXT_F512_in, EXT_F512_out and WCLK_OUT0. 00: GPIO5 (I/O) 01: EXT_F512 In (I) 10: EXT_F512 Out (O) 11: WCLK_OUT0
GPIO4	7:6	0	RW	Selects between GPIO4, EXT_FBR In or EXT_FBR out. 00: GPIO4 (I/O) 01: EXT_FBR In (I) (or WCLK_IN0) 10: EXT_FBR Out (O) 11: Reserved
GPIO3	5	0	RW	Selects between GPIO3 and SRAM_READY. 0: GPIO3 (I/O) 1: SRAM_READY (I)
GPIO2	4	0	RW	Selects between GPIO2 and CS3, this setting is ignored if pins configured for SPI (SPI=01) or (ENC1=0). 0: GPIO2 (I/O) 1: CS3* (O)

Name	Bit	Reset	Dir	Description
GPIO1	3	0	RW	Selects between GPIO1 and CS2, this setting is ignored if pins configured for SPI or ENC1. 0: GPIO1 (I/O) 1: CS2* (O)
SPI	2:1	00	RW	Selects the pins used for SPI. On TCD2210 there are two alternative sets of pins for SPI. 00: No SPI (default) 01: SPI_a (overrules setting for ENC1 and GPIO1/2) 10: SPI_b (overrules setting for ENC2 and GPIO12/13) 11: Reserved
GPIO0	0	0	RW	Selects between GPIO0 and CLKE functionality. 0: GPIO0 (I/O) 1: CLKE (O)

4.1.7 GPCSR_CHIP_ID

Address - 0xc700 0014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ID								Rev							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0												Type			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
CHIP_ID	31:24	01h	RO	Chip ID, always 01h for DICE_JR Family
CHIP_REV	23:16	00h	RO	Chip Revision, 00h
CHIP_TYPE	3:0	xx	RO	Chip Type 1111: TCD2220 1110: TCD2210 xxxx: All other types reserved for future use

4.1.8 Considerations for Chip Selects

Chip select 2 and 3 (CS2-CS3) are by default programmed to have an alternative function as an input. If external memories or peripherals are connected to those pins, a pull-up should be added in order not to select those devices during boot.

4.1.9 GPCSR_IRQ/FIQ_SEL – 0xc700 0024 – 0xc700 0038

The Interrupt controller defined in section 4.8 has 32 vectored, prioritized IRQ sources and 8 FIQ sources. Only IRQ sources 0 to 18 are used by the TCD22XX. Each of those

19 logical IRQ sources and 8 logical FIQ can be connected to any of the 19 physical interrupt sources. The IRQ_SEL registers controls this routing.

Each register is 5 bits wide and the assignment is as follows:

Value	Interrupt source
00000	SPI
00001	WatchDog
00010	1394Link_on
00011	1394Link
00100	Gray
00101	GPIO
00110	Timer
00111	UART0
01000	UART1
01001	I2C
01010	Reserved
01011	AVS_IRQ0
01100	AVS_IRQ1
01101	ARM_AUDIO_OVERFLOW
01110	ARM_AUDIO_IRQ
01111	Jet^{TM} PLL
10000	MIXER_OVL
10001	POWER_MGR
10010	MIDI
10011 – 11111	Reserved

Table 9: Physical Interrupt Sources

4.1.10 IRQ_SEL0_5

Address - 0xc700 0024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved		IRQ5					IRQ4					IRQ3			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ3	IRQ2					IRQ1					IRQ0				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.1.11 IRQ_SEL6_11

Address - 0xc700 0028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved		IRQ11					IRQ10					IRQ9			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ9	IRQ8					IRQ7					IRQ6				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.1.12 IRQ_SEL12_17

Address - 0xc700 002c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved		IRQ17					IRQ16					IRQ15			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ15	IRQ14					IRQ13					IRQ12				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.1.13 IRQ_SEL18

Address - 0xc700 0030

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											IRQ18				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.1.14 FIQ_SEL0_5

Address - 0xc700 0034

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved		FIQ5					FIQ4					FIQ3			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIQ3	FIQ2					FIQ1					FIQ0				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.1.15 FIQ_SEL6_7

Address - 0xc700 0038

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						FIQ7					FIQ6				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.2 External Bus Interface

The External Bus Interface (EBI) is a highly configurable generic memory interface supporting a wide variety of static and dynamic memories as well as memory mapped peripherals.

Below is a list of the main features:

- Memory interface Unit is clocked from the ARM system clock enabling 49.152MHz (typical) memory accesses.
- Support for 8 bit and 16 bit memories.
- Support for both SDRAM and static memory types
- 20 address bits (lsb addresses 16bit data word) and 16bit data on memory interface. Byte lane enables/masks are available for byte wide accesses into 16bit memory.
- Supports 4 chip selects. Each chip select is assigned a memory type: SDRAM, SRAM, FLASH or ROM. Both chip select assignment and memory type timing characteristics are runtime reconfigurable.
- Base address and block size for each chip select is configurable at runtime.
- Address aliasing will be available for both chip select 0 and 1. The feature allows two concurrent AHB bus address mappings to be available for each chip select. This feature can be enabled or disabled at runtime.
- Address remapping will be available for both chip select 0 and chip select 1. A control signal **remap** on dictates which of two AHB bus address mappings to apply for chip select 0 and chip select 1.
- Supports SDRAM precharge.

The memory map illustrated in Table 11 is the default after reset. The FLASH type access chosen as default for CS_0 is set-up using a very conservative timing. As CS_0 will have a default base address at 0x0000_0000 a 16bit FLASH/ROM or any similar for ARM SW booting should be mounted here.

The memory controller contains two main control modules; one for SDR SDRAM control and one for static memory including asynchronous SRAM, ROM and FLASH memory. The memory controller can be connected to 4 different memory devices at a time, with the choice of SDRAM, SRAM, FLASH or ROM for each. The memory type, size, addressing, and timing are all programmable. The data bus for both the SDRAM and static memory controllers is shared between the two controllers. The address bus for the two controllers is also shared.

4.2.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
Data Bus					
D0	144	1	I/O (S[7])	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU[8],5V[9])
D1	1	2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D2	2	3	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D3	3	4	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D4	4	5	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D5	5	6	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D6	6	7	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D7	7	8	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D8	8	9	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D9	9	10	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D10	10	11	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D11	13	14	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D12	14	15	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)

[7] S indicates Schmitt Trigger Input

[8] PU indicates that internal Pull-Up resistor is present on PAD

[9] 5V indicates that the input is 5V tolerant

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
D13	15	16	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D14	16	17	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D15	17	18	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
Address Bus					
A0	18	19	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A1	19	20	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A2	20	21	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A3	21	22	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A4	22	23	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A5	23	24	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A6	26	27	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A7	27	28	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A8	28	29	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A9	29	30	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A10	30	31	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A11	31	32	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A12	32	33	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A13	33	34	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A14	34	35	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A15	35	36	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A16	36	37	O	8	Address Bus. Shared address pins

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
					for SDRAM and Static memory.
A17	37	38	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A18	38	39	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A19	39	40	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
Chip Selects					
CS0*	136	121	O	4	Shared SDRAM and Static Memory Chip Selects
CS1*	46	45	O	4	Shared SDRAM and Static Memory Chip Selects
CS2*	138 (shared)	123 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
CS3*	139 (shared)	124 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
RAM Clock					
CLKO	40	41	O	8	SDRAM Interface AHB Bus Clock (Z[10])
SDRAM Dedicated Signals					
CLKE	42 (shared)	N/A	O	6	SDRAM Interface Clock Enable
RAS*	41	42	O	8	SDRAM Interface Row Address Strobe
CAS*	44	43	O	8	SDRAM Interface Column Address Strobe
SDRAM_WE	45	44	O	8	SDRAM Interface Write Enable
SDRAM_DQM0	47	46	O	8	SDRAM Interface Lower byte mask
SDRAM_DQM1	48	47	O	8	SDRAM Interface Upper byte mask
SDRAM_BNK0	52	51	O	8	SDRAM Interface Bank Address
SDRAM_BNK1	53	52	O	8	SDRAM Interface Bank Address
SDRAM_A10	43	N/A	O	8	SDRAM Precharge A10
SRAM Interface					
SRAM_READY	137(shared)	122 (shared)	I	6	SRAM ready
SRAM_BS[0]	140	125	O	4	SRAM lower byte select
SRAM_BS[1]	141	126	O	4	SRAM upper byte select

[10] Z indicates that the output is Z-stateable

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
SRAM_WE*	142	127	O	8	SRAM write enable
SRAM_OE*	143	128	O	8	SRAM output enable

Note that several pins used in the memory interface module are multi-purpose or shared. The function of these pins is software configurable via the GPCSR module, specifically GPCSR_GPIO_SELECT – 0xc700 0008. Refer to the GPCSR module documentation for more information.

Note that in general, to set the shared pins to function as memory pins, the register mentioned above should be set as shown in 4.1.6.

4.2.2 Functional Description

The memory controller consists of two main functional blocks, the Host Interface Unit (HIU) and the Memory Interface Unit (MIU). The HIU is the interface between the MIU and the AMBA Advanced High-performance Bus (AHB). The HIU generates memory read/write requests or control register read/write requests to the MIU block, which correspond to transfers on the AMBA bus. The MIU is the interface for both SDRAM and Static memories. It generates appropriate address, data, and control signals corresponding to memory read/write transfers.

4.2.3 Host Interface Unit

The HIU has the following functions:

- Buffers register/memory access requests and sends them to the memory controller MIU.
- Converts an AMBA burst size into a memory burst size.
- Supports AMBA early-burst termination.
- Breaks AMBA wrapping burst into two separate memory bursts.
- Supplies the wrapping address before the slave sees it on the AMBA in order to save overhead cycles between two bursts.
- Detects memory page boundaries; terminates the current burst and reissues a new burst.
- Masks invalid bytes for transfers that are narrower than the width of the AMBA bus.

The HIU consists of the following sub-blocks:

- Address FIFO – Buffers the request of the AMBA AHB and sends memory/register access requests to the MIU; also contains some control information for a read/write transfer.
- Write Data FIFO – Buffers write data to the memory and control registers.
- Read Data Buffer – Buffers the read data from the memory.
- Burst Control – Controls all the HIU sub-blocks by generating the control logic for read and write transfers.

4.2.4 Memory Interface Unit

The MIU includes the following modules:

- SDRAM control unit – Generates the SDRAM control signals

- Static control unit – Generates the SRAM/FLASH/ROM control signals
- Refresh unit – Generates the SDRAM refresh request at appropriate intervals
- Address decoder unit – For SDRAM generates the row, column, and bank addresses that correspond to the logical address provided by the AHB host interface. For SRAM/FLASH/ROM generates and decodes the memory address that corresponds to the logical address provided by the AHB host interface.
- Control register unit – Holds the memory controller SDRAM/SRAM/FLASH/ROM control and configuration registers, as well as address decoder registers, and three sets of static memory timing registers.

You can use three separate Static memories (SRAM/FLASH/ROM) or multiple memories of the same type, perhaps with different timings.

4.2.5 Internal Functional Diagram

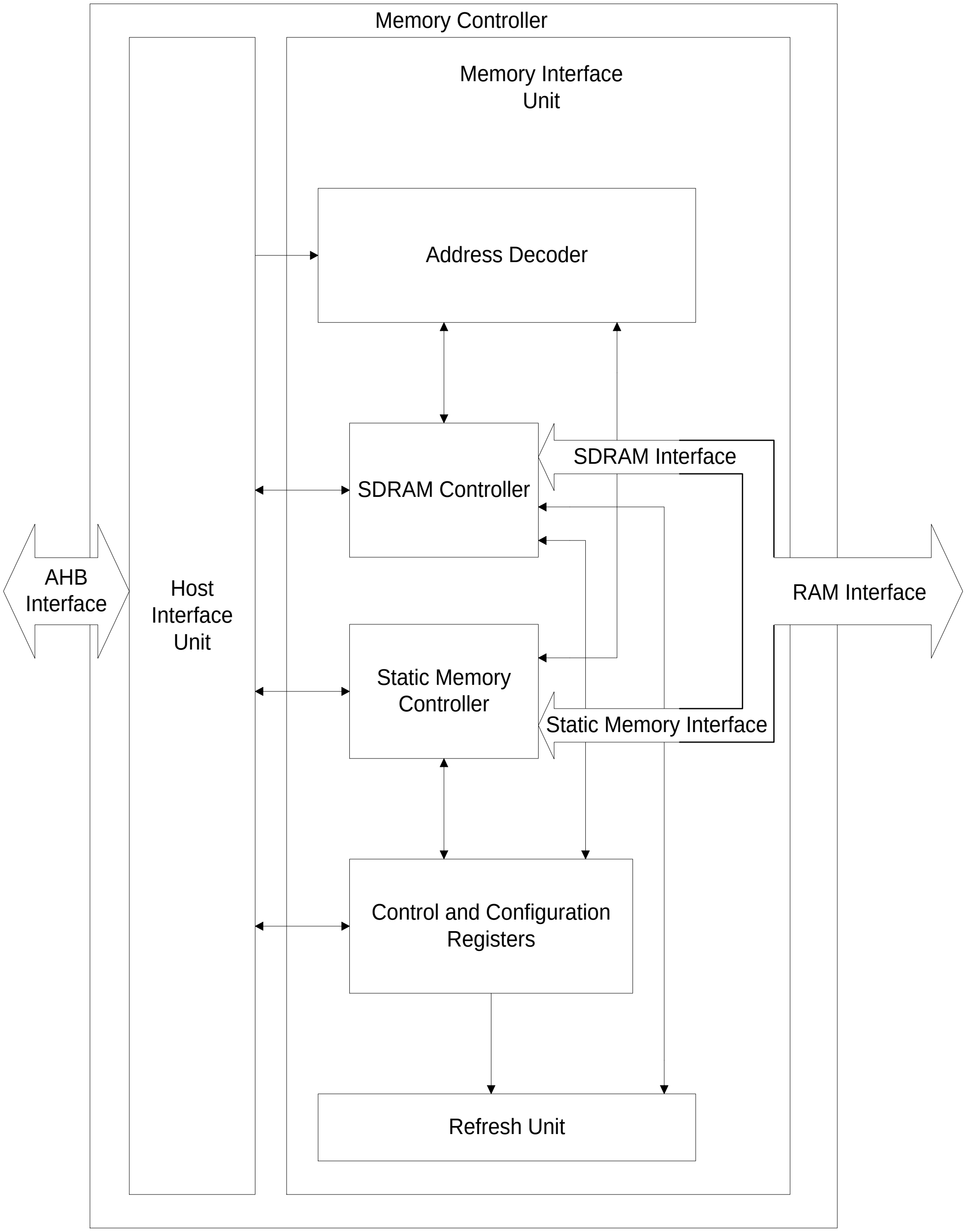


Figure 7: Internal Functional Diagram

4.2.6 Considerations for Chip Selects

Chip select 2 to 3 (CS2-CS3) are default programmed to have an alternative function as an input. If external memories or peripherals are connected to those pins a pull-up should be added in order not to select those devices during boot.

4.2.7 Module Configuration

Address	Register
0x8100 0000	EBI_SCONR
0x8100 0004	EBI_STMG0R
0x8100 0008	EBI_STMG1R
0x8100 000c	EBI_SCTLR
0x8100 0010	EBI_SREFR
0x8100 0014	EBI_SCSLR0
0x8100 0018	EBI_SCSLR1
0x8100 001c	EBI_SCSLR2
0x8100 0020	EBI_SCSLR3
0x8100 0024	EBI_SCSLR4
0x8100 0028	EBI_SCSLR5
0x8100 002c	EBI_SCSLR6
0x8100 0030	EBI_SCSLR7
0x8100 0054	EBI_SMSKR0
0x8100 0058	EBI_SMSKR1
0x8100 005c	EBI_SMSKR2
0x8100 0060	EBI_SMSKR3
0x8100 0064	EBI_SMSKR4
0x8100 0068	EBI_SMSKR5
0x8100 006c	EBI_SMSKR6
0x8100 0070	EBI_SMSKR7
0x8100 0074	EBI_CSALIAS0
0x8100 0078	EBI_CSALIAS1
0x8100 0084	EBI_CSREMAP0
0x8100 0088	EBI_CSREMAP1
0x8100 0094	EBI_SMTMGR_SET0
0x8100 0098	EBI_SMTMGR_SET1
0x8100 009c	EBI_SMTMGR_SET2
0x8100 00a0	EBI_FLASH_TRPDR
0x8100 00a4	EBI_SMCTLR

Table 10: EBI Module Description

4.2.8 SCONR – SDRAM Configuration Register

Address - 0x8100 0000

Reset values are considered to be the default and are the maximum configurable values. Programmed values should always be less than or equal to the reset values. This applies to all the programmable registers.

<table>
<tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td></td><td colspan="11">Reserved</td><td>s_sda_
oe_n</td><td>s_sd</td><td>s_scl</td><td colspan="2">s_sa</td></tr>
<tr><td>Reset:</td><td colspan="11">0</td><td>1</td><td>0</td><td>1</td><td colspan="2">0</td></tr>
<tr><td></td><td colspan="11">R</td><td>RW</td><td>RW</td><td>RW</td><td colspan="2">RW</td></tr>
</table>

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>s_sa</td><td colspan="2">s_data_width</td><td colspan="4">s_col_addr_width</td><td colspan="4">s_row_addr_width</td><td colspan="2">s_bank_addr_width</td><td colspan="3">Reserved</td></tr>
<tr><td>Reset:</td><td>0</td><td colspan="2">16</td><td colspan="4">15</td><td colspan="4">16</td><td colspan="2">4</td><td colspan="3">0</td></tr>
<tr><td></td><td>RW</td><td colspan="2">RW</td><td colspan="4">RW</td><td colspan="4">RW</td><td colspan="2">RW</td><td colspan="3">R</td></tr>
</table>

Name	Bit	Reset	Dir	Description
s_sda_oe_n	20	1	RW	Output enable for bi-directional data pin for I2C serial presence detect (SPD) logic 1 – corresponds to data written into bit 19 0 – programs for data reads
s_sd	19	0	RW	Bi-directional data for serial presence detect (SPD) logic; data written into bit goes in as data for SPD. During reads to this register, bit represents data read back from the SPD logic.
s_scl	18	1	RW	Clock for serial presence detect logic.
s_sa	17:15	0	RW	Serial presence detect address bits.
s_data_width	14:13	16	RW	Specifies SDRAM data width in bits 00 – 16 bits. 16bits is the only allowable value
s_col_addr_width	12:9	15	RW	Number of address bits for column address 15 – reserved, 7-14 – correspond to 8-15 bits, 0-6 – reserved
s_row_addr_width	8:5	16	RW	Number of address bits for row address 10-15 – correspond to 11-16 bits, 0-10 – reserved
s_bank_addr_width	4:3	4	RW	Number of bank address bits; values of 0-3 correspond to 1-4 bits, and therefore select 2-16 banks.

4.2.9 STMG0R – SDRAM Timing Register 0

Address - 0x8100 0004

Reset values are considered to be the default values. Programmed values should always be less than or equal to the reset values. The STMG0R and STMG1R registers hold the SDRAM timing parameters.

The memory controller uses the CAS latency value during the initialization sequence in order to program the mode register of the SDRAM. The user can also specifically force the memory controller to do a mode register update by programming the set_mode_reg bit (bit 9 of SCTLR). If you want to change the value of CAS latency during normal operation, you should first program the STMG0R timing register, and then program bit 9 of SCTLR to 1. The memory controller will reset this bit once it has updated the mode register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	unused						t_rc				t_xsr				t_rcar	
Reset:	0						10				11				10	
	R						RW				RW				RW	

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	t_rcar		t_wr		t_rp			t_rcd			t_ras_min				cas_latency	
Reset:	10		2		3			3			6				2	
	RW		RW		RW			RW			RW				RW	

Name	Bit	Reset	Dir	Description
t_rc	25:22	10	RW	Active-to-active command period; values of 0-15 correspond to t_rc of 1-16 clocks.
t_xsr	21:18	11	RW	Exit self-refresh to active or auto-refresh command time; minimum time controller should wait after taking SDRAM out of self-refresh mode before issuing any active or auto-refresh commands; values 1-512 correspond to t_xsr of 1-512 clocks.
t_rcar	17:14	10	RW	Auto-refresh period; minimum time between two auto-refresh commands; values 0-15 correspond to t_rcar of 1-16 clocks.
t_wr	13:12	2	RW	For writes, delay from last data in to next precharge command; values 0-3 correspond to t_wr of 1-4 clocks.
t_rp	11:9	3	RW	Precharge period; values of 0-7 correspond to t_rp of 1-8 clocks
t_rcd	8:7	3	RW	Minimum delay between active and read/write commands; values 0-7 correspond to t_rcd values of 1-8 clocks.
t_ras_min	5:2	6	RW	Minimum delay between active and precharge commands; values of 0-15 correspond to t_ras_min of 1-16 clocks.

Name	Bit	Reset	Dir	Description
cas_latency	1:0	3	RW	Delay in clock cycles between read command and availability of first data. 0 – 1 clock, 1 – 2 clocks, 2 – 3 clocks, 3 – 4 clocks

4.2.10 STMG1R – SDRAM Timing Register 1

Address - 0x8100 0008

Reset values are considered to be the default values. Programmed values should always be less than or equal to the reset values. The STMG0R and STMG1R registers hold the SDRAM timing parameters. See section 2.1.3 for more details.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved										unused		num_init_ref			
Reset:	0										1		8			
	R										R		RW			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	t_init															
Reset:	8															
	RW															

Name	Bit	Reset	Dir	Description
num_init_ref	19:16	8	RW	Number of auto-refreshes during initialization; values 0-15 correspond to 1-16 auto-refreshes
t_init	15:0	8	RW	Number of clock cycles to hold SDRAM inputs stable after power up, before issuing any commands

4.2.11 SCTLR – SDRAM Control Register

Address - 0x8100 000C

You can program SDRAM control registers at any time after power-up. However, the SDRAM controller does not poll the registers until the SDRAM controller finishes current and pending SDRAM accesses in the write FIFO.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved													exn_mode_reg_update	s_rd_ready_mode	num_open_banks
Reset:	0													0	0	2
	RW													RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	num_open_banks				self_refresh_status	sync_flash_soft_seq	set_mode_reg	read_pipe			full_refresh_after_sr	full_refresh_before_sr	precharge_algorithm	power_down_mode	self_refresh_mode	initialize
Reset:	2				0	0	0	2			0	0	1	0	0	0
	RW				R	RW	RW	RW			RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	31:19	0	RW	
exn_mode_reg_update	18	0	RW	Commands controller to update Mobile-SDRAM extended-mode register; once mode register update is done, controller automatically clears bit
s_rd_ready_mode	17	0	RW	SDRAM read-data-ready mode; set to 1, indicates SDRAM read data is sampled after s_rd_ready goes active
num_open_banks	16:12	2	RW	Number of SDRAM internal banks to be open at any time; values of 0-15 correspond to 0-15 banks open
self_refresh_status	11	0	RW	Read only. When “1,” indicates SDRAM is in self refresh mode. When “self_refresh/deep_power_mode” bit (bit 1 of SCTLR) is set, it may take some time before SDRAM is put into self-refresh mode, depending on whether all rows or one row are refreshed before entering self refresh mode defined by full_refresh_before_sr bit. Before gating clock in self-refresh mode, ensure this bit is set
sync_flash_soft_seq	10	0	RW	Specify type of command sequences used forSyncFlash operations: 1 – Software Command Sequence (SCS) 0 – Hardware Command Sequence (HCS) SyncFlash operation is not supported
set_mode_reg	9	0	RW	Set to 1, forces controller to do update of SDRAM mode register; bit is cleared by controller once it has finished mode register update

Name	Bit	Reset	Dir	Description
read_pipe	8:6	2	RW	Indicates number of registers inserted in read data path for SDRAM in order to correctly latch data; values 0-7 indicate 0-7 registers
full_refresh_after_sr	5	0	RW	Controls number of refreshes done by the memory controller after SDRAM is taken out of self-refresh mode. 1 – Refresh all rows before entering self-refresh mode 0 – Refresh just 1 row before entering self-refresh mode
full_refresh_before_sr	4	0	RW	Controls number of refreshes done by memory controller before putting SDRAM into self-refresh mode. 1 – Refresh all rows before entering self-refresh mode 0 – Refresh just one row before entering self-refresh mode
precharge_algorithm	3	1	RW	Determines when row is precharged. 0 – Immediate precharge; row precharged at end of read/write operation 1 – Delayed precharge; row kept open after read/write operations
power_down_mode	2	0	RW	Forces memory controller to put SDRAM in power-down mode
self_refresh_mode	1	0	RW	Forces memory controller to put SDRAM in self-refresh mode. Bit can be cleared by writing to this bit or by clear_sr_dp pin, generated by external power management unit.
initialize	0	0	RW	Forces memory controller to initialize SDRAM; bit reset to 0 by memory controller once initialization sequence is complete.

4.2.12 SREFR – SDRAM Refresh Interval Register

Address - 0x8100 0010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	gpi								gpo							
Reset:	-								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	t_ref															
Reset:	see Section Auto-Refresh															
	RW															

Name	Bit	Reset	Dir	Description
gpi	31:24	-	RW	General purpose inputs; directly connected to gpi. Connects status bits from FLASH memory to bits 2:0 of gpi; three bits of gpi used for FLASH status because three separate FLASH memories can be connected to memory controller.

gpo	23:16	0	RW	General purpose output signals; directly connected to gpo
t_ref	15:0	Sec.	RW	Number of clock cycles between consecutive refresh cycles. For details on programming this register refer to Section Auto-Refresh.

4.2.13 SCSLR (0-7) – Chip Select Registers

Address - 0x8100 0014 - 0x8100 0030

The memory controller has eight chip selects and a 32-bit AHB address width. There are four external chip select registers, each one holding the base address value that corresponds to its chip select.

The memory controller uses mask registers that specify the size of the memory connected to each chip select. Starting address of each memory segment (connected to a chip select) should be full multiple of the segment size. For example, a segment of 0x20000000 size (32mB) can only start at the addresses 0x00000000 or 0x20000000 or 0x40000000 etc. So, the starting address defined here should be in accordance with memory segment size defined in SMSKRn register.

The memory controller also supports aliasing and remapping, but these features are available only for chip select0 and chip select1.

The memory map illustrated in Table 11 below is the default after reset. You can later change a default chip select addresses by programming it. The FLASH type access chosen as default for chip select 0 (CS_0) is set-up using a very conservative timing. As CS_0 will have a default base address at 0x0000_0000, a 16bit FLASH/ROM or similar, that requires ARM, SW booting should be mounted here.

0x7FD0_0000	ROM (CS_3)
0x7FA0_0000	ROM (CS_2)
0x7F90_0000	ROM (CS_1)
0x4000_0000	Unassigned Space
0x0000_0000	FLASH (CS_0)

Table 11: Default Chip Select Memory Map

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Upper bits of chip select base address															
Reset:	see above															
	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	Unused
Reset:	0
	RW

Name	Bit	Reset	Dir	Description
Chip Select Address	31:16	see above	RW	The address of the selected one of eight memory chips

4.2.14 SMSKR (0 – 7) – Address Mask Registers

Address - 0x8100 0054 - 0x8100 0070

There are eight address mask registers, one for each chip select. They specify the size, type and timing mode of their corresponding memory chip.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved					reg_select			mem_type			mem_size				
Reset:	0					see table			see table			see table				
	RW					RW			RW			RW				

chip_select	reg_select	mem_type	mem_size
0	set 2	FLASH	1 GB
1	set 0	ROM	16 MB
2	set 0	ROM	16 MB
3	set 0	ROM	16 MB
4	set 0	ROM	16 MB
5	set 0	ROM	16 MB
6	set 0	ROM	16 MB
7	set 0	ROM	16 MB

Table 12: Default Chip Select Memory Attributes

Name	Bit	Reset	Dir	Description

Name	Bit	Reset	Dir	Description
reg_select	10:8	see table	RW	Register determines which timing parameters of memory connect to associated chip select; primarily used for specifying static memories. 0 – register set 0, 1 – register set 1, 2 – register set 2 This is “don't care” if mem_type is SDRAM
mem_type	7:5	see table	RW	Type of memory connected to corresponding chip select. 0 – SDRAM, 1 – SRAM, 2 – FLASH, 3 – ROM Others – Reserved
mem_size	4:0	see table	RW	Size of memory connected to corresponding chip select. Value of 0 specifies that no memory is connected to chip select. 0 – No memory is connected to the chip select 1 – 64KB, 2 – 128KB, 3 – 256KB, 4 – 512KB, 5 – 1MB, 6 – 2MB, 7 – 4MB, 8 – 8MB, 9 – 16MB, 10 – 32MB, 11 – 64MB, 12 – 128MB, 13 – 256MB, 14 – 512MB, 15 – 1GB, 16 – 2GB, 17 – 4GB

4.2.15 CSALIAS (0 – 1) – Alias Register

Address - 0x8100 0074 - 0x8100 0078

This register holds the aliasing address value for the given chip select. Note that only chip selects 0 and 1 support aliasing. When aliasing is enabled, the chip select becomes active when the AHB address matches either the base address for that chip select or the alias address for the chip select.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	csalia (0 – 1)															
Reset:	see table															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	csalia (0 – 1)	Unused														
Reset:	see table	0														
	RW	RW														

Chip Select	Alias Address
0	0x0
1	0x7f900000

Table 13: Alias Addresses

Name	Bit	Reset	Dir	Description
csalias (0 – 1)	31:15	see table	RW	Aliasing register bits for the chip select. Compared with corresponding AHB address to generate the chip select. The number of bits compared depends on size of memory selected by chip select (specified in mask register). 64 KB – bits 31:15 compared, 128 KB – bits 31:16 compared
Unused	14:0	0	RW	Unused since memory smaller than 64KB is not supported.

4.2.16 CSREMAP (0 – 1) – Remap Register

Address - 0x8100 0084 - 0x8100 0088

This register holds the remapping address value for the given chip select. Note that only chip select 0 and 1 support remapping. The chip select will be generated under the following conditions:

- When the remap input is 1 and the AHB address matches the remap address for the chip select.
- When the remap input is 0 and the AHB address matches the base address for chip select.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	csremap (0 – 1)															
Reset:	see table															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	csremap (0 – 1)	Unused														
Reset:	see table	0														
	RW	RW														

Chip Select	Remap Address
0	0x0
1	0x7f900000

Table 14: Remap Addresses

Name	Bit	Reset	Dir	Description
csremap (0 – 1)	31:15	see table	RW	Remap register bits for chip select. Compared with corresponding AHB address to generate chip select. The number of compared bits depends on size of memory selected by chip select (specified in mask register). 64KB – bits 31:15 compared, 128 KB – bits 31:16 compared

4.2.17 SMTMGR (0 – 2) – Static Memory Timing Register

Address - 0x8100 0094 - 0x8100 009C

There are 3 timing registers available to hold 3 sets of the various parameters for static memory. This allows the programming of 3 different timing modes, 1 in each register.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved		sm_read_pipe		low_freq_sync_device	ready_mode	page_size		page_mode	t_prc				t_bta		
Reset:	0		see table		see table	see table	see table		see table	see table				see table		
	RW		RW		RW	RW	RW		RW	RW				RW		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	t_wp						t_wr		t_as		t_rc					
Reset:	see table						see table		see table		see table					
	RW						RW		RW		RW					

Set	sm_read_pipe	low_freq_sync_device	ready_mode	page_size	page_mode	t_prc	t_bta	t_wp	t_wr	t_as	t_rc
0	1	0	0	4	0	1	1	2	0	1	2
1	1	0	0	4	0	16	4	20	3	1	28
2	1	0	0	4	0	15	4	20	3	3	63

Table 15: Default Static Memory Timing Parameters

Name	Bit	Reset	Dir	Description
sm_read_pipe	29:28	see table	RW	Number of registers inserted in the read data path for latching the data correctly, in the case of Static memory associated with register set0
low_freq_sync_device	27	see table	RW	Valid if register set0 is used to control low-frequency synchronous device; instructs the memory controller to sample sm_clken before starting any Static memory operation. Synchronous memory device could be same or sub-multiple of AMBA clock
ready_mode	26	see table	RW	Indicates if the static memory associated with register set 0 is a data-ready device (valid data indicated by a ready signal)
page_size	25:24	see table	RW	Page size. 0 – 4-word page, 1 – 8-word page, 2 – 16-word page, 3 – 32-word page
page_mode	23	see table	RW	Page-mode device. 0 – device does not support page mode 1 – device supports page mode

Name	Bit	Reset	Dir	Description
t_prc	22:19	see table	RW	Page mode read cycle time. Values of 0 – 15 correspond to read cycle time of 1 - 16 clock cycles.
t_bta	18:16	see table	RW	Idle cycles between read to write, or write to read, for memory data bus turn around time. Values of 0 - 7 correspond to 0 - 7 idle clock cycles.
t_wp	15:10	see table	RW	Write pulse width. Values of 0 - 63 correspond to write pulse width of 1 - 64 clock cycles.
t_wr	9:8	see table	RW	Write address/data hold time. Values of 0 – 3 correspond to write address/data hold time of 0 – 3 clock cycles.
t_as	7:6	see table	RW	Write address setup time. Values of 0 – 3 correspond to address setup time of 0 – 3 clock cycles. Value of 0 is only valid in case of SSRAM.
t_rc	5:0	see table	RW	Read cycle time. Values of 0 – 63 correspond to read cycle time of 1 – 64 clock cycles.

4.2.18 FLASH_TRPDR – Flash Timing Register

Address - 0x8100 00A0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0															
	R															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				t_rpd											
Reset:	0				200 cycles											
	R				RW											

Name	Bit	Reset	Dir	Description
t_rpd	11:0	200 cycles	RW	FLASH reset/power-down high to read/write delay. Values correspond to sm_rp_n high to read/write delay minus one.

4.2.19 SMCTLR – Static Memory Control Register

Address - 0x8100 00A4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	sm_data_width_set2			sm_data_width_set1			sm_data_width_set0			Unused			wp_n			sm_rp_n
Reset:	16			16			16			0			0			0
	RW			RW			RW			RW			RW			RW

Name	Bit	Reset	Dir	Description
sm_data_width_set2	15:13	16	RW	Width of Static memory data bus for Set2. Maximum of 16 bits. 000 – 16 bits, 100 – 8 bits
sm_data_width_set1	12:10	16	RW	Width of Static memory data bus for Set1. Maximum of 16 bits. 000 – 16 bits, 100 – 8 bits
sm_data_width_set0	9:7	16	RW	Width of Static memory data bus for Set0. Maximum of 16 bits. 000 – 16 bits, 100 – 8 bits
wp_n	3:1	0	RW	FLASH write-protection mode. Writing 0 forces FLASH memory boot block to write protect. The three bits correspond to three register sets.
sm_rp_n	0	0	RW	FLASH reset/power-down mode. After reset, controller internally performs a power-down for FLASH and then sets this bit to 1. To force FLASH to power-down mode during normal operation the following applies. 0 – Forces FLASH to power-down mode 1 – Takes FLASH out of power-down mode

4.2.20 SDRAM Power ON Initialization

The SDR-SDRAM controller follows the JEDEC-recommended SDR-SDRAM power-on initialization sequence as follows:

- Apply power and start clock; maintain a NOP condition at the inputs.
- Maintain stable power, stable clock, and NOP input conditions for a minimum of t_init clock cycles.
- Issue precharge commands for all banks of the device.
- Issue auto-refresh commands, depending on the value num_init_ref in the programmable register.
- Issue a set-mode register command to initialize the mode register.

The memory controller performs a power-on sequence of the SDRAM under these circumstances:

- Immediately after reset.
- When the programmable initialize bit (bit 0 of SCTLR) is set, the memory controller resets the bit when it comes out of initialization.

All SDRAM read/write requests that occur during initialization are queued in the memory controller.

The memory controller initializes the SDRAM after reset using the default timing parameters, shown in the figure below. After reset, if you feel that these timing parameters are not adequate, then you can program them accordingly using the SDRAM timing registers and then program the initialize bit (set bit 0 of SCTLR to 1), which forces the memory controller to re-initialize the SDRAM. If you feel that the reset time of the system is long enough to take care of the t_init time, then you can assign a value of zero to this parameter. The t_mrd is fixed at a value of 3 clock cycles, according to the JEDEC standard.

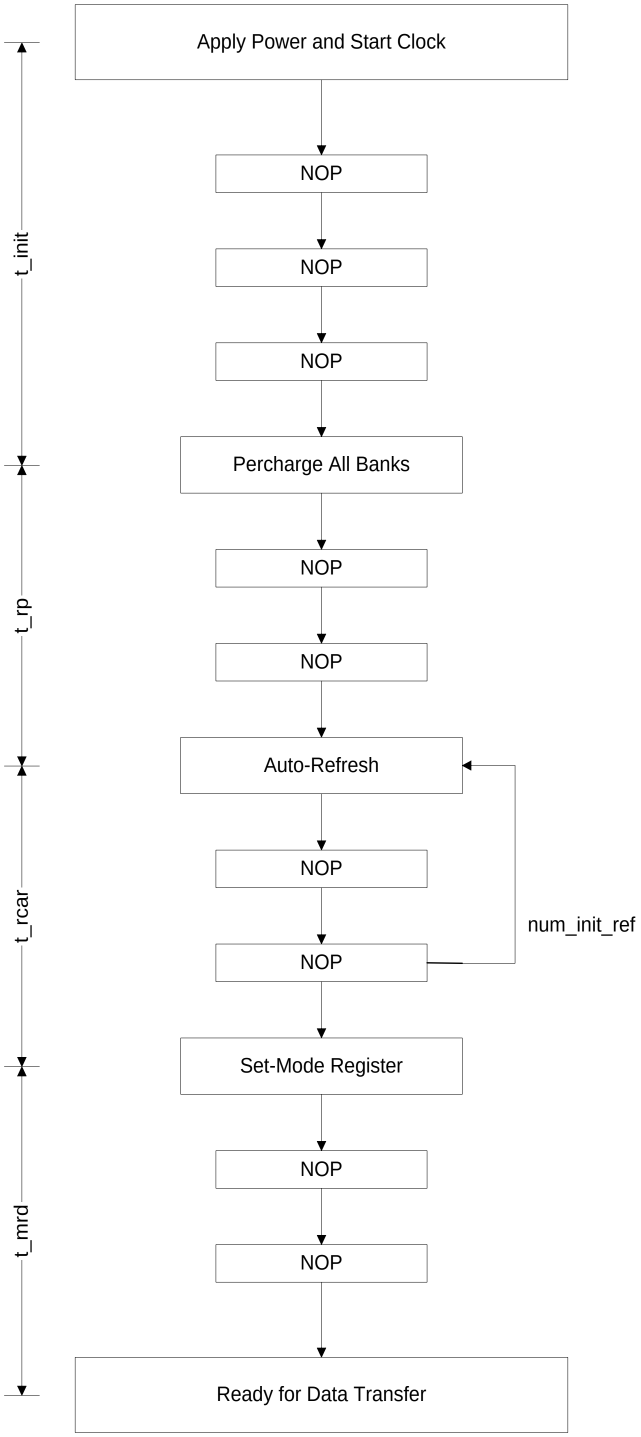


Figure 8: SDRAM Power ON Initialization Diagram

4.2.21 SDRAM Read and Write

The memory controller converts all AHB bursts to 4-word bursts on the SDRAM side. The memory bursts are concatenated to achieve continuous data flow for long AHB bursts. You can terminate the memory read/write burst with either a precharge command or terminate command, depending on which precharge mode (immediate precharge or delayed precharge) that you program. You can also terminate the write burst with a subsequent write burst.

The memory controller does not use auto-precharge mode. If you program for an immediate precharge mode, then the memory controller closes the open row after a read or write access. If you program for a delayed precharge mode, then the memory controller keeps the row open after an access. The memory controller can keep multiple banks open at the same time, depending on the value of num_open_bank in the programmable register. When the number of open banks reaches the num_open_bank and an access to a new bank comes, the memory controller will close the oldest bank (the bank opened first) before opening the new bank. The table below lists the memory controller performance during read/writes to the SDRAM in various circumstances, with the following assumptions:

- Memory controller is idle; that is, no pending read/write request is in the address FIFO.
- In the case of a read, timing information is relative to the latency from the time the select signal is asserted on the AMBA bus to when the first data is available on the AMBA bus.
- In the case of a write, timing information is relative to the latency from the time the select signal is asserted on the AMBA bus to when the first data is written to the SDRAM.
- Timing parameters in used in the table are:
    1. t_init – internal delay before a command is sent to the memory device
    2. t_rcd – active-to-read/write command time; assumed to be 3
    3. t_cas – CAS latency; assumed to be 3

Transfer Type	Condition	Timing	Latency
Writes	Page hit	t_init = 2	2
	Page miss	(t_init = 2) + (t_rcd = 3)	5
Reads	Page hit	(t_init = 2) + (t_cas = 3)	5
	Page miss	(t_init = 2) + (t_rcd = 3) + (t_cas = 3)	8

Table 16: Read/Write Timing Delays

4.2.22 SDRAM Set Mode Register

The memory controller automatically sets the SDR-SDRAM mode register during the power-up initialization. During normal operation, if you want to set the mode register you need to set set_mode_reg (bit 9) in the control register (SCTLR).

After the memory controller finishes the mode register setting, it clears the set_mode_reg to 0. The "burst length" field and the "burst type" field of the SDR-SDRAM-mode register are fixed by the memory controller to "010" (burst length 4) and "0" (sequential burst), respectively. The memory controller programs the "CAS latency" field and the "operating mode" field of the mode register according to the values provided by the user in the control and timing registers.

4.2.23 SDRAM Refresh

4.2.23.1 Auto-Refresh Mode

During normal refresh operations, the memory controller always refreshes one row at a time. It is important for the user to program the t_ref refresh interval register after a reset.

If you need to refresh the SDRAM while a burst is active, normally the memory controller will issue the refresh command after the ongoing burst completes. However, if the ongoing burst is an AHB INCR burst, the memory controller will stop the burst, issue the refresh command, and then resume the burst.

The memory controller takes into account the maximum time it takes to complete a worst-case burst. This is the time to complete a read burst corresponding to an INCR16 burst on the AMBA bus, and with an AMBA-to-SDRAM data width ratio of 2:1. It is reasonable to assume 50 cycles for this worst-case burst, with 32 cycles for the data and the remaining 17 cycles for various latencies for the worst case.

The t_ref value can be calculated using the following equation:

t_ref = refresh_period / clock_period

where refresh_period = typically 7.8 or 15.6 µs (see table)

Number of Rows	t_ref	Minimum Frequency
64K	(64ms - (50 / f)) / 65536	51 MHz
32K	(64ms - (50 / f)) / 32768	26 MHz
16K	(64ms - (50 / f)) / 163904	13 MHz
8K	(64ms - (50 / f)) / 8192	6 MHz
4K	(64ms - (50 / f)) / 4096	3 MHz
2K	(64ms - (50 / f)) / 2048	1.5 MHz

Table 17: Calculating t_ref

The t_ref is the value of a free-running counter that the refresh logic in the memory controller operates on. When the count expires, the refresh logic gives a refresh request to the SDRAM control.

Since the 64 ms refresh period is the same for most SDRAMs, the total number of rows in the SDRAM limits the minimum operating frequency for the memory controller. While calculating the minimum frequency, use the following equation:
t_ref > 50*(1/f)

The refresh logic in the memory controller is inactive when the memory controller forces the SDRAM into self-refresh or power-down mode.

4.2.23.2 Self-Refresh Mode

You can put the SDRAM into self-refresh mode, at which point the SDRAM retains data without external clocking and auto-refresh.

You can force the memory controller to enter self-refresh mode by programming bit 1 of the SDRAM control register (SCTLR). The memory controller forces the SDRAM to come out of self-refresh mode when bit 2 of the SCTLR is set to 0. You can set this bit to 0 by either programming the SDRAM control register or driving the clear_sr_dp pin high. You can use the clear_sr_dp pin when the code resides in the SDRAM, and the SDRAM itself is in self-refresh mode.

Bits 4 and 5 of the SCTLR specify the type of refresh done by the memory controller just prior to entering self-refresh mode and just after entering self-refresh mode. Programming bit 4 of the SCTLR to 0 forces the memory controller to refresh only one row before putting the SDRAM into self-refresh mode. The default value of 1 forces the memory controller to perform auto-refreshes for all rows. Bit 5 does the same, except that it controls the refresh pattern just after coming out of self-refresh mode.

Since it takes time between programming the control register bit, to the SDRAM entering self-refresh mode, the memory controller provides a read-only register bit (bit 11 of the SDRAM control register) to indicate that the SDRAM is already in self-refresh mode. If you want to gate off the clock to the memory controller when the SDRAM is in self-refresh mode, you should ensure this bit is set to 1 before you stop the clock.

The SDRAM must remain in self-refresh mode for a minimum period of t_ras and can remain in self-refresh mode for an indefinite period of time. After the SDRAM exits self-refresh mode, the memory controller issues NOP commands for t_xsr before it issues any other command. The t_ras and t_xsr are programmable register values and have default values. These registers can be programmed only once after reset.

When an AHB read/write request to the SDRAM occurs while the SDRAM is in self-refresh mode, the memory controller generates dummy ready signals to the AHB without accessing external memory; no error response is generated on the AHB bus.

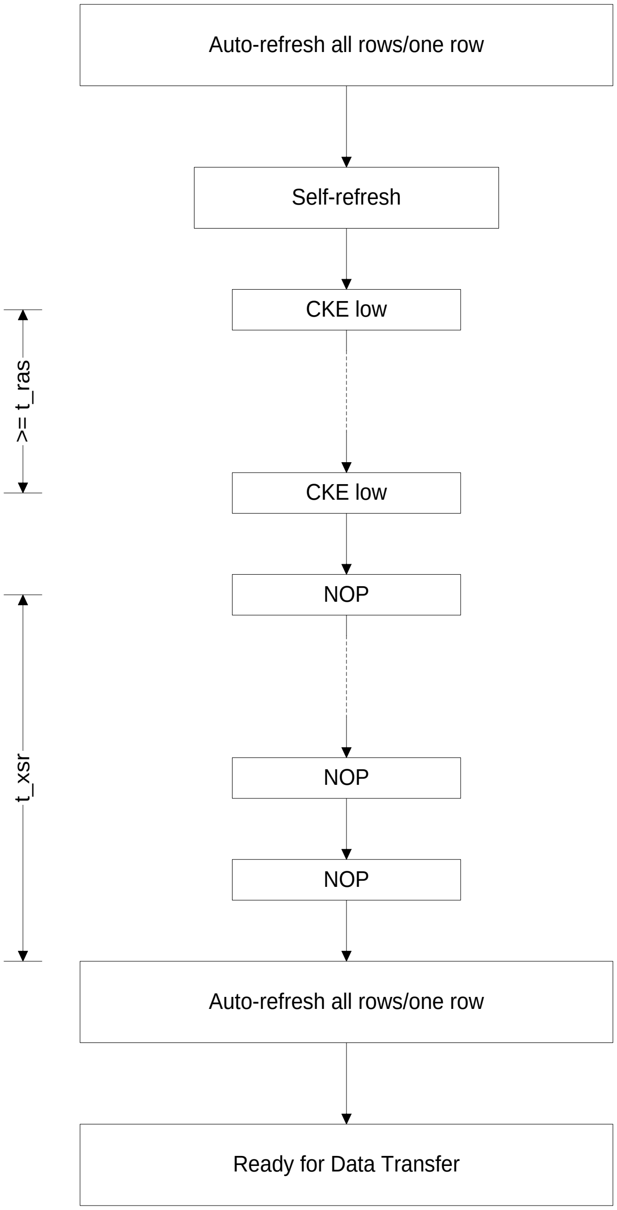


Figure 9: Auto-Refresh Diagram

4.2.24 SDRAM Power DOWN

The SDRAM can be put into power-down mode to save power. There are two ways to force the memory controller to put the SDRAM in power-down mode:

- Program bit 2 of SCTLR to 1; should be 0 to bring the SDRAM out of power-down mode.

- Use the power-down input pin; can be driven by an external power management unit; the SDRAM will be in power-down mode as long as this signal stays high.

When in SDRAM power-down mode, the memory controller keeps switching the device back and forth between power-down and refresh mode. It remains in power-down for a t_ref period of time, then comes out of power-down and does a single-row refresh, then it again goes into power-down mode. The memory controller keeps the SDRAM in this periodical power-down/refresh/power-down sequence until it is commanded to exit power-down mode (set bit 2 of SCTLR to 0).

When an AHB read/write request to the SDRAM occurs while the SDRAM is in power-down mode, the memory controller brings the SDRAM out of power-down mode and issues the read/write access to the SDRAM. The memory controller then puts the SDRAM back to power-down mode after the read/write access.

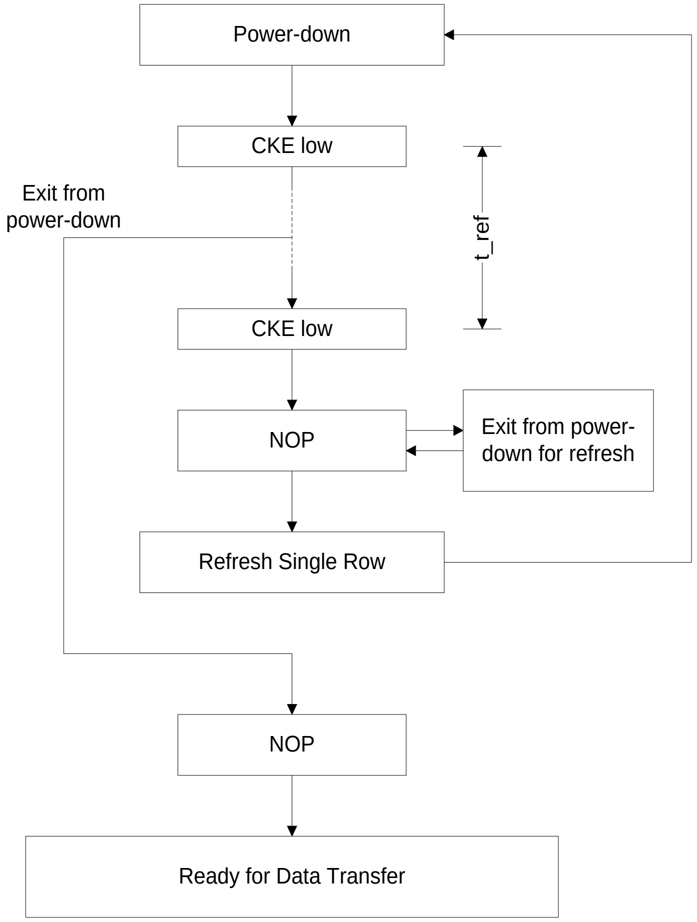


Figure 10: Power DOWN Diagram

4.2.25 SDRAM Chip Select Decoding

There can be a maximum of 8 chip selects. You should specify the size of the memory connected to each chip select in bits 4:0 of the corresponding Address Mask Registers (SMSKRn). The table lists the number of address bits that are used to generate a chip select, which is dependent on the block size programmed in the mask register. Only these bits will be compared with the host address for generating chip selects.

The memory controller supports address aliasing and remapping, which can be done only on chip select0 and chip select 1.

In normal application aliasing is always enabled, so the chip select becomes active when the AHB address matches either the chip select base address or the chip select alias address. If for some reason aliasing is not wanted the alias register should be set equal to the base address i.e. CSALIASn = SCSLRn.

When remapping is enabled, the chip select becomes active when the AHB address matches the chip select base address and remap pin is 0, or when the AHB address matches the chip select remap address and the remap pin is 1.

Mask Register Bits	Block Size	Number of Address Bits for Addressing a Block	Bits Used in Address Comparison

00000	Unused	Unused	Unused
00001	64 KB	16	31:16
00010	128 KB	17	31:17
00011	256 KB	18	31:18
00100	512 KB	19	31:19
00101	1 MB	20	31:20
00110	2 MB	21	31:21
00111	4 MB	22	31:22
01000	8 MB	23	31:23
01001	16 MB	24	31:24
01010	32 MB	25	31:25
01011	64 MB	26	31:26
01100	128 MB	27	31:27
01101	256 MB	28	31:28
01110	512 MB	29	31:29
01111	1 GB	30	31:30
10000	2 GB	31	31:31
10001	4 GB	32	No Address Comparison

Table 18: Chip Select Decoding

4.2.26 SDRAM Read/Write Timing

Timing Parameter	Register and Bit	Description
t_init	STMG1R (15:0)	Internal delay before a command is sent to the memory device
t_rcd	STMG1R (8:7)	Active-to-read/write command time.
t_cas	STMG0R (1:0)	CAS latency. Delay in clock cycles between read command and availability of first data.

Table 19: SDRAM Read/Write Timing Parameters

4.2.26.1 SDRAM Page-Hit Single Write (hburst = Single)

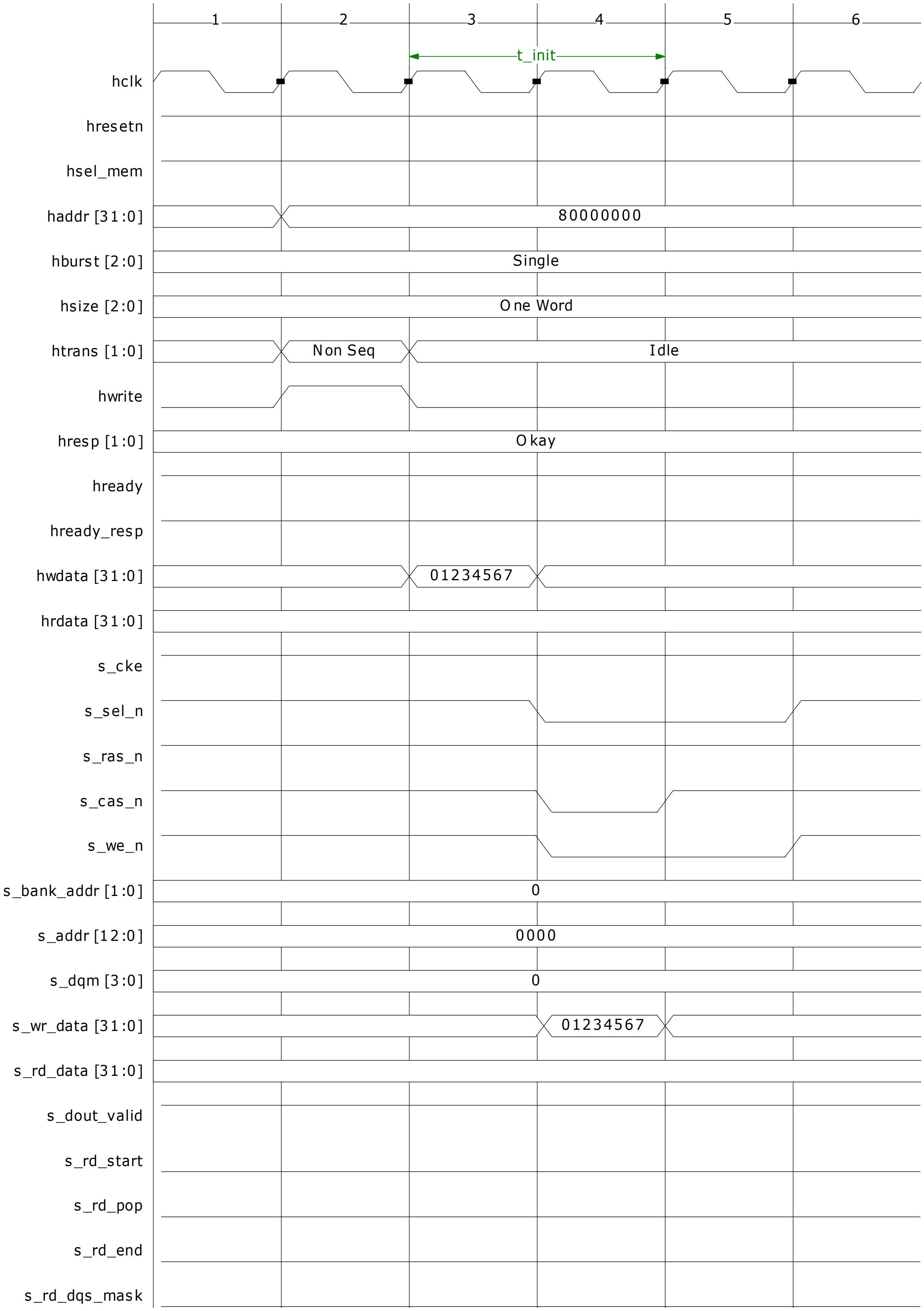


Figure 11: SDRAM Page-Hit Single Write

4.2.26.2 SDRAM Page-Miss Single Write (hburst = Single)

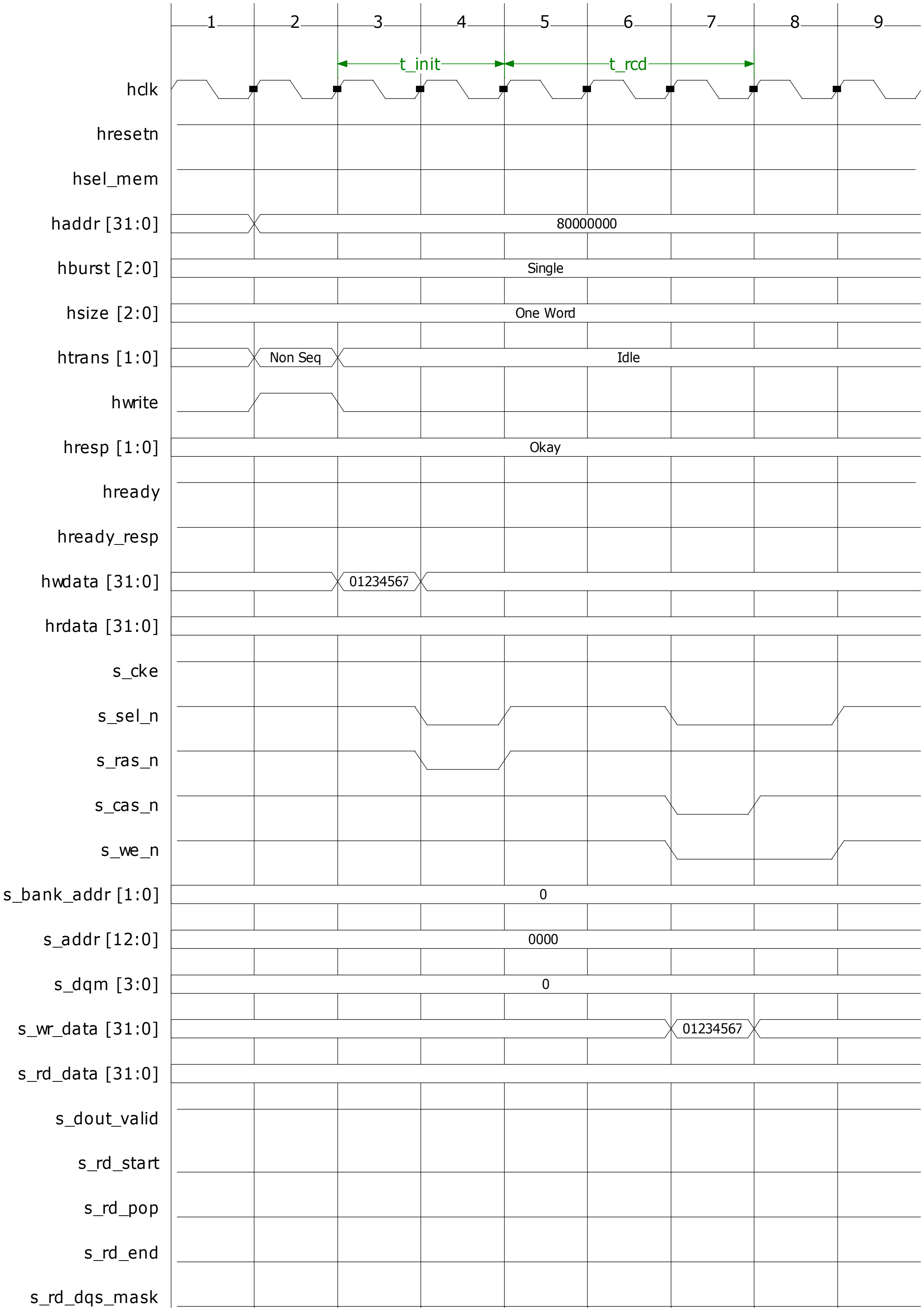


Figure 12: SDRAM Page-Miss Single Write

4.2.26.3 SDRAM Page-Hit Burst Write (hburst = Incr 8)

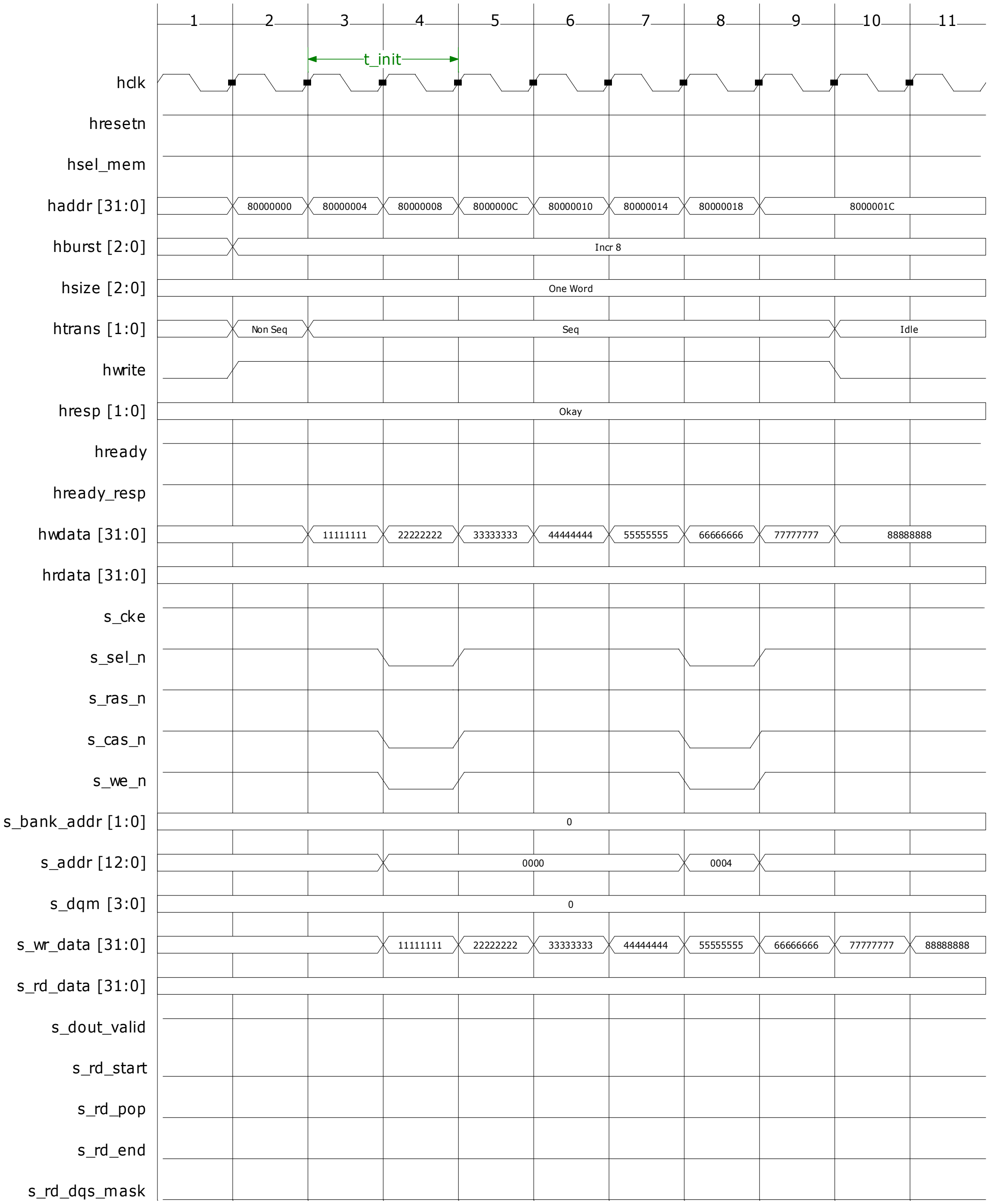


Figure 13: SDRAM Page-Hit Burst Write

4.2.26.4 SDRAM Page-Hit Single Read (hburst = single)

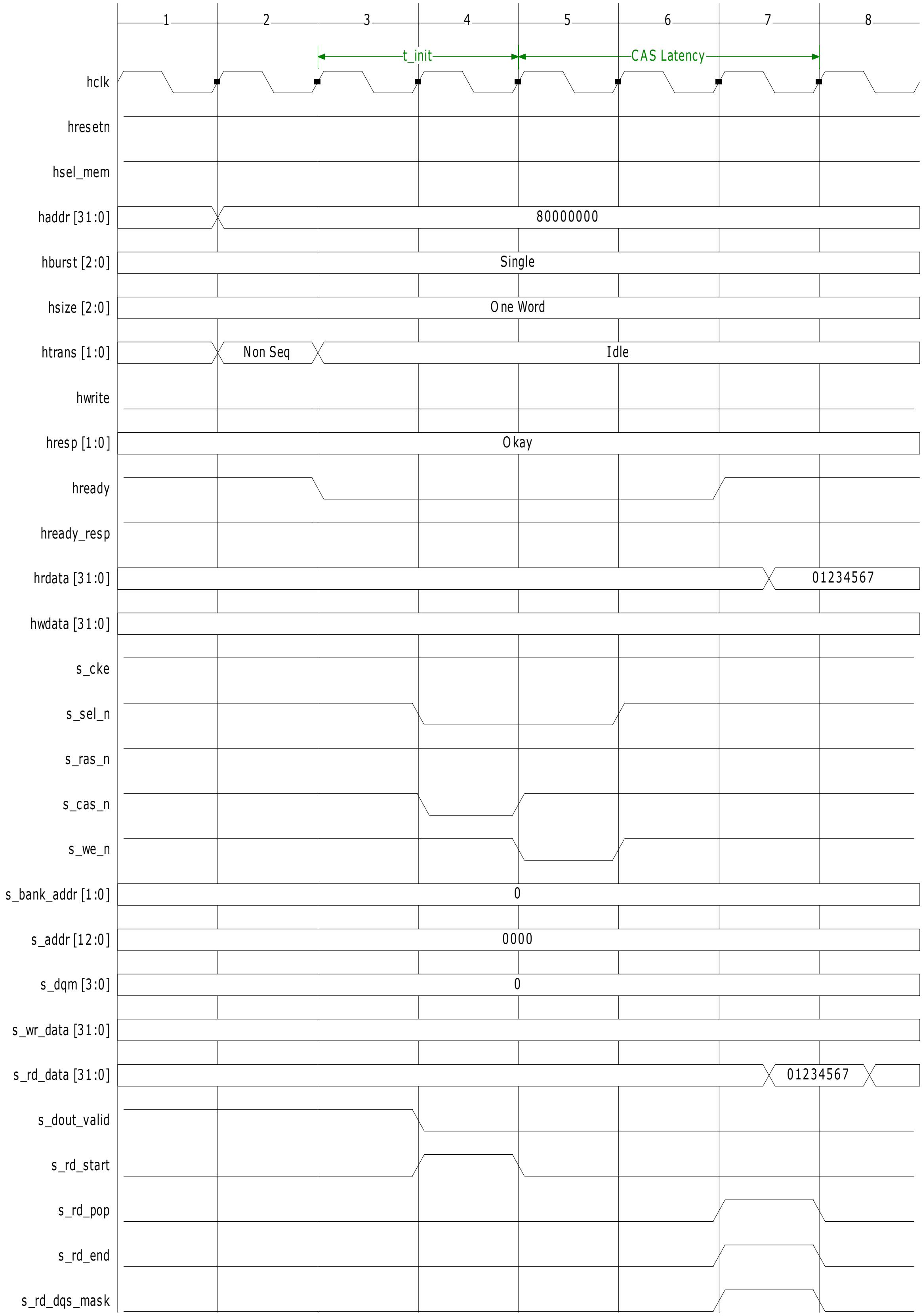


Figure 14: SDRAM Page-Hit Single Read

4.2.26.5 SDRAM Page-Miss Single Read (hburst = single)

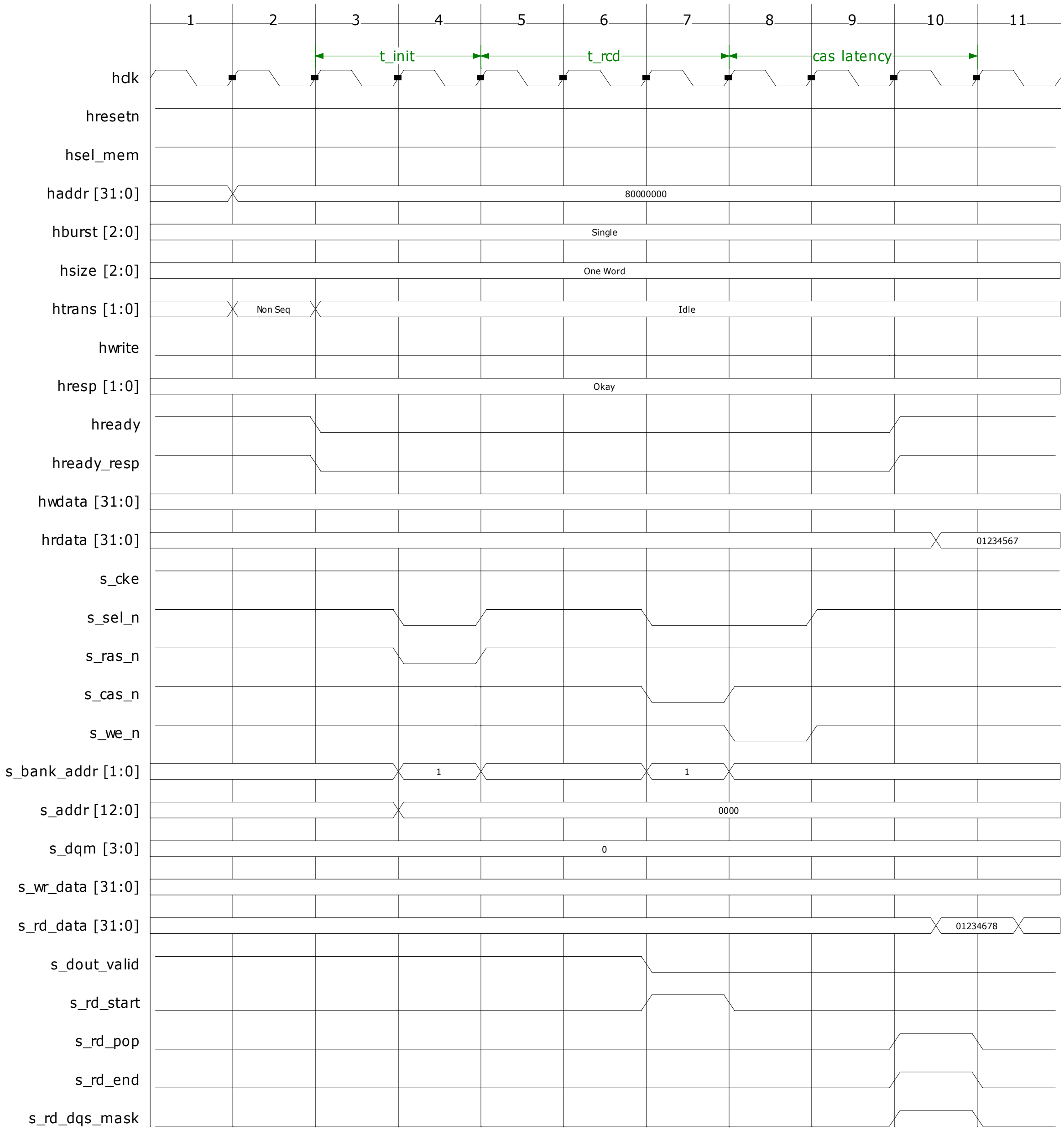


Figure 15: SDRAM Page-Miss Single Read

4.2.26.6 SDRAM Page-Hit Burst Read (hburst = Incr 8)

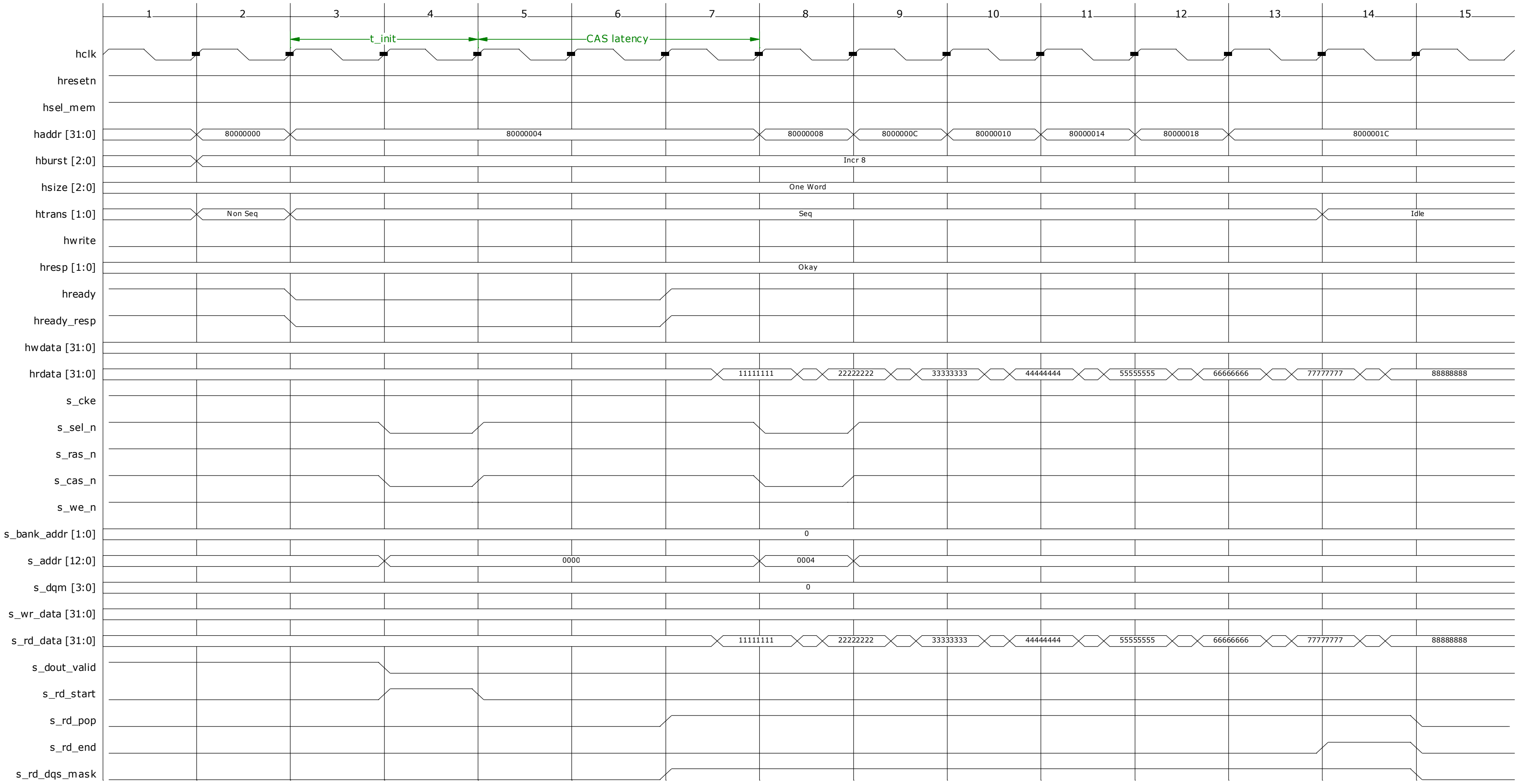


Figure 16: SDRAM Page-Hit Burst Read

4.2.27 Static Memory Configuration

This chapter describes the functional details of the static memory controller. Under the Static memory category, the memory controller supports asynchronous SRAMs, asynchronous FLASH memories, asynchronous ROMs, synchronous SRAMs (non-ZBT) and non-memory devices with ready pin.

The memory controller supports Static memories of various data widths. The memory controller has three sets of timing registers for controlling the Static memory:

- Static Memory Timing Register - Set 0 (SMTMGR_SET0)
- Static Memory Timing Register - Set 1 (SMTMGR_SET1)
- Static Memory Timing Register - Set 2 (SMTMGR_SET2)

The following compile-time and programmable parameters apply when configuring the memory controller for static memories:

- The maximum data width for the static memory data bus is 16 bits.
- You can dynamically specify the Static memory data widths by programming the Static Memory Control Register (SMCTLR). Bits 9:7 specify the static memory data bus width of the memory associated with all three timing registers. If you configure the memory controller as both a static memory controller and an SDRAM controller, then the static memory data width must be less than or equal to the SDRAM data width.
- The maximum address width for the static memory data bus is 24 bits.
- You can dynamically specify the read latencies – which are required if you put extra flip-flops in the read data path to the Static memory – by programming the Static Memory Timing Registers.
- The write latency for the static memory controller is 1cc. A write latency is required if you put extra flip-flops in the write data path to the static memory.

4.2.28 Static Memory Chip Selection

The memory controller supports up to eight chip selects. The Address Mask Registers (SMSKR*n*) and the Chip Select Registers (SCSLR*n)* control the chip select selection.

Bits 4:0 of the mask register specify the size of the memory assigned to a particular chip select. The memory controller supports static memory sizes from 64KB to 4GB. Bits 6:5 specify the type of memory connected to that particular chip select. Bits 8:7 specify the Static Memory Timing Register set which this particular memory is associated with.

The chip select base address registers hold the base address values that correspond to each chip select. The memory controller compares the AHB address with the chip select base address register values in order to generate the chip select.

4.2.29 FLASH Memory

4.2.29.1 Reset/Power DOWN

When the sm_rp_n pin (connected to a FLASH memory module) is driven low, the following happens:

- FLASH internal status registers are cleared.
- Many internal circuits are turned off.
- Device goes into power-down mode.

In this mode, all inputs to the FLASH except sm_rp_n have a value of "Don't Care," and all outputs from the FLASH are high-impedance.

During reset, the memory controller asserts sm_rp_n, which is de-asserted by the memory controller immediately after reset. After reset, all requests to the FLASH will be queued until the t_rpd timer for the FLASH expires. During normal operation, the FLASH Timing Register (FLASH_TRPDR), the sm_power_down pin, and the Static Memory Control Register (SMCTLR) enable you to control the reset/power-down mode of the FLASH.

Even though the memory controller can support up to three different FLASH memories with different timing parameters, there is only one register for specifying t_rpd.

There are two ways to control the reset/power-down mode of a FLASH memory:

- Program bit 0 of Static Memory Control Register. A 0 commands the memory controller to put the FLASH in reset/power-down mode. A 1 commands the memory controller to take the FLASH out of reset/power-down mode.
- Use sm_power_down input pin. FLASH will be in reset/ power-down mode as soon as this signal stays high.

4.2.29.2 Write Protection

Some FLASH memories have a write protection pin that protects important system information in the boot block. For the memory controller, you can control this pin by programming the Static Memory Control Register (SMCTLR). Writing 0 to bits 3:1 of SMCTLR forces a 0 on the WP pin of the FLASH memory.

4.2.29.3 Status Information

Some FLASH memories have a status pin that indicates the status of the internal state machine of the FLASH memory. The memory controller does not have dedicated pins for the status inputs from the FLASH memories. However, you can connect the FLASH status pins to the General Purpose Input (GPI) pins of the memory controller and get the status information by reading corresponding bits in the SDRAM Refresh Interval Register (SREFR).

4.2.30 Static Memory Read/Write Timing

This section explains how the timing parameter specified in the static memory timing registers affect the functioning of static memory read/write operation. The following table gives brief descriptions of the various timing parameters.

Timing Parameter	Register and Bits	Description
t_rc	SMTMGR_SETn (5:0)	Read cycle time
t_prc	SMTMGR_SETn (22:19)	Page mode read cycle time
t_as	SMTMGR_SETn (7:6)	Write address setup time
t_wp	SMTMGR_SETn (15:10)	Write pulse width
t_wr	SMTMGR_SETn (9:8)	Write address/data hold time
t_bta	SMTMGR_SETn (18:16)	Idle cycles between read to write / write to read
t_rpd	FLASH_TRPDR (11:0)	FLASH reset/power-down

Table 20: Static Memory Read/Write Timing Parameters

4.2.30.1 Read Timing of SRAM, FLASH and ROM

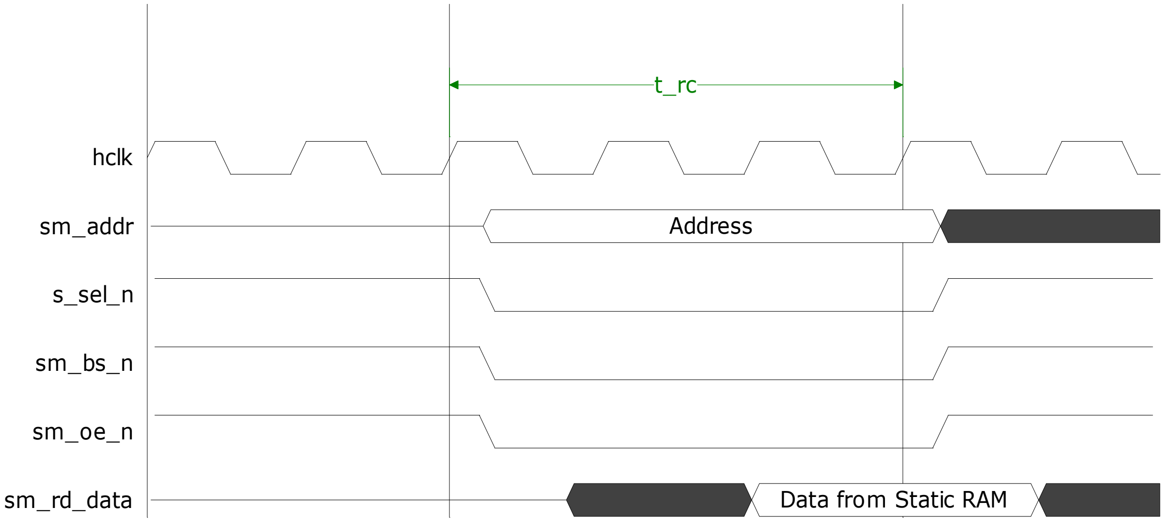


Figure 17: Static Read Timing

4.2.30.2 Page Read Timing of FLASH and ROM

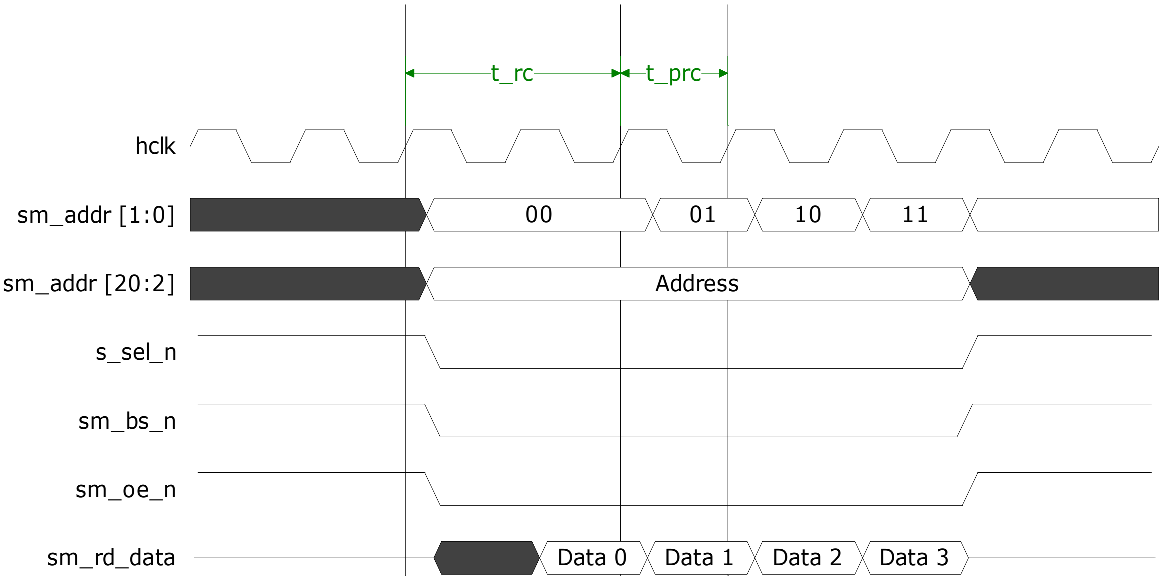


Figure 18: FLASH and ROM Page Read Timing

4.2.30.3 Write Timing of SRAM and FLASH

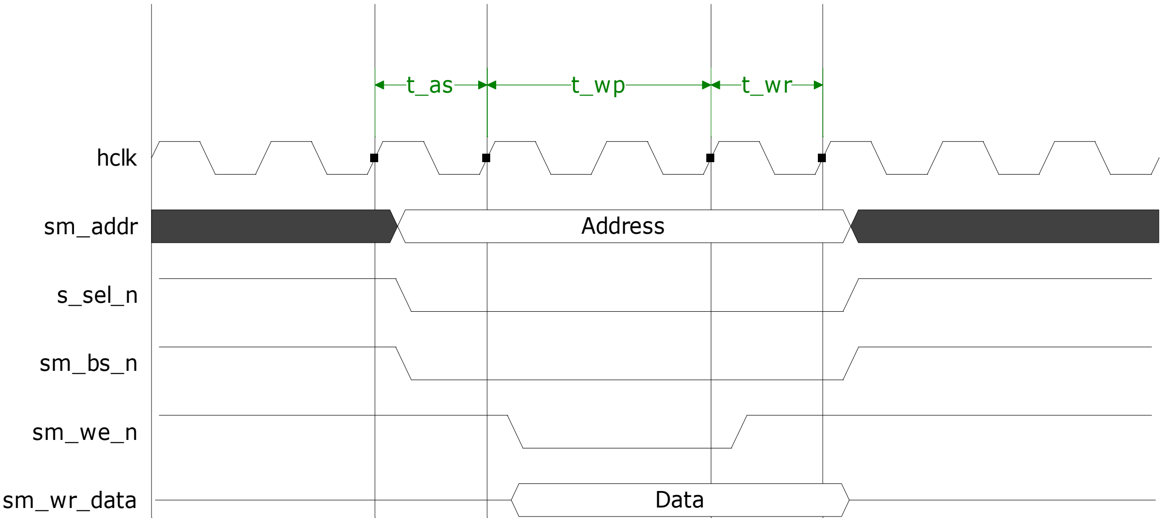


Figure 19: SRAM and FLASH Write Timing

4.2.30.4 External Memory Data Bus Turnaround Timing

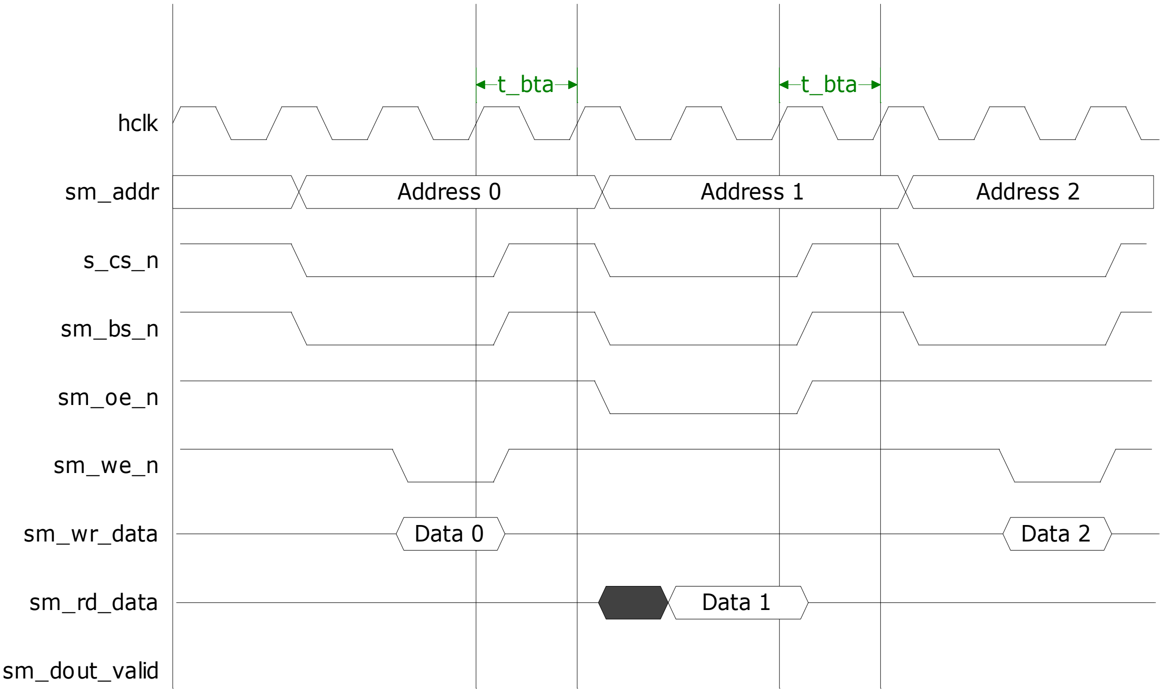


Figure 20: Turnaround Timing

4.2.30.5 First Read After Reset/Power-Down

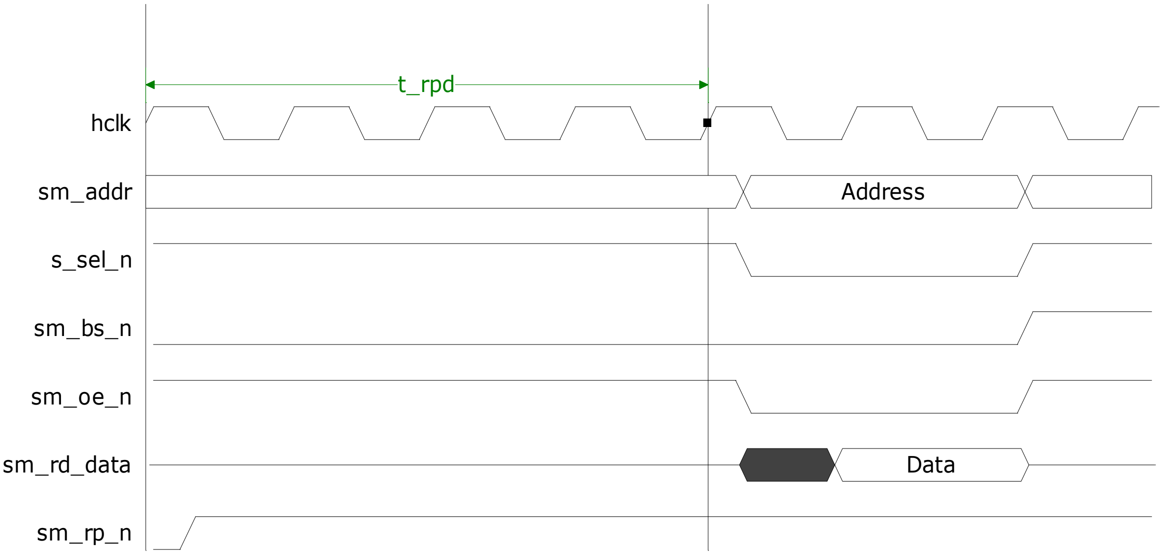


Figure 21: Reset/Power-Down Timing

4.2.31 Interfacing to Non-Memory Devices with Ready Pin (DSP)

The Memory Controller supports non-memory devices with a ready pin, such as a DSP. This type of device has the same interface as an asynchronous SRAM, except that it has a ready pin to indicate that the read data is available on the data bus or that the write data is accepted by the device.

The ready pin of the device should be connected to the SRAM_READY pin on the DICEJR/Mini. Note that the SRAM_READY pin is a multi-function pin, so the SRAM_READY bit of the GPCSR_GPIO_SELECT register must be set to 1.

The READY_MODE bit of the Static Timing Register (SMTMGR_SET0/1/2 – bit 26) should be set to 1.

4.2.31.1 I/O Interface between Non-Memory Device and DICE JR/Mini

The following are conditions for interfacing Non-Memory device pins to the DICE JR/Mini Memory Controller pins:

- Address pins – Connect the address pins to the DICE JR/Mini SDRAM/SRAM shared address pins.
- Chip select pin – Connect the chip select pin to one of the chip select pins. Specify which chip select is connected to the Non-Memory device in the memory type bits (bits 6:5) of the mask register (SMSKR0-7) that correspond to a particular chip select.
- Output enable pin – Connect the output enable pin of the Non-Memory device to the SRAM_OE pin.
- Write enable pin – Connect the write enable pin of the Non-Memory device to the SRAM_WE pin.
- Byte control pins – Connect the byte enable pins to the SRAM_BS pins. The lower byte enable should be connected to SRAM_BS[0] and the upper byte enable should be connected to SRAM_BS[1].
- Data inputs/outputs – Drive the bidirectional data pins of the Non-Memory device using the DICE JR/Mini data bus pins.

4.2.31.2 Timing Diagrams of Read/Write Accesses

The figure below shows the timing diagram of a read access. The Memory Controller checks SRAM_READY after the tRC read access time. When SRAM_READY is high, the Memory Controller latches read data at the next rising clock edge.

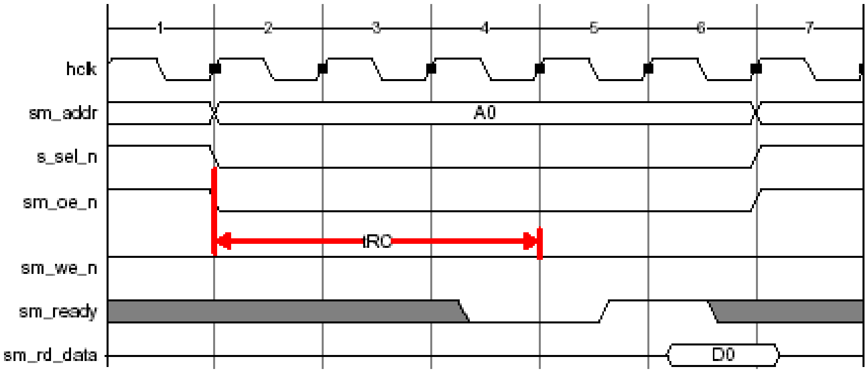


Figure 22: Read Access of the Device with Ready Signal

The figure below shows the timing diagram of a write access. The Memory Controller checks SRAM_READY after a time equal to "tAS (address setup time) + tWP (write period)." When the SRAM_READY is high, the write is finished.

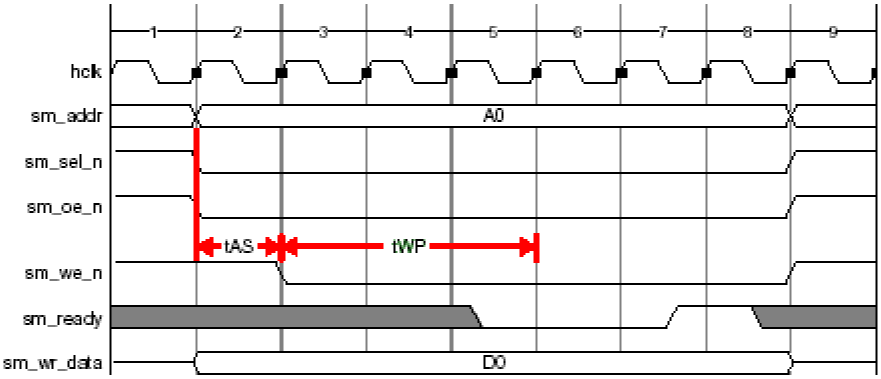


Figure 23: Write Access of the Device with Ready Signal

4.2.31.3 Specifying the Timing Parameters

You can specify timing parameters of the Non-Memory device by programming the Static Memory Timing Register - Set 0 (SMTMGR_SET0), Static Memory Timing Register - Set 1 (SMTMGR_SET1), or Static Memory Timing Register - Set 2 (SMTMGR_SET2), depending on which of the three register sets should control the device.

You can use the following timing parameters:

- Read cycle time (tRC) – Bits 5:0 of SMTMGR_SET0/1/2 specify the read cycle time.
- Address setup time (tAS) – Bits 7:6 of the Static Memory Timing Register SMTMGR_SET0/1/2.
- Write period (tWP) – Bits 15:10 of SMTMGR_SET0/1/2.
- Bus turnaround time (tBTA) – Bits 18:16 of SMTMGR_SET0/1/2 force the Memory Controller to insert tBTA number of cycles between back-to-back read/writes.

4.3 I^2C

4.3.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
I2C_CLK	104 (shared)	97 (shared)	I/O (S)	6	I2C Clock (OD[11], 5V)
I2C_DATA	105 (shared)	98 (shared)	I/O (S)	6	I2C Data (OD, 5V)

Table 21: Signal Description

4.3.2 Features

For a full description of the I2C standard, including timing and frame format diagrams, see the Phillips I2C specification.

The I2C interface implemented in the TCD22XX is fully compliant with Philips I2C definitions. The following is a list of general features of the DICE JR I2C Interface:

- The APB data width is 32 bits.
- The highest I2C speed mode supported is high (standard and fast modes are also supported).
- Supports 10-bit addressing in both master and slave mode
- 8-bit receive and transmit buffers
- 100pF bus loading

Note that all pins used by the I2C module are multi-purpose or shared. The function of these pins is software configurable via the GPCSR module, specifically register GPCSR_GPIO_SELECT – 0xc700 0008. Refer to the GPCSR module documentation for more information.

[11] OD indicates Open Drain pad type. External Pull-Up resistor required.

4.4 I2C Overview

The I2C bus is a two-wire serial interface. The I2C Interface module can operate in both standard mode (with data rates up to 100 Kb/s), fast mode (with data rates up to 400 Kb/s), and high-speed mode (with data rates up to 3.4 Mb/s). The I2C Interface can communicate with devices only of these modes as long as they are attached to the bus. The I2C serial clock determines the transfer rate.

The I2C interface protocol is setup with a master and slave. The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of data is sent by the device that is receiving data, which can be either the master or the slave. The protocol also allows multiple masters to reside on the I2C bus, which requires the masters to arbitrate for ownership.

The slaves each have a unique address that is determined by the system designer. When the master wants to communicate with a slave, the master transmits a start condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends acknowlegment (ACK) pulse after the address and R/W bit is received to notify the master that the slave has received the request.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver receives a byte of data. This transaction continues until the master terminates the transmission with a stop condition. If the master is reading from a slave, the slave transmits a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging the transaction after the last byte is received, and then the master issues a stop condition or addresses another slave after issuing a restart condition.

The I2C Interface is a synchronous serial interface. The data signal (SDA) is a bidirectional signal and changes only while the serial clock signal (SCL) is low. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

4.4.1 I2C START and STOP Condition Protocol

When the bus is IDLE both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1.

4.4.2 I2C Addressing Slave Protocol

There are two address formats: the 7-bit address format and the 10-bit address format. During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit When Bit 8 is set to 0, the master writes to the slave. When Bit 8 (R/W) is set to 1, the master reads from the slave. Data is transmitted most significant bit (MSB) first. During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (Bit 8) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address.

4.4.3 I2C Transmitting and Receiving Protocol

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal. When a slave-receiver does not respond with an acknowledge pulse, the master aborts the transfer by issuing a STOP condition. The slave shall leave the SDA line high so the master can abort the transfer.

4.4.4 I2C START BYTE Transfer Protocol

The START BYTE transfer protocol is set up for systems that do not have an on board dedicated I2C hardware module. When the I2C Interface is addressed as a slave, it always samples the I2C bus at the highest speed supported so that it never requires a START BYTE transfer. However, when the I2C Interface is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave device requires it. The START BYTE protocol consists of seven zeros being transmitted followed by a 1. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master. The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse.
4. No slave sets the ACK signal to 0.
5. Master generates a repeated START (Sr) condition.

A hardware receiver does not respond to the START BYTE because it is a reserved address and resets after the Sr (restart condition) is generated.

4.4.5 Operation Modes

Slave Mode Operation
Initial configuration

To use the I2C Interface as a slave, perform the following steps:

1. Disable the I2C Interface by writing a 0 to the IC_ENABLE register.
2. Write to the IC_SAR register to set the slave address. This is the address to which the I2C Interface responds.
3. Write to the IC_CON register to specify whether 10-bit addressing is supported and whether the I2C Interface is in slave-only or master-slave mode.
4. Enable the I2C Interface with the IC_ENABLE register.

Note: Depending on the reset values chosen, Steps 2 and 3 may not be necessary. If the I2C Interface is configured to use a default reset address, these registers do not need to be programmed. The values stored are static and do not need to be reprogrammed if the I2C Interface is disabled.

Slave-Transmitter Operation
When another master addresses the I2C Interface and requests data, the I2C Interface acts as a slave-transmitter and the following steps occur:

1. The other master initiates an I2C transfer with an address that matches the slave address in the IC_SAR register of the I2C Interface.
2. The I2C Interface acknowledges the sent address and recognizes the direction of transfer to indicate that it is acting as a slave-transmitter.
3. The I2C Interface asserts the RD_REQ interrupt (IC_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds.
4. If there is any data remaining in the TX FIFO before receiving the read request, then the I2C Interface asserts a TX_ABRT interrupt (IC_RAW_INTR_STAT register) to flush the old data from the TX FIFO.
5. Software then writes the IC_DATA_CMD register with the data to be written. The CMD bit, Bit 8, should be set to write (0).
6. Software should clear the RD_REQ and TX_ABRT interrupts before proceeding.
7. The I2C Interface releases the SCL and transmits the byte.
8. The master may hold the I2C bus by issuing a restart condition or release the bus by assuming a stop condition.

Slave-Receiver operation
When another master addresses the I2C Interface and is sending data, the I2C Interface acts as a slave-receiver and the following steps occur:

1. The other master initiates an I2C transfer with an address that matches the I2C Interface’s slave address in the IC_SAR register.
2. The I2C Interface acknowledges the sent address and recognizes the direction of transfer to indicate that the I2C Interface is acting as a slave-receiver.
3. The I2C Interface receives the transmitted byte and place it in the receive buffer, assuming there is room.
4. The status and interrupt bits corresponding to the receive buffer is updated.
5. Software may read the byte from the IC_DATA_CMD register.

6. The other master may hold the I2C bus by issuing a restart condition or release the bus by issuing a stop condition.

Slave Bulk Transfer Mode

In the standard I2C protocol, all transaction are single byte transactions and the programmer responds to a remote master read request by writing one byte into the TXFIFO. For the Slave Bulk Transfer mode, if the remote master acknowledged the sent byte to request more data, then the slave must hold the I2C SCL line low and request the next byte from the processor side.

If the programmer knows in advance that the remote master is requesting a packet of *n* bytes, then when another master addresses the I2C Interface and request data, the TXFIFO could be written with *n* number bytes and the remote master will receive it as a continuous stream of data. For example, the I2C Interface slave will continue to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the TX_FIFO. There is no need to hold the SCL line low or to issue RD-REQ again.

If the remote master is to receive *n* bytes from the I2C Interface but the programmer wrote a number of bytes larger than *n* to the TX-FIFO then when the slave finishes sending the requested *n* bytes, it will clear the TX-FIFO and ignore any excess bytes.

Master Mode Operation

Initial configuration

To use the I2C Interface as a master, perform the following steps:

1. Disable the I2C Interface by writing 0 to the IC_ENABLE register.
2. Write to the IC_SAR register to set the slave address, which is the address to which the I2C Interface responds.
3. Write to the IC_CON register to set the maximum speed mode supported for slave operation and the desired speed of the I2C Interface master-initiated transfers, either 7-bit or 10-bit addressing.
4. Write to the IC_TAR register to the address of the I2C device to be addressed. It also indicates whether adding a START BYTE or issuing a general call is going to occur.
5. *Only applicable for high-speed mode transfers*. Write to the IC_HS_MADDR register the desired master code for the I2C Interface.
6. Enable the I2C Interface with the IC_ENABLE register.
7. Commands and data to be sent may be written now to the IC_DATA_CMD register. If the IC_DATA_CMD register is written before the I2C Interface is enabled, the data and commands are lost as the buffers are kept cleared when I2C Interface is not enabled.

Note: Depending on the reset values chosen, Steps 2, 3, 4, and 5 may not be necessary because the reset values can be configured. The values stored are static and do not need to be reprogrammed if the I2C Interface is disabled, with the exception of the commands and data.

Master Transmit and Master Receive

The I2C Interface supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the IC_DATA_CMD register. The CMD bit, Bit 8, should be written to 0 for write operations.

Subsequently, a read command may be issued by writing “don’t cares” to the lower byte of the IC_DATA_CMD register, and a 1 should be written to the CMD bit. As data is transmitted and received, the transmit and receive buffer status bits and interrupts change.

4.4.6 I2C IC_CLK Frequency Configuration

The *CNT registers must be set when configured as a master before any I2C bus transaction can take place to ensure proper I/O timing. The *CNT registers are:

- IC_SS_SCL_HCNT
- IC_SS_SCL_LCNT
- IC_FS_SCL_HCNT
- IC_FS_SCL_LCNT
- IC_HS_SCL_HCNT
- IC_HS_SCL_LCNT

Setting the *_LCNT registers, configures the number of IC_CLKs that are required for setting the low time of the SCL clock in each speed mode. Setting the *_HCNT* registers, configures the number of IC_CLKs that are required for setting the high time of the SCL clock in each speed mode. Setting the registers to the correct value is described as follows.

The equation to calculate the proper number of IC_CLKs required for setting the proper SCL clocks high and low times is as follows:

IC_xCNT = (ROUNDUP(MIN_SCL_xxxtime*OSCFREQ,0))

ROUNDUP is an explicit Excel function call that is used to roundup the results of the division to an integer.

MIN_SCL_HIGHtime = Minimum High Period
MIN_SCL_HIGHtime = 4000 ns for 100 kbps
600 ns for 400 kbps
60 ns for 3.4 Mbs, bus loading = 100pF
160 ns for 3.4 Mbs, bus loading = 400pF

MIN_SCL_LOWtime = Minimum Low Period
MIN_SCL_LOWtime = 4700 ns for 100 kbps
1300 ns for 400 kbps
120 ns for 3.4Mbs, bus loading = 100pF
320 ns for 3.4Mbs, bus loading = 400pF

OSCFREQ = IC_CLK Clock Frequency (Hz).

For example:
OSCFREQ = 100 MHz
I2Cmode = fast, 400 kbit/s
MIN_SCL_HIGHtime = 600 ns.
MIN_SCL_LOWtime = 1300 ns.

IC_xCNT = (ROUNDUP(MIN_SCL_HIGH_LOWtime*OSCFREQ,0))

IC_HCNT = (ROUNDUP(600 ns * 100 MHz,0))
IC_HCNTSCL PERIOD = 60
IC_LCNT = (ROUNDUP(1300 ns * 100 MHz,0))
IC_LCNTSCL PERIOD = 130

Actual MIN_SCL_HIGHtime = 60*(1/100 MHz) = 600 ns
Actual MIN_SCL_LOWtime = 130*(1/100 MHz) = 1300 ns

4.4.7 I2C General Notes

When the I2C Interface is configured in the master mode of operation, the minimum value for *_LCNT is 8 and the minimum *_HCNT is 6. Also, because of the digital filtering on the receiver, the actual SCL high and low times are slightly longer than the specified count value—8 more ic_clks for SCL high and 1 more ic_clk for SCL low period. You may subtract 8 from your low count and 1 from the high count values to account for this. The following six points describe why this occurs:

The minimum ic_clk oscillator frequency for standard mode is 2 MHz; fast mode is 10 MHz; and for high-speed mode is 100 MHz. According to the I2C specifications, the minimum time period to be able to generate or detect at a 3.4 Mb/s data rate is 60 ns (HIGH), which means theoretically that the minimum ic_clk clock frequency should be >= 33 MHz. Given this:

You do not have to have the I2C module running at the clock speeds listed previously to support all different modes. However, you have to run at only those speeds if you are willing to operate your master at a 3.4 Mb/s data rate or at the highest supported clock rate. These MIN_SCL_HIGHtime and MIN_SCL_LOWtime values for high-speed mode depend on loading in the system as described in the previous equation. Please see the *I2C-BUS Specification* and information from Phillips for more detail.

The final values calculated in the equation for IC_*_HCNT and IC_*_LCNT (where * represents SS, FS, or HS) are decimal values. For programming the actual registers, the values must be converted to hexadecimal. The 16-bit range on these registers allows a wide range of input clock frequencies to be used.

4.4.8 Module Configuration

Address	Register
0xc400 0000	IC_CON
0xc400 0004	IC_TAR
0xc400 0008	IC_SAR
0xc400 000c	IC_HS_MAR
0xc400 0010	IC_DATA_COMMAND
0xc400 0014	IC_SS_HCNT
0xc400 0018	IC_SS_LCNT
0xc400 001c	IC_FS_HCNT
0xc400 0020	IC_FS_LCNT
0xc400 0024	IC_HS_HCNT
0xc400 0028	IC_HS_LCNT
0xc400 002c	IC_INTR_STAT
0xc400 0030	IC_INTR_MASK
0xc400 0034	IC_RAW_INTR_STAT
0xc400 0038	IC_RX_TL
0xc400 003c	IC_TX_TL
0xc400 0040	IC_CLR_INTR
0xc400 0044	IC_CLR_RX_UNDER
0xc400 0048	IC_CLR_RX_OVER
0xc400 004c	IC_CLR_TX_OVER
0xc400 0050	IC_CLR_RD_REQ
0xc400 0054	IC_CLR_TX_ABRT
0xc400 0058	IC_CLR_RX_DONE
0xc400 005c	IC_CLR_ACTIVITY
0xc400 0060	IC_CLR_STOP_DET
0xc400 0064	IC_CLR_START_DET
0xc400 0068	IC_CLR_GEN_CALL
0xc400 006c	IC_ENABLE
0xc400 0070	IC_STATUS
0xc400 0074	IC_TXFLR
0xc400 0078	IC_RXFLR
0xc400 007c	IC_SRESET
0xc400 0080	IC_TX_ABRT_SOURCE

Table 22: I2C Register set

4.4.9 Programming the I2C Interface

Some registers may only be written when the I2C module is disabled as corresponding to Register IC_ENABLE. Software should not disable the I2C module while it is active. If the module was transmitting it will stop as well as delete the contents of the transmit buffer after the current transfer is complete. If the module was receiving, it will stop the current transfer at the end of the current byte and not acknowledge the transfer. Registers that cannot be written to when the I2C module is disabled are indicated in their descriptions.

4.4.10 IC_CON register – I2C Control

Address – 0xC400 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									IC_SLAVE_DISABLE	IC_RESTART_EN	IC_10BITADDR_MASTER	IC_10BITADDR_SLAVE	SPEED		IC_MASTER_MODE
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	3		1
	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW		RW

This register can be written only when the I2C interface is disabled. That condition corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect on the register.

Name	Bit	Reset	Dir	Description
IC_SLAVE_DISABLE	6	0	RW	This bit controls whether I2C has its slave disabled after reset. The slave can be disabled by programming a ‘1’ into IC_CON[6]. By default the slave is enabled. 0: slave is enabled 1: slave is disabled
IC_RESTART_EN	5	1	RW	Determines whether re-start conditions may be sent when acting as a master. Some older slaves do not support handling re-start conditions. Re-start conditions are used in several I2C operations. Disabling re-start will not allow the master to perform these functions: - send multiple bytes per transfer (split) - change direction within a transfer (split) - send a start byte - perform any high speed mode operation - perform combined format transfers in 7 or 10 bit addressing modes (split for 7 bit) - perform a read operation with a 10 bit address Operations which are split are broken down into multiple I2C transfers with a stop and start condition in between. The other operations will not be performed at all, and will result in setting TX_ABRT.
IC_10BITADDR_MASTER	4	1	RW	Controls whether the I2C module starts its transfers in 10-bit addressing mode. 0 = 7-bit addressing, 1 = 10-bit addressing

Name	Bit	Reset	Dir	Description
IC_10BITADDR_SLAVE	3	1	RW	When acting as a slave, this bit controls whether the I2C module responds to 7 or 10 bit addresses. 0: 7 bit addressing, the I2C module will ignore transactions which involve 10 bit addressing, for 7 bit addressing only the lower 7 bits of Register IC_SAR will be compared. 1: 10 bit addressing, the I2C module will only respond to 10 bit addressing transfers which match the full 10 bits of Register IC_SAR.
SPEED	2:1	3	RW	Controls what speed the I2C module will operate at. 0 = illegal, writing a 0 will result in setting SPEED to 3 1 = standard mode (100 kbit/s) 2 = fast mode (400 kbit/s) 3 = high speed mode (3.4 Mbit/s)
MASTER_MODE	0	1	RW	This bit controls whether the I2C master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled

4.4.11 IC_TAR register – I2C Target Address

Address – 0xC400 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				SPECIAL	GC_OR_START	IC_TAR									
Reset:	0	0	0	0	0	0	0									
	R	R	R	R	RW	RW	RW									

This register can be written only when the I2C interface is disabled. That condition corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect on the register.

Name	Bit	Reset	Dir	Description
SPECIAL	11	0	RW	This bit indicates whether software would like to perform a general call or start byte I2C command. 0 = ignore bit 10 GC_OR_START and use IC_TAR normally 1 = perform special I2C command as specified in GC_OR_START bit
GC_OR_START	10	0	RW	This bit indicates whether a general call or start byte command is to be performed by the I2C module. 0 = General Call Address – after issuing a general call, only writes may be performed. Attempting to issue a read command will result in setting TX_ABRT. After the I2C module is disabled by writing logic 0 to Register IC_ENABLE the I2C module will revert back to normal operation. 1 = Start Byte

Name	Bit	Reset	Dir	Description
IC_TAR	9:0	0	RW	This is the target address for any master transactions. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect.

4.4.12 IC_SAR register – I2C Slave Address

Address – 0xC400 0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						IC_SAR									
Reset:	0	0	0	0	0	0	0x55									
	R	R	R	R	R	R	RW									

Name	Bit	Reset	Dir	Description
IC_SAR	9:0	0x055	RW	The IC_SAR holds the slave address when the I2C module is operating as a slave. IC_SAR holds the slave address that the I2C module will respond to. For 7 bit addressing only IC_SAR[6:0] will be used. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect.

4.4.13 IC_HS_MAR register – I2C Master Mode Code Address

Address – 0xC400 000C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													IC_HS_MAR		
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
	R	R	R	R	R	R	R	R	R	R	R	R	R	RW		

Name	Bit	Reset	Dir	Description
IC_HS_MAR	2:0	1	RW	IC_HS_MAR holds the value of the I2C HIGH SPEED mode master code. Valid values are from 1-7, 0 being reserved. Note that the value 0 should not be used since that code is reserved according to the I2C specification. Writing a value of 0 to this register will be ignored. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect.

4.4.14 IC_DATA_CMD register – I2C RX/TX Data Buffer and Command

Address – 0xC400 0010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							CMD	DAT							
Reset:	0	0	0	0	0	0	0	0	0							
	R	R	R	R	R	R	R	W	RW							

Name	Bit	Reset	Dir	Description
CMD	8	0	W	This bit controls whether a read or a write is performed. Logic 1 corresponds to read. Logic 0 corresponds to write. For reads the lower 8 (DAT) bits are ignored by the I2C. Reading this bit returns logic 0. Attempting to perform a read operation after a general call command has been sent will result in TX_ABRT if the I2C module has not been previously disabled. This bit is ignored if the write to the tx buffer is in response to a RD_REQ.
DAT	7:0	0	RW	This register contains the data to be transmitted or received on the I2C bus. Read these bits to read out the data received on the I2C interface. Write these bits to send data out on the I2C interface.

4.4.15 IC_SS_HCNT register – Standard Speed IC_CLK High Count

Address – 0xC400 0014

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_SS_HCNT															
Reset:	0x0190															
	RW															

Name	Bit	Reset	Dir	Description
IC_SS_HCNT	15:0	0x0190	RW	The IC_SS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for STANDARD speed. It is not used for HIGH SPEED mode but must be set correctly since HIGH SPEED mode is initiated at lower speeds that may use these values. Sample I2C STANDARD SPEED high period count calculations are shown in Table 2. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 6, values less than that will result in 6 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	SCL High required min (US)	H_CNT (HEX)	SCL High Time (US)
100	2	4	0008	4.00
100	6.6	4	001B	4.09
100	10	4	0028	4.00
100	75	4	012C	4.00
100	100	4	0190	4.00
100	125	4	01F4	4.00
100	1000	4	0FA0	4.00

4.4.16 IC_SS_LCNT register – Standard Speed IC_CLK Low Count

Address – 0xC400 0018

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_SS_LCNT															
Reset:	0x01d6															
	RW															

Name	Bit	Reset	Dir	Description
IC_SS_LCNT	15:0	0x01d6	RW	The IC_SS_LCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for STANDARD speed. It is not used for HIGH SPEED mode but must be set correctly since HIGH SPEED mode is initiated at lower speeds that may use this value. Sample I2C STANDARD SPEED low period count calculations are shown in Table 3. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 8, values less than that will result in 8 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	SCL Low required min (US)	L_CNT (HEX)	SCL Low Time (US)
100	2	4.7	000A	5.00
100	6.6	4.7	0020	4.85
100	10	4.7	002F	4.70
100	75	4.7	0161	4.71
100	100	4.7	01D6	4.70
100	125	4.7	024C	4.70
100	1000	4.7	125C	4.70

4.4.17 IC_FS_HCNT register – Fast Speed IC_CLK High Count

Address – 0xC400 001C

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td></td><td colspan="16">IC_FS_HCNT</td></tr>
<tr><td>Reset:</td><td colspan="16">0x003c</td></tr>
<tr><td></td><td colspan="16">RW</td></tr>
</table>

Name	Bit	Reset	Dir	Description
IC_FS_HCNT	15:0	0x003c	RW	The IC_FS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for FAST SPEED. It is not used for HIGH SPEED mode but must be set correctly since HIGH SPEED mode is initiated at lower speeds that may use these values. Sample I2C FAST SPEED high period count calculations are shown in Table 4. This register goes away and becomes read-only returning 0's if MAX_SPEED_MODE = STANDARD. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 6, values less than that will result in 6 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	SCL High required min (US)	H_CNT (HEX)	SCL High Time (US)
400	10	0.6	0006	0.60
400	25	0.6	000F	0.60
400	50	0.6	001E	0.60
400	75	0.6	002D	0.60
400	100	0.6	003C	0.60
400	125	0.6	004B	0.60
400	1000	0.6	0258	0.60

4.4.18 IC_FS_LCNT register – Fast Speed IC_CLK Low Count

Address – 0xC400 0020

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_FS_LCNT															
Reset:	0x0082															
	RW															

Name	Bit	Reset	Dir	Description
IC_FS_LCNT	15:0	0x0082	RW	The IC_FS_LCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for FAST SPEED. It is not used for HIGH SPEED mode but must be set correctly since HIGH SPEED mode is initiated at lower speeds that may use this value. Sample I2C FAST SPEED low period count calculations are shown in Table 5. This register goes away and becomes read-only returning 0's if MAX_SPEED_MODE = STANDARD. This register can only be written when the I2C interface is disabled which corresponds Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 8, values less than that will result in 8 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	SCL Low required min (US)	L_CNT (HEX)	SCL Low Time (US)
400	10	1.3	000D	1.30
400	25	1.3	0021	1.32
400	50	1.3	0041	1.30
400	75	1.3	0062	1.31
400	100	1.3	0082	1.30
400	125	1.3	00A3	1.30
400	1000	1.3	0514	1.30

4.4.19 IC_HS_HCNT register – High Speed IC_CLK High Count

Address – 0xC400 0024

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_HS_HCNT															
Reset:	0x0006															
	RW															

Name	Bit	Reset	Dir	Description
IC_HS_HCNT	15:0	0x0006	RW	The IC_HS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for HIGH SPEED. Sample I2C HIGH SPEED high period count calculations are shown in Table 6. The SCL high time is 60ns. This register goes away and becomes read-only returning 0's if MAX_SPEED_MODE != HIGH. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 6, values less than that will result in 6 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	I2C bus loading (pF)	SCL High required min (US)	H_CNT (HEX)	SCL High Time (US)
3400	100	100	60	0006	60
3400	125	100	60	0008	64
3400	1000	100	60	003C	60
3400	100	400	120	000C	120
3400	125	400	120	000F	120
3400	1000	400	120	0078	120

4.4.20 IC_HS_LCNT register – High Speed IC_CLK Low Count

Address – 0xC400 0028

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC_HS_LCNT															
Reset:	0x0010															
	RW															

Name	Bit	Reset	Dir	Description
IC_HS_LCNT	15:0	0x0010	RW	The IC_HS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for HIGH SPEED. Sample I2C HIGH SPEED low period count calculations are shown in Table 7. The SCL low time is 160ns. This register goes away and becomes read-only returning 0's if MAX_SPEED_MODE != high. This register can only be written when the I2C interface is disabled which corresponds to Register IC_ENABLE being set to 0. Writes at other times will have no effect. The minimum valid value is 8, values less than that will result in 8 being set.

I2C Data Rate (kbps)	ic_clk (MHz)	I2C bus loading (pF)	SCL Low required min (US)	L_CNT (HEX)	SCL Low Time (US)
3400	100	100	160	0010	160
3400	125	100	160	0014	160
3400	1000	100	160	00A0	160
3400	100	400	320	0020	320
3400	125	400	320	0028	320
3400	1000	400	320	0140	320

4.4.21 IC_INTR_STAT register – I2C Interrupt Status

Address – 0xC400 002C

Each bit in this register has a corresponding mask bit in Register IC_INTR_MASK. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in Register IC_RAW_INTR_STAT.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				GEN_CALL	START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABRT	RE_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
GEN_CALL	11	0	R	Indicates that a general call request was received. The I2C module will store the received data in the RX buffer.
START_DET	10	0	R	Indicates whether a start condition has occurred on the I2C interface.
STOP_DET	9	0	R	Indicates whether a stop condition has occurred on the I2C interface.
ACTIVITY	8	0	R	Indicates whether the I2C block is idle. A logic 1 indicates the I2C module is processing data.
RX_DONE	7	0	R	When the I2C module is acting as a slave-transmitter this bit will be set to logic 1 if the master does not acknowledge a transmitted byte. This will occur on the last byte of the transmission, indicating that the transmission is done.

Name	Bit	Reset	Dir	Description
TX_ABRT	6	0	R	In general this bit will be set to logic 1 when the I2C module acting as a master is unable to complete a command that the processor has sent. The conditions which set TX_ABRT are: · no slave acknowledges after the address is sent · the addressed slave does not acknowledge a byte of data · arbitration is lost · attempting to send a master command when configured only to be a slave · IC_RESTART_EN bit in Register IC_CON is set to logic 0 (re-start condition disabled) and the processor attempts to issue an I2C function which is impossible to perform without using re-start conditions. · high speed master code is acknowledged · start byte is acknowledged · general call address is not acknowledged (impossible condition because slave module is always active and always acknowledges general call) · when a RD_REQ occurs and the processor has previously placed data in the TX buffer that has not been transmitted yet. This data could have been intended to service a multi-byte RD_REQ which ended up having fewer numbers of bytes requested. Or, if IC_RESTART_EN is disabled and the I2C module loses control of the bus between transfers and is then accessed as a slave-transmitter. · if a read command is issued after a general call command has been issued. Disabling the I2C module reverts it back to normal operation. · if the processor attempts to issue read command before a RD_REQ is serviced Anytime this bit is set the contents of the transmit buffer will be flushed.
RE_REQ	5	0	R	This bit will be set to logic 1 when the I2C module is acting as slave and another I2C master is attempting to read data from our module. The I2C module will hold the I2C bus in waiting until this interrupt is serviced. The processor must acknowledge this interrupt and then write the requested data to the Register IC_DATA.
TX_EMPTY	4	0	R	This bit will be set to logic 1 when the transmit buffer is at or below the threshold value set in Register IC_TX_TL. Automatically cleared by hardware when buffer level goes above the threshold.
TX_OVER	3	0	R	Set during transmit if the transmit buffer is filled to 8 and the processor attempts to issue another I2C command by writing to the Register IC_DATA_CMD.

Name	Bit	Reset	Dir	Description
RX_FULL	2	0	R	Set when the transmit buffer reaches or goes above the RX_TL threshold in Register IC_RX_TL. Automatically cleared by hardware when buffer level goes below the threshold.
RX_OVER	1	0	R	Set if the receive buffer was completely filled to 8 and more data arrived. That data will be lost.
RX_UNDER	0	0	R	Set if the processor attempts to read the receive buffer when it is empty by reading from Register IC_DATA_CMD.

4.4.22 IC_INTR_MASK register – I2C Interrupt Mask

Address – 0xC400 0030

These bits mask their corresponding interrupt status bits. They are active high, a value of logic 0 prevents a bit from generating an interrupt.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				M_GEN _CALL	M_STA RT_DE T	M_STO P_DET	M_ACT IVITY	M_RX_ DONE	M_TX_ ABRT	M_RE_ REQ	M_TX_ EMPTY	M_TX_ OVER	M_RX_ FULL	M_RX_ OVER	M_RX_ UNDER
Reset:	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
M_GEN_CALL	11	1	RW	Masks this bit in the register IC_INTR_STAT
M_START_DET	10	0	RW	Masks this bit in the register IC_INTR_STAT
M_STOP_DET	9	0	RW	Masks this bit in the register IC_INTR_STAT
M_ACTIVITY	8	0	RW	Masks this bit in the register IC_INTR_STAT
M_RX_DONE	7	1	RW	Masks this bit in the register IC_INTR_STAT
M_TX_ABRT	6	1	RW	Masks this bit in the register IC_INTR_STAT
M_RE_REQ	5	1	RW	Masks this bit in the register IC_INTR_STAT
M_TX_EMPTY	4	1	RW	Masks this bit in the register IC_INTR_STAT
M_TX_OVER	3	1	RW	Masks this bit in the register IC_INTR_STAT
M_RX_FULL	2	1	RW	Masks this bit in the register IC_INTR_STAT
M_RX_OVER	1	1	RW	Masks this bit in the register IC_INTR_STAT
M_RX_UNDER	0	1	RW	Masks this bit in the register IC_INTR_STAT

4.4.23 IC_RAW_INTR_STAT register – I2C Raw Status

Address – 0xC400 0034

Unlike the Register IC_INTR_STAT register, these bits are not masked so they always show the true status of the I2C module.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				R_GEN_CALL	R_START_DET	R_STOP_DET	R_ACTIVITY	R_RX_DONE	R_TX_ABRT	R_RE_REQ	R_TX_EMPTY	R_TX_OVER	R_RX_FULL	R_RX_OVER	R_RX_UNDER
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
R_GEN_CALL	11	0	R	Indicates that a general call request was received. The I2C module will store the received data in the RX buffer.
R_START_DET	10	0	R	Indicates whether a start condition has occurred on the I2C interface
R_STOP_DET	9	0	R	Indicates whether a stop condition has occurred on the I2C interface
R_ACTIVITY	8	0	R	The ACTIVITY bit indicates whether the I2C block is idle. A logic 1 indicates the I2C module is processing data.
R_RX_DONE	7	0	R	When the I2C module is acting as a slave-transmitter this bit will be set to logic 1 if the master does not acknowledge a transmitted byte. This will occur on the last byte of the transmission, indicating that the transmission is done.

Name	Bit	Reset	Dir	Description
R_TX_ABRT	6	0	R	In general this bit will be set to logic 1 when the I2C module acting as a master is unable to complete a command that the processor has sent. The conditions which set TX_ABRT are: · no slave acknowledges after the address is sent · the addressed slave does not acknowledge a byte of data · arbitration is lost · attempting to send a master command when configured only to be a slave · IC_RESTART_EN bit in Register IC_CON is set to logic 0 (re-start condition disabled) and the processor attempts to issue an I2C function which is impossible to perform without using re-start conditions. · high speed master code is acknowledge · start byte is acknowledged · general call address is not acknowledged (impossible condition because I2C Interface slave module is always active and always acknowledges general call) · when a RD_REQ occurs and the processor has previously placed data in the TX buffer that has not been transmitted yet. This data could have been intended to service a multi-byte RD_REQ which ended up having fewer numbers of bytes requested. Or, if IC_RESTART_EN is disabled and the I2C module loses control of the bus between transfers and is then accessed as a slave-transmitter. · if a read command is issued after a general call command has been issued. Disabling the I2C module reverts it back to normal operation. · if the processor attempts to issue read command before a RD_REQ is serviced Anytime this bit is set the contents of the transmit buffer will be flushed.
R_RE_REQ	5	0	R	This bit will be set to logic 1 when the I2C module is acting as slave and another I2C master is attempting to read data from our module. The I2C module will hold the I2C bus in waiting until this interrupt is serviced. The processor must acknowledge this interrupt and then write the requested data to the Register IC_DATA.
R_TX_EMPTY	4	0	R	This bit will be set to logic 1 when the transmit buffer is at or below the threshold value set in Register IC_TX_TL. Automatically cleared by hardware when buffer level goes above the threshold.
R_TX_OVER	3	0	R	Set during transmit if the transmit buffer is filled to 8 and the processor attempts to issue another I2C command by writing to the Register IC_DATA_CMD.

Name	Bit	Reset	Dir	Description
R_RX_FULL	2	0	R	Set when the transmit buffer reaches or goes above the RX_TL threshold in Register IC_RX_TL. Automatically cleared by hardware when buffer level goes below the threshold.
R_RX_OVER	1	0	R	Set if the receive buffer was completely filled to 8 and more data arrived. That data will be lost.
R_RX_UNDER	0	0	R	Set if the processor attempts to read the receive buffer when it is empty by reading from Register IC_DATA_CMD.

4.4.24 IC_RX_TL register – I2C RX Threshold Level

Address – 0xC400 0038

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								RX_TL							
Reset:	0								1							
	R								RW							

Name	Bit	Reset	Dir	Description
RX_TL	7:0	1	RW	Receive Buffer Threshold Level. Controls the level of entries (or above) that will trigger the RX_FULL interrupt. The valid range is 0-255 with the additional restriction that it may not be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set with the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry and a value of 255 sets the threshold for 256 entries.

4.4.25 IC_TX_TL register – I2C TX Threshold Level

Address – 0xC400 003C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								TX_TL							
Reset:	0	0	0	0	0	0	0	0	0							
	R	R	R	R	R	R	R	R	RW							

Name	Bit	Reset	Dir	Description
TX_TL	7:0	0	RW	Transmit Buffer Threshold Level. Controls the level of entries (or below) that will trigger the TX_EMPTY interrupt. The valid range is 0-255 with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set with the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries and a value of 255 sets the threshold for 255 entries.

4.4.26 IC_CLR_INTR register – Clear All Interrupts

Address – 0xC400 0040

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR _INTR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_INTR	0	0	R	Read this register to clear the combined interrupt and all individual interrupts.

4.4.27 IC_CLR_ RX_UNDER register – Clear RX_UNDER Interrupt

Address – 0xC400 0044

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_RX_UNDER
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_RX_UNDER	0	0	R	Read this register to clear the RX_UNDER interrupt

4.4.28 IC_CLR_RX_OVER register – Clear RX_OVER Interrupt

Address – 0xC400 0048

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_RX_OVER
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_RX_OVER	0	0	R	Read this register to clear the RX_OVER interrupt

4.4.29 IC_CLR_TX_OVER register – Clear TX_OVER Interrupt

Address – 0xC400 004C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_TX_OVER
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_TX_OVER	0	0	R	Read this register to clear the TX_OVER interrupt

4.4.30 IC_CLR_RD_REQ register – Clear RD_REQ Interrupt

Address – 0xC400 0050

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_RD_REQ
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_RD_REQ	0	0	R	Read this register to clear the RD_REQ interrupt

4.4.31 IC_CLR_TX_ABRT register – Clear TX_ABRT Interrupt

Address – 0xC400 0054

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR _TX_A BRT
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_TX_ABRT	0	0	R	Read this register to clear the TX_ABRT interrupt

4.4.32 IC_CLR_RX_DONE register – Clear RX_DONE Interrupt

Address – 0xC400 0058

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR _TX_D ONE
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_RX_DONE	0	0	R	Read this register to clear the RX_DONE interrupt

4.4.33 IC_CLR_ACTIVITY register – Clear ACTIVITY Interrupts

Address – 0xC400 005C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_ACTIVITY
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_ACTIVITY	0	0	R	Read this register to clear the ACTIVITY interrupt

4.4.34 IC_CLR_STOP_DET register – Clear STOP_DET Interrupts

Address – 0xC400 0060

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_STOP_DET
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_STOP_DET	0	0	R	Read this register to clear the STOP_DET interrupt

4.4.35 IC_CLR_START_DET register – Clear START_DET Interrupt

Address – 0xC400 0064

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_START_DET
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_START_DET	0	0	R	Read this register to clear the START_DET interrupt

4.4.36 IC_CLR_GEN_CALL register – Clear General Call Interrupt

Address – 0xC400 0068

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_CLR_GEN_CALL
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_CLR_GEN_CALL	0	0	R	Read this register to clear the GEN_CALL interrupt

4.4.37 IC_ENABLE register – I2C Enable

Address – 0xC400 006C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															IC_ENABLE
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_ENABLE	0	0	RW	Controls whether the I2C module is enabled. Writing a logic 1 enables the I2C and writing a logic 0 disables it. Software should not disable the I2C module while it is active. The ACTIVITY bit can be polled to determine if the I2C module is active. If the module was transmitting it will stop as well as delete the contents of the transmit buffer after the current transfer is complete. If the module was receiving the module will stop the current transfer at the end of the current byte and not acknowledge the transfer..

4.4.38 IC_STATUS register – I2C Status

Address – 0xC400 0070

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											RFF	RFNE	TFE	TFNF	ACTIVITY
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
RFF	4	0	R	**Receive FIFO Completely Full**. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full

Name	Bit	Reset	Dir	Description
RFNE	3	0	R	**Receive FIFO Not Empty**. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty
TFE	2	1	R	**Transmit FIFO Completely Empty**. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
TFNF	1	1	R	**Transmit FIFO Not Full**. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
ACTIVITY	0	0	R	**I2C Activity Status**.

4.4.39 IC_TXFLR register – I2C Transmit FIFO Level Register

Address – 0xC400 0074

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared when the I2C is disabled, whenever there is a transmit abort, or whenever the Slave Bulk Transfer mode is aborted. It increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															TXFLR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
TXFLR	0	0	R	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.

4.4.40 IC_RXFLR register – I2C Receive FIFO Level Register

Address – 0xC400 0078

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared when the I2C is disabled or whenever there is a transmit abort. It increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															TXFLR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
TXFLR	0	0	R	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.

4.4.41 IC_SRESET register – I2C Soft Reset Register

Address – 0xC400 007c

This register is used to issue a soft reset to the master and/or the slave state machines. Reading this register does not clear it; but is automatically cleared by hardware.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													IC_SLAVE_SRST	IC_MASTER_SRST	IC_SRST
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
IC_SLAVE_SRST	2	0	W	Issues a soft reset to the slave state machines. 1 = perform the reset
IC_MASTER_SRST	1	0	W	Issues a soft reset to the master state machines. 1 = perform the reset

Name	Bit	Reset	Dir	Description
IC_SRST	0	0	W	Issues a soft reset to both the master and slave state machines. 1 = perform the reset

4.4.42 IC_ABRT_SOURCE register – I2C Transmit Abort Source Register

Address – 0xC400 0080

This register has 16 bits that indicate the source of the tx_abrt signal, This register is cleared whenever the processor reads it or when the processor issues a clear signal to all interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														IC_SLAVE_SRST	IC_MASTER_SRST	IC_SRST
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
ABRT_SLVRD_INTX	15	0	RW	1 = Slave requesting data to TX and the user wrote a read command into the tx_fifo (9th bit is a 1).
ABRT_SLV_	14	0	RW	1 = Slave lost the bus while it is transmitting data to a remote master. IC_TX_ABRT[12] will be set at the same time.
ARB_MASTER_DIS	13	0	RW	1 = Slave has received a read command and some data exists in the tx_fifo so the slave issues a TX_ABRT to flush old data in tx_fifo.
ABRT_10B_RD_NORSTRT	12	0	RW	1 = Master has lost arbitration, or if TX_ABRT_SRC[12] is also set, then the slave transmitter has lost arbitration.
ABRT_SBYTE_NORSTRT	11	0	RW	1 = User attempted to use disabled Master.
ABRT_HS_NORSTRT	10	0	RW	1 = The restart is disabled (IC_RESTART_EN bit (ic_con[5]) = 0) and the Master sends a read command in 10-bit addressing mode.
ABRT_SBYTE_ACKDET	9	0	RW	1 = The restart is disabled (IC_RESTART_EN bit (ic_con[5]) = 0) and the user is trying to send a Start Byte.
ABRT_HS_ACKDET	8	0	RW	1 = The restart is disabled (IC_RESTART_EN bit (ic_con[5]) = 0) and the user is trying to use the master to send data in High Speed mode.

Name	Bit	Reset	Dir	Description
ABRT_SBYTE_ACKDET	7	0	RW	1 = Master has sent a Start Byte and the Start Byte was acknowledged (wrong behavior).
ABRT_HS_ACKDET	6	0	RW	1 = Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).
ABRT_GCALL_READ	5	0	RW	1 = Master sent a general call but the user programmed the byte following the G.CALL to be a read from the bus (9th bit is set to 1).
ABRT_GCALL_NOACK	4	0	RW	1 = Master sent a general call and no slave on the bus responded with an ack.
ABRT_TXDATA_NOACK	3	0	RW	1 = Master has received an acknowledgement for the address, but didn't receive an acknowledgement for data.
ABRT_10ADDR2_NOACK	2	0	RW	1 = Master is in 10-bit address mode and the 2nd address byte of the 10-bit address was not acknowledged by any slave.
ABRT_10ADDR1_NOACK	1	0	RW	1 = Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.
ABRT_7B_ADDR_NOACK	0	0	RW	1 = Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.

4.5 UART

4.5.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
UART0_TX	130	115	O	4	Serial output; active-high
UART0_RX	131	116	I	-	Serial input; active-high (5V)
UART1_TX	134	119	O	4	Serial output; active-high
UART1_RX	135	120	I	-	Serial input; active-high (5V)

Table 23: UART Signal Description

4.5.2 Features

The UART in the DICE JR is implemented in compliance with industry standard type 16550. The UART uses an internal baud generator clocked by APB clock 'pclk' (connected to ARM system clock – typically 49.152 MHz). 32bit data access is required for the APB bus interface.

The TCD2210 and TCD2220 does not support the UART handshake pins, only the serial communication pins. In applications where the handshake pins are required GPIO signals can be used instead.

4.5.3 Internal Functional Description

This section describes each of the functional blocks that make up the UART. A diagram showing the connections between these functional blocks is given in Figure 2.

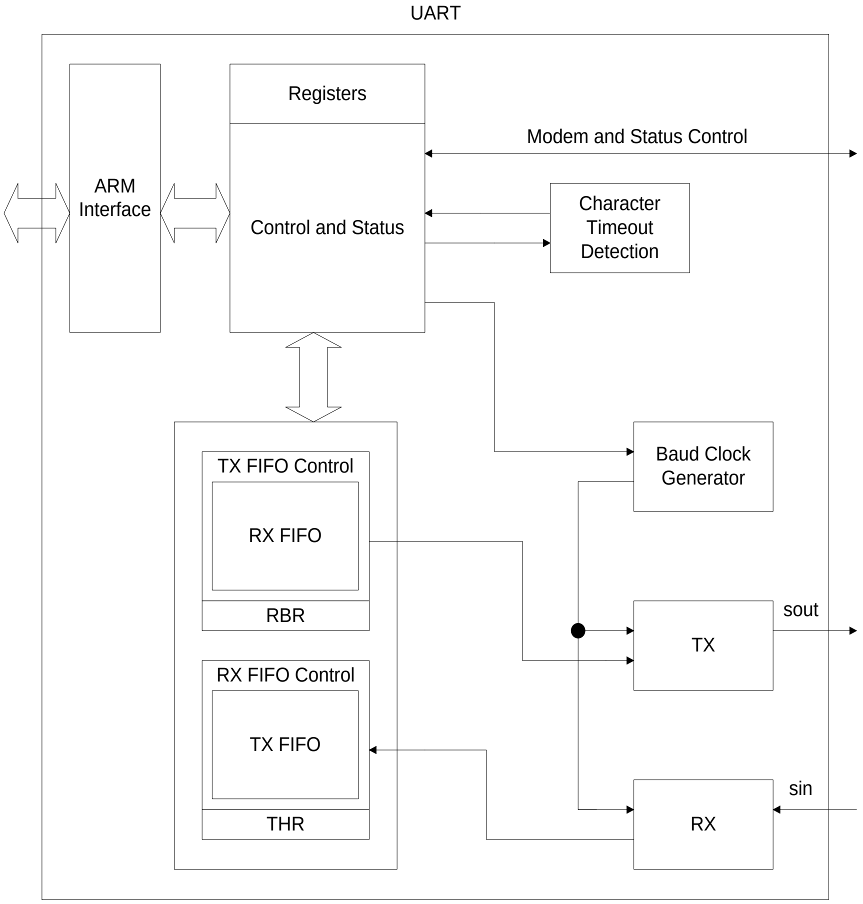


Figure 24: Generalized Functional Diagram

4.5.4 Registers, Control and Status

Primary control and status registers exist in this module, as well as the main UART functionality. Control registers are stored here and are used for serial data control and status generation. This module is also responsible for interrupt generation based on transmitter and receiver status, as well as which interrupts are enabled.

4.5.5 RX and TX FIFO Controllers

These FIFO controllers implement specially designed logic to access data elements before they reach the top of the FIFO. This guarantees correct status and data at all times.

4.5.6 Character Timeout Detection

This module sets and clears a timeout counter. This counter is then used to generate Character Timeout Interrupts when enabled.

4.5.7 Baud Clock Generator

This module uses the values in the Divisor Latch registers to generate the divide by 16 Baud Clock.

4.5.8 TX (transmitter)

The transmitter converts parallel data that has been programmed into the Transmitter Holding Register into a serial data stream. This serial data stream is built according to conditions specified in the Line Control Register. The serial data then exits the design on the Sout port. The parallel data will be sourced from the TX FIFO.

4.5.9 RX (receiver)

The receiver converts serial data that has been sent to the uart on the sin port and converts it to a parallel data character, based on Line Control Register settings. Once a complete character is received, it is then sent to the RX FIFO.

4.5.10 Serial Frame Format

The Line Control Register allows a number of different options with regards to frame format, all within the UART standard 16550. Four different character lengths are supported (5, 6, 7, 8 data bits). There is also an option for 1 or 2 stop bits, and an option for 0 or 1 parity bits. Figure 3 shows the frame formats supported.

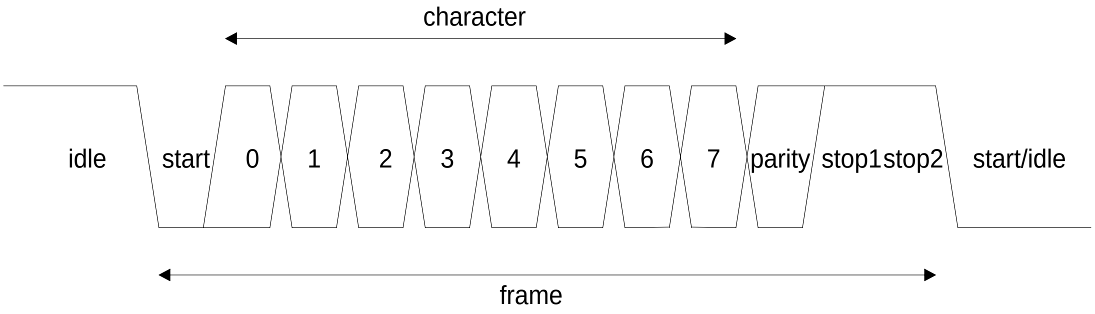


Figure 25: Serial Frame Format (using 8 data bit configuration)

4.5.11 Module Configuration

An address map for the programmable registers is given the table below. Note that several of the registers are accessed with the same address. In these cases control signals such as DLAB and pwrite determine which register is accessed at that time.

Address	Register
0xbd00 0000	UART#1 RBR, THR, DLL[a]
0xbd00 0004	UART#1 IER, DLH[a]
0xbd00 0008	UART#1 IIR, FCR
0xbd00 000c	UART#1 LCR
0xbd00 0010	UART#1 MCR
0xbd00 0014	UART#1 LSR
0xbd00 0018	UART#1 MSR
0xbd00 001c	UART#1 SCR
0xbe00 0000	UART#0 RBR, THR, DLL
0xbe00 0004	UART#0 IER, DLH
0xbe00 0008	UART#0 IIR, FCR
0xbe00 000c	UART#0 LCR
0xbe00 0010	UART#0 MCR
0xbe00 0014	UART#0 LSR
0xbe00 0018	UART#0 MSR
0xbe00 001c	UART#0 SCR

Table 24: UART Memory Map

[a] Bit 7 of the Line Control Register (LCR) enables reading and writing of the Divisor Latch Registers (DLL, DLH).

The UART contains 12 registers that are programmable via the 5-bit APB address bus. Note that these are actually 8-bit registers. They have 32-bit data boundaries to simplify access to the APB. When reading from the APB the upper 24 bits are ignored, whereas when writing to the APB the 8 bit registers are padded with 24 zeros automatically.

4.5.12 Receive Buffer Register (RBR)

Address – 0xBD00 0000

The RBR is a read-only register that contains the data byte received on the serial input port (sin). The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. This register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								RBR							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Receive Buffer Register (RBR)	7:0	0	R	Contains data character from serial input port.

4.5.13 Transmit Holding Register (THR)

Address – 0xBD00 0000

The THR is a write-only register that contains data to be transmitted on the serial output port (sout). Data can be written to the THR any time that the THR Empty (THRE) bit of the Line Status Register (LSR) is set. If THRE is set, 16 characters of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								THR							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Name	Bit	Reset	Dir	Description
Transmit Holding Register (THR)	7:0	0	W	Contains data byte for serial transmission.

4.5.14 Divisor Latch Register (DLL)

Address – 0xBD00 0000

The DLL register in conjunction with the DLH register forms a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. It is accessed by first setting the DLAB bit (bit 7) in the Line Control Register (LCR). The output baud rate is equal to the APB clock frequency (pclk) divided by sixteen times the value of the baud rate divisor as follows (see section 3.3 for details):

baud rate = (APB clock freq) / (16 * divisor)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DLH/DLL							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Divisor Latch High (High Byte)	7:0	0	RW	High byte Divisor Latch Register
Divisor Latch Low (Low Byte)	7:0	0	RW	Low byte Divisor Latch Register

4.5.15 Interrupt Enable Register (IER)

Address – 0xBD00 0004

The IER is a read/write register that contains four bits that enable the generation of interrupts. Note that the IER enables inputs, whereas the IIR actually registers those interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												EDSSI	ELSI	ETBEI	ERBFI
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
EDSSI	3	0	RW	Enable the Modem Status Interrupt
ELSI	2	0	RW	Enable the Receiver Line Status Interrupt
ETBEI	1	0	RW	Enable the Transmitter Holding Register Empty Interrupt
ERBFI	0	0	RW	Enable the Received Data Available Interrupt

4.5.16 Interrupt Identity Register (IIR)

Address – 0xBD00 0008

The Interrupt Identity Register is a read-only register that identifies the source of an interrupt. The upper two bits of the register are FIFO-enabled bits. These bits will be "00" if the FIFOs are disabled, and "11" if they are enabled. The lower four bits identify the highest priority pending interrupt. A full description of the interrupt control functions is given in Table 2 below.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								FIFO enable bits		Reserved		Interrupt Identity bits.			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Fifo enabled bits	7:6	0	R	2'b00: FIFO's disabled, 2'b11: FIFO's enabled
Interrupt Identity bits	3:0	0	R	See Table 2 for details

IIR bits				Interrupt Set and Reset Functions			
3	2	1	0	Priority	Type	Source	Reset and Control
0	0	0	1	-	-	-	-
0	1	1	0	first	Receiver line status	Overrun/parity/ framing errors or break interrupt	Reading the line status register
0	1	0	0	second	Received data available	Receiver data available or read data FIFO trigger level reached	Reading the receiver buffer register or the FIFO drops below the trigger level
1	1	0	0	second	Character timeout indication	During the last four character times there were no characters in or out of receiver FIFO and at least one character in it already	Reading the receiver buffer register
0	0	1	0	third	Transmitter holding register empty	Transmitter holding register empty	Reading the IIR (if source of interrupt) or writing into THR
0	0	0	0	forth	Modem status	Clear to send or data set ready or ring indicator or data center detect	Reading the MSR

Table 25: Interrupt Control Functions

4.5.17 FIFO Control Register (FCR)

Address – 0xBD00 0008

The FIFO control register is a write-only register. It controls the read and write data FIFO. The FIFOs are reset anytime bit 0 of the FCR changes value. Only when the FIFOs are enabled (bit 0 of FCR is set to 1) are bits 3, 6 and 7 active.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Receiver Trigger		Reserved			TX FIFO Reset	Receiver FIFO Reset	FIFO Enable
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	W	W	R	R	R	W	W	W

Name	Bit	Reset	Dir	Description
Receiver Trigger (RT)	7:6	0	W	Sets the trigger level in the receiver FIFO for the Enable Received Data Available Interrupt (ERBFI) 00 = 1 character in FIFO, 01 = 4 characters in FIFO 10 = 8 characters in FIFO, 11 = 14 characters in FIFO
Transmitter FIFO Reset	2	0	W	Resets and flushes transmit FIFO (self-clearing)
Receiver FIFO Reset	1	0	W	Resets and flushes receive FIFO (self-clearing)
FIFO Enable	0	0	W	Allows operation of transmit and receive FIFOs

4.5.18 Line Control Register (LCR)

Address – 0xBD00 000C

The Line Control Register controls the format of the data that is serially transmitted and received by the UART.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DLAB	Break Control	Stick Parity	EPS	PEN	STOP bits	CLS	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
DLAB (Divisor Latch Address bit)	7			Enables reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.

Name	Bit	Reset	Dir	Description
Break Control	6			Sends break signal by holding the sout line low, until cleared. When in Loopback Mode, the break condition is internally looped back to the receiver
Stick Parity	5			Not used
EPS	4			Parity Select bit: 0=odd number of ones, 1=even number of ones
PEN	3			Enables the a parity bit in outgoing serial data
STOP bits	2			Number of stop bits transmitted: 0=1bit, 1=2bits. If there are only 5 bits per character then there will be 1.5 stop bits.
CLS	1:0			Number of bits per character: 00=5bits, 01=6bits, 10=7bits, 11=8bits

4.5.19 Modem Control Register (MCR)

Address – 0xBD00 0010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											Loop Back bit	OUT2	OUT1	RTS	DTR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Loop Back bit	4	0	RW	This feature is used for diagnostic purposes. When set, data on the sout line is held HIGH, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the four modem control outputs (dtr_n, rts_n, out1_n, out1_n) are looped back to the inputs, internally.
OUT2	3	0	RW	Bit is inverted and then used to drive the UART output out2_n
OUT1	2	0	RW	Bit is inverted and then used to drive the UART output out1_n
RTS	1	0	RW	Bit is inverted and then used to drive the UART output rts_n
DTR	0	0	RW	Bit is inverted and then used to drive the UART output dtr_n

4.5.20 Line Status Register (LSR)

Address – 0xBD00 0014

The Line Status Register contains status of the receiver and transmitter data transfers. This status can be read by the programmer at anytime.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								FERR	TEMT	THRE	BI	FE	PE	OE	DR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
FERR	7	0	R	Error in Receiver FIFO: Set when there is at least one parity error, framing error, or break indication in the FIFO. This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
TEMT	6	0	R	Transmitter Empty bit: Set whenever the Transmitter Shift Register and the FIFO are both empty.
THRE	5	0	R	Transmitter Holding Register Empty bit: Indicates the UART can accept a new character for transmission. This bit is set whenever data is transferred from the THR to the transmitter shift register and no new data has been written to the THR. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled.
BI	4	0	R	Break Interrupt bit: Set whenever the serial input (sin) is held in a logic ‘0’ state for longer than the sum of (start time+data bits+parity+stop bits). A break condition on sin causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit.
FE	3	0	R	Framing Error Bit: Set whenever there is a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. Since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. The OE, PE and FE bits are reset when a read of the LSR is performed.
PE	2	0	R	Parity Error Bit: Set whenever there is a parity error in the receiver if the Parity Enable (PEN) bit in the LCR is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO.
OE	1	0	R	Overrun bit: A new data character was received before the previous data was read. An overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost.
DR	0	0	R	Data Ready bit: Indicates the receiver contains at least one character in the RBR or the receiver FIFO. Bit cleared when the receiver FIFO is empty.

4.5.21 Modem Status Register (MSR)

Address – 0xBD00 0018

The Modem Status Register contains the current status of the modem control input lines and if they changed. DCTS (bit 0), DDSR (bit 1) and DDCD (bit 3) bits record whether the modem control lines (cts_n, dsr_n and dcd_n) have changed since the last time the CPU read the MSR. The CTS, DSR, RI and DCD Modem Status bits contain information on the current state of the modem control lines.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
DCD	7	0	R	Compliment of dcd_n. In Loopback Mode, DCD is the same as MCR bit 3 (Out2).

Name	Bit	Reset	Dir	Description
RI	6	0	R	Compliment of ri_n. In Loopback Mode, RI is the same as MCR bit 2 (Out1).
DSR	5	0	R	Compliment of dsr_n. In Loopback Mode, DSR is the same as MCR bit 0 (DTR).
CTS	4	0	R	Compliment of cts_n. In Loopback Mode, CTS is the same as MCR bit 1 (RTS).
DDCD	3	0	R	Record whether the modem control line dcd_n has changed since the last time the CPU read the MSR. In Loopback Mode DDCD reflects changes on MCR bit 3 (OUT2)
TERI	2	0	R	Indicates ri_n has changed from an active low, to an inactive high state since the last time the MSR was read. In loopback modeTERI reflects when MCR bit 2 (OUT1) has changed state from a high to a low.
DDSR	1	0	R	Record whether the modem control line dsr_n has changed since the last time the CPU read the MSR. In Loopback Mode DDSR reflects changes on MCR bit 0 (DTR)
DCTS	0	0	R	Record whether the modem control line cts_n has changed since the last time the CPU read the MSR. In Loopback Mode DCTS reflects changes on MCR bit 1 (RTS).

4.5.22 Scratchpad Register (SCR)

Address – 0xBD00 001C

The SCR register is an 8-bit read/write register for programmers to use as a temporary storage space. It has no defined purpose in this UART.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Temp Storage							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
	7:0	0	RW	Temporary storage for programmers

4.5.23 Serial Baud Rate

The serial baud rate of the UART can be user defined by entering a 2 byte divisor in the Divisor Latch Register (two 8-bit registers; one high and one low byte register). The divisor effectively divides the APB clock rate to give a baud rate that is 16 x divisor. Table 5 gives the divisor values that are required to specify several common serial baud rates, assuming the APB 'pclk' rate is 49.152 MHz.

Desired Baud Rate	Divisor	Obtained Rate	Deviation (%)
1200	2560	1200.00	0.00
2400	1280	2400.00	0.00

4800	640	4800.00	0.00
9600	320	9600.00	0.00
19200	160	19200.00	0.00
31250	98	31346.94	0.31
38400	80	38400.00	0.00
57600	53	57962.26	0.63
115200	27	113777.78	-1.23
1024000	3	1024000.00	0.00

Table 26: Determining Baud Rate

4.6 GPIO

The General Purpose I/O (GPIO) module has a 32 bit interface to the APB data bus. It consists of one port with a data width of 16 bits. The default direction of the GPIO is input. The GPIO module includes logic to support the debouncing of glitches. It also includes logic to support interrupt detection. The active level or edge for interrupt detection is active high. The GPIO also includes metastability registers to synchronize read back data.

The GPIO pins share functionality. Consult the GPCSR section for further details.

On TCD2220 GPIO0-14 are available, on TCD2210 GPIO1-8 are available.

4.6.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
GPIO0	42 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO1	138 (shared)	123 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO2	139 (shared)	124 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO3	137 (shared)	122 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO4	85 (shared)	78 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO5	86 (shared)	79 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO6	117 (shared)	106 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO7	118 (shared)	107 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO8	119 (shared)	108 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO9	55	N/A	I/O (S)	8	General Purpose I/O (5V)
GPIO10	56	N/A	I/O (S)	8	General Purpose I/O (5V)
GPIO11	57 (shared)	N/A	I/O (S)	8	General Purpose I/O (5V)
GPIO12	65 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO13	66 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)
GPIO14	67 (shared)	N/A	I/O (S)	6	General Purpose I/O (5V)

Table 27: GPIO Signal Description

Note that all pins used by the GPIO module are multi-purpose or shared. The function of these pins is software configurable via the GPCSR module, specifically register GPCSR_IO_SELECT0 – 0xc700 0004. Refer to the GPCSR module documentation for more information.

Note that each GPIO signal can be configured as an input or output using GPIO_DDR, the GPIO Data Direction Register at address 0xc300 0004. See section 3.

4.6.2 Module Configuration

Address	Register
0xc300 0000	GPIO_DR
0xc300 0004	GPIO_DDR
0xc300 0030	GPIO_INTEN
0xc300 0034	GPIO_INTMSK
0xc300 0038	GPIO_INTSENSE
0xc300 003c	GPIO_INTPOL
0xc300 0040	GPIO_INTSTAT
0xc300 0044	GPIO_RAWINTSTAT
0xc300 0048	GPIO_DEBOUNCE
0xc300 004c	GPIO_EOI
0xc300 0050	GPIO_EXT
0x3c00 0060	GPIO_SYNC

Table 28: GPIO Memory Map

Note that all programmable registers are actually 32 bits wide. However, the upper 17 bits of all the registers are Reserved. Therefore, only the lower 15 bits of each register is shown in this document.

4.6.3 GPIO_DR Data Register

Address – 0Xc300 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Data															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Data	15:0		RW	Values written to this register are output on the I/O pins. If the corresponding data direction bits are set to “output” mode. The value read back is equal to the last value written to this register.

4.6.4 GPIO_DDR Data Direction Register

Address – 0Xc300 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Data Direction															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Data Direction	15:0		RW	Values written to this register independently control the direction of the corresponding data bit.

4.6.5 GPIO_INTEN Interrupt Enable Register

Address – 0Xc300 0030

This register is available only if GPIO port is configured to generate interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Interrupt Enable															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Interrupt Enable	15:0		RW	Allows each bit to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit to become an interrupt. Otherwise, the GPIO operates as a normal GPIO port. Interrupts are disabled on the corresponding bits if the corresponding data direction register is set to “output”. 0: configure bit as normal GPIO port (default) 1: configure bit as interrupt

4.6.6 GPIO_INTMASK Interrupt Mask Register

Address – 0Xc300 0034

This register is available only if GPIO port is configured to generate interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Interrupt Mask															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Interrupt Mask	15:0		RW	Controls whether an interrupt can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for the whole port, otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking. 0: interrupt bits are unmasked (default) 1: mask interrupt

4.6.7 GPIO_INTSENSE Interrupt Level Register

Address – 0Xc300 0038

This register is available only if GPIO port is configured to generate interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Interrupt Level															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Interrupt Level	15:0		RW	Controls the type of interrupt that can occur. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive. 0: level-sensitive (default) 1: edge-sensitive

4.6.8 GPIO_INTPOL Interrupt Polarity Register

Address – 0Xc300 003c

This register is available only if GPIO port is configured to generate interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Interrupt Polarity															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Interrupt Polarity	15:0		RW	Controls the polarity of edge or level sensitivity that can occur on input. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 0: active-low (default) 1: active-high

4.6.9 GPIO_INTSTAT Interrupt Status Register

Address – 0Xc300 0040

This register is available only if GPIO port is configured to generate interrupts.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Interrupt Status															
Reset:																
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Interrupt Status	15:0		R	Interrupt status of each bit

4.6.10 GPIO_RAWINTSTAT Raw Interrupt Status (Premasking) Register

Address – 0Xc300 0044

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Raw Interrupt Status															
Reset:																
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Raw Interrupt Status	15:0		R	Raw interrupt status or “premasking” of each bit.

4.6.11 GPIO_DEBOUNCE Debounce Enable Register

Address – 0Xc300 0048

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Debounce Enable															
Reset:																
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Debounce Enable	15:0		RW	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0: no debounce (default) 1: enable debounce

4.6.12 GPIO_EOI Clear Interrupt Register

Address – 0Xc300 004c

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Clear Interrupt															
Reset:																
	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Name	Bit	Reset	Dir	Description

Name	Bit	Reset	Dir	Description
Clear Interrupt	15:0		W	Controls the clearing of edge type interrupts. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when the port is not configured for interrupts. 0: no interrupt clear (default) 1: clear interrupt

4.6.13 GPIO_EXT External Port Register

Address – 0Xc300 0050

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	External Port															
Reset:																
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
External Port	15:0		R	When the port is configured as "input", then reading this location reads the values on the port. When the data direction of the port is set as "output", reading this location reads the data register for the port.

4.6.14 GPIO_SYNC Level Sensitive Synchronization Enable Register

Address – 0Xc300 0060

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Sync Level
Reset:																

Name	Bit	Reset	Dir	Description
Synchronization Level	0	0	RW	Writing a 1 to this register results in all level sensitive interrupts being synchronized to pclk_intr. 0: no synchronization to pclk_intr (default) 1: synchronize to pclk_intr

4.7 1394 Link

4.7.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
SCLK	54	53	I (S)	-	49.152MHz PHY Clock
PHD0	58	54	I/O (S)	8	PHY tristatable data line bit 0
PHD1	59	55	I/O (S)	8	PHY tristatable data line bit 1
PHD2	60	56	I/O (S)	8	PHY tristatable data line bit 2
PHD3	63	59	I/O (S)	8	PHY tristatable data line bit 3
PHD4	64	60	I/O (S)	8	PHY tristatable data line bit 4
PHD5	68	61	I/O (S)	8	PHY tristatable data line bit 5
PHD6	69	62	I/O (S)	8	PHY tristatable data line bit 6
PHD7	70	63	I/O (S)	8	PHY tristatable data line bit 7
PHCT0	71	64	I/O (S)	8	PHY tristatable control line bit 0
PHCT1	72	65	I/O (S)	8	PHY tristatable control line bit 1
PHDI	73	66	I (S)	-	A high indicates isolation barrier is not present (PU, 5V)
PHLR	74	67	O	8	Serial request output from S-LINK (Z)
PHLP	75	68	O	4	Link power status. Pulsing if isol. barrier present
PHLO	76	69	I (S)	-	Link on indication from PHY. Pulsing when asserted (PU, 5V)

Table 29: Signal Description

4.7.2 Module Configuration

Address	Register
0x8200 0000	VERSION_REG_DP
0x8200 0004	ND_ID_REG_DP
0x8200 0008	LNK_CTRL_REG_DP
0x8200 000c	LCSR_REG_DP
0x8200 0010	CY_TMR_REG_DP
0x8200 0014	ATFIFO_STAT_REG_DP
0x8200 0018	ITFIFO_STAT_REG_DP
0x8200 001c	ARFIFO_STAT_REG_DP
0x8200 0020	IRFIFO_STAT_REG_DP
0x8200 0024	ISOC_RX_ENB_REG_1_DP
0x8200 0028	ISOC_RX_ENB_REG_2_DP
0x8200 002c	ISO_TX_STAT_REG_DP

Address	Register
0x8200 0030	ASY_TX_STAT_REG_DP
0x8200 0044	PHY_CTRL_REG_DP
0x8200 0048	INTERRUPT_REG_SET_DP
0x8200 004c	INTERRUPT_REG_CLEAR_DP
0x8200 0050	INTR_MASK_REG_SET_DP
0x8200 0054	INTR_MASK_REG_CLEAR_DP
0x8200 0058	DIAG_REG_DP
0x8200 005c	BUS_STAT_REG_DP
0x8200 0060	ASY_TX_FIFO_SPACE_REG_DP
0x8200 0064	ASY_RX_FIFO_QLETS_REG_DP
0x8200 0068	ISO_TX_FIFO_SPACE_REG_DP
0x8200 006c	ISO_RX_FIFO_QLETS_REG_DP
0x8200 0070	ISO_DATA_PATH_REG_DP
0x8200 0074	ASY_TX_FIRST_REG_DP
0x8200 0078	ASY_CONTINUE_REG_DP
0x8200 007c	ASY_CONTINUE_UPDATE_REG_DP
0x8200 0080	ASY_TX_FIFO_DEPTH_REG_DP
0x8200 0084	ASY_RX_FIFO_REG_DP
0x8200 0088	ASY_RX_FIFO_DEPTH_REG_DP
0x8200 008c	ISO_TX_FIRST_REG_DP
0x8200 0090	ISO_CONTINUE_REG_DP
0x8200 0094	ISO_CONTINUE_UPDATE_REG_DP
0x8200 0098	ISO_TX_FIFO_DEPTH_REG_DP
0x8200 009c	ISO_RX_FIFO_REG_DP
0x8200 00a0	ISO_RX_FIFO_DEPTH_REG_DP
0x8200 00a4	HST_ACC_ERR_REG_DP
0x8200 00a8	RET_CT_REG_DP
0x8200 00ac	DIG_FSM_STAT_REG
0x8200 00b0	ISO_TX_ENB_REG_1_DP
0x8200 00b4	ISO_TX_ENB_REG_2_DP
0x8200 00b8	ISO_HDR_REG_DP
0x8200 00bc	LPS_REG_DP
0x8200 00c0	PING_REG_DP
0x8200 00c4	ISOC_EXPC_CHAN_REG1
0x8200 00c8	ISOC_EXPC_CHAN_REG2
0x8200 00cc	DUP_EXPC_STAT_REG
0x8200 00d0	ASYN_RX_ENB_REG_1_DP
0x8200 00d4	ASYN_RX_ENB_REG_2_DP

Table 30: 1394 LLC Memory Map

4.8 GRAY, Rotary Encoder Interface

This module can decode the input from 2 rotary encoders. Each interface consists of 2 pins, A and B.

4.8.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
EN1_A	138(shared)	123(shared)	I (S)	6	Rotary Encoder Input (5V)
EN1_B	139(shared)	124(shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_A	65(shared)	N/A	I (S)	6	Rotary Encoder Input (5V)
EN2_B	66 (shared)	N/A	I (S)	6	Rotary Encoder Input (5V)

Table 31: GRAY Encoder Signal Description

4.8.2 Module Configuration

Address	Register
0xc600 0000	GRAY_STAT
0xc600 0004	GRAY_CTRL
0xc600 0008	GRAY_CNT

Table 32: GRAY Memory Map

4.8.3 GRAY_STAT

Address – 0Xc600 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														INT1	INT0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
reserved	31:2	0	R	Reads back as zero.
INT1	1	0	R	This bit indicates the changed status of counter 1. This bit will be set whenever the counter is changed and cleared when the GRAY_CNT register is read.
INT0	0	0	R	This bit indicates the changed status of counter 0. This bit will be set whenever the counter is changed and cleared when the GRAY_CNT register is read.

4.8.4 GRAY_CTRL

Address – 0Xc600 0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	Reserved														INTE1	INTE0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
reserved	31:2	0	R	This bit enables the interrupt on change for counter 2.
INTE1	1	0	RW	This bit enables the interrupt on change for counter 1.
INTE0	0	0	RW	This bit enables the interrupt on change for counter 0.

4.8.5 GRAY_CNT

Address – 0Xc600 0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT1								CNT0							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
reserved	31:16	0	R	Reads back as zeros
CNT1	15:8	0	R	Count indicating amount of change since last read. The value range is -128 to 127 with saturation logic.
CNT0	7:0	0	R	Count indicating amount of change since last read. The value range is -128 to 127 with saturation logic.

4.9 Interrupt Controller

4.9.1 Features

AIC supports the following features:

- 32 IRQ normal interrupt sources
- 8 FIQ fast interrupt sources
- Vectored interrupts
- Software interrupts
- Priority filtering
- Masking

4.9.2 Functional Description

AIC is a configurable, vectored interrupt controller. It supports 32 normal interrupts (IRQ) sources that are processed to produce a single IRQ interrupt to the processor. It supports 8 fast interrupts (FIQ) sources that are processed to produce a single FIQ interrupt to the processor.

AIC supports IRQ interrupts, software interrupts, priority filtering, and vector generation. FIQ interrupts are similar to IRQ interrupts with the exception that priority filtering and vector generation are not included. Figure 1 shows a block diagram of the AIC.

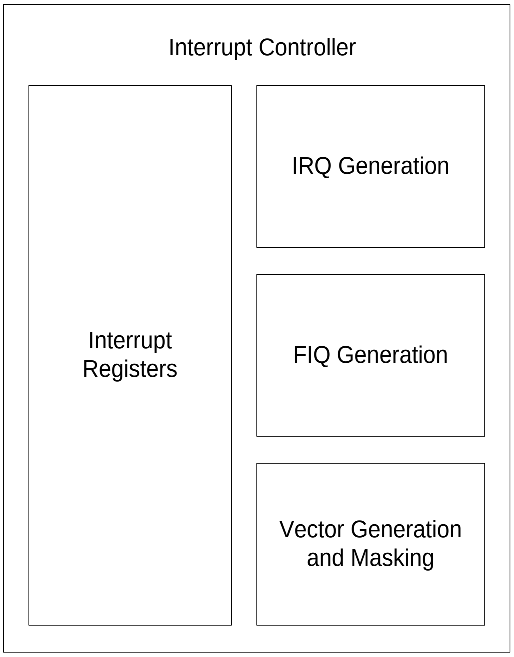


Figure 26: Block Diagram of AIC

4.9.3 IRQ Processing

The AIC processes 32 interrupt sources to produce a single IRQ interrupt to the processor. The processing of the interrupt sources is shown in Figure 2 and described in the following sections.

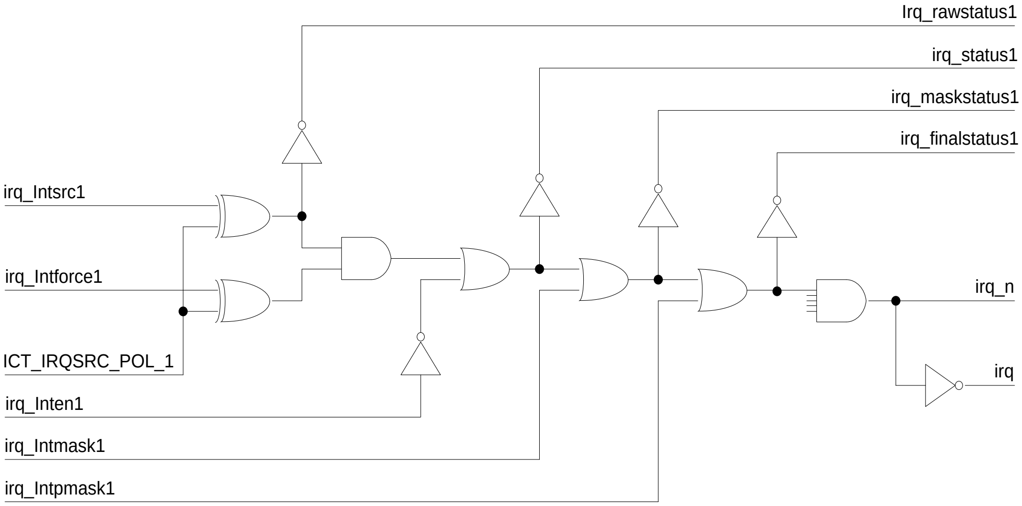


Figure 27: IRQ Internal Diagram (Interrupt 1)

4.9.4 IRQ Software Programmable Interrupts

The AIC supports forcing interrupts from software. To force an interrupt to be active, write to the corresponding bit in the INTCTRL_FORCE registers

4.9.5 IRQ Enable and Masking

To enable each interrupt source independently, write a 1 to the corresponding bit of the INTCTRL_ENABLE registers. Writing 1 in the INTCTRL_MASK register masks an interrupt.

4.9.6 IRQ Priority Filter

The AIC supports priority filtering. Each interrupt source has one of 16 priority levels (0 to 15), where 0 is the lowest priority. A system priority level can be programmed into the INTCTRL_SYSTEM_PRIORITY_LEVEL register, which holds values from 0 to 15. The reset value of this register is set to 0. AIC filters out any interrupt source with a configured priority level less than the priority currently programmed in this register.

4.9.7 IRQ Interrupt Status Registers

The AIC includes up to four status registers used for querying the current status of any interrupt at various stages of the processing. A 1 indicates that an interrupt is active; a 0 indicates it is inactive.

- INTCTRL_RAW

  This register contains the state of the interrupt sources. Each bit of this register is set to 1 if the corresponding interrupt source bit is active and is set to 0 if it is inactive

- INTCTRL_STATUS

  This register contains the state of all interrupts after the enabling stage, meaning that an active-high bit indicates that particular interrupt source is active and enabled.

- INTCTRL_MASKSTAT

  This register contains the state of all interrupts after the masking stage, meaning that an active-high bit indicates that particular interrupt source is active, enabled, and not masked.

- INTCTRL_FINALSTAT

  This register contains the state of all interrupts after the priority filtering stage, meaning an active-high bit indicates that particular interrupt source is active, enabled, not masked, and its configured priority level is greater or equal to the value programmed in the INTCRTL_SYSTEM_PRIORITY_LEVEL register.

4.9.8 IRQ Interrupt Vectors

The AIC supports interrupt vectors. The AIC has one vector register associated with each of the 16 interrupt priority levels: INTCTRL_VECTOR0 to INTCTRL_VECTOR15. These registers are 32 bits wide. The value of each interrupt vector register is hard-coded.

Vector processing proceeds as follows:

- Active interrupts are conditioned by their enable and mask control bits.

- All active interrupts with priority level less than the current value programmed into the INTCTRL_SYSTEM_PRIORITY_LEVEL register are filtered out.

- The highest priority level from among the remaining active interrupts is used to select one of the 16 interrupt vectors.

- The user retrieves the vector associated with the highest priority level that has an active interrupt source by reading the interrupt vector register The register is “read coherent,” you need to be guaranteed that you are reading a valid value for the entire vector. The contents of the register will be stored in a shadow location, when the user starts to read the register, so that the register can be read without being corrupted by it being changed by subsequent interrupts occurring.

4.9.9 FIQ Interrupt Processing

AIC supports 8 FIQ interrupt sources. AIC processes these interrupt sources to produce a single FIQ interrupt to the processor. FIQ interrupt processing is similar to IRQ interrupt processing except that priority filtering and interrupt vectors are not supported for the FIQ interrupts. This section describes how the AIC handles the FIQ interrupt processing. Figure 3 for further detail

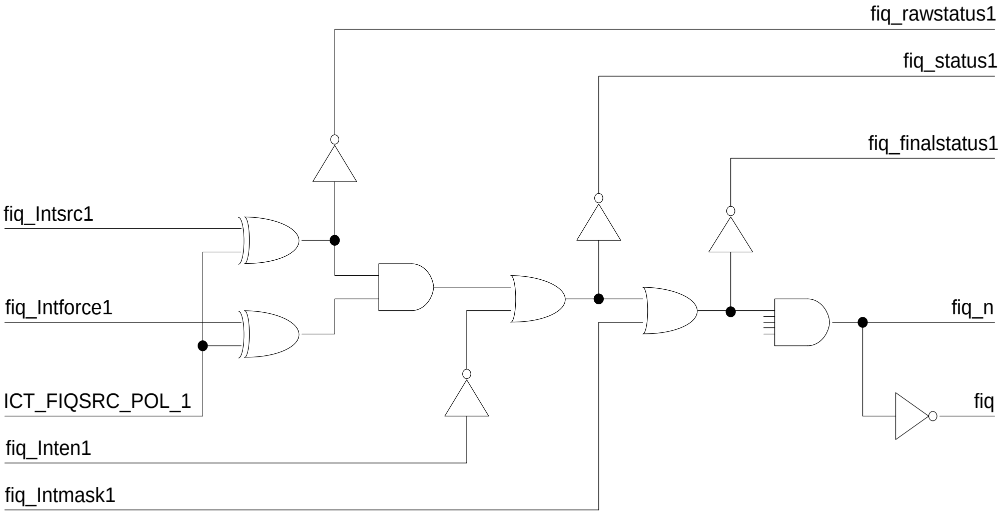


Figure 28: FIQ Internal Diagram (Interrupt 1)

4.9.10 FIQ Software-Programmable Interrupts

AIC supports forcing interrupts from software. You force an interrupt to be active by writing to the corresponding bit in the INTCTRL_FIQ_FORCE register

4.9.11 FIQ Enable and Masking

You can enable each interrupt source independently by writing a 1 to the corresponding bit of the INTCTRL_FIQ_FORCE register. At reset all interrupts are disabled. You can mask each interrupt source independently by writing a 1 to the corresponding bit of the INTCTRL_FIQ_MASK register. The reset value for each mask bit is 0(unmasked).

4.9.12 FIQ Interrupt Status Registers

AIC includes three status registers that you can use to query the current status of any FIQ interrupt at various stages of the processing. A 1 indicates that in interrupt is active, a 0 indicates inactive

- INTCTRL_FIQ_RAW

This register contains the state of the interrupt sources. Each bit of this register is set to 1 if the corresponding interrupt source bit is active and is set to 0 if it is inactive.

- INTCTRL_FIQ_STAT

This register contains the state of all interrupts after the enabling stage, meaning that an active-high bit indicates that particular interrupt source is active and enabled.

- INTCTRL_FIQ_FINALSTAT

This register contains the state of all interrupts after the masking, meaning that an active-high bit indicates that particular interrupt source is active, enabled, and unmasked.

4.9.13 Module Configuration

Address	Reset Value	Priority Level	Register
0xc100 0000	0x00		INTCTRL_ENABLE
0xc100 0008	0x00		INTCTRL_MASK
0xc100 0010	0x00		INTCTRL_FORCE
0xc100 0018	0x00		INTCTRL_RAW
0xc100 0020	0x00		INTCTRL_STAT
0xc100 0028	0x00		INTCTRL_MASKSTAT
0xc100 0030	0x00		INTCTRL_FINALSTAT
0xc100 0038	0x00		INTCTRL_INTVECTOR
0xc100 0040	0x00	15	INTCTRL_VECTOR0
0xc100 0048	0x01	14	INTCTRL_VECTOR1
0xc100 0050	0x02	13	INTCTRL_VECTOR2
0xc100 0058	0x03	12	INTCTRL_VECTOR3
0xc100 0060	0x04	11	INTCTRL_VECTOR4
0xc100 0068	0x05	10	INTCTRL_VECTOR5
0xc100 0070	0x06	9	INTCTRL_VECTOR6
0xc100 0078	0x07	8	INTCTRL_VECTOR7
0xc100 0080	0x08	7	INTCTRL_VECTOR8
0xc100 0088	0x09	6	INTCTRL_VECTOR9
0xc100 0090	0x0a	5	INTCTRL_VECTOR10
0xc100 0098	0x0b	4	INTCTRL_VECTOR11
0xc100 00a0	0x0c	3	INTCTRL_VECTOR12
0xc100 00a8	0x0d	2	INTCTRL_VECTOR13
0xc100 00b0	0x0e	1	INTCTRL_VECTOR14
0xc100 00b8	0x0f	0	INTCTRL_VECTOR15
0xc100 00c0	0x00		INTCTRL_FIQ_ENABLE
0xc100 00c4	0x00		INTCTRL_FIQ_MASK
0xc100 00c8	0x00		INTCTRL_FIQ_FORCE
0xc100 00cc	0x00		INTCTRL_FIQ_RAW
0xc100 00d0	0x00		INTCTRL_FIQ_STAT
0xc100 00d4	0x00		INTCTRL_FIQ_FINALSTAT
0xc100 00d8	0x00		INTCTRL_SYSTEM_PRIORITY_LEVEL

Table 33: ICTL Memory Map

4.9.14 INTCTRL_ENABLE

This is a Read/Write Register to enable/disable interrupts. Writing 1 in the corresponding bit enables interrupt and 0 disables it. At Reset all interrupts are disabled.

4.9.15 INTCTRL_MASK

This is a Read/Write Register to mask interrupts. A 0 indicates the corresponding interrupt is unmasked and 1 indicates that it’s masked. At Reset all interrupts are unmasked.

4.9.16 INTCTRL_FORCE

This is a Read/Write Register to force interrupts. Writing 1 to a bit location forces the interrupt to occur. At Reset this register is initialized to all zeros.

4.9.17 INTCTRL_RAW

This is a Read Only Register which shows the actual state of interrupt as generated by the corresponding device. A 1 indicates that an interrupt occurred. At Reset this register is initialized to all zeros.

4.9.18 INTCTRL_STAT

This is a Read Only Register. Only those bits are set for which the interrupts are enabled. At Reset this register is initialized to all zeros.

4.9.19 INTCTRL_MASKSTAT

This is a Read Only Register. Only those bits are set for which the interrupts are enabled and unmasked. At Reset this register is initialized to all zeros.

4.9.20 INTCTRL_FINALSTAT

This is a Read Only Register. Only those bits are set for which the interrupts are enabled, unmasked and whose priority level is higher than the system level priority or in case of two interrupts occurring at the same instant the one with the highest priority. At Reset this register is initialized to all zeros.

4.9.21 INTCTRL_INTVECTOR

This is a Read Only Register which contains the address of interrupt vector corresponding to the highest priority interrupt source.

4.9.22 INTCTRL_VECTOR0 to INTCTRL_VECTOR15

These are 16 Read/Write Registers which contain the interrupt vectors for interrupts corresponding to priority level 0 to 15.

4.9.23 INTCTRL_FIQ_ENABLE

This is a Read/Write Register to enable/disable fast interrupts. Writing 1 in the corresponding bit enables that interrupt and 0 disables it. At Reset all interrupts are disabled.

4.9.24 INTCTRL_FIQ_MASK

This is a Read/Write Register to mask interrupts. A 0 indicates the corresponding interrupt is unmasked and 1 indicates that it's masked. At Reset all interrupts are unmasked.

4.9.25 INTCTRL_FIQ_FORCE

This is a Read/Write Register to force interrupts. Writing 1 to a bit location forces the interrupt to occur. At Reset this register is initialized to all zeros.

4.9.26 INTCTRL_FIQ_RAW

This is a Read Only Register which shows the actual state of interrupt as generated by the corresponding device. A 1 indicates that an interrupt occurred. At Reset this register is initialized to all zeros.

4.9.27 INTCTRL_FIQ_STAT

This is a Read Only Register. Only those bits are set for which the interrupts are enabled. At Reset this register is initialized to all zeros.

4.9.28 INTCTRL_FIQ_FINALSTAT

This is a Read Only Register. Only those bits are set for which the interrupts are enabled and unmasked. At Reset this register is initialized to all zeros.

4.9.29 INTCTRL_SYSTEM_PRIORITY_LEVEL

This is a Read/Write Register. Only interrupts having priority levels higher than this figure are served.

4.10 Watch Dog

The watchdog is basically a counter that is capable of resetting the ARM core on a counter timeout. In order to avoid a reset the software must access the watchdog on a regular basis. The benefit of the watchdog functionality is that software dead locks, software runaway and corrupted RAM will be caught by the watchdog and the ARM core will be re-initialized.

Watchdog functionality may not be required in all applications. For theses occasions watchdog reset generation can be disabled, and the watchdog can be utilized as a periodic interrupt generator or timer.

4.10.1 Functional Description

The watchdog internal modules are illustrated in Figure 29. Two separate counters **prescale_cnt** (16bit) and **wd_12bit_cnt** (12bit) are used for what effectively becomes a single free running 28bit counter. Two status registers are maintained during operation, **wd_int_reg** which drives the interrupt signal *wd_int* and **wd_reset_reg** who drives the reset signal *wd_reset*, when the setup signal *wd_reset_en_reg* has been set.

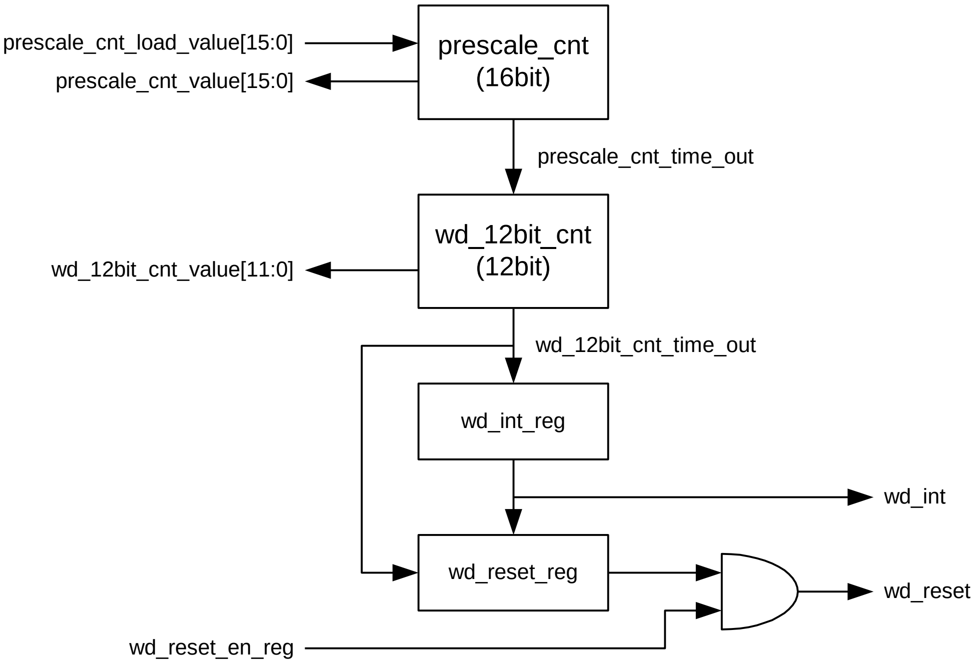


Figure 29: Basic illustration of the watchdog

prescale_cnt decrements every *pclk* cycle. When reaching zero *prescale_cnt_time_out* is set and **prescale_cnt** loads the value on *prescale_cnt_load_value[15:0]* into the counter register. *prescale_cnt_load_value[15:0]* is driven by a register mapped into the

APB bus memory space. The *prescale_cnt_load_value[15:0]* register initializes to 0xFFFF when wd_reset pulses.

wd_12bit_cnt differs from **prescale_cnt** in that the counter register only decrements when *prescle_cnt_time_out* is set. When it reaches zero *wd_12bit_cnt_time_out* is set and the counter register initializes to 0xFFF.

Understanding of signals *wd_int* and *wd_reset* is best achieved by studying the two scenarios in 26.

In "Scenario 1" generation of an ARM reset pulse *wd_reset* is illustrated. First *wd_12bit_cnt_time_out* pulses, which sets **wd_int_reg**. *wd_12bit_cnt_time_out* pulses again while **wd_int_reg** is set, which sets wd_reset_reg. In "Scenario 1" *wd_reset_en_reg* is enabled, and hence *wd_reset* will pulse, and the ARM core gets reset. At the same time **wd_int_reg** is cleared.

In "Scenario 2" *wd_12bit_cnt_time_out* pulses and sets the **wd_int_reg.** Next **wd_int_reg** is accessed from the APB bus and cleared. Continuing this pattern of operation will ensure that the watchdog will never reset the ARM core. If *wd_reset_en_reg* was not set "Scenario 2" would have illustrated timer operation or operation of a periodic interrupt generator.

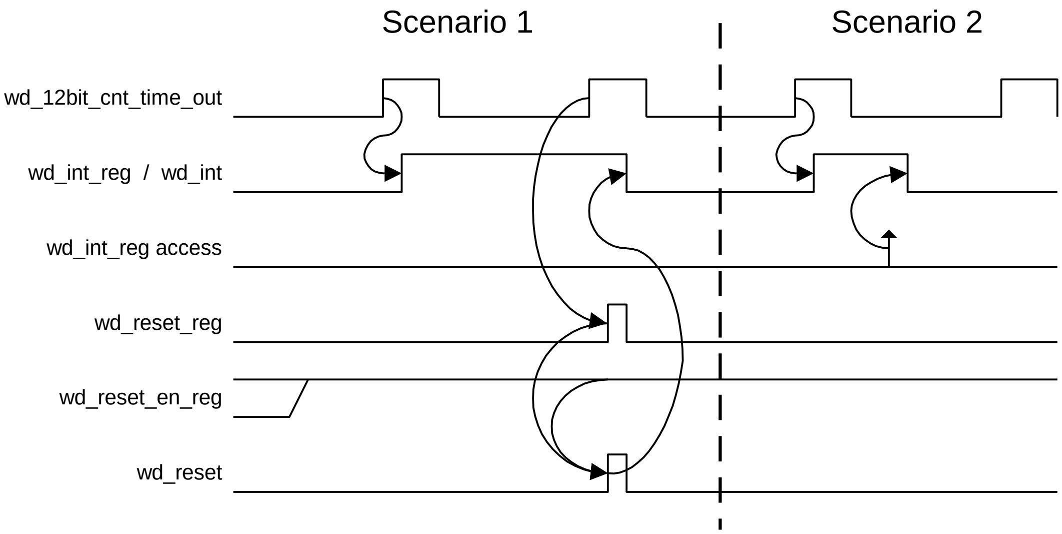


Figure 30: Wave form illustrating update of wd_int and wd_reset.

The wd_int_reg_access is a virtual signal illustrating that the wd_int_reg gets cleared.

4.10.2 Module Configuration

Address	Register
0xbf00 0000	WD_RESET_EN
0xbf00 0004	WD_INT
0xbf00 0008	WD_PRESCALE_LOAD
0xbf00 000c	WD_PRESCALE_CNT
0xbf00 0010	WD_COUNT

Table 34: Watch Dog Memory Map

4.10.3 WD_RESET_EN

0xbf00 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	unlock_wd															wd_reset_en
Reset:	0															0
	RW															RW

Name	Bit	Reset	Dir	Description
unlock_wd	15:1	0	RW	This feature helps prevent accidental enabling and disabling of the watchdog timer *reset function*. To disable the watchdog reset function, unlock_wd must be set to 0x91A (equivalent to setting whole register to 0x1234). To enable the watchdog reset function, unlock_wd must be set to 0x0. The watchdog reset function may then be disabled/enabled using wd_reset_en. See below.
wd_reset_en	0	0	RW	Used to enable/disable the watchdog timer *reset function*: enable = high, disable = low IMPORTANT: To disable the watchdog reset function, unlock_wd must first be set to 0x91A (equivalent to setting whole register to 0x1234). To enable the watchdog reset function, unlock_wd must first be set to 0x0. See above.

4.10.4 WD_INT

0xbf00 0004

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="15">unlock_wd_int</td><td>wd_int</td></tr>
<tr><td>Reset:</td><td colspan="15">0</td><td>0</td></tr>
<tr><td></td><td colspan="15">RW</td><td>RW</td></tr>
</table>

Name	Bit	Reset	Dir	Description
unlock_wd_int	15:1	0	RW	This feature helps prevent accidental enabling and disabling of the watchdog timer *interrupt*. To disable the watchdog interrupt, unlock_wd_int must be set to 0x2B3C (equivalent to setting whole register to 0x5678). To enable the watchdog interrupt, unlock_wd_int must be set to 0x0. The watchdog interrupt may then be disabled/enabled using wd_int. See below.
wd_int	0	0	RW	Used to enable/disable the watchdog timer *interrupt*: enable = high, disable = low IMPORTANT: To disable the watchdog interrupt, unlock_wd_int must first be set to 0x2B3C (equivalent to setting whole register to 0x5678). To enable the watchdog interrupt, unlock_wd_int must first be set to 0x0. See above.

4.10.5 WD_PRESCALE_LOAD

0xbf00 0008

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td colspan="16">prescale_cnt_load_value</td></tr>
<tr><td>Reset:</td><td colspan="16">0</td></tr>
<tr><td></td><td colspan="16">RW</td></tr>
</table>

Name	Bit	Reset	Dir	Description
prescale_cnt_load_value	15:0	0	RW	Write to this register to load a watchdog timer prescaler count.

4.10.6 WD_PRESCALE_CNT

0xbf00 000c

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	prescale_cnt_value															
Reset:	0															
	RW															

Name	Bit	Reset	Dir	Description
prescale_cnt_value	15:0	0	R	Read the current watchdog timer prescaler count value.

4.10.7 WD_COUNT

0xbf00 0010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				wd_count											
Reset:	0				0											
	R				R											

Name	Bit	Reset	Dir	Description
wd_count	11:0	0	R	Read the current watchdog timer count.

4.11 Dual Timer

4.11.1 Introduction

Timers count down from a programmed value and generate an interrupt when the count reaches zero. DICE JR provides 2 programmable timers that can be configured independently.

4.11.2 Features

The timer module has following features:

- Two programmable timers
- Configurable timer width: 32 bits
- Support for two operation modes: free-running and user-defined count

4.11.3 Internal Functional Description

This section describes each of the functional blocks that make up the Timers.
The timer component implements two identical but separately programmable timers.
The timers are accessed through a single AMBA APB interface.

A combined interrupt is also provided, which is active if any of the individual timer interrupts is active. Each loadable down counter is clocked by the ARM system clock (typically 49.152Mhz). The width of the counter is 32 bits. The initial value for each timer (the value it counts down from) is loaded into the counter by writing the desired value into the timer Local Count register (Timer*N*LoadCount, where *N* is in the range 1 to 2). Two events can cause the timer to load the initial count from its Timer*N*LoadCount register as follows:

- Timer is enabled after being reset or disabled
- Timer counts down to zero
-

Enabling/Disabling a Timer

Timers are disabled on reset. To enable a timer, write a 1 to bit 0 of its control register (Timer*N*ControlReg, where *N* is in the range 1 to 2). To disable a timer, write a 0 to bit 0 of its control register. When a timer is enabled, its counter decrements on each rising edge of its clock signal. When a timer transitions from disabled to enabled, the current value of its Timer*N*LoadCount register is loaded into the counter on the next rising edge of the timer clock*.*

When the timer enable (timer_en) goes low, it asynchronously resets the timer counter and any associated registers that exist in the timer clock domain, such as the toggle register and the at_zero register that is used to detect interrupts. When the timer enable is set, then a rising edge on the timer enable is used to load the initial value into the counter. One always reads back 0 when the timer is not enabled; otherwise, one reads back the current value of the timer (Timer*N*CurrentValue register).

If the timer reset is asserted when the timer rolls over, the timer is reset to all 1s, the interrupt register is cleared, and the toggle register is cleared.

Setting a Timer Operating Mode

When a timer counts down to 0, it loads one of two values depending on the timer operating mode. In user-defined count mode, the timer loads the current value of the Timer*N*LoadCount register. In free-running mode, the timer loads the maximum value depending on the timer width (2**32 – 1).

Use the user-defined count mode if you want a fixed, timed interrupt. Use the free running mode if you want a single-timed interrupt. When in free-running mode, the counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt occurs.

Select the user-defined count mode by writing a 1 to bit 1 of the timer control register. Select the free-running mode by writing a 0 to bit 1 of the timer control register.

Normal operation of the timer is as follows:

1. Disable the timer and program its operating mode by writing to its control register.

2. Load the Timer*N*LoadCount register.

3. Enable the timer.

Note

Before writing to a Timer*N*LoadCount register, you must disable the timer by writing a 0 to bit 0 of its control register.

Toggle Generation

A timer can be configured to generate a toggle output that toggles each time the timer reaches 0, the toggle signal is not available on DICE JR.

Interrupt Handling and Generation

In both the free-running and user-defined count modes of operation, a timer generates an interrupt when its count changes from 0 to its maximum count value. The setting of the internal interrupt occurs synchronous to the timer clock domain. This interrupt is transferred to the system clock domain in order to set the actual interrupt. The internal and actual interrupt are not generated if the timer is disabled; if the actual interrupt is set, then it is cleared when the timer is disabled.

The timer interrupt, once set, remains asserted until it is cleared by reading one of two registers, provided the timer is enabled. When the timer is disabled, the timer interrupt is cleared. You can clear an individual timer interrupt by reading its End of Interrupt register (Timer*N*EOI). You can clear all active timer interrupts at once by reading the global End of Interrupt register (TimersEOI) or by disabling the interrupt.

When reading the TimersEOI register, an interrupt is cleared at the rising edge of pclk, and when penable is low.

If the TimersEOI register is read during the time when the internal interrupt pulse is high, the interrupt is set. This occurs because setting the interrupts is of higher precedence than clearing the interrupts.

You can query the interrupt status of an individual timer without clearing the interrupt by reading the Timer*N*IntStatus register. You can query the interrupt status of all timers without clearing the interrupts by reading the global TimersIntStatus register.

Each individual timer interrupt can be masked using its control register. To mask an interrupt, write a 1 to bit 2 of the Timer*N*ControlReg control register. If all individual timer interrupts are masked, then the combined interrupt is also masked.

The two timer interrupts are combined into one global interrupt signal which is fed to the interrupt controller through the interrupt switching block described in the GP_CSR section of the DICE JR User Guide.

4.11.4 APB Interface

Standard AMBA 2.0 compliant APB interface is provided for reading and writing the internal registers. This component is configured for 32 bits bus width.

4.11.5 Module Configuration

The Timer module is little-endian. All timers are disabled on reset and can be enabled only by writing 1 to the Timer Enable Select bit of the timer control register.

Timer module contains both timer-specific and system registers. Table 1 show the address range of the registers of each timer, which are aligned to 32-bit boundaries.

The TimerLoadCount register and the Timer Mode Select bit of the Timer Control Register can be written only when the timer is disabled. Writing these registers while a timer is active results in undefined behavior. The proper sequence for programming these registers is as follows:

1. Write the Timer Control Register to set the Timer Mode and to disable the timer.

2. Write the TimerLoadCount register to program a new terminal count for the .

3. Write the TimerControlRegister to enable the timer.

All interrupt status and clearing registers can be accessed at any time.

The address range of timer is listed below:

Address Range	Function
0xc200 0000 to 0xc200 0010	Timer 1 Registers
0xc200 0014 to 0xc200 0024	Timer 2 Registers
0xc200 00a0 to 0xc200 00a4	Timer System Registers

Table 35: Timer Memory Map

4.11.6 Timer Registers

Address	Name/Type	Description
Base + 0x0	TimerNLoadCount Read/Write	Width: *32* Range: 0 to 2***31* Default value: 0 Description: Value to be loaded into Timer1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.
Base + 0x4	TimerNCurrentValue Read-only	Width: *32* bits wide Range: 0 to 2***31* Default value: *0* Description: Current Value of Timer1. This register is supported only when timer_*N*_clk is tied to the system clock (pclk). Reading this register when using independent clocks results in an undefined value.
Base + 0x8	TimerNControlReg Read/Write	Width: 3 bits Default value: 0 Description: Control Register for TimerN. Controls enabling, operating mode (free-running or defined-count), and interrupt mask of TimerN.
Base + 0xc	TimerNEOI Read-only	Width: 1 bit Default value: 0 Description: Reading from this register clears the interrupt from Timer N. It is set when a timer terminal count is reached
Base + 0x10	TimerNIntStatus Read-only	Width: 1 bit Default value: 0 Description: This register contains the interrupt status for Timer N. Reading from this register does not clear the interrupt from Timer N.

Table 36: Timer N Registers

4.11.6.1 TimerNLoadCount

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Uppe16 bits of Timer N load count value															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lower 16 bits of Timer N load count value															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

4.11.6.2 TimerNCurrentValue

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Uppe16 bits of Timer N's Current Value															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lower 16 bits Timer N's Current Value															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

4.11.6.3 TimerNControl

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													Timer_Int_Mask	Timer_Mode	Timer_En
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Field	Function	Post-Reset Value
Timer_En RW	Timer Enable Select	0: disabled 1: enabled
Timer_Mode RW	Timer Mode Select	0: free-running mode 1: user defined count mode
Timer_Int_Mask RW	Timer Interrupt Mask	0: timer interrupt not masked, 1: timer interrupt masked

4.11.6.4 Timer_N_EOI

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Timer_ EOI
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Field	Function	Description
Timer_EOI RO	Clear Timer N’s interrupt	Reading from this register clears the interrupt from Timer N. It is set when a timer terminal count is reached

4.11.6.5 TimerNIntStatus

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Timer_Int_Status
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Field	Function	Description
Timer_Int_Status RO	Timer N’s interrupt status	This register contains the interrupt status for Timer N. Reading from this register does not clear the interrupt from Timer N.

4.11.7 Timer System Registers

Address	Name/Type	Description
0xc200 00a0	TimersIntStatus Read-only	Width: *2* Default value: 0 Description: The register contains the interrupt status of all timers in the component. Reading from this register does not clear any active interrupts: 0 = either timer_intr or timer_intr_n is not active after masking 1 = either timer_intr or timer_intr_n is active after masking
0xc200 00a4	TimersEOI Read-only	Width: *2* Default value: 0 Description: Reading this register returns all zeroes (0) and clears all active interrupts.
0xc200 00a8	TimersRawIntStatus Read-only	Width: *2* Default value: 0 Description: The register contains the unmasked interrupt status of all timers in the component. 0 = either timer_intr or timer_intr_n is not active prior to masking 1 = either timer_intr or timer_intr_n is active prior to masking
0xc200 00ac	TIMERS_ COMP_VERSION	Width: *32* bits Description: Current revision number of the Timer component. This is a read-only register.

Table 37: Timer System Registers

4.11.7.1 TimersIntStatus

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														Timer1_int_Status	Timer0_int_Status
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Field	Function	Description
Timer0_Int_Status RO	Timer 0's interrupt status after masking	This bit contains the interrupt status for Timer 0 after masking. Reading from this register does not clear the interrupt from Timer 0.
Timer1_Int_Status RO	Timer 1's interrupt status after masking	This bit contains the interrupt status for Timer 1 after masking. Reading from this register does not clear the interrupt from Timer 1.

4.11.7.2 TimersEOI

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														EOI_1	EOI_0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Field	Function	Description
EOI_0 RO	Clear Timer 0's interrupt	Reading this bit clears the interrupt from Timer 0. It is set when a timer terminal count is reached
EOI_1 RO	Clear Timer 1's interrupt	Reading this bit clears the interrupt from Timer 1. It is set when a timer terminal count is reached

4.11.7.3 TimersRawIntStatus

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														INT_1	INT_0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Field	Function	Description
Timer0_Raw_Int_Status, RO	Timer 0's raw interrupt status	This bit contains the interrupt status for Timer 0 before masking. Reading from this register does not clear the interrupt from Timer 0.
Timer1_Raw_Int_Status, RO	Timer 1's raw interrupt status	This bit contains the interrupt status for Timer 1 before masking. Reading from this register does not clear the interrupt from Timer 1.

4.11.8 Interrupt Handling

The Timer*N*IntStatus and Timer*N*EOI registers handle interrupts to ensure safe operation of the interrupt clearing. If the system bus (AHB) can perform a write to clear an interrupt, it could continue with another transfer on the bus without knowing whether the write has occurred because of the hclk/pclk ratio. Therefore, it is much safer to clear the interrupt by a read operation.

To detect and service an interrupt, the system clock must be active. The timer_en output bus from this block is used to activate the necessary timer clocks and to ensure that the component is supplied with an active system clock while timers are running.

4.12 SPI Interface

4.12.1 SPI features

The SPI interface implemented in TCD22xx can be programmed to be master or slave and supports four different combinations of clock phases and polarity. The clock polarity and clock phase should be identical for the master device and all slave devices involved in the communication link. The transfer format from the master may be changed between transfers to accommodate various requirements of the slave device. The SPI bit rate can be controlled in master mode.

Several Data Lengths are supported: 8-bit, 16-bit, 24-bit & 32 bit. MSB can be transferred first or last, depending on the configuration.

4.12.2 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
SPIA_SS	138 (shared)	123 (shared)	I/O (S)	6	SPI Slave Select
SPIA_MISO	139 (shared)	124 (shared)	I/O (S)	6	SPI Master. In, Slave Out
SPIA_MOSI	104 (shared)	97 (shared)	I/O (S)	6	SPI Master. Out, Slave In
SPIA_CK	105 (shared)	98 (shared)	I/O (S)	6	SPI Clock
SPIB_SS	57 (shared)	N/A	I/O (S)	6	Alt. SPI Slave Select
SPIB_MISO	65 (shared)	N/A	I/O (S)	6	Alt. SPI Master. In, Slave Out
SPIB_MOSI	66 (shared)	N/A	I/O (S)	6	Alt. SPI Master. Out, Slave In
SPIB_CK	67 (shared)	N/A	I/O (S)	6	Alt. SPI Clock

Table 38: SPI Signal description

The SPI clock is generated by the master, and the SPI_SS signal represents the slave device select from the SPI master. The SPI clock, MISO & MOSI pins are directly connected between master and slave. The MISO signal is the output from the slave (slave transmission) and the MOSI signal is the output from the master (master transmission). On the chip interface SPI signals can be mapped to two locations depending on GPCSR register configuration (see 4.1.6). Location B is not available in TCD2210.

4.12.3 SPI Transfer formats

CLKPL	CLKPHASE	Sample On
0	0	Positive edge
0	1	Negative edge
1	0	Negative edge
1	1	Positive edge

Table 39: CLKPHASE & CLKPL configurations

The four SPI formats are controlled by CLKPHASE and CLKPL bits in the SPI control register. Any combination of those two bits is valid. Table 39 shows the edge of SPICLK on which data is sampled in various combinations of CLKPHASE & CLKPL.

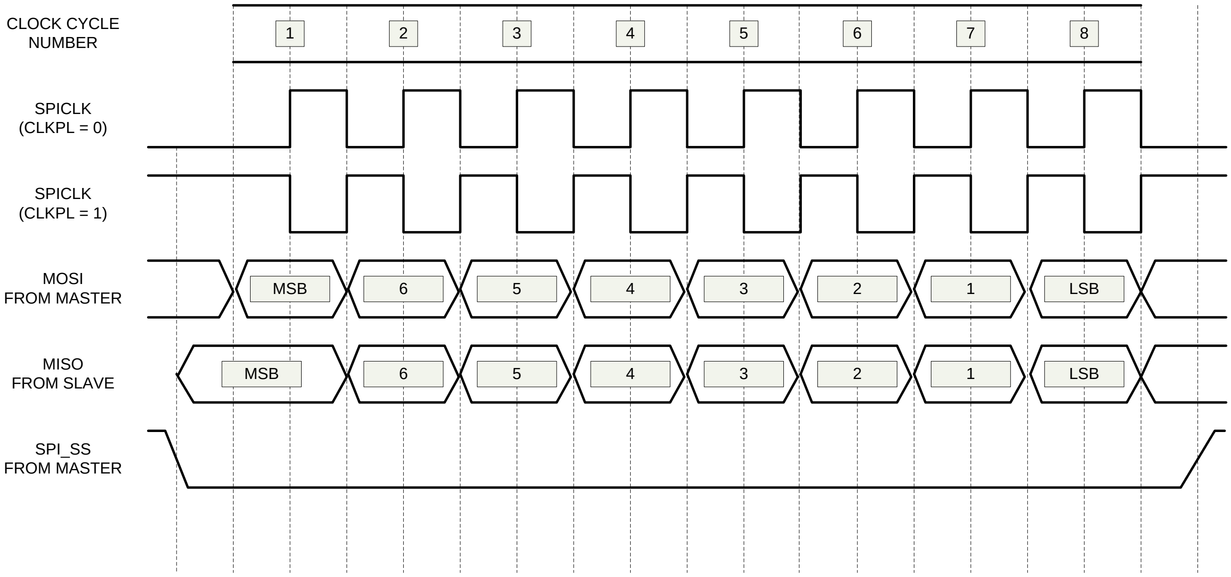


Figure 31, SPI CLOCK & Data for CLKPHASE = 0

Figure 31 shows the SPI transfer protocol for CLKPHASE = 0. Figure 32 shows the SPI transfer for CLKPHASE = 1. Note that in both cases the transfer length is 8-bit and MSB is transferred first.

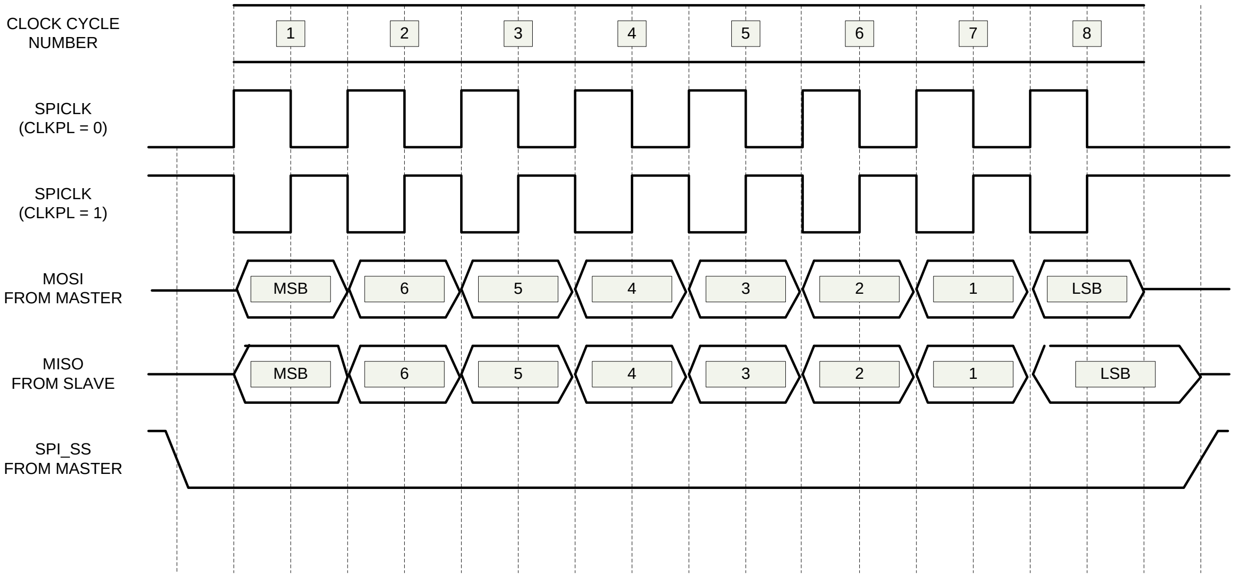


Figure 32: SPI CLOCK & Data for CLKPHASE = 1

4.12.4 SPI Data Formats

The SPI interface is able to transfer 8, 16, 24 or 32 bits of data, if configured so through the SPICTL register.

When configured to be 8-bit, SPI sends and receives one byte of data in each transfer. If transmitting, only lower byte of 32-bit data register is transmitted. In 16-bit mode, SPI transmits 2 lower bytes of the SPI transmit buffer. In 24-bit mode three lower bytes of the SPI data register are transmitted.

4.12.5 SPI Interrupts

SPI status is reported trough Status register or through interrupts (if not masked). SPI reports the following interrupt conditions:

TX underrun – happens when there is no relevant data to be loaded into TX register (not possible in master mode)
RX overrun - happens when new data is ready, but the old data has not yet been read by the processor. If configured through SPI_CTL register, the old data could be either kept or overwritten by the new one.
RX full - Data is ready to be read
TX empty - Data was pushed to shift register
TX done - Actual shifting is done

First two conditions are cleared, when status is read by the processor.

4.12.6 SPI Module Configuration

Address	Register
0xc500 0000	SPI Control register
0xC500 0004	SPI Status Register
0xC500 0008	SPI Interrupt Mask Register
0xC500 000c	SPI Data Register
0xC500 0010	SPI Baud Rate Register

Table 40: SPI Memory Map

4.12.7 SPI programming model

4.12.7.1 SPI_CNTL Register

0xc500 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								WSIZE		CLKPL	CLKPHASE	KPOLD	CLRIUR	MSBF	MSTR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	31:8	0	R	Reads back as 0
WSIZE	7:6	0	RW	Word size for SPI Transfer 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit
CLKPL	5	0	RW	SPI Clock polarity (see **Error! Reference source not found.**, Figure 32)
CLKPHASE	4	0	RW	SPI Clock phase (see **Error! Reference source not found.**, Figure 32)
KPOLD	3	1	RW	Keep old data or overwrite it in case of overrun in RX 0: Overwrite 1: Keep old data
CLRIUR	2	1	RW	Data fill in case of underrun 0: transmit last 1: transmit all zeroes
MSBF	1	0	RW	Transmit MSB first 0: MSB first 1: LSB first
MSTR	0	1	RW	Master/Slave mode 0: slave 1: master

4.12.7.2 SPI Status Register

0xc500 0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											TX DONE	TX URUN	RX ORUN	TX EMPTY	RX FULL
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Reserved	31:5	0	R	Reads back as 0
TX_DONE	4	0	R	Actual Shifting out is done
TX_URUN	3	0	R	Underrun condition detected
RX_ORUN	2	0	R	Overrun condition detected
TX_EMPTY	1	0	R	No Data in SPI TX data buffer
RX_FULL	0	0	R	Data is ready in SPI RX buffer

4.12.7.3 SPI Interrupt Mask Register

0xc500 0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												TX URUN	RX ORUN	TX EMPTY	RX FULL
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Reserved	31:4	0	R	Reads back as 0
TX_URUN MASK	3	0	R	Underrun interrupt mask 0: Mask 1: Enable
RX_ORUN MASK	2	0	R	Overrun Interrupt mask 0: Mask 1: Enable
TX_EMPTY MASK	1	0	R	Empty interrupt Mask 0: Mask 1: Enable
RX_FULL MASK	0	0	R	Full Interrupt mask 0: Mask 1: Enable

4.12.7.4 SPI Data register

0xc500 000c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	rW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Data	31:0	0	RW	Writes TX data, Reads RX data

4.12.7.5 SPI Baud Rate Register

0xc500 0010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	rW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	31:16	0	R	Reads back as zeroes
Baud Rate	15:0	0x3e8	RW	spiclk = Fsys/(8(baud_rate+1))

4.13 Remap Block

4.13.1 Remap features

The Remap block contains an address decoder that supports up to two memory maps: a boot memory map, and a normal memory map. A typical use of this feature is to allow ROM to be mapped to 0x0000 on system reset (boot memory map) and RAM to be remapped to the same memory space after initialization (normal memory map). The DW_apb_rap can be configured to include a remap control register that is used to switch the DW_ahb address decoder from boot mode to normal mode operation.

With power-on reset, the boot memory map is selected. This is reprogrammed some time later to the normal memory map by writing to the remap register in the Remap block. This in turn sends a signal to the AHB address decoder to change its memory map. The memory map cannot be changed back to the boot map by writing to this register. It is reset only when a power-on reset occurs.The remap register is cleared by the power-on reset.

Remap block also implements a configurable, read-only, identification register. This is used to store a processor-accessible system ID of the whole DICE family.

4.13.2 Remap block Memory map

Address Range	Function
0xc000 0004	**DICE Family ID Register**
0xc000 0008	**Remap Register**

Table 41: Address Remap Memory map

4.13.3 Address Remap Register description

4.13.3.1 DICE Family ID Register

0xc000 0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	1	0	1	0	1	0	0	0	1	0	0	0	0	1	1
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	rW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
DICEFamily ID	31:0	54430000	R0	Should be used with chip ID in GPCSR (0xc700 0014)

4.13.3.2 Address Remap Register

0xc000 0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	rW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Remap
Reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	31:1	0	R0	Reads back as zerroes
Remap	0	0	RW	Should be set to switch memory map from boot (default) to application Couldn't be reset by SW

Chapter 5 DICE

5.1 Router

The DICE router module handles all audio transfers from all inputs to all outputs. It is able to handle up to 32 units (16 Rx and 16 Tx) that match the standard DICE router interface. Each Rx/Tx is defined to contain up to 16 channels of 32 bit audio samples.

The DICE Router module has the following features:

- Handles all audio transfers between Rx and Tx modules
- The router contains a router table of 128 entries making it possible to make up to 128 data moves within one sample.
- Each router entry has an associated peak value which will be updated with the max magnitude of every sample. This value is cleared when read from the ARM host.
- Router contains error detection to indicate that the number of selected data moves cannot be made within one sample period. (This is only applicable at high sample rates (176.4-192kHz in which case only 100 channels can be routed).

Peak Detection description

Peak value is calculated per channel and basically means the highest data value read by the router from the channel. All peak values are initialized to zero and stored in the router RAM in upper 12 bits [27:16]

The router reads the RAM to get the next destination and source address located in bits [15:0] of the same RAM address. With that Router gets a previous peak value for that channel. In the next clock router fetches the data from the source address. This Data is compared to a peak value as well as being delivered to destination address. If present value of data is less or equal to the peak value, peak does not change. If the value is bigger, new data is written to the peak location for that channel.

Peak operation is enabled when “Peak_ON” bit in the Router CTRL register is set. Peak value is cleared by ARM Read from memory.

SW should initialize the peak portion of all router entries to zeroes prior to use.

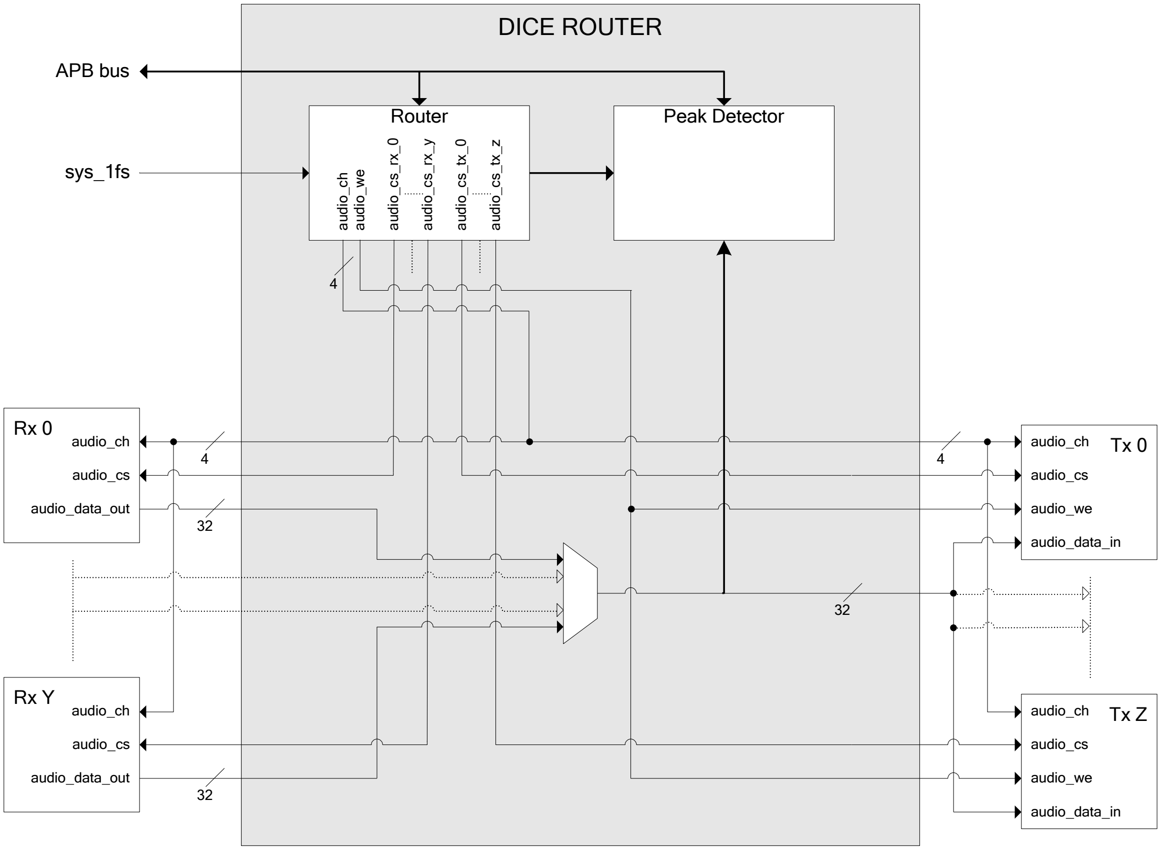


Figure 33: DICE Router Block Diagram

5.1.1 Module Configuration

Note that there are 128 entry points (moves) in the router.

Address	Register
0xce00 0000	ROUTER_CTRL
0xce00 0400	ROUTER_ENTRY0
0xce00 0404	ROUTER_ENTRY1
:	:
0xce00 07fc	ROUTER_ENTRY127

Table 42: ROUTER Memory Map

5.1.2 ROUTERn_CTRL

Address - 0xce00 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNT								Reserved					Peak ON	ERR	Router ON
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW

Name	Bit	Reset	Dir	Description
COUNT	15:8	0	RW	Selects the number of valid entries for this router. The router will handle COUNT+1 entries.
Peak_ON	2	0	RW	Enables PEAK operation
ERR	1	0	R	This read-only bit indicates that the router was not able to complete the routing within on cycle.
Router_ON	0	0	RW	This bit enables this router.

5.1.3 ROUTERn_ENTRYm

Address - 0xce00 0400 - 0xce00 07fc

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRC_BLK				SRC_CH				DST_BLK				DST_CH			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
PEAK Value	27:16	0	RW	Peak value for that particular channel
SRC_BLK	15:12	0	RW	Selects the source block for this router entry.
SRC_CH	11:8	0	RW	Selects the source channel for this router entry.
DST_BLK	7:4	0	RW	Selects the destination block for this router entry.
DST_CH	3:0	0	RW	Selects the destination channel for this router entry.

5.1.4 Source Block ID’s

ID	Block	Channels
0	AES	8
1	ADAT	16
2	Mixer	16
3	Reserved	N/A
4	INS0	16
5	INS1	16
6	Reserved	N/A
7	Reserved	N/A
8	Reserved	N/A

ID	Block	Channels
9	Reserved	N/A
10	ARM APB Audio	8
11	AVS-0	16
12	AVS-1	16
13	Reserved	N/A
14	Reserved	N/A
15	Mute	Any

Table 43: Router Source Block codes

5.1.5 Destination Block ID's

ID	Block	Channels
0	AES	8
1	ADAT	16
2	Mixer TX0	16
3	Mixer Tx1	2
4	INS0	16
5	INS1	16
6	Reserved	N/A
7	Reserved	N/A
8	Reserved	N/A
9	Reserved	N/A
10	ARM APB Audio	8
11	AVS-0	16
12	AVS-1	16
13	Reserved	N/A
14	Reserved	N/A
15	Reserved	N/A

Table 44: Router Destination Block Codes

5.2 Clock Controller

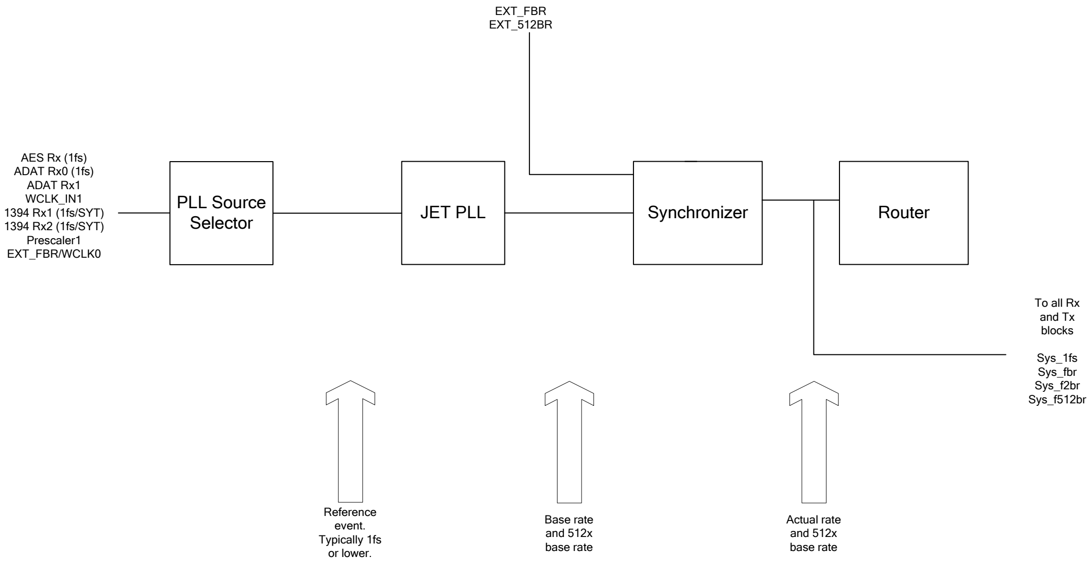


Figure 34: Clock Controller Block Diagram

The clock controller contains the logic to handle selection of clock sources, clock domain memberships, block sync selection for the 60958 and AES receivers and transmitters, as well as setup for receiver clock regeneration (onboard VCO's), sample rate/phase detection and other clock related functions of the TCD22XX.

The clock system consists of three types of blocks.

- Router
- Synchronizer
- Jet^{TM} PLL

The router is described in the section above. It handles the routing of audio data from source devices (Rx Modules) to sink devices (Tx Modules). Router gets its clocks from a synchronizer. To operate correctly, router requires a 1fs and system clock.

Each synchronizer takes as input a base rate clock and a 512 x base rate clock and generates at its output the actual 1fs (sys_1fs) and base rate signals for use with ADAT and dual wire AES modes (sys_fbr and sys_f2br). The sys_f512br clock is also used in Tx modules to generate bit clocks. In most cases the synchronizer will get its input from one of the Jet^{TM} PLL's, but in certain cases the synchronizer will slave to other sources such as the external slave interface.

The Jet^{TM} PLL takes any reference input and generates a base rate and 512 times the base rate clocks. The base rate can be programmed to have a fractional relationship to the incoming reference. The reference is called the event.

The clock controller also contains two measurement blocks, each programmable to act as either a sample rate counter or a phase detector. Each block outputs a 32 bit value that can be read by the ARM, and each block can be programmed to count a maximum number of cycles before outputting this value. The counters/detectors count in cycles at the frequency of the ARM system clock (typically 49.152MHz). Each block contains two multiplexers used to select either the two input values when in phase detector mode or the one input value when in sample rate counter mode. When in sample rate counter mode the blocks will count the sample rate of the signal at input 1.

The Clock doubler is part of the Clock Controller, and is used to double the clock frequency generated by the oscillator circuit whose inputs are pins xtal 1 and xtal2 (typically 25.000MHz at xtal1/xtal2 doubled to 50.000MHz). Figure 27 illustrates how it is connected inside DICE 22xx. As seen in the figure both sclk and the clock doubler output can be selected as main clock for the Jet^{TM} PLL. The flexibility provided by the design allows convenient support of both 1394 applications and non-1394 applications, without sacrificing Jet^{TM} PLL performance.

The typical setup for 1394 applications will be that 'clk_out_hpll' feeds from the clock doubler and 'clk_out_system' from the sclk (PHY clock) input. This is so because the 1394 LLC and PHY need to be ‵synchronous′. Since the sclk (from the PHY) has been generated within the PHY device using "unknown" PLL technology the quality of this clock cannot be guaranteed. For this reason, the Jet^{TM} PLL by default uses the output of the clock doubler, since we are in "full control" of the quality of this clock. Furthermore, we have the possibility to choose an X-tal frequency that is "out of sync" with the normal audio sample rates (recommended 25.000 MHz) which avoids beating and improves Jet^{TM} PLL performance. For 1394 applications we also want to be able to power-up/power-down on a request sent through 1394. When powered down, the sclk from the PHY will be disabled. To be able to detect a wake-up both our Power Manager and the 1394 LLC require a "second clock". This clock is in our case the direct input from 'xtl1'.

The typical setup for non-1394 applications will be that both clocks (Jet^{TM} PLL and system) feed from the clock doubler and the sclk input is unused (hardwired). This requires only one external clock source. The flexibility is still there to be able to distinguish between the 2 clocks by forcing a clock on the sclk input if it is requested.

“clk_out_hpll' is dedicated for the Jet^{TM} PLL block only, which means that the rest (including the prescalers within the Clock Controller) are fed with 'clock_out_system'.

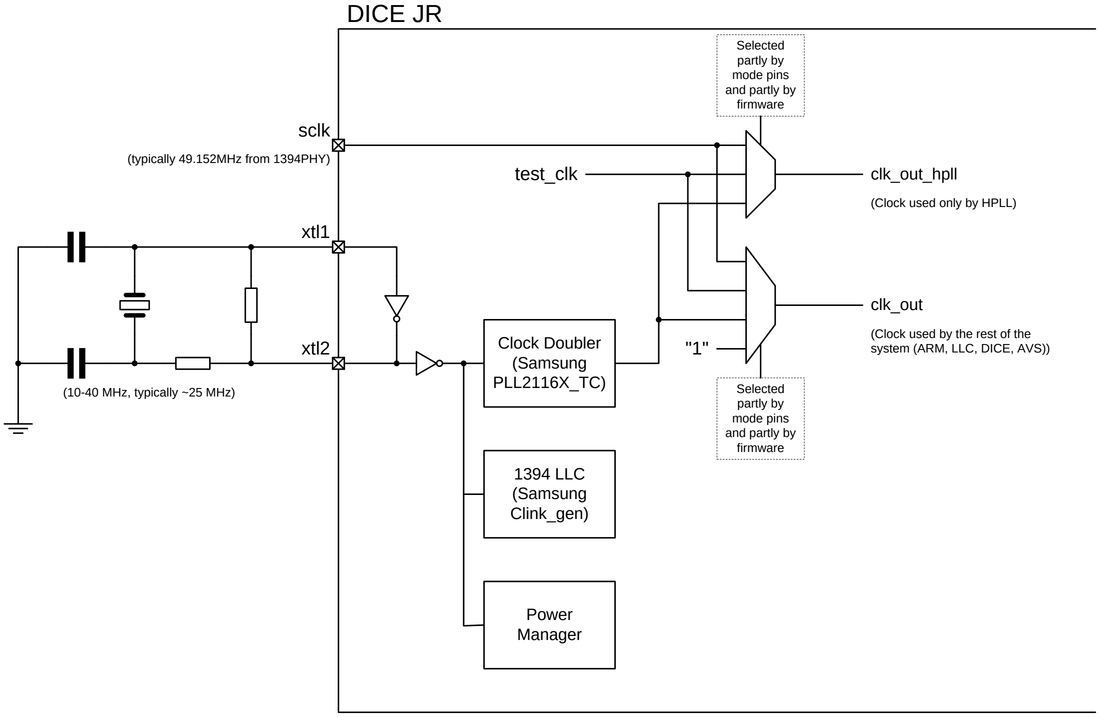

DICE JR
sclk
(typically 49.152MHz from 1394PHY)
xtl1
xtl2
(10-40 MHz, typically ~25 MHz)
test_clk
Selected partly by mode pins and partly by firmware
clk_out_hpll
(Clock used only by HPLL)
clk_out
(Clock used by the rest of the system (ARM, LLC, DICE, AVS))
"1"
Selected partly by mode pins and partly by firmware
Clock Doubler (Samsung PLL2116X_TC)
1394 LLC (Samsung Clink_gen)
Power Manager

Figure 35: Clock Doubler Block Diagram

5.2.1 Signal Description

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
EXT_FBR WCLK_IN0	85	78	B	6	External 1fs base rate clock (5V)
EXT_512BR WCLK_OUT0	86	79	B	6	External 512 x base rate clock (5V)/ Word clock out
XTAL2	102	95	O	-	XTAL - clock doubler/power manager/LLC
XTAL1	103	96	I	-	XTAL - clock doubler/power manager/LLC
WCLK_IN1	65	N/A	B	6	Word Clock In (5V)
WCLK_OUT1	66	N/A	B	6	Word Clock Out

Table 45: Clock Controller Signal Description

5.2.2 Module Configuration

Address	Register
0xce01 0000	SYNC_CTRL
0xce01 0004	DOMAIN_CTRL
0xce01 0008	EXTCLK_CTRL
0xce01 000c	BLK_CTRL
0xce01 0010	REFEVENT_CTRL
0xce01 0014	SRCNT_CTRL
0xce01 0018	SRCNT_MODE
0xce01 001c	Reserved
0xce01 0020	Reserved
0xce01 0024	AES_VCO_SETUP
0xce01 0028	Reserved
0xce01 002c	Reserved
0xce01 0030	Reserved
0xce01 0034	PRESCALER
0xce01 0038	Reserved
0xce01 003c	HPLL_REF
0xce01 0040	SRCNT1
0xce01 0044	SRCNT2
0xce01 0048	SR_MAX_CNT1
0xce01 004c	SR_MAX_CNT2

Table 46: Clock Ctrl Memory Map

5.2.3 SYNC_CTRL

Address - 0xce01 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														SYNC1_SRC	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
SYNC_SRC	1:0	0x3	RW	Selects the clock source for Synchronizer 1 00: aes_1fs (or ~ aes_1fs, res. for test and debug) 01: reserved 10: Slave Inputs (EXT_FBR and EXT_512FB) 11: hPLL xx: Reserved for internal use

5.2.4 DOMAIN_CTRL

Address - 0xce01 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved										RTR_FS		Reserved			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
reserved	7:6	0	RW	Reserved
RTR_FS (SYS_MODE)	5:4	0	RW	Selects the FS source for Router (rate mode) 00: base rate from Synchronizer 01: 2 x base rate from Synchronizer 10: 4 x base rate from Synchronizer 11: reserved
Reserved	3:2	0	RW	Reserved
Reserved	1:0	0	RW	Reserved

5.2.5 EXTCLK_CTRL

Address - 0xce01 0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved										WCLK1		EXTBR		WCLK0	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
WCLK_OUT1	5:3	0	RW	Selects the source for the word clock output (WCLK_OUT1) 000: off(default) 001: base rate from Synchronizer 010: 2 x base rate from Synchronizer 011: 4 x base rate from Synchronizer 100: sys_1fs
EXT_FBR / WCLK_OUT0	2:0	0	RW	Selects the source for the external master mode output for base rate rate (EXT_FBR / WCLK_OUT0) 000: off (default) 001: base rate from Synchronizer 010: 2 x base rate from Synchronizer 011: 4 x base rate from Synchronizer 100: sys_1fs 101: ext_fbr from synchronizer input (for master to slave op.)

5.2.6 BLK_CTRL

Address - 0xce01 000c

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TXDI2BLK			TXDI1BLK			AESTXBLK			BLKO		
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
TXDI2BLK	11:9	000	RW	Selects the source of the block sync for AVS ATX2 transmitter 000: Blocksync from AES Rx 001: Blocksync from AVS ARx1 010: Blocksync from AVS ARx2 011: Reserved 100: Reserved 101: Blocksync from AES Tx Blocksync Generator 111: Blocksync from AVS ATx1 110: Reserved
TXDI1BLK	8:6	000	RW	Selects the source of the block sync for AVS ATX1 transmitter 000: Blocksync from AES Rx 001: Blocksync from AVS ARx1 010: Blocksync from AVS ARx2 011: Reserved 100: Reserved 101: Blocksync from AES Tx Blocksync Generator 110: Reserved 111: Blocksync from AVS ATx2

Name	Bit	Reset	Dir	Description
AESTXBLK	5:3	000	RW	Selects the block sync source for the AES Tx. 000: Blocksync from AES Rx 001: Blocksync from AVS ARx1 010: Blocksync from AVS ARx2 011: Reserved 100: Reserved 101: Reserved 110: Blocksync from AVS ATx1 111: Blocksync from AVS ATx2
BLKO	2:0	000	RW	Selects the source for the block sync output pin. 000: Blocksync from AES Rx 001: Blocksync from AVS ARx1 010: Blocksync from AVS ARx2 011: Reserved 100: Reserved 101: Blocksync from AES TX Blocksync Generator 110: Blocksync from AVS ATx1 111: Blocksync from AVS ATx2

5.2.7 REFEVENT_CTRL

Address - 0xce01 0010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											REF				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
REF	4:0	0x8	RW	Selects the ref. event source for PLL1. 00000: AES Rx (1fs) 00001: ADAT Rx0 (1fs) 00010: ADAT Rx1 (1fs) 00011: WCLK_IN1 00100: 1394 Rx1 (1fs/SYT_INTERVAL) 00101: 1394 Rx2 (1fs/SYT_INTERVAL) 00110: 00111: 01000: Prescaler 01001: EXT_FBR / WCLK_IN0 xxxxx: Reserved

5.2.8 SRCNT_CTRL

Address - 0xce01 0014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													SRC2_IN2			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRC2_IN2	SRC2_IN1					SRC1_IN2					SRC1_IN1				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
SRC2_IN2	15:12	0x0	RW	Selects source 2 for sample rate counter 2 See table below.
SRC2_IN1	11:8	0x8	RW	Selects source 1 for sample rate counter 2 See table below.
SRC1_IN2	7:4	0x0	RW	Selects source 2 for sample rate counter 1 See table below.
SRC1_IN1	3:0	0x8	RW	Selects source 1 for sample rate counter 1 See table below.

Source Select Value	Source
0000	AES Rx (1fs)
0001	ADAT Rx0 (1fs)
0010	ADAT Rx1 (1fs)
0011	WCLK_IN1
0100	1394 Rx1 (1fs/SYT_INTERVAL)
0101	1394 Rx2 (1fs/SYT_INTERVAL)
0110	Reserved
0111	Reserved
1000	Router 1fs
1001	Reserved
1010	EXT_FBR / WCLK_IN0 pin
1011	AES Rx0 (1fs)
1100	AES Rx1 (1fs)
1101	AES Rx2 (1fs)
1110	AES Rx3 (1fs)
1111	Reserved

Table 47: Sample Rate Counter Input Selection

5.2.9 SRCNT_MODE

Address - 0xce01 0018

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														SRM2	SRM1
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
SRM2	1	0	RW	Selects the sample rate counter mode for counter 2 0: Phase counting 1: Period Counting
SRM1	0	0	RW	Selects the sample rate counter mode for counter 1 0: Phase counting 1: Period Counting

5.2.10 AES_VCO_SETUP

Address - 0xce01 0024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved												aes_down_pol	aes_up_pol	aes_clk_regen_s	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	aes_clk_regen_m								aes_clk_regen_p						aes_clk_regen_vco_pwrdn	aes_clk_regen_vco_pwrdn
Reset:	248								62						1	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
aes_down_pol	19	0	RW	Select polarity of DOWN signal to on-chip AES VCO 0: Down signal to AES VCO is active low 1: Down signal to AES VCO is active high
aes_up_pol	18	1	RW	Select polarity of UP signal to on-chip AES VCO 0: Up signal to AES VCO is active low 1: Up signal to AES VCO is active high
aes_clk_regen_s	17:16	2	RW	Set the value for the Post - Scaler (S) 00: Divide by 1 01: Divide by 2 10: Divide by 4 11: Divide by 8
aes_clk_regen_m	15:8	248	RW	Set the value for the Main - Divider (M) Range is 1 to 248
aes_clk_regen_p	7:2	62	RW	Set the value for the Pre - Divider (P) Range is 1 to 62
aes_clk_regen_vco_pwrdn	1	1	RW	Disable/Enable the internal VCO for the AES Receiver 0: Enable internal AES Receiver VCO 1: Disable internal AES Receiver VCO

Name	Bit	Reset	Dir	Description
aes_clk_regen_vco_ext_clk_sel	0	0	RW	Selects source of VCO clock for AES Receiver either int. or ext. 0: From internal VCO / Clock 1: From external AES Receiver VCO

5.2.11 PRESCALER

Address - 0xce01 0034 – 0xce010038

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRE_DIVn															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRE_DIVn															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
PREDIVn	31:0	0	RW	Sets the divider for the prescaler. Fs = pclk/PREDIVn Note: Values lower than 2 are illegal. Pclk typical freq. 49.152MHz

5.2.12 HPLL_REF

Address - 0xce01 003c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															system_clk_vco_s
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	system_clk_vco_s	system_clk_vco_m								system_clk_vco_p						PLLCLK
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description

Name	Bit	Reset	Dir	Description
system_clk_vco_s	16:15	3	RW	Set the value for the Post - Scaler (S) 00: Divide by 1 01: Divide by 2 10: Divide by 4 11: Divide by 8
system_clk_vco_m	14:7	40	RW	Set the value for the Main - Divider (M) Range is 1 to 248
system_clk_vco_p	6:1	1	RW	Set the value for the Pre - Divider (P) Range is 1 to 62
PLLCLK	0	0	RW	Sets the reference clock for the PLL's. 0: Locally doubled clock (typical 49.152Mhz) from pins XTAL1/XTAL2 1: PHY clock, (SCLK pin)

5.2.13 SRCNTn

Address - 0xce01 0040

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COUNTn															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNTn															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
COUNTn	31:0	0	R	This register holds the last count for Sample Rate Counter n. The counter runs at the freq. of the ARM system clock (typically 49.152MHz). When set for phase detection, an edge on input 1 will reset the counter and an edge on input 2 will latch it into this register. When set for sample rate counting every edge on input 1 will latch the count into this register and restart the counter.

5.2.14 SR_MAX_CNTn

Address - 0xce01 0048 - 0xce01 004c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SR_count_max_count															
Reset:	0x6db															
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SR_count_max_count															
Reset:	0x6db															
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
SR_count_max_countn	31:0	0x6DB	R	Max no. of 50 MHz cycles the counter will count to before value is output. For Sample Rate counter n Reset value

5.3 Jet^{TM} PLL

5.3.1 Jet^{TM} PLL Background

The Jitter Elimination Technologies (Jet)^{TM} PLL on the TCD22XX chips feature state-of-the-art jitter rejection abilities and extremely low intrinsic jitter levels. The PLL is ideally suited for the clock and data boundaries between any analog or digital source and destination.

Like all phase-locked loops, Jet^{TM} PLL use feedback to lock an oscillator to a timing reference. They track slow reference changes, but effectively free-run through rapid modulations of the reference. From a jitter transfer point of view, they provide increasing jitter attenuation above some chosen corner frequency.

Jitter attenuation is just one aspect of PLL design. Other considerations include frequency range and intrinsic jitter. It can be shown that conventional designs are bound by a fundamental trade-off between these three aspects. For example, specifying a frequency range of one octave means using a low-Q oscillator. But that makes for high intrinsic jitter when the loop corner frequency is held down. Conversely, good jitter attenuation and low intrinsic jitter can be had by using a voltage-controlled crystal oscillator (VCXO). But the frequency range is then tiny. A further consideration is that only low-Q oscillators are easy to integrate on chip.

Jet^{TM} PLL sidesteps the above-mentioned trade-off. It incorporates two loops. One is largely or wholly numeric, and has its corner frequency set low enough to give good reference-jitter attenuation. The other regulates the analog oscillator and has its corner frequency set much higher, to moderate the intrinsic jitter. The two corner frequencies might be around 10 Hz and 100 kHz, for example.

Another benefit of having a high corner frequency in the analog loop is that interference, e.g. via the oscillator's supply rail, is suppressed more effectively.

Jet^{TM} PLL requires a fast, stable, fixed-frequency clock. It is this that gives it stability in the band between the two corner frequencies. (Equally, in this band any jitter on this clock passes straight through to the Jet^{TM} PLL's clock output.) The stable clock is usually derived from a free-running crystal oscillator.

Jet^{TM} PLL contains a number-controlled oscillator, which can also be called a fractional frequency divider. Like the analog oscillator, this injects jitter. Typically, spectrum shaping is used to push most of that jitter up to frequencies where it will be heavily attenuated by the analog loop.

As well as frequency-locking the analog oscillator to the provided reference, Jet^{TM} PLL can also phase-lock an associated frame sync to the reference.

Jet^{TM} PLL can generate internal master clock rates with extremely fine frequency resolution and precision e.g. 44.056 kHz +/- 0.4 PPM. Depending on the precision of the master XTAL connected to the TCD22XX.

The Jet^{TM} PLL also allows clock “slew rate” to be controlled when the operating frequency is changing (e.g. 44.1kHz to 48kHz or when experiencing a clock source phase shift). With a slow slew rate, downstream equipment might not need to go into “unlock” state and back to lock state during such a shift.

Jet^{TM} PLL has the ability to lock to frequencies as low as 15Hz (e.g. 24/25/Drop Frame video rates), and as high as 256xFs “super clock” (e.g. 12.288MHz).

The Jet^{TM} PLL intrinsic jitter performance can be lowered even further by overriding the internal analog VCO with an ultra-high performance external TCXO if desired.

Jet[TM] PLL have additional facilities for measuring frequency and phase of the incoming reference signal and posting events to firmware if clock quality falls outside acceptable limits (e.g. reference signal disappears).

The Jet[TM] technologies in are covered by several patents.

5.3.2 Block Diagram

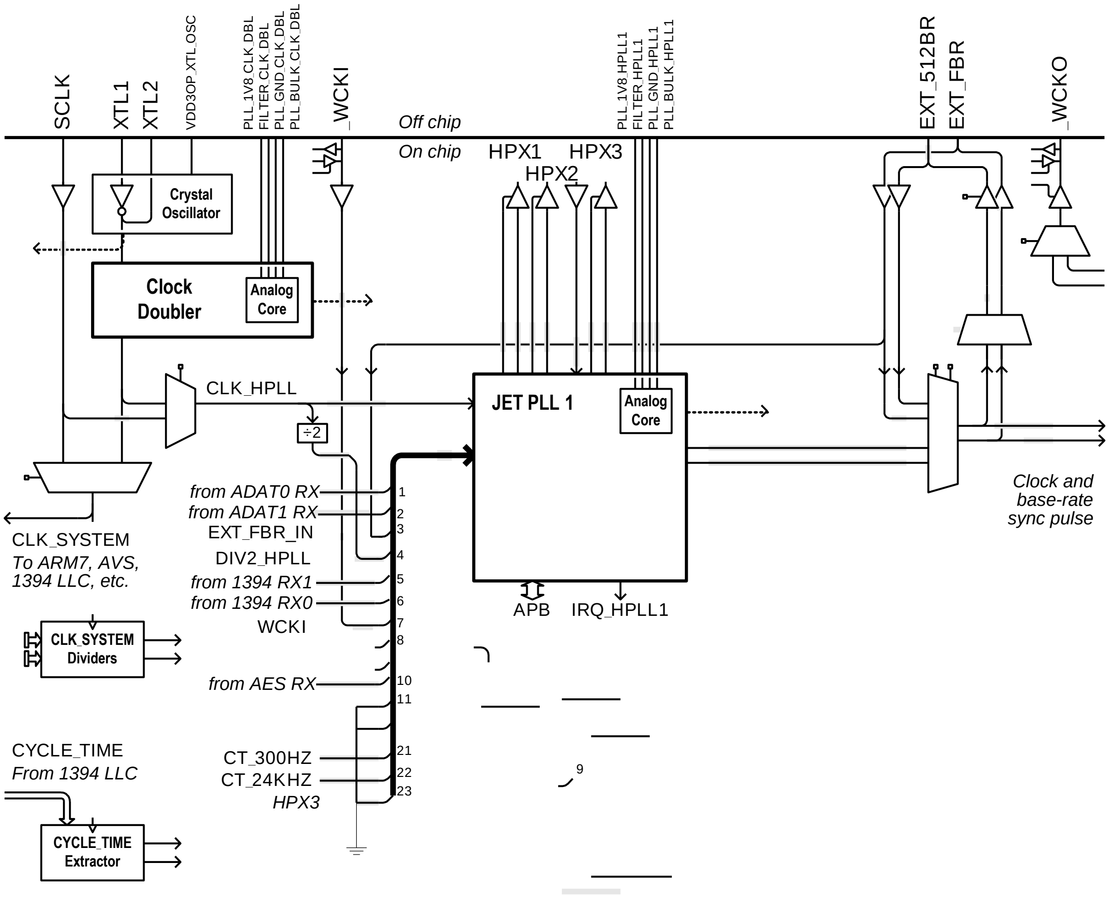


Figure 36: Detailed Jet[TM] PLL Block Diagram

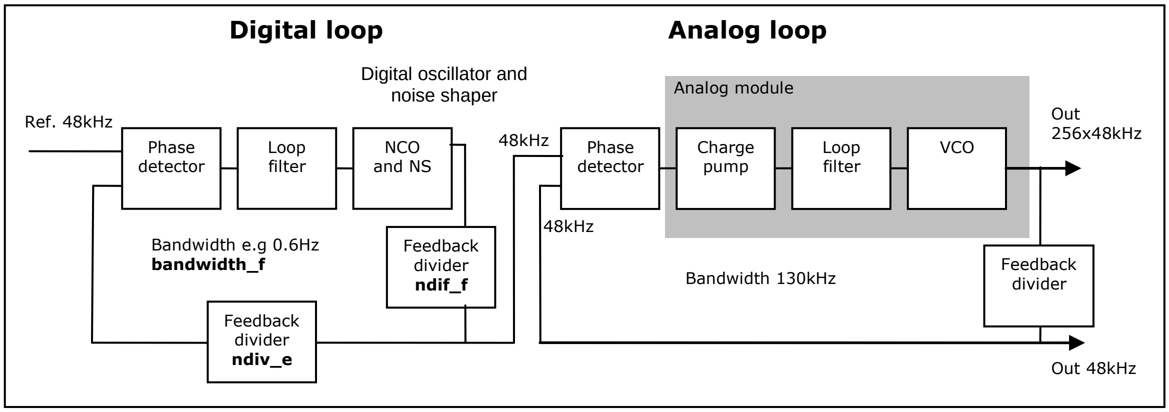


Figure 37: Simple block diagram of Jet[TM] PLL including location of dividers.

5.3.3 Basic registers

The Jet^{TM} PLL is handled by control registers and status registers.

5.3.4 Control registers

In the control registers the Jet^{TM} PLL can be set to lock to different reference frequencies and generate different output frequencies.

The Jitter rejection bandwidth can be set to different frequencies, e.g. 1 Hz or 100Hz. The lock response time is somewhat related to this.

Name	Bit	Reset	Dir	Description

register / field name	r, w *	addr. hex	Value decimal	Width bits	brief description
bandwidth_f	W	‘h60	‘d4	4	Loop bandwidth floor. Enforced while the loop is locked.
bandwidth_c	W	‘h64	‘d9	4	Loop bandwidth ceiling. Has a role in acquisition mode.
ndiv_f	W	‘hA0	’d255	11	Controls NL Divider stage F ('frame_b').
ndiv_e	W	‘hB4	’d0	12	Controls NL Divider stage E ('fbk_event').

Figure 38: Basic Jet^{TM} PLL registers

5.3.5 Status Registers

At address …308hex the Jet^{TM} PLL is presenting its main status register. The 2 least significant bits are presented here:

Name	Bit	Reset	Dir	Description
Unlocked	1	0	R	Triggered if phase offset wraps or exceeds ‘u_threshold’.
Ref_flawed	0	0	R	Triggered by reference discontinuities. Hi when auto coasting.

Figure 39: Jet^{TM} PLL Status Registers

The register read value “unlocked” going high means that the Jet^{TM} PLL is unlocked. The ref_flawed is triggered when the reference is discontinuous e.g. by being disconnected. If e.g. an AES input (id#) feeds the Jet^{TM} PLL, ‘unlocked’ bit will go high as well when the AES receiver is unlocked.

5.3.6 Frequency Reconstruction Generation.

When locking to frequency range 30 – 50 kHz the Jet^{TM} PLL should be set to lock to 1Fs. This is done by setting the divider ndiv_f to 255dec and ndiv_e to 0dec. When locking to 60 – 100 kHz range the divider ndiv_f should be set to 127dec and ndiv_e to 0dec.

When locking to a 1394 ISOC stream the Jet^{TM} PLL should refer to the SYT_match signal which is a sub 8KHz rate linked to the sample rate. Sample rates of 44.1kHz and 88.2kHz respectively both have a SYT_match signal of 5.513kHz = 44100/8 = 88200/16. This means that the ndiv_e should be set to 8dec if the sample rate is 44.1kHz and 16dec if the sample rate is 88.2kHz.

ndiv_f is still 255dec for 44.1kHz and 127dec for 88.2kHz. This means that the ndiv_f and ndiv_e dividers should be set to 255dec and 8dec for sample rates 30 – 50 kHz and 127 and 16 for samples rates 60 – 100kHz respectively.

When generating internal master sample rates on DICE JR the prescalers (outside the Jet^{TM} PLL modules) can be used together with the ndiv_e divider. This way a 49.152MHz (2 times the reference 24.576MHz XTAL at the 1394 phy chip) can be used to generate multiple extremely low jitter extremely high precision internal sample rates using only one XTAL.

The formula used is: FS = 49.152MHz / 'Prescaler' * 'ndiv_e divider'

Required FS (Hz)		Prescaler (32bit)	ndiv_e divider (12bit)	Actual FS (Hz)	Deviation (ppm)
44100		* 491520	441	44100	0
44144,1	(44.1k + 0.1%)	10021	9	44144,0974	0,059
44055,9	(44.1k - 0.1%)	51321	46	44055,88356	0,3732
45864	(44.1k + 4%)	76090	71	45864,00315	0,0688
42336	(44.1k - 4%)	* 512000	441	42336	0
44283,75	(44.1k + 4.1666%)	16649	15	44283,74077	0,2085
42262,5	(44.1k - 4.1666%)	68618	59	42262,49672	0,0776
48000		1024	1	48000	0
48048	(48k + 0.1%)	45011	44	48047,98827	0,2441
47952	(48k - 0.1%)	41001	40	47952,00117	0,0244
49920	(48k + 4%)	12800	13	49920	0
46080	(48k - 4%)	3200	3	46080	0
50000	(48k + 4.1666%)	24576	25	50000	0
46000	(48k - 4.1666%)	24576	23	46000	0

Figure 40: Internal sampling rates generated with the Jet^{TM} PLL.

* When generating internal rates with great precision the register bandwidth_c should be set to 2dec. This will ensure stability in the Jet^{TM} PLL while the reference is a very low frequency e.g 100Hz – the very low reference frequency is the key for the high precision on the resulting internal sample rates.

5.3.7 Jitter Transfer Function JTF, BW and peaking.

The bandwidth in the Jet^{TM} PLL can be set by firmware. This is the -3dB frequency of the jitter rejection low pass filter. It can be set to 0hex to Fhex, which corresponds to approx. 0.1Hz to 2.8kHz in steps of an octave.

bandwidth_f	Hz
00hex	0.085
01hex	0.17
02hex	0.34
03hex	0.68
04hex	1.4
05hex	2.7

06hex	5.5
07hex	10.9
08hex	21.9
09hex	43.8
0Ahex	87.5
0Bhex	175
0Chex	350
0Dhex	700
0Ehex	1400
0Fhex	2800

Figure 41: Jitter rejection bandwidth set in the Jet^{TM} PLL.

The -60dB frequency is at approx 13 times the -3dB frequency. The roll off is of 4th order approaching a rejection ability that increases by 80dB/decade.

Peaking is maximum 1.5dB at a frequency approx ¼ of the -3dB point.

5.3.8 Jet^{TM} PLL Performance

- Frequency range: 15.8 MHz to 27.7 MHz (scalable)
- Jitter attenuation: more than 60 dB above 100 Hz
- Period jitter: less than 50ps RMS
- Wideband jitter: less than 200ps RMS (100 Hz highpass)
- Baseband jitter: less than 20ps RMS (100 Hz to 40 kHz)
- Jitter density: less than 0.1ps/rootHz above 100 Hz

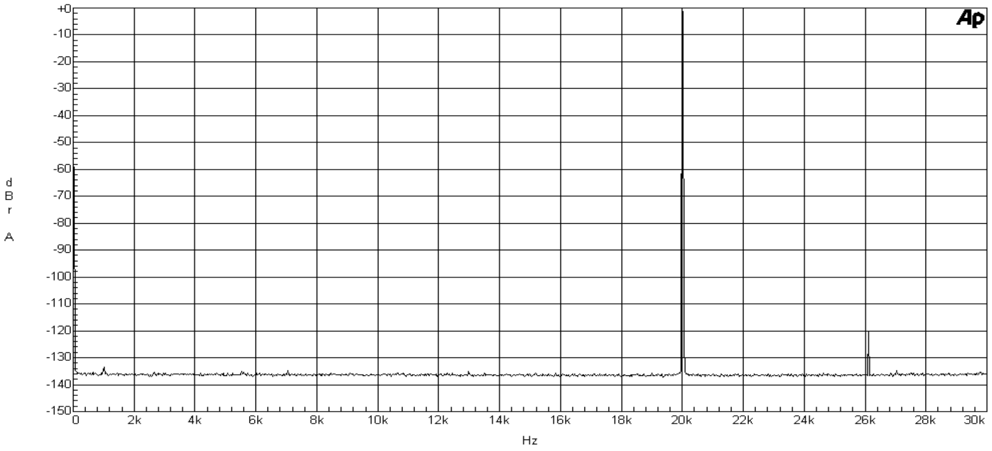


Figure 42: Resulting spectrum when converting a 20 kHz audio tone. The spectrum looks the same with and without incoming jitter being removed.

5.3.9 Jet^{TM} PLL Registers

The Jet^{TM} PLL registers are categorized as Basic or Advanced. Application firmware typically reads and writes the values of the basic registers. Firmware provided by TC

Applied Technologies in the DICE 22xx SDK handles these registers and provides a simple API that hides the complexity. **To ensure a reliable operation the advanced registers should remain untouched.**
All hex addresses should be multiplied by 4 to get the real address.

5.3.9.1 Control registers:

register / field name	r, w *	Adr. hex	Value decimal	Width bits	brief description
caf_enable	W	`h00	1	1	Internal. Initialized by constant in the basic SW.
caf_select	W	`h01	0	2	Internal. Initialized by constant in the basic SW.
coast	W	`h02	0	1	Internal. Initialized by constant in the basic SW.
ref_select	W	`h06	1	5	Internal. Initialized by constant in the basic SW.
ref_edges	W	`h07	0	2	Internal. Initialized by constant in the basic SW.
rdiv	W	`h0A	0	16	Internal. Initialized by constant in the basic SW.
throttle_r	W	`h0B	0	1	Internal. Initialized by constant in the basic SW.
gravity	w	`h11	1	1	Internal. Initialized by constant in the basic SW.
u_threshold	W	`h16	100	8	Internal. Initialized by constant in the basic SW.
bandwidth_f	W	`h18	4	4	Loop bandwidth floor. Enforced while the loop is locked.
bandwidth_c	W	`h19	9	4	Loop bandwidth ceiling. Has a role in acquisition mode.
shape_f	W	`h1A	0	2	Internal. Initialized by constant in the basic SW.
shape_v	W	`h1B	3	2	Internal. Initialized by constant in the basic SW.
max_slew_f	W	`h1C	15	4	Internal. Initialized by constant in the basic SW.
Max_slew_v	W	`h1D	15	4	Internal. Initialized by constant in the basic SW.
descent_lin	W	`h1E	4	3	Internal. Initialized by constant in the basic SW.
descent_exp	W	`h1F	4	3	Internal. Initialized by constant in the basic SW.
loose_thr	W	`h22	10	8	Internal. Initialized by constant in the basic SW.
Min_period	W	`h26	58	8	Internal. Initialized by constant in the basic SW.
Max_period	W	`h27	111	8	Internal. Initialized by constant in the basic SW.
ndiv_f	W	`h2C	255	11	Controls NL Divider stage F ('frame_b').
ndiv_e	W	`h2D	0	12	Controls NL Divider stage E ('fbk_event').
ndiv_b	W	`h2E	1	7	Internal. Initialized by constant in the basic SW.
bypass_f	W	`h2F	0	1	Internal. Initialized by constant in the basic SW.
phase_lag	W	`h30	0	11	Internal. Initialized by constant in the basic SW.
fract_res	W	`h32	1	2	Internal. Initialized by constant in the basic SW.
burst_len	W	`h34	3	6	Internal. Initialized by constant in the basic SW.
gpo_grant	W	`h36	0	3	Not available in TCD22xx
x1_gpo	W	`h37	0	2	Not available in TCD22xx

register / field name	r, w *	Adr. hex	Value decimal	Width bits	brief description
x2_gpo	W	`h38	0	2	Not available in TCD22xx
x3_gpo	W	`h39	0	2	Not available in TCD22xx
x1x2_mode	W	`h3C	1	3	Internal. Initialized by constant in the basic SW.
chain_i	W	`h40	0	1	Internal. Initialized by constant in the basic SW.
sink_i	W	`h41	0	1	Internal. Initialized by constant in the basic SW.
anchor_i	W	`h42	0	1	Internal. Initialized by constant in the basic SW.
i_anc_val	W	`h43	4	5	Internal. Initialized by constant in the basic SW.
unbind_i	W	`h44	0	1	Internal. Initialized by constant in the basic SW.
idet	W	`h46	0	1	Internal. Initialized by constant in the basic SW.
idiv_c	W	`h48	1	3	Internal. Initialized by constant in the basic SW.
idiv_f	W	`h49	511	13	Internal. Initialized by constant in the basic SW.
idiv_s	W	`h4A	3	4	Internal. Initialized by constant in the basic SW.
invert_cdi	W	`h4C	0	1	Internal. Initialized by constant in the basic SW.
hobble_cdi	W	`h4D	0	1	Internal. Initialized by constant in the basic SW.
sink_e	W	`h51	0	1	Internal. Initialized by constant in the basic SW.
anchor_e	W	`h52	0	1	Internal. Initialized by constant in the basic SW.
e_anc_val	W	`h53	4	5	Internal. Initialized by constant in the basic SW.
unbind_e	W	`h54	0	1	Internal. Initialized by constant in the basic SW.
edet_x1	W	`h56	0	4	Internal. Initialized by constant in the basic SW.
edet_x2	W	`h57	0	4	Internal. Initialized by constant in the basic SW.
ediv_c	W	`h58	1	3	Internal. Initialized by constant in the basic SW.
ediv_f	W	`h59	511	13	Internal. Initialized by constant in the basic SW.
ediv_s	W	`h5A	3	4	Internal. Initialized by constant in the basic SW.
invert_cde	W	`h5C	0	1	Internal. Initialized by constant in the basic SW.
hobble_cde	W	`h5D	0	1	Internal. Initialized by constant in the basic SW.
divide_cj	W	`h60	0	1	Internal. Initialized by constant in the basic SW.
invert_cj	W	`h61	0	1	Internal. Initialized by constant in the basic SW.
config_ac	W	`hF8	1	2	Internal. Initialized by constant in the basic SW.
shutdown_m	W	`hFC	1	1	Internal. Initialized by constant in the basic SW.
shutdown_i	W	`hFD	0	1	Internal. Initialized by constant in the basic SW.
shutdown_e	W	`hFE	1	1	Internal. Initialized by constant in the basic SW.

register / field name	r, w *	Adr. hex	Value decimal	Width bits	brief description

Table 48: Jet^{TM} PLL Control Register Map

5.3.9.2 Status registers

register / field name	r, w *	Adr. hex	Value decimal	Width bits	brief description
family_id	R	`hA0	“SB”	16	Internal. Not used in the basic SW.
form_id	R	`hA1	“A ”	16	Internal. Not used in the basic SW.
revision_id	R	`hA2	“ 0”	16	Internal. Not used in the basic SW.
instance_id	R	`hA3	?*	16	Internal. Not used in the basic SW.
mtr_select	W	`hAE	0	5	Internal. Not used in the basic SW.
mtr_edges	W	`hAF	0	2	Internal. Not used in the basic SW.
res_ex	W	`hB0	7	4	Internal. Not used in the basic SW.
punc_mp	W	`hB1	-*	1	Internal. Not used in the basic SW.
mtr_period	R	`hB3	-	16	Internal. Not used in the basic SW.
greatest_mp	R	`hB4	-	16	Internal. Not used in the basic SW.
greatest_mp_$	R$	`hB5	-	16	Internal. Not used in the basic SW.
smallest_mp	R	`hB6	-	16	Internal. Not used in the basic SW.
smallest_mp_$	R$	`hB7	-	16	Internal. Not used in the basic SW.
tick_rate	W	`hC0	3	4	Internal. Not used in the basic SW.
turn_rate	W	`hC1	0	1	Internal. Not used in the basic SW.
main_status	R	`hC2	-	16	Internal. Not used in the basic SW.
main_status_$	R$	`hC3	-	16	Internal. Not used in the basic SW.
detect_r	W	`hC8	0	16	Internal. Not used in the basic SW.
detect_f	W	`hC9	0	16	Internal. Not used in the basic SW.
sticky_bits	RW*	`hCA	-	16	Internal. Not used in the basic SW.
sticky_bits_$	R$	`hCB	-	16	Internal. Not used in the basic SW.
irq_enables	W	`hD4	0	16	Internal. Not used in the basic SW.
nco_period	R	`hE3	-	16	Internal. Not used in the basic SW.
greatest_np	R	`hE4	-	16	Internal. Not used in the basic SW.
greatest_np_$	R$	`hE5	-	16	Internal. Not used in the basic SW.
smallest_np	R	`hE6	-	16	Internal. Not used in the basic SW.
smallest_np_$	R$	`hE7	-	16	Internal. Not used in the basic SW.
gpi	R	`hF6	-	1	For reading one or more pins as gen-purpose inputs.

Table 49: Jet^{TM} PLL Status Memory map

5.3.9.3 Main_status

bit	name	stretched?	brief description
15	turn	no	Internal. Not used in the basic SW.
14	gmp_over	no	Internal. Not used in the basic SW.
13	gathered	no	Internal. Not used in the basic SW.
12	mp_flushed	no	Internal. Not used in the basic SW.
11	(Unspecified.)	-	Internal. Not used in the basic SW.
10	slew_is_max	yes	Internal. Not used in the basic SW.
9	period_is_max	yes	Internal. Not used in the basic SW.
8	period_is_min	yes	Internal. Not used in the basic SW.
7	e_shaky	yes	Internal. Not used in the basic SW.
6	e_slipping	yes	Internal. Not used in the basic SW.
5	i_shaky	yes	Internal. Not used in the basic SW.
4	i_slipping	yes	Internal. Not used in the basic SW.
3	loose	yes	Internal. Not used in the basic SW.
2	varying	yes	Internal. Not used in the basic SW.
1	unlocked	yes	Triggered if phase offset wraps or exceeds ‘u_threshold’.
0	ref_flawed	yes	Triggered by reference discontinuities. Hi when auto coasting.

5.3.9.4 Module Configuration

Address	Register
0xcc00 0000	PLL1_CAF_ENABLE
0xcc00 0004	PLL1_CAF_SELECT
0xcc00 0008	PLL1_COAST
0xcc00 0018	PLL1_REF_SEL
0xcc00 001c	PLL1_REF_EDG
0xcc00 0028	PLL1_RDIV
0xcc00 002c	PLL1_THROTTLE
0xcc00 0058	PLL1_U_THRESHOLD
0xcc00 0060	PLL1_BW_FLOOR
0xcc00 0064	PLL1_BW_CEILING
0xcc00 0068	PLL1_SHP_FIX
0xcc00 006c	PLL1_SHP_VAR
0xcc00 0070	PLL1_MAX_SLW_FIX
0xcc00 0074	PLL1_MAX_SLW_VAR

Address	Register
0xcc00 0078	PLL1_DCNT_LIN
0xcc00 007c	PLL1_DCNT_EXP
0xcc00 0088	PLL1_LOOSE_THR
0xcc00 0098	PLL1_MIN_PER
0xcc00 009c	PLL1_MAX_PER
0xcc00 00b0	PLL1_NDIV_F
0xcc00 00b4	PLL1_NDIV_E
0xcc00 00b8	PLL1_NDIV_B
0xcc00 00bc	PLL1_BYP_F
0xcc00 00c0	PLL1_PHASE_LAG
0xcc00 00c8	PLL1_FRACT_RES
0xcc00 00d0	PLL1_BURST_LEN
0xcc00 00d8	PLL1_GPO_EN
0xcc00 00dc	PLL1_GPO_1
0xcc00 00e0	PLL1_GPO_2
0xcc00 00e4	PLL1_GPO_3
0xcc00 00f0	PLL1_X1X2_MODE
0xcc000100	PLL1_CHAIN_I
0xcc000104	PLL1_SINK_I
0xcc000108	PLL1_ANCHOR_I
0xcc00010c	PLL1_IANCHOR_VAL
0xcc000110	PLL1_UNBND_I
0xcc000118	PLL1_IDET
0xcc000120	PLL1_IDIV_C
0xcc000124	PLL1_IDIV_F
0xcc000128	PLL1_IDIV_S
0xcc000130	PLL1_INV_CDI
0xcc000134	PLL1_HBL_CDI
0xcc000144	PLL1_SINK_E
0xcc000148	PLL1_ANCHOR_E
0xcc00014c	PLL1_E_ANC_VAL
0xcc000150	PLL1_UNBIND_E
0xcc000158	PLL1_EDET_X1
0xcc00015c	PLL1_EDET_X2
0xcc000160	PLL1_EDIV_C
0xcc000164	PLL1_EDIV_F
0xcc000168	PLL1_EDIV_S
0xcc000170	PLL1_INV_CDE
0xcc000174	PLL1_HBL_CDE

Address	Register
0xcc000180	PLL1_DIVIDE_CJ
0xcc000184	PLL1_INVERT_CJ
0xcc000280	PLL1_FAMILY_ID
0xcc000284	PLL1_FORM_ID
0xcc000288	PLL1_REVISION_ID
0xcc00028c	PLL1_INSTANCE_ID
0xcc0002b8	PLL1_MTR_SELECT
0xcc0002bc	PLL1_MTR_EDGES
0xcc0002c0	PLL1_RES_EX
0xcc0002c4	PLL1_PUNC_MP
0xcc0002cc	PLL1_MTR_PERIOD
0xcc0002d0	PLL1_GREATEST_MP
0xcc0002d4	PLL1_GREATEST_MP_$
0xcc0002d8	PLL1_SMALLEST_MP
0xcc0002dc	PLL1_SMALLEST_MP_$
0xcc000300	PLL1_TICK_RATE
0xcc000304	PLL1_TURN_RATE
0xcc000308	PLL1_MAIN_STATUS
0xcc00030c	PLL1_MAIN_STATUS_$
0xcc000320	PLL1_DETECT_R
0xcc000324	PLL1_DETECT_F
0xcc000328	PLL1_STICKY_BITS
0xcc00032c	PLL1_STICKY_BITS_$
0xcc000350	PLL1_IRQ_ENABLES
0xcc00038c	PLL1_NCO_PERIOD
0xcc000390	PLL1_GREATEST_NP
0xcc000394	PLL1_GREATEST_NP_$
0xcc000398	PLL1_SMALLEST_NP
0xcc00039c	PLL1_SMALLEST_NP_$
0xcc0003d8	PLL1_GPI
0xcc0003e0	PLL1_CONFIG_AC
0xcc0003f0	PLL1_SHUTDOWN_M
0xcc0003f4	PLL1_SHUTDOWN_I
0xcc0001f8	PLL1_SHUTDOWN_E

Table 50: Jet^{TM} PLL Memory Map

5.4 AES Receivers

The DICE TCD22xx contains 4 independent, fully compliant AES/EBU Receivers. The main features of these receivers are:

- Handling / buffering (4 layers) of up to 8 channels of audio and control data.
- Handling of CS/USER bits through both memory-mapping and AM824 format. First 4 bytes of CS from each channel and one full block of Channel Status from a selected channel can be accessed by the ARM. User bits from all 8 channels are serially output and can be routed to relevant transmitter modules.
- Slipped sample detection in case of differences in clock rate of the incoming data compared to the clock of the interfacing system
- Slipped sample detection in case of phase/frequency differences between the AES input chosen to be clock master and other AES inputs.
- Memory-mapped error/lock indication.

5.4.1 Signal Description

AES receives data from AUDIO ports depending on configuration (see 4.1.1)

5.4.2 Module Configuration

Address	Register
0xce02 0000	CTRL
0xce02 0004	STAT_ALL
0xce02 0008	STAT_RX0
0xce02 000c	STAT_RX1
0xce02 0010	STAT_RX2
0xce02 0014	STAT_RX3
0xce02 0018	V_BIT
0xce02 0040	PLL_PULSE_WIDTH
0xce02 0044	FORCE_VCO
0xce02 0048	VCO_MIN_LSB
0xce02 004c	VCO_MIN_MSB
0xce02 0080	CHSTAT_0_BYTE0
0xce02 0084	CHSTAT_0_BYTE1
0xce02 0088	CHSTAT_0_BYTE2
0xce02 008c	CHSTAT_0_BYTE3
0xce02 0090 - 0xce02 009c	CHSTAT_1_BYTE0-3
0xce02 00a0 - 0xce02 00ac	CHSTAT_2_BYTE0-3
0xce02 00b0 - 0xce02 00bc	CHSTAT_3_BYTE0-3
0xce02 00c0 - 0xce02 00cc	CHSTAT_4_BYTE0-3
0xce02 00d0 - 0xce02 00dc	CHSTAT_5_BYTE0-3
0xce02 00e0 - 0xce02 00ec	CHSTAT_6_BYTE0-3

Address	Register
0xce02 00f0 - 0xce02 00fc	CHSTAT_7_BYTE0-3
0xce02 0100 – 0xce02 015c	CHSTAT_FULL_BYTE0-23

Table 51: AES Receiver Memory Map

Note that all registers are 8 bits wide aligned to 32 bit word addresses. The upper 24 bits of the data will be ignored and will be read as ‘0’.

5.4.3 CTRL

Address - 0xce02 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								MASTER		Reserved		Dual Wire	CSCH		
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	R	R	R	RW	RW	RW

Name	Bit	Reset	Dir	Description
MASTER	7:6	0	RW	Selects the master receiver.
DUAL WIRE	3	0	RW	1’b0 – Single wire mode 1’b1 – Dual wire mode
CSCH	2:0	0	RW	Selects the channel to receive full Channel Status from.

5.4.4 STAT_ALL

Address - 0xce02 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								OU_R	U_R	O_R	Reserved				
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
OU_R	7	0	R	An or’ed function of O_R and U_R and therefore indicates that a slipped sample or resampling condition has occurred.
U_R	6	0	R	Indicates resampling that typically happen when the system 1FS is faster than 1FS from the master receiver. Can also be due to jitter and phase differences between the router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.
O_R	5	0	R	Indicates slipped sample which typically happen when the system 1FS is slower than 1FS from the master receiver. Can also be due to jitter and phase differences between router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.

5.4.5 STAT_RXn

Address - 0xce02 0008 - 0xce02 0014

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								OU_R	U_R	O_R	VAL	PRTY	CRC	NLOCK	LOCK
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
OU_R	7	0	R	An or'ed function of O_R and U_R and therefore indicates that a slipped sample or resampling condition has occurred.
U_R	6	0	R	Indicates resampling from a specific receiver. Typically caused by jitter and phase differences between the failing receiver and the master receiver. Can also be due to a very small difference in sample rate between failing receiver and the master receiver. This bit is sticky and will be cleared immediately after a read.
O_R	5	0	R	Indicates slipped sample from a specific receiver. Typically caused by jitter and phase differences between the failing receiver and the master receiver. Can also be due to a very small difference in sample rate between failing receiver and the master receiver. This bit is sticky and will be cleared immediately after a read.
VAL	4	0	R	Indicates that the v bit has been detected as a 1 in either of the channels in the receiver since last time the register was read. This bit is sticky and will be cleared immediately after a read.
PRTY	3	0	R	Indicates that a parity error has been detected in either of the channels in the receiver since last time the register was read. This bit is sticky and will be cleared immediately after a read.
CRC	2	0	R	Indicates that a crc error has been detected in either of the channels in Receiver X since last time the register was read. This bit is sticky and will be cleared immediately after a read.
NLOCK	1	0	R	Indicates that Receiver X has been out of lock since last time the register was read. This bit is sticky and will be cleared immediately after a read.
LOCK	0	0	R	Indicates whether or not the receiver is currently locked.

5.4.6 V_BIT

Address - 0xce02 0018

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									V_BIT							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
V_BIT	7:0	0	RW	V-bits for AES channels 7-0

5.4.7 PLL_PULSE_WIDTH

Address - 0xce02 0040

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								UP_PULSE_WIDTH				DOWN_PULSE_WIDTH			

Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
UP_PULSE_WIDTH	7:4	0	RW	Up Pulse Width 1 – 16 cycles wide. Sets the width of the up signal when receiver is out of lock.
DOWN_PULSE_WIDTH	3:0	0	RW	Down Pulse Width 1 – 16 cycles wide. Sets the width of the down signal when receiver is out of lock.

5.4.8 FORCE_VCO

Address - 0xce02 0044

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														FORCE _UP	FORCE _DOW N
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
FORCE_UP	1	0	RW	AES VCO force up.
FORCE_DOWN	0	0	RW	AES VCO force down.

5.4.9 VCO_MIN_LSB

Address - 0xce02 0048

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								MIN_FREQ_LSB							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
MIN_FREQ_LSB	7:4	0	RW	8 LSB’s for setting minimum frequency on VCO. Minimum VCO Sample frequency (1FS) = ARM System clock (typ. 49.152MHz) / [MSB,LSB]

5.4.10 VCO_MIN_MSB

Address - 0xce02 004c

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								MIN_FREQ_MSB							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
MIN_FREQ_MSB	3:0	0	RW	8 MSB's for setting minimum frequency on VCO. Minimum VCO Sample frequency (1FS) = ARM System clock (typ. 49.152MHz) / [MSB,LSB]

5.4.11 CHSTAT_n_BYTE0-3

0xce02 0080 - 0xce02 008c

The four bytes represents the first 32 bits of channel status for channel n. BYTE0 bit 0 corresponds to CS bit 0 and BYTE3 bit 7 corresponds to CS bit 31.

5.4.12 CHSTAT_FULL_BYTE0-23

0xce02 0100 - 0xce02 015c

The 24 bytes represent the full 192 bits of channel status for the channel selected by CSCH in the CTRL register. BYTE0 bit 0 corresponds to CS bit 0 and BYTE23 bit 7 corresponds to CS bit 191.

5.5 AES Transmitters

The DICE TCD22xx contains 4 independent, fully compliant AES/EBU transmitters. The main features of these transmitters are:

- Sampling and buffering of up to 8 channels of audio to be transmitted at a common sample rate.
- Transmission of Channel Status (CS) bits from either memory mapped bits (master mode) or through AM824 frames (slave mode).
- Individual setting for each channel of first 4 bytes of Channel Status (master mode)
- U bit directly sourced from a selected AES Receiver channel (master mode) or from AM824 frames (slave mode).

For a given channel, block sync can be configured to be generated internally (free running), or synchronized to an external source. Note that the particular external source is selected by the BLKCTRL register described in the section titled Clock Controller. Block sync is used as the synchronization signal for the CS information, defining the beginning and end of each CS block.

5.5.1 Signal Description

AES transmits through AUDIO Ports, for usage & configuration please see 4.1.2

5.5.2 Module Configuration

Address	Register
0xce03 0000	MODE_SEL
0xce03 0004	CBL_SEL
0xce03 0008	CS_SEL0
0xce03 000c	CS_SEL1
0xce03 0010	CS_SEL2
0xce03 0014	MUTE
0xce03 0018	V_BIT
0xce03 0040	USR_SEL0
0xce03 0044	USR_SEL1
0xce03 0048	USR_SEL2
0xce03 004c	USR_SEL3
0xce03 0080	CHSTAT_0_BYTE0
0xce03 0084	CHSTAT_0_BYTE1
0xce03 0088	CHSTAT_0_BYTE2
0xce03 008c	CHSTAT_0_BYTE3
0xce03 0090 - 0xce03 009c	CHSTAT_1_BYTE0-3
0xce03 00a0 - 0xce03 00ac	CHSTAT_2_BYTE0-3
0xce03 00b0 - 0xce03 00bc	CHSTAT_3_BYTE0-3
0xce03 00c0 - 0xce03 00cc	CHSTAT_4_BYTE0-3
0xce03 00d0 - 0xce03 00dc	CHSTAT_5_BYTE0-3

Address	Register
0xce03 00e0 - 0xce03 00ec	CHSTAT_6_BYTE0-3
0xce03 00f0 - 0xce03 00fc	CHSTAT_7_BYTE0-3
0xce03 0100 – 0xce03 015c	CHSTAT_FULL_BYTE0-23

Table 52: AES Transmitter Memory Map

5.5.3 MODE_SEL

Address - 0xce03 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CRC4	CRC3	CRC2	CRC1	Reserved		Dual Wire	MSTR
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R	R	R	RW

Name	Bit	Reset	Dir	Description
CRC4	7	0	RW	Enables auto CRC for transmitter 4. 0: Auto CRC Enabled 1: Auto CRC Disabled
CRC3	6	0	RW	Enables auto CRC for transmitter 3. 0: Auto CRC Enabled 1: Auto CRC Disabled
CRC2	5	0	RW	Enables auto CRC for transmitter 2. 0: Auto CRC Enabled 1: Auto CRC Disabled
CRC1	4	0	RW	Enables auto CRC for transmitter 1. 0: Auto CRC Enabled 1: Auto CRC Disabled
Dual Wire	1	0	RW	Dual Wire Mode Enable/Disable 0: Disabled 1: Enabled
MSTR	0	0	RW	Selects the transmitter mode. Refer to 5.5 0: Master (only 24 bits from the audio stream is used) 1: Slave (upper bits can be used for sync, U, C and V)

5.5.4 CBL_SEL

Address - 0xce03 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CBL_MSTR				CBL_SLAVE			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
CBL_MSTR	7:4	0	RW	Selects the Block Sync source in master mode. 0000: Internal CBL (free running) 0001: External CBL (Selected in Clock Controller) xxxx: All other values are reserved
CBL_SLAVE	3:0	0	RW	Selects the Block Sync in slave mode. The block sync is extracted from the AM824 defined frame (bit 29). 0000: CBL from audio channel 0 0001: CBL from audio channel 1 0010: CBL from audio channel 2 0011: CBL from audio channel 3 0100: CBL from audio channel 4 0101: CBL from audio channel 5 0110: CBL from audio channel 6 0111: CBL from audio channel 7 xxxx: All other values are reserved

5.5.5 CS_SEL0

Address - 0xce03 0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CH3		CH2		CH1		CH0	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
CH3	7:6	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 3 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH2	5:4	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 2 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH1	3:2	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 1 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved

Name	Bit	Reset	Dir	Description
CH0	1:0	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 0 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved

5.5.6 CS_SEL1

Address - 0xce03 000c

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CH7		CH6		CH5		CH4	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
CH7	7:6	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 7 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH6	5:4	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 6 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH5	3:2	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 5 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH4	1:0	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 4 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved

5.5.7 CS_SEL2

0xce03 0010

When the memory mapped Channel Status mode is selected for a channel, the first 4 channel status bytes (bit 0-31) can come from either a common memory mapped file or an individual per channel file. The remaining 20 bytes (bit 32-192) are always defined by the common file.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
CH7	7	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH6	6	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH5	5	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH4	4	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH3	3	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH2	2	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH1	1	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file
CH0	0	0	RW	Selects Memory mapped CS source for bit 0-31. 0: Individual file. 1: Common file

5.5.8 MUTE

Address - 0xce03 0014

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								MUTE 0 - 7							
Reset:	0								0xFF							
	RW								RW							

Name	Bit	Reset	Dir	Description
MUTE 0 - 7	7:0	0	RW	One bit for each channel selects whether the audio should be muted (all bits zero) or not. Default configuration after reset is that all channels are muted. **0** = Channel not muted. **1** = Channel muted.

5.5.9 V_BIT

Address - 0xce03 0018

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								V_BIT 0 - 7							
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
V_BIT 0 - 7	7:0	0	RW	One bit for each channel selects whether the V bit should indicate valid audio (V=0) or invalid audio (V=1). Default configuration after reset is to indicate valid audio. **0** = Audio valid. **1** = Audio invalid.

5.5.10 USR_SEL0

Address - 0xce03 0040

In master mode the user bits can be programmed to 0 or be sourced from any of the 8 AES receiver channels. The U bits are not block aligned and are considered a raw bit stream.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								USR1				USR0			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
USR1	7:4	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = “000” => Rx channel 0, ..., nnn = “111” => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = “00” => AVS Rx 0, ..., nn = “11” => AVS Rx 3). **11xx =** USER bit set to 0
USR0	3:0	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = “000” => Rx channel 0, ..., nnn = “111” => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = “00” => AVS Rx 0, ..., nn = “11” => AVS Rx 3). **11xx =** USER bit set to 0

5.5.11 USR_SEL1

Address - 0xce03 0044

In master mode the user bits can be programmed to 0 or be sourced from any of the 8 AES receiver channels. The U bits are not block aligned and are considered a raw bit stream.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								USR3				USR2			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
USR3	7:4	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0
USR2	3:0	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0

5.5.12 USR_SEL2

Address - 0xce03 0048

In master mode the user bits can be programmed to 0 or be sourced from any of the 8 AES receiver channels. The U bits are not block aligned and are considered a raw bit stream.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								USR5				USR4			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
USR5	7:4	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0
USR4	3:0	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0

5.5.13 USR_SEL3

Address - 0xce03 004c

In master mode the user bits can be programmed to 0 or be sourced from any of the 8 AES receiver channels. The U bits are not block aligned and are considered a raw bit stream.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								USR7				USR6			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
USR7	7:4	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0
USR6	3:0	0	RW	4 bits for each channel selects the USER bit source. Slave mode: **0xxx** = USER bit from Audio register file (same as audio data). **1xxx =** USER bit set to 0 Master mode: **0nnn** = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => Rx channel 7). **10nn =** USER bit from AVS Receiver nn (nn = "00" => AVS Rx 0, ..., nn = "11" => AVS Rx 3). **11xx =** USER bit set to 0

5.5.14 CHSTAT_n_BYTE0-3

Address - 0xce03 0080 - 0xce03 008c

The four bytes represents the first 32 bits of channel status for channel n. BYTE0 bit 0 corresponds to CS bit 0 and BYTE3 bit 7 corresponds to CS bit 31.

5.5.15 CHSTAT_FULL_BYTE0-23

Address - 0xce03 0100 - 0xce03 015c

The 24 bytes represents the full 192 bits of channel status. In memory mapped CS mode the last 20 bytes are always used for all channels. Usage of the first 4 bytes depends on the setting of the CS_SEL3 register. BYTE0 bit 0 corresponds to CS bit 0 and BYTE23 bit 7 corresponds to CS bit 191.

5.5.16 Slave Mode

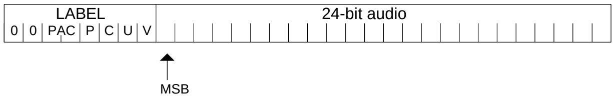
In order to have full control of the extra data send in an AES sub-frame, two basic modes are provided. The Parity bit and the Left/Right sub-frame bits are always calculated by the transmitter.

Master Mode:

In this mode the upper 8 bits of the incoming audio data are not used. The sync pattern is generated in the transmitter and transmission thereof can optionally be synced to the block sync signal from the DICE Clock Controller. The Channel Status bits are taken from the memory files. The first 4 bytes of CS can be specified individually per channel. The remaining bytes are specified in a common file.

Slave Mode:

In this mode the upper 8 bits of the audio data are used. The bits are interpreted using the AM824 specification.



The block sync is generated from the PAC bits in the audio stream. All 4 transmitters are aligned to the block sync in the audio channel selected by the CBL_SEL register. The User bits are sourced from the U bit in the corresponding audio frame or set to zero. The validity bits are sourced from the V bit in the corresponding audio frame. The Channel Status bits are either sourced from the C bit in the corresponding audio frame, the C bit in the audio frame selected for block sync or from the register files.

5.6 I^{n}S Transmitters

5.6.1 System view

These are highly configurable serial audio interface transmitters that can be set to comply with a number of different formats, ensuring compatibility with most DAC's and SRC's as well as other serial audio devices. The DICE JR (TCD2220) contains two I^{n}S transmitter modules, the DICE Mini (TCD2210) contains one, each consists of 4 Tx instances. Each module is connected to the DICE Router.

Each Tx module is capable of receiving up to 16 channels, depending on configuration & system speed. Depending on the configuration of the 4 Tx instancies all 16 channels might not be used. An example is when all 4 modules are configured for I2S. In that case only 8 channels are used.

Audio ch	I2S	I4S	I8S
0	1	1	1
1			
2	2		
3			
4	3	2	
5			
6	4		
7			
8	N/A	3	2
9			
10			
11			
12		4	
13			
14			
15			

Table 53: InS instance to channel mapping

It is also obvious from the above that the last two instances can't be configured for I8S. It is considered an illegal configuration and in that case for the Tx module, only muted audio should be delivered and in the case of the Rx module the received data should be ignored.

In addition each of the 16 channels in Tx module could be muted by software, using the "mute" register located in each of the two Tx modules

Each Tx instance can work in 3 different modes which is 2, 4 or 8 channels per frame. In high rate mode 8 channels per frame is not possible.

Name	Ch/frame	Bits per frame

Name	Ch/frame	Bits per frame
I2S	2	64
I4S	4	128
I8S	8	256

Table 54: InS modes

InS communication is frame based and driven by sys_f512br. This signal has a fixed relation to the frame clock which depends on the system rate_mode, sys_mode. Number of 512br clocks in each frame in different sytem modes is described Table 55

Name	Freq	System Clocks per Frame
Low Rate	48	512
Medium Rate	96	256
High Rate	192	128

Table 55: InS Clocks in frame

5.6.2 Clock Ports

Eternal modules also receive clocks and sync signals from CLOCK port located inside Tx Modules. DICE JR has two clock ports that are located in InS Transmiter modules to utilize similar logic.

Each clock port has 3 external pins, fsync, bclk and mck. They all have edges aligned with the internal sys_1fs positive edge and they all have an invert option before the actual pin output. Please note that wcko output is alaigned to the negative edge of internal sys_1fs, so it causes half a clock delay between wcko and InS external clocks in case no delay was programmed.

Configuration options for fsync

fsync_length	1 bit	1 or 32 bclk's
fsync_invert	1 bit	Invert if set

Configuration options for bclk

Bclk_rate	2 bits	64fs, 128fs, 256fs
Bclk_invert	1 bit	Invert if set

Configuration options for mck

Mck_rate	2 bits	256xbr, 512xbr, 128xfs, 256xfs
Mck_invert	1 bit	Invert if set

The bit clock always has a fixed relation to the sys_1fs regardless of the sys_mode.

	sys_mode		
bclk_rate	Low	mid	High

	sys_mode		
bclk_rate	Low	mid	High
00 =64fs	sys_f512br/8	sys_f512br/4	sys_f512br/2
01 =128fs	sys_f512br/4	sys_f512br/2	sys_f512br/1
10 =256fs	sys_f512br/2	sys_f512br/1	N/A
11 =reserved	N/A	N/A	N/A

Table 56: bclk relation to sys_f512fs

The mck relation to the sys_1fs and base rate depends on the system configuration. Most codec's expect it to have a fixed relation to the base rate and that is handled by the two first options.

The last two options are for systems which require the mck to have a fixed relation to the actual 1fs.

	sys_mode		
mck_rate	low	mid	High
00 =256br	sys_f512br/2	sys_f512br/2	sys_f512br/2
01 =512br	sys_f512br/1	sys_f512br/1	sys_f512br/1
10 =128fs	sys_f512br/4	sys_f512br/2	sys_f512br/1
11 =256fs	sys_f512br/2	sys_f512br/1	N/A

Table 57: bclk relation to sys_f512fs

As DICE JR has only two clock ports, only two types of communication out of possible three (2channels/4channels/8channels) could be used simultaneously.

5.6.3 Data transmission

Each Tx instance inside Tx module can be configured to transfer 2/4/8 audio channels. Data for each chnnel is stored in RAM shared by all Tx instances inside one module. One RAM is used to utilize common logic in address generation & data storage. If several instancies require data at the same time, requests are arbitrated using fixed priority mechanism. Each Tx instance could shuffle audio channel data if programmed to do so, according to the following list of shuffle schemes:

data_shuffle[3:0]	Order of transmission
0000	data[31:0] -> b31,..,b8, b7,..,b0
0001	data[31:0] -> b31,..,b8, b0,..,b7
0010	data[31:0] -> b8,..,b31, b7,..,b0
0011	data[31:0] -> b8,..,b31, b0,..,b7
0100	data[31:0] -> b7,..,b0, b31,..,b8
0101	data[31:0] -> b7,..,b0, b8,..,b31
0110	data[31:0] -> b0,..,b7, b31,..,b8
0111	data[31:0] -> b0,..,b7, b8,..,b31

1000	data[31:0] -> b31,..,b24, b23,..,b0
1001	data[31:0] -> b31,..,b24, b0,..,b23
1010	data[31:0] -> b24,..,b31, b23,..,b0
1011	data[31:0] -> b24,..,b31, b0,..,b23
1100	data[31:0] -> b23,..,b0, b31,..,b24
1101	data[31:0] -> b23,..,b0, b24,..,b31
1110	data[31:0] -> b0,..,b23, b31,..,b24
1111	data[31:0] -> b0,..,b23, b24,..,b31

Table 58: Data Shuffle table

Shuffled data is loaded into shift register so that first bit would be shifted out exactly on the positive edge of sys_1fs if no delay was configured.

If some delay was configured for current TX, then it would be added so the first out bit would be late.

5.6.4 Signal Description

InS Tx instancies transmit data through AUDIO Ports. AUDIO Port 0 is connected to Tx Module 0 (Instances 0-3) and Audio Port1 is connected to Tx Module 1(Instances 0-3), when configured.

For AUDIO port configuration options please see 4.1.2

Sync and Clock signals have dedicated pins described in Table 59

Signal	TCD2220	TCD2210	I/O	Drive (mA)	Description
Mck0	106	99	O	8	Master clock from Clock Port0
Fck0	107	100	O	8	Sync from Clock port0
Bck0	108	101	O	8	Bit clock from Clock port0
Mck1_gp6	117 (shared)	106 (shared)	B	6	Master clock from Clock Port1
fck1_gp7	118 (shared)	107 (shared)	B	6	Sync from Clock port1
Bck1_gp8	119 (shared)	108 (shared)	B	6	Bit clock from Clock port1

Table 59: InS Clock & sync signals

5.6.5 Module Configuration

Address	Register
0xce09 0000	INS0_TX0_SETUP
0xce09 0020	INS0_TX1_SETUP
0xce09 0040	INS0_TX2_SETUP
0xce09 0060	INS0_TX3_SETUP
0xce09 0080	INS0_CLKP_SETUP
0xce09 0fe0	INS0_MUTE
0xce0b 0000	INS1_TX0_SETUP
0xce0b 0020	INS1_TX1_SETUP
0xce0b 0040	INS1_TX2_SETUP
0xce0b 0060	INS1_TX3_SETUP
0xce0b 0080	INS1_CLKP_SETUP
0xce0b 0fe0	INS1_MUTE

Table 60: I^2S Transmitter Memory Map

5.6.6 INSm_CLKP_SETUP

Address - 0xce09 0080 - InS0_CLKP_SETUP
Address - 0xce0b 0080 - InS1_CLKP_SETUP

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							enable	Fsy_inv	Bck_inv	Mck_inv	fsy_len	Bck_rate		Mck_rate	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	15:9	0	R	Reserved
ENABLE	8	0	RW	Enables the clock interface. If disabled all outputs are 0.
FSY_INV	7	0	RW	0: pos. edge aligned with 1fs frame 1: neg. edge aligned with 1fs frame
BCK_INV	6	0	RW	0: pos. edge aligned with 1fs frame 1: neg. edge aligned with 1fs frame
MCK_INV	5	0	RW	0: pos. edge aligned with 1fs frame 1: neg. edge aligned with 1fs frame
FSY_LEN	4	0	RW	0: 1 bclk length 1: 32 bclk length
BCK_RATE	3:2	0	RW	0: 64fs 1: 128fs 2: 256fs 3: reserved
MCK_RATE	1:0	0	RW	0: 256br 1: 512br 2: 128fs 3: 256fs

5.6.7 INSm_TXn_SETUP

Address - 0xce09 0000 - InS0_Tx0
Address - 0xce09 0020 - InS0_Tx1
Address - 0xce09 0040 - InS0_Tx2
Address - 0xce09 0060 - InS0_Tx3
Address - 0xce0b 0000 - InS1_Tx0
Address - 0xce0b 0020 - InS1_Tx1
Address - 0xce0b 0040 - InS1_Tx2
Address - 0xce0b 0060 - InS1_Tx3

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							Enable	DATA SHUFFLE				Ins_Delay		Ins_Mode	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
ENABLE	8	0	RW	This bit enables the Operation of particular Tx instance inside INS module. 0: TX instance disabled 1: TX instance Enabled
DATA SHUFFLE	7:4	0	RW	See Shuffle table Table 58
Ins DELAY	3:2	0	RW	This bit selects the delay from posedge of sys_1fs to the beginning of fsync transmission 00: no delay 01: 1 bclk delay 10: 2 bclk delay 11: 3 bclk delay
Ins MODE	1:0	0	RW	Those bits select mode for Tx instance 00: I2S 01: I4S 10: I8S 11: reserved

5.6.8 InSn_MUTE

Address - 0xce09 0fe0 - InS0 Module

Address - 0xce0b 0fe0 - InS1 Module

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Mute15	Mute14	Mute13	Mute12	Mute11	Mute10	Mute9	Mute8	Mute7	Mute6	Mute5	Mute4	Mute3	Mute2	Mute1	Mute0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Name	Bit	Reset	Dir	Description
Mutex	15:0	1	RW	This field controls the muting of individual channels in the transmitter. 0: Not muted 1: Mute

5.6.9 I^{2}S compliant operation

In order to increase system flexibility by developing standardized communication structures between different digital audio system IC's, Philips developed the Inter-IC sound bus (I^2S), a serial link especially for digital audio.

The bus has three lines:

- Continuous serial clock SCK (BCK)
- Word Select WS (FSYNC)
- Serial data SD (SDATA)

The most distinguishing feature of the I^2S bus is the one serial bit clock delay after a transition on the word clock. The transmitter always sends the MSB of the next word one clock period after a transition on LRCK_OUT. This means that the MSB has a fixed position.

Our transmitter can be programmed as a fully compliant I^2S transmitter and is designed to operate as a master on the I^2S bus, which means that it must output the Word Select (through the FSY_OUT pin) clock and the continuous serial clock (output BCK) to the I^2S transmitter operating in slave mode.

Figure 43 illustrates I^2S operation with a 32 bit word length.

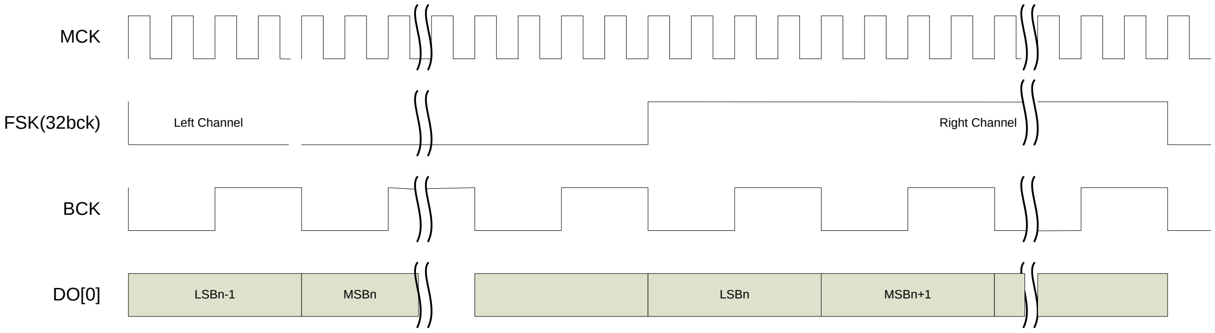


Figure 43: I2S compliant operation – 32 bit word length

To achieve I^2S compliant operation, we must therefore make sure that the receiver operates in the correct mode by programming the input pins as follows:

- Only 2 channels are transmitted, so **InS_MODE = 00**
- MSB transmitted one serial clock (BCK) period after transition on FSYNC i.e. **InSDelay = 01**.
- MSB should be transmitted first **Shuffle cheme = 0000**.
- BCK should correspond to 64 bits per frame (32 bits in one channel, 32 bits in another, so **BCK_RATE = 00**
- MCK should be 256 clocks in frame, so **MCK_RATE= 00**

- Inversion of the bit clock so that each bit is transmitted beginning on the falling edge of the bit clock and clocked in on the rising edge of BICK. i.e. **BCK_INV = 1**.
- FSync should resemble LRCLK_OUT, so we should choose 32bclk length **FSY_LEN = 1**
- Inversion of FSYNC so that reception of left channel data corresponds to FSYNC=0 i.e. **FSY_INV = 1**.

5.6.10 TDM compliant operation

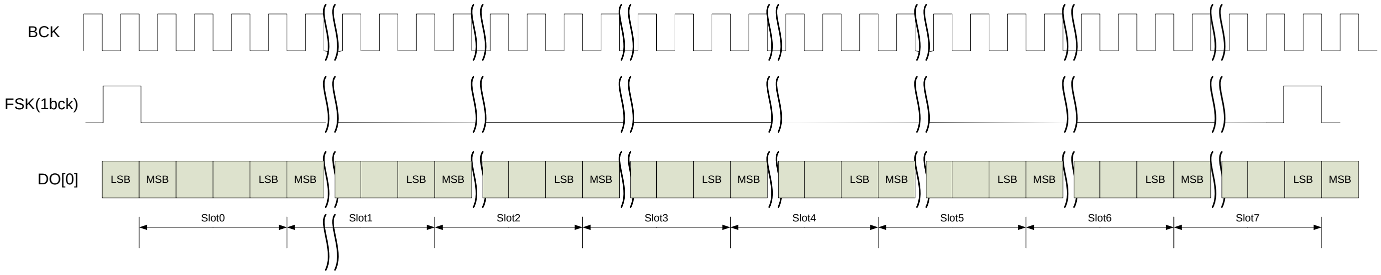


Figure 44: TDM functional diagram

To achieve TDM compliant operation the following bits need to be set:

- 8 channels are transmitted, so **InS_MODE = 10**
- MSB transmitted one serial clock (BCK) period after transition on FSYNC i.e. **InSDelay = 01**.
- MSB should be transmitted first **Shuffle cheme = 0000**.
- BCK should correspond to 32bitx8slots = 256 bits per frame so **BCK_RATE = 10**
- MCK is not used in this scheme, but for convenience **MCK_RATE = 00**
- Inversion of the bit clock so that each bit is transmitted beginning on the falling edge of the bit clock and clocked in on the rising edge of BICK. i.e. **BCK_INV = 1**.
- Only rising edge of FSync is meaningfull, so we could choose choose either 32bclk length or 1bclk length. In this example 1bclk length is chosen. **FSY_LEN = 0**
- No inversion of FSYNC is needed so **FSY_INV = 0**.

5.7 InS Receivers

5.7.1 System view

InS Receiver design is very similar to InS Transmitter. The only differences are as follows:

- No Clock port in InS RX modules
- No Mute Register in InS Rx Modules
- All Rx instancies operate using inverted version of sys_f512br (sys_f512br_inv)
- All Ins0 Rx instances are connected to AUDIO IN port0, and InS1 Rx instances are connected to AUDIO IN port 1. for AUDIO Port configuration please see 4.1.1. If AES is enabled on one of the ports, InS reception on this port is corrupted. If AES is disabled, then no additional configuration is needed

External modules that work with InS receivers inside the chip are supposed to use the clock ports that were described in previous section for their clock signal supply.

5.7.2 Signal Description

InS Rx instancies Receive data through AUDIO Ports. AUDIO Port 0 is connected to Rx Module 0 (Instances 0-3) and Audio Port1 is connected to Rx Module 1(Instances 0-3),

Audio Ports are always connected to the InS RX modules as specified, no additional configuration is needed. Note, that input AUDIO ports are shared between AES, InS & ADAT interfaces, and only for AES input can be disabled on AUDIO port level. For further information on AUDIO Input ports please see 4.1.1. For TCD2210 product AUDIO Port1 is not available, consequently, InS1 is not available either.

5.7.3 Module Configuration

Address	Register
0xce08 0000	INS0_RX0_SETUP
0xce08 0020	INS0_RX1_SETUP
0xce08 0040	INS0_RX2_SETUP
0xce08 0060	INS0_RX3_SETUP
0xce0a 0000	INS1_RX0_SETUP
0xce0a 0020	INS1_RX1_SETUP
0xce0a 0040	INS1_RX2_SETUP
0xce0a 0060	INS1_RX3_SETUP

Table 61: InS Receivers Memory Map

5.7.4 INSm_RXn_SETUP

Address - 0xce08 0000 - InS0_Rx0
Address - 0xce08 0020 - InS0_Rx1
Address - 0xce08 0040 - InS0_Rx2
Address - 0xce08 0060 - InS0_Rx3
Address - 0xce0a 0000 - InS1_Rx0
Address - 0xce0a 0020 - InS1_Rx1
Address - 0xce0a 0040 - InS1_Rx2

Address - 0xce0a 0060 - InS1_Rx3

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							Enable	DATA SHUFFLE				Ins_Delay		Ins_Mode	
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
ENABLE	8	0	RW	This bit enables the Operation of particular Tx instance inside INS module. 0: RX instance disabled 1: RX instance Enabled
DATA SHUFFLE	7:4	0	RW	See Shuffle table Table 58
Ins DELAY	3:2	0	RW	This bit selects the delay from posedge of sys_1fs to the beginning of data reception 00: no delay 01: 1 bclk delay 10: 2 bclk delay 11: 3 bclk delay
Ins MODE	1:0	0	RW	Those bits select mode for Rx instance 00: I2S 01: I4S 10: I8S 11: reserved

5.8 ADAT Receiver

The DICE JR contains two Alesis ADAT compatible receivers. The receiver can receive 8 channels of audio at the base rates (low system rate) and 4 channels of audio at twice the base (medium system rate) rate and 2 channels at four times base rate (high system rate).

S-Mux mode is selected automatically when system mode is set to medium or high rate.

Data is demultiplexed into the system buffers from a single ADAT lightpipe as follows:

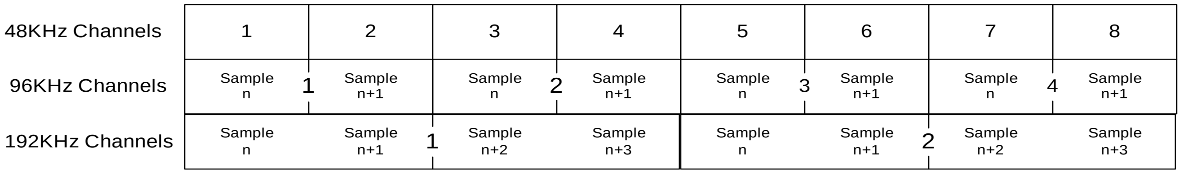


Figure 45 : ADAT lightpipe channel configuration

5.8.1 Signal Description

ADAT is connected to AUDIO port 0 bits [3:2]. For AUDIO Port operation description & programming see 4.1.1

5.8.2 Module Configuration

Address	Register
0xce04 0000	ADATRX0
0xce04 0004	ADATRX1

Table 62: ADAT Receiver Memory Map

5.8.3 ADATRX0

Address - 0xce04 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						User_bits0				U_run0	O_run0	NLock0	Nsync0	Lock0	Sync0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
reserved	31:10	0	R	Reads back as zerroes
User_bits0	9:6	0	RW	The 4 bits of user data received in the last frame.
U_RUN0	5	0	R	Indicates resampling which typically happens when the system 1FS is faster than 1FS from the master receiver. Can also be due to jitter and phase differences between the router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.

Name	Bit	Reset	Dir	Description
O_RUN0	4	0	R	Indicates slipped sample which typically happens when the system 1FS is slower than 1FS from the master receiver. Can also be due to jitter and phase differences between router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.
NLOCK0	3	0	R	Indicates that ADATRX0 is not locked
NSYNC0	2	0	R	No sync was detected by ADAT RX0
LOCK0	1	0	R	Indicates that DATARX0 was locked after 4 consecutive sync patterns
SYNC0	0	0	R	Indicates that ADAT has syncked for 4 frames

5.8.4 ADATRX1

Address - 0xce04 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved						User_bits1				U_run1	O_run1	NLock1	NSync1	Lock1	Sync1
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
reserved	31:10	0	R	Reads back as zerroes
User_bits1	9:6	0	RW	The 4 bits of user data received in the last frame.
U_R1	5	0	R	Indicates resampling which typically happens when the system 1FS is faster than 1FS from the master receiver. Can also be due to jitter and phase differences between the router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.
O_R1	4	0	R	Indicates slipped sample which typically happens when the system 1FS is slower than 1FS from the master receiver. Can also be due to jitter and phase differences between router 1FS and 1FS from master receiver. This bit is sticky and will be cleared immediately after a read.
NLOCK1	3	0	R	Indicates that ADATRX1 is not locked
NSYNC1	2	0	R	No sync was detected by ADAT RX1
LOCK1	1	0	R	Indicates that DATARX1 was locked after 4 consecutive sync patterns
SYNC1	0	0	R	Indicates that ADAT has syncked for 4 frames

5.9 ADAT Transmitter

The DICE JR chip contains two identical Alesis ADAT compatible transmitters. Each transmitter can transmit 8 channels of audio at the base rates. The transmitter can also handle 4 channels of audio at medium rate (96KHz) and 2 channels at high system rate (S-Mux mode).

S-Mux mode is selected automatically when system mode is set to medium or high rate.

When operating in S-Mux mode the router interface to the ADAT transmitter becomes a 4 channel interface, writing 4 channels of audio every 96KHz sample period. Data is multiplexed into the 48KHz transmitter buffers and transmitted on a single ADAT lightpipe with channel configuration as follows:

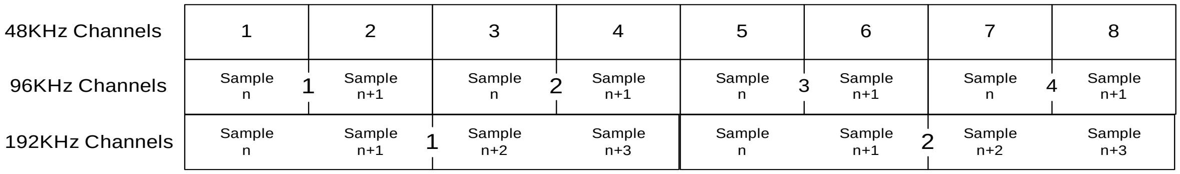


Figure 46: S-Mux ADAT lightpipe channel configuration

5.9.1 Signal Description

ADAT is connected to AUDIO port 0 bits [3:2] if programmed so. For AUDIO Port operation description & programming see 4.1.2

5.9.2 Module Configuration

Address	Register
0xce05 0000	ADATTX_CTRL1
0xce05 0004	ADATTX0_MUTE
0xce05 0008	ADATTX1_MUTE

Table 63: ADAT Transmitter Memory Map

5.9.3 ADATTX_CTRL1

Address - 0xce05 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								UDATA1				UDATA0			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	15:8	0	R	Reads back as zerros
UDATA1	7:4	0	RW	Used in non-loop mode to specify static user data for ADAT1
UDATA0	3:0	0	RW	Used in non-loop mode to specify static user data for ADAT0

5.9.4 ADATTXn_MUTE

Address - 0xce05 0004 - ADATTX0_MUTE

Address - 0xce05 0008 - ADATTX1_MUTE

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								MUTE7	MUTE6	MUTE5	MUTE4	MUTE3	MUTE2	MUTE1	MUTE0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
MUTE7-0	7:0	0	RW	Individual mute of each of the 8 audio channels in the ADAT stream.

5.10 ARM Audio Transceiver

The ARM Audio transceiver enables the ARM processor to access 8 channels of 32-bit audio from the router and to provide 8 channels of 32-bit audio to the router. The Receiver and Transmitter are synchronous to guarantee known latency.

The module consists of a 2 by 4 sample ping pong buffer system minimizing interrupt overhead. The host is interrupted every 4 samples, indicating that 4 new samples for each of the 8 channels are ready to be written/read.

5.10.1 Module Configuration

Address	Register
0xce16 0000 – 0xce16 0080	ARMAUDIO_BUF
0xce16 0100	ARMAUDIO_CTRL

Table 64: ARM Transceiver Memory Map

5.10.2 ARMAUDIO_BUF

Address - 0xce16 0000 – 0xce16 0080

The buffer is arranged as 4 32-bit samples of 8 channels of audio, the first 8 positions contain the first samples and so forth.

Even though the receive and transmit buffers share an address space, there are separate buffers. Writes will always access the transmit buffer and reads will access the receive buffer.

5.10.3 ARMAUDIO_CTRL

Address - 0xce16 0100

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														OVR	INT
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW*	RW*

Name	Bit	Reset	Dir	Description
OVR	1	0	RW	This bit indicates that the ARM did not manage to clear the interrupt condition before the next chunk was ready. This bit is cleared by a write to the register.
INT	0	0	RW	This bit indicates that a new chunk is ready for processing. The host should read the received data, write the new data to transmit and clear the interrupt by writing to the register.

5.11 Audio Mixer

The mixer creates 16 individual mono sub-mixes of up to 18 mono inputs. The mixer is part of the DICE audio system and any input channel can be routed to the mixer inputs and the mixer outputs can be routed to any output.
It is a full 18x16 mixing matrix with full 24 bit precision. I uses 288 16 bit coefficients to control the gains. The fixpoint system for the coefficients is 2:14 so each coefficient can provide a gain of 12dB (factor of 4). This provides for a channel gain mapping where each channel fader can reach +6dB and the output main fader can reach +6dB.
At unity gain the mixer is 24 bit transparent.
In case of clipping in the mixer a full per channel saturation system will limit the signal and set a sticky overload bit which can be polled by the ARM or used for interrupt generation.
All cofficients are directly addressable from the ARM processor and can be changed from cycle to cycle. Gliding and fading can be implemented in the ARM due to the high precision of the coefficients.

At high modes (176.4k or 192k) the mixer wil only be able to do 8 channels of sub-mixes (18x8).

5.11.1 Module Configuration

Address	Register
0xce06 0000	MIXER_CTRL
0xce06 0004	MIXER_OVL
0xce06 0008	MIXER_NUMOFCH
0xce06 0800	MIXER_COEFF RAM Coeff 0 for channel 0
0xce06 0804	MIXER_COEFF RAM Coeff 1 for channel 0
0xce06 0808	MIXER_COEFF RAM Coeff 2 for channel 0
…	
0xce06 0c78	MIXER_COEFF RAM Coeff 16 for channel 15
0xce06 0c7c	MIXER_COEFF RAM Coeff 17 for channel 15

Table 65: Mixer Memory Map

5.11.2 MIXER_CTRL

Address - 0xce06 0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															Enable
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW*

Name	Bit	Reset	Dir	Description
Reserved	15:1	0	R	Reserved. Read back as zerroes
Enable	0	0	RW	This bit enables/disables mixer operation 0: Disabled 1: Enabled

5.11.3 MIXER_OVERFLOW

Address - 0xce06 0004

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bit	Reset	Dir	Description
Overflow CH15	15	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH14	14	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH13	13	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH12	12	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH11	11	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH10	10	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH9	9	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH8	8	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH7	7	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH6	6	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH5	5	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH4	4	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH3	3	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH2	2	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH1	1	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed
Overflow CH0	0	0	RW	This bit shows if channel is overflowed 0: No overflow 1: Channel overflowed

5.11.4 MIXER_NUMOFCH

Address - 0xce06 0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved											Number of RX channels				
Reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Name	Bit	Reset	Dir	Description
Reserved	15:5	0	R	Reserved. Read back as zerroes
Number of RX Channels	4:0	10000	RW	Could be any value from 0 to 16. 16 is maximum In hifh rate mode is set to 8 by hardware

5.11.5 MIXER COEFFICIENTS RAM

Address - 0xce06 0800 – 0xce06 0c7c

Chapter 6 AVS

The 1394 Audio Video System (AVS) handles isochronous streaming of media. The Audio part interfaces with the DICE system described above. The Video part has access to dedicated pins on the chip.

The AVS consists of 4 1394 audio receivers and 2 1394 audio transmitters. The AVS also contains 1 1394 video receiver and 1 1394 video transmitter. Each audio receiver and transmitter can receive/send 16 channels of audio over the 1394 network. The video receiver and transmitter can receive/send 1 channel of video over the 1394 network.

The AVS contains a complex buffering system. Timestamps located in the CIP headers (for audio/video) or source packet headers (for video) of the received 1394 isochronous packets, are processed to cause each sample of each stream to be presented to the router (or the dedicated video interface) at the appropriate presentation time. Note that the appropriate presentation time is determined by the configured sample frequency.

The AVS transmitters create the timestamps which accompany the transmitted 1394 isochronous packets. As sample quadlets from audio/video streams are written to the AVS by the router/video interface, the AVS creates timestamps and associates them with the incoming sample quadlets. The AVS then organizes the sample quadlets into isochronous packets to be transmitted over the 1394 network. The associated timestamps are written to CIP/source packet headers and accompany the sample quadlets over the 1394 network.

All nodes on a 1394 network must be synchronized to one clock called the cycle timer, which is determined by the master node on the network. One cycle of the master nodes' cycle timer defines a 1394 cycle. At the beginning of each 1394 cycle the master node transmits a clock sync signal that allows all nodes on the 1394 network to be synchronized to the cycle timer. This maintains synchronicity among all the 1394 nodes. Each 1394 node receives the clock sync signal and uses it to update or correct its local timer. However, this clock correction can cause the local timer to jump forward or backward as it is updated by the clock sync signal, which can reduce the performance of the system.

The format of a quadlet of audio data passing through the AVS is configurable. The AVS can be configured to be transparent for 32-bit audio data. In this case the data will not be touched as it passes through the AVS. The AVS can also be configured to support the IEC61883-6 (AM824) steaming model. This allows the AVS to either source AM824 labels from another location, or build its own AM824 labels for the 24-bit data. The AVS can take the various label fields (block sync, user bits, channel status bits, etc) that make up each AM824 label, from different sources, and then pack them together into an AM824 label.

The AVS contains a local interrupt controller handling all the different interrupt sources and merging them before sending them on to the host system interrupt controller.

The main data buffering structure in the AVS is called the Media FIFO, and it uses 4 banks of circular buffers. Each buffer can be allocated by software configuration to a particular audio or video receiver or transmitter.

6.1 AVS Audio Receivers

The system contains 2 independent audio receivers each capable of extracting 16 audio channels and 8 MIDI plugs.

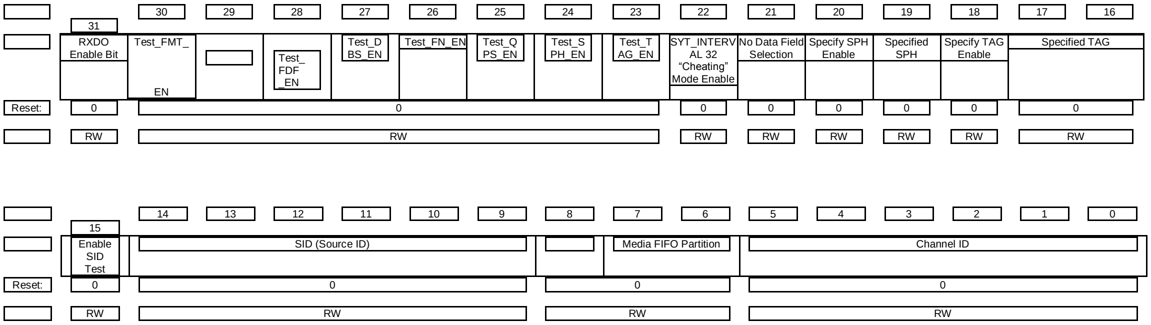
6.1.1 Module Configuration

Address	Register
0xcf00 0000	ARX1_CFG0
0xcf00 0004	ARX1_CFG1
0xcf00 0008	ARX1_QSEL0
0xcf00 000c	ARX1_QSEL1
0xcf00 0010	ARX1_QSEL2
0xcf00 0014	ARX1_QSEL3
0xcf00 0018	ARX1_QSEL4
0xcf00 001c	ARX1_PHDR
0xcf00 0020	ARX1_CIP0
0xcf00 0024	ARX1_CIP1
0xcf00 0028	ARX1_ADO_CFG
0xcf00 002c	ARX1_ADO_MIDI
0xcf00 0030	ARX2_CFG0
0xcf00 0034	ARX2_CFG1
0xcf00 0038	ARX2_QSEL0
0xcf00 003c	ARX2_QSEL1
0xcf00 0040	ARX2_QSEL2
0xcf00 0044	ARX2_QSEL3
0xcf00 0048	ARX2_QSEL4
0xcf00 004c	ARX1_PHDR
0xcf00 0050	ARX1_CIP0
0xcf00 0054	ARX1_CIP1
0xcf00 0058	ARX2_ADO_CFG
0xcf00 005c	ARX2_ADO_MIDI

Table 66: AVS Audio Receiver Memory Map

6.1.2 ARXn_CFG0

Address - 0xcf00 0000



Name	Bits	Reset	Dir	Description
RXDO Enable Bit	31	0	RW	RXDO Enable Bit. Setting this bit enables operation of the RXDO block.
Reserved	30	0	R	
TEST_FMT_EN	29	0	RW	Enables testing of locally set FMT against incoming FMT. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_FDF_EN	28	0	RW	Enables testing of locally set FDF against incoming FDF. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_DBS_EN	27	0	RW	Enables testing of locally set DBS against incoming DBS. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_FN_EN	26	0	RW	Enables testing of locally set FN against incoming FN. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_QPS_EN	25	0	RW	Enables testing of locally set QPS against incoming QPS. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_SPH_EN	24	0	RW	Enables testing of locally set SPH against incoming SPH. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
TEST_TAG_EN	23	0	RW	Enables testing of locally set TAG against incoming TAG. In case of difference CFG_FAIL interrupt is raised 0: Comparison disabled 1: Comparison Enabled
SYT_INTERVAL 32 “Cheating” Mode Enable	22	0	RW	SYT_INTERVAL 32 “Cheating” Mode Enable. Setting this bit puts the ARX DB counters into a cheat mode, allowing SYT_INTERVAL 32 streams to be output to the Router as if they were SYT_INTERVAL 16 streams with Data Blocks 2 times larger than shown in the CIP headers. Proper FORCED set up of the rest of the ARX (DBS, FDF) for SYT_INTERVAL 16 is required for this mode to

				work.
No Data Field Selection	21	0	RW	No Data Field Selection. Setting this bit causes a check of the FDF field to identify a NO_DATA packet; otherwise the FMT field is checked.
Specify SPH Enable	20	0	RW	Specify SPH Enable. Forces the ARX to obey the specified SPH field rather than the SPH received in the CIP headers of its isoch stream.
Specified SPH	19	0	RW	Specified SPH. Forced value of the SPH field.
Specify TAG Enable	18	0	RW	Specify TAG Enable. Forces the ARX to obey the specified TAG field rather than the TAG received in the Packet Headers of its isoch stream.
Specified TAG	17:16	0	RW	Specified TAG. Forced value of the TAG field.
Enable SID Test	15	0	RW	Enable SID Test. Setting this bit causes the ARX to compare the Source ID (SID) field of its isoch stream against the SID value given in this CFG register. If a mismatch occurs, and interrupt to the ARM will be signaled.
SID (Source ID)	14:9	0	RW	SID (Source ID). Value to optionally check the SID of an isoch stream against.
Reserved	8	0	R	Reserved. Read back as zerro
Media FIFO Partition Select	7:6	0	RW	Media FIFO Partition Select. Select which Media FIFO partition this RXDO block shall use.
Channel ID	5:0	0	RW	Channel ID. Tell the ARX what channel ID it shall take its isoch data from.

6.1.3 ARXn_CFG1

Address - 0xcf00 0004

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Specify DBS Enable	Specify FN Enable	Specify QPC Enable	Specified DBS								Specified FN		Specified QPC		
Reset:	0	0	0	0								0		0		
	RW	RW	RW	RW								RW		RW		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Specify FMT Enable	Specify FDF Enable	Specified FMT						Specified FDF							
Reset:	0	0	0						0							
	RW	RW	RW						RW							

Name	Bits	Reset	Dir	Description
Specify DBS Enable	31	0	RW	Specify DBS Enable. Forces the ARX to obey the specified DBS field rather than the DBS received in the CIP headers of its isoch stream.
Specify FN Enable	30	0	RW	Specify FN Enable. Forces the ARX to obey the specified FN field rather than the FN received in the CIP headers of its isoch stream.

Specify QPC Enable	29	0	RW	Specify QPC Enable. Forces the ARX to obey the specified QPC field rather than the QPC received in the CIP headers of its isoch stream.
Specified DBS	28:21	0	RW	Specified DBS. Forced value of DBS for the ARX. This sets how many data quadlets per Data Block will be expected in the isoch stream.
Specified FN	20:19	0	RW	Specified FN. Forced value of the FN field.
Specified QPC	18:16	0	RW	Specified QPC. Forced value of the QPC field. Any quadlets considered as padding will be discarded rather than stored in the Media FIFO.
Specify FMT Enable	15	0	RW	Specify FMT Enable. Forces the ARX to obey the specified FMT field rather than the FMT received in the CIP headers of its isoch stream.
Specify FDF Enable	14	0	RW	Specify FDF Enable. Forces the ARX to obey the specified FDF field rather than the FDF received in the CIP headers of its isoch stream.
Specified FMT	13:8	0	RW	Specified FMT. Forced value of the FMT field.
Specified FDF	7:0	0	RW	Specified FDF. Forced value of the FDF field.

6.1.4 ARXn_QSEL0

Address - 0xcf00 0008

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	4th Slot in each DB to allow through to the Media FIFO								3rd Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2nd Slot in each DB to allow through to the Media FIFO								1st Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

Isochronous data channels received by the ARX can include up to 64 different audio and MIDI sequences. The AVS handles a maximum of 16 audio sequences and one MIDI sequence per Isochronous data channel. The QSEL registers select which of the incoming audio and MIDI sequences for a given isochronous channel, are to be sent through the AVS to the Router. For example, if

Each data block received by the ARX inside an isoch packet, can contain a maximum of 256 quadlets of data, where each quadlet is one sample of one of 256 audio sequences (or MIDI sequence).

The AVS can handle a maximum of 17 audio sequences from each isoch channel (16 audio sequences maximum and 1 MIDI sequence maximum). The QSEL registers specify which of the 256 sequences the AVS is to receive.

Each QSEL register slot can hold a value of 0-255. These slots identify which quadlets of a data block to pull out and store in the Media FIFO, and which to ignore. Each quadlet in an incoming data block is assigned a number starting with 1 as the first quadlet. Setting a QSEL to 0 causes the all further quadlets to be ignored. This numbering scheme is then used to specify which quadlets should be stored in the Media FIFO. For example, let’s say that the Data Block Size (DBS) of the received stream is 150, and you only want to store quadlets 2, 3, 19, 101, 133. You would assign the following values to the QSEL slots:

QSEL Slots 1-17:

0x02, 0x03, 0x13, 0x65, 0x85, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

Setting a QSEL Slot to 0x00 causes all further quadlets in the data block to be ignored. Slots must contain numbers in ascending order…further manipulation of data quadlet ordering must be done by the DICE JR/Mini Router. At most, 17 quadlets can be pulled out of a data block.

Note that the ARX does not care about MIDI quadlets. It does not matter whether MIDI is enabled or not, the ARX will still just pass through whatever quadlets are referenced to in the QSEL registers.

In the ADO, if MIDI is enabled, the *last* quadlet of each “QSEL defined” data block will always be stripped off and sent to the MIDI interface. All other quadlets will then be sent to the DICE JR/Mini Router. If MIDI is not enabled the ADO will assume that all quadlets in the “QSEL defined” data block are audio and will send them all to the DICE JR/Mini router.

The reason for having 17 QSEL entries is to be able to handle 16 audio and 1 MIDI. Note that this does not mean that QSEL 17 is reserved for MIDI. Rather, MIDI must be referenced by the last valid QSEL entry. Note that this could even be the first QSEL, in the case of having 1 MIDI sequence and NO audio sequences. Also note, even though there are 17 QSEL entries, the ADO can only send out a maximum of 16 audio sequences to the router.

If a data block were to arrive with 256 entries, only quadlets in the first 255 can be pulled out. This is a limitation of the numbering scheme, but not one to likely cause problems since no accepted audio format has data block sizes anywhere near 256.

Name	Bits	Reset	Dir	Description
QSEL 3	31:24	0	RW	4th Slot in each DB to allow through to the Media FIFO.
QSEL 2	23:16	0	RW	3rd Slot in each DB to allow through to the Media FIFO.
QSEL 1	15:8	0	RW	2nd Slot in each DB to allow through to the Media FIFO.
QSEL 0	7:0	0	RW	1st Slot in each DB to allow through to the Media FIFO.

6.1.5 ARXn_QSEL1

Address - 0xcf00 000c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	8th Slot in each DB to allow through to the Media FIFO								7th Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	6th Slot in each DB to allow through to the Media FIFO								5th Slot in each DB to allow through to the Media FIFO							

Reset:	0	0
	RW	RW

Name	Bits	Reset	Dir	Description
QSEL 7	31:24	0	RW	8th Slot in each DB to allow through to the Media FIFO.
QSEL 6	23:16	0	RW	7th Slot in each DB to allow through to the Media FIFO.
QSEL 5	15:8	0	RW	6th Slot in each DB to allow through to the Media FIFO.
QSEL 4	7:0	0	RW	5th Slot in each DB to allow through to the Media FIFO.

6.1.6 ARXn_QSEL2

0xcf00 0010

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	12th Slot in each DB to allow through to the Media FIFO								11th Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	10th Slot in each DB to allow through to the Media FIFO								9th Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

Name	Bits	Reset	Dir	Description
QSEL 11	31:24	0	RW	12th Slot in each DB to allow through to the Media FIFO.
QSEL 10	23:16	0	RW	11th Slot in each DB to allow through to the Media FIFO.
QSEL 9	15:8	0	RW	10th Slot in each DB to allow through to the Media FIFO.
QSEL 8	7:0	0	RW	9th Slot in each DB to allow through to the Media FIFO.

6.1.7 ARXn_QSEL3

Address - 0xcf00 0014

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	16th Slot in each DB to allow through to the Media FIFO								15th Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

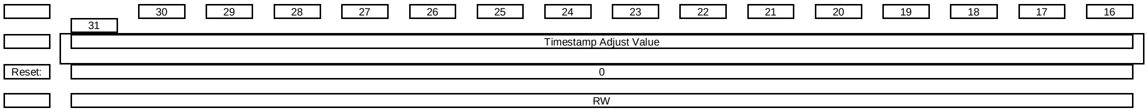
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	14th Slot in each DB to allow through to the Media FIFO								13th Slot in each DB to allow through to the Media FIFO							
Reset:	0								0							
	RW								RW							

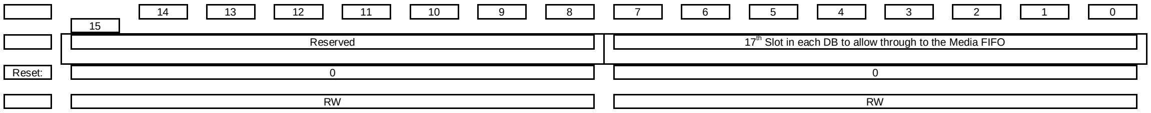
Name	Bits	Reset	Dir	Description
QSEL 15	31:24	0	RW	16th Slot in each DB to allow through to the Media FIFO.
QSEL 14	23:16	0	RW	15th Slot in each DB to allow through to the Media FIFO.

QSEL 13	15:8	0	RW	14th Slot in each DB to allow through to the Media FIFO.
QSEL 12	7:0	0	RW	13th Slot in each DB to allow through to the Media FIFO.

6.1.8 ARXn_QSEL4

Address - 0xcf00 0018





Name	Bits	Reset	Dir	Description
Timestamp Adjust Value	31:16	0	RW	Allows skew of presentation time, both forwards and backwards. MSB is a sign bit, positive causes forward skew, and vice-versa.
QSEL 16	7:0	0	RW	17th Slot in each DB to allow through to the Media FIFO.

NOTE: How to use the QSEL fields to receive data quadlets

There are 17 "slots" you can specify in the QSEL registers. Each QSEL register slot can hold a value of 0-255. Since Data Blocks in the received stream can have up to 256 quadlets, these slots identify which quadlets of a data block to pull out and store in the Media FIFO, and which to ignore. If you assign numbers to the quadlets in a data block (starting with 1 as the first quadlet) then use this numbering scheme to specify which quadlets should be stored in the Media FIFO. For example, let's say that the Data Block Size (DBS) of the received stream is 16, and you only want to store quadlets 2, 4, 6, …, 16. You would assign the following values to the QSEL slots:

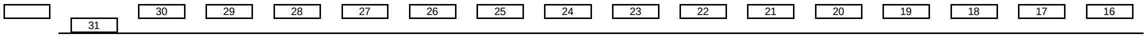
QSEL Slots 1-8: 8'h02, 8'h04, 8'h06, 8'h08, 8'h0A, 8'h0C, 8'h0E, 8'h10 (respectively).

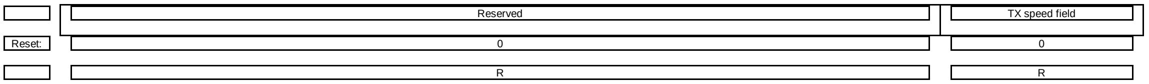
QSEL Slots 9-17: 8'h00.

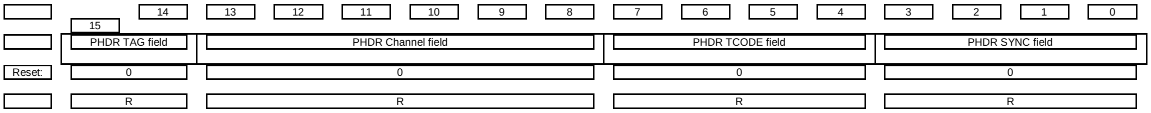
Setting a QSEL Slot to 8'h00 causes all further quadlets in the data block to be ignored. Slots must contain numbers in ascending order… further manipulation of data quadlet ordering must be done by the DICE2 Router. At most, 17 quadlets can be pulled out of a data block, but only the first 16 of these can be passed through to the DICE2 Router—the last quadlet is for MIDI only. Also, if a data block were to arrive with 256 entries, only quadlets in the first 255 can be pulled out (this is a limitation of the numbering scheme, but not one to likely cause problems since no accepted audio format has data block sizes anywhere near 256).

6.1.9 ARXn_PHDR

Address - 0xcf00 001c





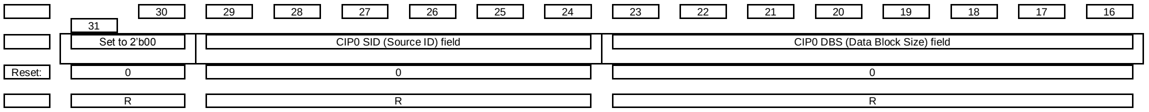


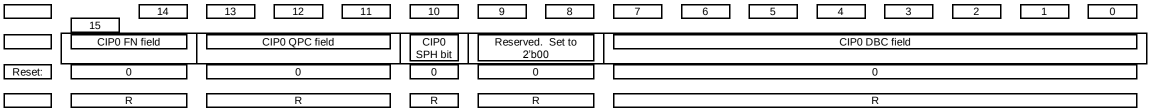
Name	Bits	Reset	Dir	Description
TX speed field	18:16	0	R	Set to 3’h0 for S100, 3’h1 for S200, and any other value indicates S400 isochronous transmit speed
PHDR TAG field	15:14	0	R	
PHDR Channel field	13:8	0	R	
PHDR TCODE field	7:4	0	R	
PHDR SYNC field	3:0	0	R	

This register is read only. It contains the last received PHDR quadlet.

6.1.10 ARXn_CIP0

Address - 0xcf00 0020





Name	Bits	Reset	Dir	Description
Reserved	31:30	0	R	
CIP0 SID (Source ID) field	29:24	0	R	CIP0 SID (Source ID) field
CIP0 DBS (Data Block Size) field	23:16	0	R	CIP0 DBS (Data Block Size) field

CIP0 FN field	15:14	0	R	CIP0 FN field
CIP0 QPC field	13:11	0	R	CIP0 QPC field
CIP0 SPH bit	10	0	R	CIP0 SPH bit
Reserved	9:8	0	R	
CIP0 DBC field	7:0	0	R	CIP0 DBC field

This register is read only. It contains the last received CIP0 quadlet.

6.1.11 ARXn_CIP1

Address - 0xcf00 0024

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Set to 2’b10		CIP1 FMT field						CIP1 FDF field							
Reset:	0		0						0							
	R		R						R							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIP1 SYT Timestamp field															
Reset:	0															
	R															

Name	Bits	Reset	Dir	Description
Set to 2’b10	31:30	0	R	
CIP1 FMT field	29:24	0	R	
CIP1 FDF field	23:16	0	R	
CIP1 SYT Timestamp field	15:0	0	R	

This register is read only. It contains the last received CIP1 quadlet.

6.1.12 ARXn_ADO_CFG

Address - 0xcf00 0028

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Mute_n Bit	MIDI Enable Bit	Automatic Muting Enable	LOCK	DUAL Wire	Reserved										
Reset:	0	0	0	0												
	RW	RW	RW	RW												

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lock Count Field															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Mute_n Bit	31	0	RW	Mute_n Bit (Active Low). When low, all data output from the ADO will be muted. This bit must be set active to get data values through the ADO.
MIDI Enable Bit	30	0	RW	MIDI Enable Bit. This bit tells the ADO whether the last quadlet in every DB is MIDI, or if there is no MIDI in the stream to deal with.
Automatic Muting Enable	29	0	RW	Automatic Muting Enable. Setting this bit causes an automatic mute of the ADO data stream when the ADO is not locked.
Transmission is LOCKED	28	0	R	SM is in the “LOCKED” state
Dual Wire mode	27	0	RW	Dual wire mode set (as in standard) 0: Mode disabled 1: Mode enabled
Reserved	26:16	0	R	Reads back as zerro
Lock Count Field	15:0	0	RW	Lock Count Field. This sets the number of 1934 clocks that must pass without ADO slipping or repeating sample before the ADO will signal it is locked. If a slip or repeat occurs, the lock is lost and this count must again be satisfied.

6.1.13 ARXn_ADO_MIDI

Address - 0xcf00 002c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Enable MIDI Port 7	Plug to Port mapping for MIDI Port 7			Enable MIDI Port 6	Plug to Port mapping for MIDI Port 6			Enable MIDI Port 5	Plug to Port mapping for MIDI Port 5.			Enable MIDI Port 4	Plug to Port mapping for MIDI Port 4		
Reset:	0	0			0	0			0	0			0	0		
	RW	RW			RW	RW			RW	RW			RW	RW		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Enable MIDI Port 3	Plug to Port mapping for MIDI Port 3			Enable MIDI Port 2	Plug to Port mapping for MIDI Port 2			Enable MIDI Port 1	Plug to Port mapping for MIDI Port 1			Enable MIDI Port 0	Plug to Port mapping for MIDI Port 0		
Reset:	0	0			0	0			0	0			0	0		
	RW	RW			RW	RW			RW	RW			RW	RW		

Name	Bits	Reset	Dir	Description
Enable MIDI Port 7	31	0	RW	Enable MIDI Port 7.

Plug to Port mapping for MIDI Port 7	30:28	0	RW	Plug to Port mapping for MIDI Port 7.
Enable MIDI Port 6	27	0	RW	Enable MIDI Port 6.
Plug to Port mapping for MIDI Port 6	26:24	0	RW	Plug to Port mapping for MIDI Port 6.
Enable MIDI Port 5	23	0	RW	Enable MIDI Port 5.
Plug to Port mapping for MIDI Port 5	22:20	0	RW	Plug to Port mapping for MIDI Port 5.
Enable MIDI Port 4	19	0	RW	Enable MIDI Port 4.
Plug to Port mapping for MIDI Port 4	18:16	0	RW	Plug to Port mapping for MIDI Port 4.
Enable MIDI Port 3	15	0	RW	Enable MIDI Port 3.
Plug to Port mapping for MIDI Port 3	14:12	0	RW	Plug to Port mapping for MIDI Port 3.
Enable MIDI Port 2	11	0	RW	Enable MIDI Port 2.
Plug to Port mapping for MIDI Port 2	10:8	0	RW	Plug to Port mapping for MIDI Port 2.
Enable MIDI Port 1	7	0	RW	Enable MIDI Port 1.
Plug to Port mapping for MIDI Port 1	6:4	0	RW	Plug to Port mapping for MIDI Port 1.
Enable MIDI Port 0	3	0	RW	Enable MIDI Port 0.
Plug to Port mapping for MIDI Port 0	2:0	0	RW	Plug to Port mapping for MIDI Port 0.

6.2 AVS Audio Transmitters

The system contains 2 independent audio transmitters each capable of sending 16 audio channels and 8 MIDI plugs.

6.2.1 Module Configuration

Address	Register
0xcf00 00c0	ATX1_CFG
0xcf00 00c4	ATX1_TSTAMP
0xcf00 00c8	ATX1_PHDR
0xcf00 00cc	ATX1_CIP0
0xcf00 00d0	ATX1_CIP1
0xcf00 00d4	ATX1_ADI_CFG
0xcf00 00d8	ATX1_ADI_MIDI
0xcf00 00dc	ATX2_CFG
0xcf00 00e0	ATX2_TSTAMP
0xcf00 00e4	ATX2_PHDR
0xcf00 00e8	ATX2_CIP0
0xcf00 00ec	ATX2_CIP1
0xcf00 00f0	ATX2_ADI_CFG
0xcf00 00f4	ATX2_ADI_MIDI

Table 67: AVS Audio Transmitter Memory Map

6.2.2 ATXn_CFG

Address - 0xcf00 00c0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXDI Enable Bit	NO_DATA Enable Bit	FMT NO_DATA Enable	FDF NO_DATA Enable	SPH Enable Bit	Multiple Source Packet Enable Bit	SPH Timestamp Enable Bit	CIP Timestamp Enable Bit	23b Timestamp Enable Bit	25b Timestamp Enable Bit	CFG_SYS_MODE		Reserved			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0			
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							Data Block Size					Reserved		Media FIFO Partition Select	
Reset:	0							0					0	0		
	RW							RW					RW	RW		

Name	Bits	Reset	Dir	Description
TXDI Enable Bit	31	0	RW	Setting this bit enables operation of the TXDI block.
NO_DATA Enable Bit	30	0	RW	Setting this bit will cause the ATX to send no-data packets when there isn't sufficient data to send a real packet. Otherwise empty packets will be sent.
FMT NO_DATA Enable	29	0	RW	Setting this bit will cause the FMT field of NO_DATA packets to be 6'h3F (all 1's).
FDF NO_DATA Enable	28	0	RW	Setting this bit will cause the FDF field of NO_DATA packets to be 8'hFF (all 1's).
SPH Enable Bit	27	0	RW	Enables Source Packet Headers in the isoch stream (not meant for standard audio streams).
Multiple Source Packet Enable Bit	26	0	RW	Setting this bit enables multiple Source Packets (SYT_INTERVAL Data Blocks) to be sent per isoch period (not meant for standard audio streams).
SPH Timestamp Enable Bit	25	0	RW	Tells the ATX to put timestamps in the Source Packet Header (SPH) instead of the CIP header (not meant for standard audio streams).
CIP Timestamp Enable Bit	24	0	RW	Tells the ATX to put timestamps in the CIP header (standard audio stream format).
23b Timestamp Enable Bit	23	0	RW	Sets timestamp width at 23 bits. NOTE: leaving bits 22 and 23 low will cause timestamp width of 16, which is standard for audio streaming.
25b Timestamp Enable Bit	22	0	RW	Sets timestamp width at 25 bits. NOTE: leaving bits 22 and 23 low will cause timestamp width of 16, which is standard for audio streaming.
CFG_SYS_MODE	21:20	0	RW	System rate configuration 00: Low rate (48KHz) 01: Medium Rate (96KHz) 10: High Rate (192KHz) 11: Reserved
Data Block Size	8:4	0	RW	Even though the DBS is part of the CIP header, this is the field that drives all counters in the ATX just in case there are logical errors in the final DICE2 chip.
Reserved	2	0	R	Reserved.
Media FIFO Partition	1:0	0	RW	Select which Media FIFO partition this TXDI block shall use.

Select

6.2.3 ATXn_TSTAMP

Address - 0xcf00 00c4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Timestamp Adjust Value															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Timestamp Adjust Value	15:0	0	RW	Value is added to the timestamp value for every isoch packet to account for network transmission time. This value also controls the amount of buffering in the destination node.

6.2.4 ATXn_PHDR

Address - 0xcf00 00c8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved													TX speed field		
Reset:	0													0		
	W													W		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PHDR TAG field		PHDR Channel field						PHDR TCODE field				PHDR SYNC field			
Reset:	0		0						0				0			
	W		W						W				W			

Name	Bits	Reset	Dir	Description
TX speed field	18:16	0	W	Set to 3’h0 for S100, 3’h1 for S200, and any other value will set S400 isochronous transmit speed
PHDR TAG field	15:14	0	W	Used when constructing PHDR quadlet to send
PHDR Channel field	13:8	0	W	Used when constructing PHDR quadlet to send
PHDR TCODE field	7:4	0	W	Used when constructing PHDR quadlet to send
PHDR SYNC field	3:0	0	W	Used when constructing PHDR quadlet to send

This register is Write Only.

6.2.5 ATXn_CIP0

Address - 0xcf00 00cc

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Set to 2'b00		CIP0 SID (Source ID) field						CIP0 DBS (Data Block Size) field							
Reset:	0		0						0							
	W		W						W							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIP0 FN field		CIP0 QPC field			CIP0 SPH bit	Reserved. Set to 2'b00		CIP0 DBC field							
Reset:	0		0			0	0		0							
	W		W			W	W		W							

Name	Bits	Reset	Dir	Description
Set to 2’b00	31:30	0	W	Set to 2’b00
CIP0 SID (Source ID) field	29:24	0	W	CIP0 SID (Source ID) field. (used when constructing CIP0 quadlet to send)
CIP0 DBS (Data Block Size) field	23:16	0	W	CIP0 DBS (Data Block Size) field. (only used when constructing CIP0 quadlet to send)
CIP0 FN field	15:14	0	W	CIP0 FN field. (used when constructing CIP0 quadlet to send)
CIP0 QPC field	13:11	0	W	CIP0 QPC field. (used when constructing CIP0 quadlet to send)
CIP0 SPH bit	10	0	W	CIP0 SPH bit. (used when constructing CIP0 quadlet to send)
Reserved. Set to 2’b00	9:8	0	W	Reserved. Set to 2’b00
CIP0 DBC field	7:0	0	W	CIP0 DBC field. This will be filled in by the ATX when sending the CIP0 quadlet.

This register is Write Only.

6.2.6 ATXn_CIP1

Address - 0xcf00 00d0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Set to 2'b10		CIP1 FMT field						CIP1 FDF field							
Reset:	0		0						0							
	W		W						W							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIP1 SYT Timestamp field															
Reset:	0															
	W															

Name	Bits	Reset	Dir	Description
Set to 2'b10	31:30	0	W	Set to 2'b10
CIP1 FMT field	29:24	0	W	(used when constructing CIP1 quadlet to send)
CIP1 FDF field	23:16	0	W	(used when constructing CIP1 quadlet to send)
CIP1 SYT Timestamp field	15:0	0	W	This will be filled in by the ATX when sending the CIP1 quadlet.

This register is Write Only.

6.2.7 ATXn_ADI_CFG

Address - 0xcf00 00d4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Mute_n Bit	MIDI Enable Bit	Reserved													
Reset:	0	0	0													
	RW	RW	RW													

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Mute_n Bit	31	0	RW	(Active Low). When low, all data output from the ADI will be muted. This bit must be set active to get data values through the ADI.
MIDI Enable Bit	30	0	RW	This bit tells the ADI whether the last quadlet in every DB should be filled with MIDI data, or if there is no MIDI in the stream to deal with.

6.2.8 ATXn_ADI_MIDI

Address - 0xcf00 00d8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Enable MIDI Port 7	Plug to Port mapping for MIDI Port 7			Enable MIDI Port 6	Plug to Port mapping for MIDI Port 6			Enable MIDI Port 5	Plug to Port mapping for MIDI Port 5			Enable MIDI Port 4	Plug to Port mapping for MIDI Port 4.		
Reset:	0	0			0	0			0	0			0	0		
	RW	RW			RW	RW			RW	RW			RW	RW		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Enable MIDI Port 3.	Plug to Port mapping for MIDI Port 3			Enable MIDI Port 2	Plug to Port mapping for MIDI Port 2			Enable MIDI Port 1	Plug to Port mapping for MIDI Port 1			Enable MIDI Port 0	Plug to Port mapping for MIDI Port 0		
Reset:	0	0			0	0			0	0			0	0		
	RW	RW			RW	RW			RW	RW			RW	RW		

Name	Bits	Reset	Dir	Description
MIDI Enable 7	31	0	RW	Enable MIDI Port 7.
MIDI Mapping 7	30:28	0	RW	Plug to Port mapping for MIDI Port 7.
MIDI Enable 6	27	0	RW	Enable MIDI Port 6.
MIDI Mapping 6	26:24	0	RW	Plug to Port mapping for MIDI Port 6.
MIDI Enable 5	23	0	RW	Enable MIDI Port 5.
MIDI Mapping 5	22:20	0	RW	Plug to Port mapping for MIDI Port 5.
MIDI Enable 4	19	0	RW	Enable MIDI Port 4.
MIDI Mapping 4	18:16	0	RW	Plug to Port mapping for MIDI Port 4.
MIDI Enable 3	15	0	RW	Enable MIDI Port 3.
MIDI Mapping 3	14:12	0	RW	Plug to Port mapping for MIDI Port 3.
MIDI Enable 2	11	0	RW	Enable MIDI Port 2.
MIDI Mapping 2	10:8	0	RW	Plug to Port mapping for MIDI Port 2.
MIDI Enable 1	7	0	RW	Enable MIDI Port 1.
MIDI Mapping 1	6:4	0	RW	Plug to Port mapping for MIDI Port 1.
MIDI Enable 0	3	0	RW	Enable MIDI Port 0.
MIDI Mapping 0	2:0	0	RW	Plug to Port mapping for MIDI Port 0.

6.3 AVS ITP (Internal Time Processor)

The ITP maintains an internal representation of the cycle timer and keeps track of time base changes. This enables isoc. streams to be immune to change of cycle master node and short arbitrated bus resets.

6.3.1 Module Configuration

Address	Register
0xcf00 01f8	ITP_CFG

Table 68: AVS ITP Memory Map

6.3.2 ITP_CFG

Address - 0xcf00 01f8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITP Enable Bit	Clear Offsets Bit	Reserved													
Reset:	0	0	0													
	RW	RW	RW													

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
ITP Enable Bit	31	0	RW	Setting this bit enables operation of the ITP block. Once the ITP is enabled, it must not be disabled without clearing the broadcast offset values that it generates. To clear these values, use the Clear Offsets Bit in this configuration register.
Clear Offsets Bit	30	0	RW	Setting this bit will cause the ITP to clear all broadcasted offset values immediately after APB writes which set this bit. This must be done when disabling the ITP after it has been enabled. The ITP does not need to be enabled to perform the clearing.

6.4 AVS Audio Transmitter Format Handler

Handles the transmission of IEC 60958 conformant data, which is compatible with AES/SPDIF and is the most important format. The DICE JR should handle Channel Status, User Bits, Validity and Block Sync in a similar way as is done by the DICE AES transceivers.

6.4.1 Module Configuration

Address	Register
0xcf00 02c0	FMT_TXDI1_CFG0
0xcf00 02c4	FMT_TXDI1_CFG1
0xcf00 02c8	FMT_TXDI1_CFG2
0xcf00 02cc	FMT_TXDI1_CFG3
0xcf00 02d0	FMT_TXDI1_CFG4
0xcf00 02d4	FMT_TXDI1_CFG5
0xcf00 02d8	FMT_TXDI1_CFG6
0xcf00 02dc	FMT_TXDI1_CSBLOCK_BYTEn
0xcf00 02f4	FMT_TXDI1_CHANNELn_CS/LABEL
0xcf00 0340	FMT_TXDI2_CFG0
0xcf00 0344	FMT_TXDI2_CFG1
0xcf00 0348	FMT_TXDI2_CFG2
0xcf00 034c	FMT_TXDI2_CFG3
0xcf00 0350	FMT_TXDI2_CFG4
0xcf00 0344	FMT_TXDI2_CFG5
0xcf00 0348	FMT_TXDI2_CFG6
0xcf00 034c	FMT_TXDI2_CSBLOCK_BYTEn
0xcf00 0374	FMT_TXDI2_CHANNELn_CS/LABEL

Table 69: AVS Audio Transmitter Format Handler Memory Map

6.4.2 FMT_TXDIn_CFG0

Address - 0xcf00 02c0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Ch 16 Label Cfg		Ch 15 Label Cfg		Ch 14 Label Cfg		Ch 13 Label Cfg		Ch 12 Label Cfg		Ch 11 Label Cfg		Ch 10 Label Cfg		Ch 9 Label Cfg	
Reset:	0		0		0		0		0		0		0		0	
	RW		RW		RW		RW		RW		RW		RW		RW	

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Ch 8 Label Cfg		Ch 7 Label Cfg		Ch 6 Label Cfg		Ch 5 Label Cfg		Ch 4 Label Cfg		Ch 3 Label Cfg		Ch 2 Label Cfg		Ch 1 Label Cfg	
Reset:	0		0		0		0		0		0		0		0	
	RW		RW		RW		RW		RW		RW		RW		RW	

Name	Bits	Reset	Dir	Description
Channel 16 Label Configuration	31:30	0	RW	Channel 16 Label Configuration. (see below for detail of bits)
Channel 15 Label Configuration	29:28	0	RW	Channel 15 Label Configuration. (see below for detail of bits)
Channel 14 Label Configuration	27:26	0	RW	Channel 14 Label Configuration. (see below for detail of bits)
Channel 13 Label Configuration	25:24	0	RW	Channel 13 Label Configuration. (see below for detail of bits)
Channel 12 Label Configuration	23:22	0	RW	Channel 12 Label Configuration. (see below for detail of bits)
Channel 11 Label Configuration	21:20	0	RW	Channel 11 Label Configuration. (see below for detail of bits)
Channel 10 Label Configuration	19:18	0	RW	Channel 10 Label Configuration. (see below for detail of bits)
Channel 9 Label Configuration	17:16	0	RW	Channel 9 Label Configuration. (see below for detail of bits)
Channel 8 Label Configuration	15:14	0	RW	Channel 8 Label Configuration. (see below for detail of bits)
Channel 7 Label Configuration	13:12	0	RW	Channel 7 Label Configuration. (see below for detail of bits)
Channel 6 Label Configuration	11:10	0	RW	Channel 6 Label Configuration. (see below for detail of bits)
Channel 5 Label Configuration	9:8	0	RW	Channel 5 Label Configuration. (see below for detail of bits)
Channel 4 Label Configuration	7:6	0	RW	Channel 4 Label Configuration. (see below for detail of bits)
Channel 3 Label Configuration	5:4	0	RW	Channel 3 Label Configuration. (see below for detail of bits)
Channel 2 Label Configuration	3:2	0	RW	Channel 2 Label Configuration. (see below for detail of bits)
Channel 1 Label Configuration	1:0	0	RW	Channel 1 Label Configuration. (see below for detail of bits)

2’b00: Transparent Mode—label byte is allowed through untouched.

2’b01: Mask Mode—label byte is replaced by constant configurable value.

2’b10: IEC 60958 Conformant Mode—label byte shall be 60958 conformant.

2’b11: Reserved. (will cause label byte to be always 0)

6.4.3 FMT_TXDIn_CFG1

Address - 0xcf00 02c4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Block Sync Master/Slave Mode Enable	Master Sync Select Bit	Collect Channel Status on Slave Channel Enable Bit	Reserved	Block Sync Slave Channel Select				Block Sync Preset Value							
Reset:	0	0	0	0	0				0							
	RW	RW	RW	RW	RW				RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Auto Channel Status CRC Disable Bits for channels 16 to 1, respectively															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Block Sync Master/Slave Mode Enable	31	0	RW	0 = Master; 1 = Slave.
Master Sync Select Bit	30	0	RW	0 = Free running Block Sync Counter; 1 = Sync to input.
Collect Channel Status on Slave Channel Enable Bit	29	0	RW	Enables collection of the 192 bit Channel Status information on the channel that Block Sync is slaved to.
Block Sync Slave Channel Select	27:24	0	RW	Selects the channel to sync the Block Sync to when in Slave mode.
Block Sync Preset Value	23:16	0	RW	Value to set the Block Sync Counter to when in Master mode, synced to input, and the input Block Sync goes active.
Auto Channel Status CRC Disable Bits	15:0	0	RW	For channels 16 to 1, respectively. (bit 15 -> channel 16; bit 14 -> channel 15; etc.)

6.4.4 FMT_TXDI_CFG2

Address - 0xcf00 02c8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel 16 Channel Status Configuration		Channel 15 Channel Status Configuration		Channel 14 Channel Status Configuration		Channel 13 Channel Status Configuration		Channel 12 Channel Status Configuration		Channel 11 Channel Status Configuration		Channel 10 Channel Status Configuration		Channel 9 Channel Status Configuration	
Reset:	0		0		0		0		0		0		0		0	
	RW		RW		RW		RW		RW		RW		RW		RW	

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel 8 Channel Status Configuration		Channel 7 Channel Status Configuration		Channel 6 Channel Status Configuration		Channel 5 Channel Status Configuration		Channel 4 Channel Status Configuration		Channel 3 Channel Status Configuration		Channel 2 Channel Status Configuration		Channel 1 Channel Status Configuration	
Reset:	0		0		0		0		0		0		0		0	
	RW		RW		RW		RW		RW		RW		RW		RW	

Name	Bits	Reset	Dir	Description
Channel 16 Channel Status Configuration	31:30	0	RW	Channel 16 Channel Status Configuration. (see below for detail of bits)
Channel 15 Channel Status Configuration	29:28	0	RW	Channel 15 Channel Status Configuration. (see below for detail of bits)
Channel 14 Channel Status Configuration	27:26	0	RW	Channel 14 Channel Status Configuration. (see below for detail of bits)
Channel 13 Channel Status Configuration	25:24	0	RW	Channel 13 Channel Status Configuration. (see below for detail of bits)
Channel 12 Channel Status Configuration	23:22	0	RW	Channel 12 Channel Status Configuration. (see below for detail of bits)
Channel 11 Channel Status Configuration	21:20	0	RW	Channel 11 Channel Status Configuration. (see below for detail of bits)
Channel 10 Channel Status Configuration	19:18	0	RW	Channel 10 Channel Status Configuration. (see below for detail of bits)
Channel 9 Channel Status Configuration	17:16	0	RW	Channel 9 Channel Status Configuration. (see below for detail of bits)
Channel 8 Channel Status Configuration	15:14	0	RW	Channel 8 Channel Status Configuration. (see below for detail of bits)
Channel 7 Channel Status Configuration	13:12	0	RW	Channel 7 Channel Status Configuration. (see below for detail of bits)
Channel 6 Channel Status Configuration	11:10	0	RW	Channel 6 Channel Status Configuration. (see below for detail of bits)
Channel 5 Channel Status Configuration	9:8	0	RW	Channel 5 Channel Status Configuration. (see below for detail of bits)
Channel 4 Channel Status Configuration	7:6	0	RW	Channel 4 Channel Status Configuration. (see below for detail of bits)
Channel 3 Channel Status Configuration	5:4	0	RW	Channel 3 Channel Status Configuration. (see below for detail of bits)
Channel 2 Channel Status Configuration	3:2	0	RW	Channel 2 Channel Status Configuration. (see below for detail of bits)
Channel 1 Channel Status Configuration	1:0	0	RW	Channel 1 Channel Status Configuration. (see below for detail of bits)

2’b0x: Channel Status bit allowed through from Router untouched.

2’b10: Channel Status bit taken from common APB Channel Status data.

2’b11: Channel Status bit taken from channel-specific APB Channel Status data.

6.4.5 FMT_TXDI_CFG3

Address - 0xcf00 02cc

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PAC SB Enable Bit for Channels 16 to 1, respectively															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAC SF Enable Bit for Channels 16 to 1, respectively															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
PAC SB Enable Bit	31:16	0	RW	For Channels 16 to 1, respectively. When enabled, the SB bit of the PAC bits will indicate start of block. (bit 31 -> channel 16; bit 30 -> channel 15; etc.)
PAC SF Enable Bit	15:0	0	RW	For Channels 16 to 1, respectively. When set, the SF bit of the PAC bits will always be set indicating the second sub-frame of data. (bit 15 -> channel 16; bit 14 -> channel 15; etc.)

6.4.6 FMT_TXDI_CFG4

Address - 0xcf00 02d0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	V Bit Replacement Enable Bits for Channels 16 to 1, respectively															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	V Bits for Channels 16 to 1, respectively															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Validity Bit Replacement Enable Bits	31:16	0	RW	For Channels 16 to 1, respectively. When enabled, the V bit will be replaced by the provided value below, otherwise the V bit already present in the label will be allowed through untouched. (bit 31 -> channel 16; bit 30 -> channel 15; etc.)
Validity Bits	15:0	0	RW	For Channels 16 to 1, respectively. When enabled by the above bit of the corresponding channel, this configuration bit will be inserted into the 60958 label as the V bit. (bit 15 -> channel 16; bit 14 -> channel 15; etc.)

6.4.7 FMT_TXDI_CFG5

Address - 0xcf00 02d4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	U Bit Selection for Channel 8				U Bit Selection for Channel 7				U Bit Selection for Channel 6				U Bit Selection for Channel 5			
Reset:	0				0				0				0			
	RW				RW				RW				RW			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	U Bit Selection for Channel 4				U Bit Selection for Channel 3				U Bit Selection for Channel 2				U Bit Selection for Channel 1			
Reset:	0				0				0				0			
	RW				RW				RW				RW			

Name	Bits	Reset	Dir	Description
User Bit for Channel 8	31:28	0	RW	U Bit Selection for Channel 8. (see detail of bits below)
User Bit for Channel 7	27:24	0	RW	U Bit Selection for Channel 7. (see detail of bits below)
User Bit for Channel 6	23:20	0	RW	U Bit Selection for Channel 6. (see detail of bits below)
User Bit for Channel 5	19:16	0	RW	U Bit Selection for Channel 5. (see detail of bits below)
User Bit for Channel 4	15:12	0	RW	U Bit Selection for Channel 4. (see detail of bits below)
User Bit for Channel 3	11:8	0	RW	U Bit Selection for Channel 3. (see detail of bits below)
User Bit for Channel 2	7:4	0	RW	U Bit Selection for Channel 2. (see detail of bits below)
User Bit for Channel 1	3:0	0	RW	U Bit Selection for Channel 1. (see detail of bits below)

4’b0000: Allow U bit already present in label byte from Router through as is.

4’b0001: Take U bit input from AVS RX 1.

4’b0010: Take U bit input from AVS RX 2.

4’b0011: Take U bit input from AVS RX 3.

4’b0100: Take U bit input from AVS RX 4.

4’b0101 – 4’b0111: Set U bit to 1’b0 always.

4’b1xxx: Take U bit from input U bit bus from AES[3’bxxx].

6.4.8 FMT_TXDI_CFG6

Address - 0xcf00 02d8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	U Bit Selection for Channel 16				U Bit Selection for Channel 15				U Bit Selection for Channel 14				U Bit Selection for Channel 13			
Reset:	0				0				0				0			
	RW				RW				RW				RW			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	U Bit Selection for Channel 12				U Bit Selection for Channel 11				U Bit Selection for Channel 10				U Bit Selection for Channel 9			
Reset:	0				0				0				0			
	RW				RW				RW				RW			

Name	Bits	Reset	Dir	Description
User Bit for Channel 16	31:28	0	RW	U Bit Selection for Channel 16. (see detail of bits below)
User Bit for Channel 15	27:24	0	RW	U Bit Selection for Channel 15. (see detail of bits below)
User Bit for Channel 14	23:20	0	RW	U Bit Selection for Channel 14. (see detail of bits below)
User Bit for Channel 13	19:16	0	RW	U Bit Selection for Channel 13. (see detail of bits below)
User Bit for Channel 12	15:12	0	RW	U Bit Selection for Channel 12. (see detail of bits below)
User Bit for Channel 11	11:8	0	RW	U Bit Selection for Channel 11. (see detail of bits below)
User Bit for Channel 10	7:4	0	RW	U Bit Selection for Channel 10. (see detail of bits below)
User Bit for Channel 9	3:0	0	RW	U Bit Selection for Channel 9. (see detail of bits below)

4’b0000: Allow U bit already present in label byte from Router through as is.

4’b0001: Take U bit input from AVS RX 1.

4’b0010: Take U bit input from AVS RX 2.

4’b0011: Take U bit input from AVS RX 3.

4’b0100: Take U bit input from AVS RX 4.

4’b0101 – 4’b0111: Set U bit to 1’b0 always.

4’b1xxx: Take U bit from input U bit bus from AES[3’bxxx].

6.4.9 FMT_TXDIN_CSBLOCK_BYTEn

Address - 0xcf00 02dc

These registers are used to write the entire channel status block (192 bits) for one selected channel.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n CS byte 3n								Channel n CS byte 2n							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n CS byte 1n								Channel n CS byte 0n							
Reset:	0								0							
	RW								RW							

Name	Bits	Reset	Dir	Description
CS Byte 3n	31:24	0	RW	Channel Status byte 3n for the selected channel.
CS Byte 2n	23:16	0	RW	Channel Status byte 2n for the selected channel.
CS Byte 1n	15:8	0	RW	Channel Status byte 1n for the selected channel.
CS Byte 0n	7:0	0	RW	Channel Status byte 0n for the selected channel.

6.4.10 FMT_TXDIn_CHANNELn_CS/LABEL

Address - 0xcf00 02f4

Note that the format handler can be configured to write the AM824 label bytes for channels 1 through 16 to these 16 registers, or it can be configured to write the first 4 Channel Status bytes for each of channels 1 through 16.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel n Channel Status byte 3 **OR** AM824 Label								Channel n Channel Status byte 2							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel n Channel Status byte 1								Channel n Channel Status byte 0							
Reset:	0								0							
	RW								RW							

Name	Bits	Reset	Dir	Description
CS Byte 3 or Label Byte	31:24	0	RW	Channel Status byte 3 for Channel n OR AM824 Label byte for Channel n
CS Byte 2	23:16	0	RW	Channel Status byte 2 for Channel n
CS Byte 1	15:8	0	RW	Channel Status byte 1 for Channel n
CS Byte 0	7:0	0	RW	Channel Status byte 0 for Channel n

6.5 AVS Audio Receiver Format Handler

Handles the reception of IEC 60958 conformant data, which is compatible with AES/SPDIF and is the most important format. The DICE JR should handle Channel Status, User Bits, Validity and Block Sync in a similar way as is done by the DICE AES transceivers.

6.5.1 Module Configuration

Address	Register
0xcf00 0200	FORMAT_RXDI1_CFG
0xcf00 0204	FORMAT_RXDI1_LABELn
0xcf00 0214	FORMAT_RXDI1_CSBLOCKn
0xcf00 0230	FORMAT_RXDI2_CFG
0xcf00 0234	FORMAT_RXDI2_LABELn
0xcf00 0244	FORMAT_RXDI2_CSBLOCKn

Table 70: AVS Audio Receiver Format Handler Memory Map

6.5.2 FORMAT_RXDIn_CFG

Address - 0xcf00 0200

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved								BS_PRESET							
Reset:	0								0							
	RW								RW							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSBLKSYNC				USRC				BLKSRC				CSCH			
Reset:	0				0				0				0			
	RW				RW				RW				RW			

Name	Bit	Reset	Dir	Description
BS_PRESET	23:16	0	RW	Block Sync Preset value, loaded to Block Sync counter on external block sync.
CSBLKSYNC	15:12	0	RW	Selects the channel to take Block Sync from for collecting Channel Status - Channel 0-15
USRC	11:8	0	RW	Selects the channel to take User data from - Channel 0-15
BLKSRC	7:4	0	RW	Selects the channel to take Block Sync from - Channel 0-15
CSCH	3:0	0	RW	Selects the channel to receive full Channel Status from - Channel 0-15

6.5.3 FORMAT_RXDIn_LABELn

Address - 0xcf00 0204

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel 3 Label Byte n								Channel 2 Label Byte n							

Reset:	0								0							
	R								R							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel 1 Label Byte n								Channel 0 Label Byte n							
Reset:	0								0							
	R								R							

Name	Bit	Reset	Dir	Description
Channel 0 Label Byte n	7:0	0	R	Allows reading the latest AM824 label byte of the given channel
Channel 1 Label Byte n	15:8	0	R	Allows reading the latest AM824 label byte of the given channel
Channel 2 Label Byte n	23:16	0	R	Allows reading the latest AM824 label byte of the given channel
Channel 3 Label Byte n	31:24	0	R	Allows reading the latest AM824 label byte of the given channel

6.5.4 FORMAT_RXDIn_CSBLOCKn

Address - 0xcf00 0214

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel Status Byte 3 n								Channel Status Byte 2 n							
Reset:	0								0							
	R								R							

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel Status Byte 1 n								Channel Status Byte 0 n							
Reset:	0								0							
	R								R							

Name	Bit	Reset	Dir	Description
Channel Status Byte 0 n	3	0	R	Allows reading the latest block of channel status bits
Channel Status Byte 1 n	2	0	R	Allows reading the latest block of channel status bits
Channel Status Byte 2 n	1	0	R	Allows reading the latest block of channel status bits
Channel Status Byte 3 n	0	0	R	Allows reading the latest block of channel status bits

6.6 AVS Interrupt Controller

The AVS Interrupt controller gathers all interrupts from the AVS, handles masking and clearing and hands of two interrupts to the host interrupt controller.

6.6.1 Module Configuration

Address	Register
0xcf00 013c	AVSI_INT0_STATUS
0xcf00 0140	AVSI_INT0_MASK
0xcf00 0144	AVSI_INT1_STATUS
0xcf00 0148	AVSI_INT1_MASK

Table 71: AVS INT CTRL Memory Map

6.6.2 APBA_INT0_STATUS

Address - 0xcf00 013c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADO2_ STREAM_ UNLOCK	ADO2_ STREAM_ LOCK	ADO2_ REPEAT	ADO2_ SLIP	ARXDO2_ SYT_ OVERFLOW	ADO2_SYT_ AGEOUT	ARX2_ LONG_PKT	ARX2_PKT_ ABORT	ARX2_ STATUS_ ERR	ADO2_NOT_ COMPL	ADO2_ MISSED_ SYNC	reserved	ARX2_CFG_ FAIL	ARX2_CIP_ FAIL	ARX2_DBC_ FAIL	ARXDO2_ SYT_ UNDERFLOW
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADO1_ STREAM_ UNLOCK	ADO1_ STREAM_ LOCK	ADO1_ REPEAT	ADO1_ SLIP	ARXDO1_ SYT_ OVERFLOW	ADO1_SYT_ AGEOUT	ARX1_ LONG_PKT	ARX1_PKT_ ABORT	ARX1_ STATUS_ ERR	ADO1_NOT_ COMPL	ADO1_ MISSED_ SYNC	reserved	ARX1_CFG_ FAIL	ARX1_CIP_ FAIL	ARX1_DBC_ FAIL	ARXDO1_ SYT_ UNDERFLOW
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Reset	Dir	Description
ADO2_STREAM_UNLOCK	31	0	RW	The lock status status in ARX2_ADO_CFG reg has changed from "lock" to "unlock"
ADO2_STREAM_LOCK	30	0	RW	The lock status in ARX2_ADO_CFG reg has changed from "unlocked" to "locked"
ADO2_REPEAT	29	0	RW	The ADO 2 had to repeat a sample of data.
ADO2_SLIP	28	0	RW	The ADO 2 had to slip a sample of data.
ARXDO2_SYT_OVERFLOW	27	0	RW	The timestamp FIFO of ARXDO 2 has overflowed.
ADO2_SYT_AGEOUT	26	0	RW	The ADO 2 had to age-out a stale frame of data.
ARX2_LONG_PKT	25	0	RW	The ARX 2 module received an isoch packet that was too long to store in its local memory.
ARX2_PKT_ABORT	24	0	RW	The ARX 2 was forced to abort an isoch packet due to an error.
ARX2_STATUS_ERR	23	0	RW	The ARX 2 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
ADO2_NOT_COMPL	22	0	RW	The package transfer didn't complete before the end of the cycle
ADO2_MISSED_SYNC	21	0	RW	The missing time stamp was detected
reserved	20	0	R	Reads back as zerro
ARX2_CFG_FAIL	19	0	RW	The ARX 2 module detected that a "forced" value in the CFG registers did not match what the stream is actually sending.
ARX2_CIP_FAIL	18	0	RW	The ARX 2 module detected an error in the CIP format of the

				received stream.
ARX2_DBC_FAIL	17	0	RW	The ARX 2 module detected a discontinuity in the DBC of the received stream. This will always fire once when the stream starts up.
ARXDO2_SYT_UNDERFLOW	16	0	RW	The timestamp FIFO of ARXDO 2 has underflowed.
ADO1_STREAM_UNLOCK	15	0	RW	The lock status status in ARX1_ADO_CFG reg has changed from "lock" to "unlock"
ADO1_STREAM_LOCK	14	0	RW	The lock status in ARX1_ADO_CFG reg has changed from "unlocked" to "locked"
ADO1_REPEAT	13	0	RW	The ADO 1 had to repeat a sample of data.
ADO1_SLIP	12	0	RW	The ADO 1 had to slip a sample of data.
ARXDO1_SYT_OVERFLOW	11	0	RW	The timestamp FIFO of ARXDO 1 has overflowed.
ADO1_SYT_AGEOUT	10	0	RW	The ADO 1 had to age-out a stale frame of data.
ARX1_LONG_PKT	9	0	RW	The ARX 1 module received an isoch packet that was too long to store in its local memory.
ARX1_PKT_ABORT	8	0	RW	The ARX 1 was forced to abort an isoch packet due to an error.
ARX1_STATUS_ERR	7	0	RW	The ARX 1 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
ADO1_NOT_COMPL	6	0	RW	The package transfer didn't complete before the end of the cycle
ADO1_MISSED_SYNC	5	0	RW	The missing time stamp was detected
reserved	4	0	R	Reads back as zerro
ARX1_CFG_FAIL	3	0	RW	The ARX 1 module detected that a "forced" value in the CFG registers did not match what the stream is actually sending.
ARX1_CIP_FAIL	2	0	RW	The ARX 1 module detected an error in the CIP format of the received stream.
ARX1_DBC_FAIL	1	0	RW	The ARX 1 module detected a discontinuity in the DBC of the received stream. This will always fire once when the stream starts up.
ARXDO1_SYT_UNDERFLOW	0	0	RW	The timestamp FIFO of ARXDO 1 has underflowed.

This register is both readable and writeable, though write data isn't stored in the register. Rather, set bits in the status register matching set bits in the write data are cleared.

6.6.3 APBA_INT0_MASK

Address - 0xcf00 0140

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTERRUPT MASK BITS															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTERRUPT MASK BITS															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Interrupt Mask	31:0	0	RW	Interrupt mask bits 0: Ignore interrupt 1: Allow interrupt

This register is both readable and writeable, though write data isn’t stored in the register. Rather, set bits in the status register matching set bits in the write data are cleared.

6.6.4 APBA_INT1_STATUS

Address - 0xcf00 0144

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATX2_ STREAM_ END	ATX2_ STREAM_ START	ATXDI2_ SYT_ OVERFLOW	ATXDI2_ SYT_ UNDERFLOW	ATX2_ BOUNDRY_ ERR	ATX2_ FRAME_ AGEOUT	ATX2_PKT_ ABORT	reserved	ATX2_ STREAM_ END	ATX2_ STREAM_ START	ATXDI2_ SYT_ OVERFLOW	ATXDI2_ SYT_ UNDERFLOW	ATX2_ BOUNDRY_ ERR	ATX2_ FRAME_ AGEOUT	ATX2_PKT_ ABORT	reserved
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	

Name	Bits	Reset	Dir	Description
reserved	31:16	0	RW	Reads back as zerroes
ATX2_STREAM_END	15	0	RW	The data stream transmitted by ATX 2 has ended.
ATX2_STREAM_START	14	0	RW	The data stream transmitted by ATX 2 has started.
ATXDI2_SYT_OVERFLOW	13	0	RW	The timestamp FIFO of ATXDI 2 has overflowed.
ATXDI2_SYT_UNDERFLOW	12	0	RW	The timestamp FIFO of ATXDI 2 has underflowed.
ATX2_BOUNDRY_ERR	11	0	RW	The ATX 2 encountered a problem with the isoch packet boundary as it was sending a packet.
ATX2_FRAME_AGEOUT	10	0	RW	The ATX 2 had to age-out a frame of data that was waiting to be sent because it became stale and was not transmitted in time.
ATX2_PKT_ABORT	9	0	RW	The ATX 2 encountered a problem and had to abort transmission of an isoch packet.
Reserved	8	0	R	Reads back as zerro
ATX1_STREAM_END	7	0	RW	The data stream transmitted by ATX 1 has ended.
ATX1_STREAM_START	6	0	RW	The data stream transmitted by ATX 1 has started.
ATXDI1_SYT_OVERFLOW	5	0	RW	The timestamp FIFO of ATXDI 1 has overflowed.
ATXDI1_SYT_UNDERFLOW	4	0	RW	The timestamp FIFO of ATXDI 1 has underflowed.
ATX1_BOUNDRY_ERR	3	0	RW	The ATX 1 encountered a problem with the isoch packet boundary as it was sending a packet.
ATX1_FRAME_AGEOUT	2	0	RW	The ATX 1 had to age-out a frame of data that was waiting to be sent because it became stale and was not transmitted in time.

ATX1_PKT_ABORT	1	0	RW	The ATX 1 encountered a problem and had to abort transmission of an isoch packet.
Reserved	0	0	R	Reads back as zerro

This register is both readable and writeable, though write data isn't stored in the register. Rather, set bits in the status register matching set bits in the write data are cleared.

6.6.5 APBA_INT1_MASK

Address - 0xcf00 0148

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTERRUPT MASK BITS															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTERRUPT MASK BITS															
Reset:	0															
	RW															

Name	Bits	Reset	Dir	Description
Interrupt Mask	31:0	0	RW	Interrupt mask bits 0: Ignore interrupt 1: Allow interrupt

This register is both readable and writeable, though write data isn't stored in the register. Rather, set bits in the status register matching set bits in the write data are cleared.

6.7 AVS Media FIFO

The AVS Media FIFO handles all buffering of Isoc. Stream data. The FIFO contains 4 partitions which can be allocated freely from the memory pool.

6.7.1 Module Configuration

Address	Register
0xcf00 0184	AVSFIFO_PART0_BASE
0xcf00 0188	AVSFIFO_PART0_LIMIT
0xcf00 018c	AVSFIFO_PART0_FLUSH
0xcf00 0190	AVSFIFO_PART1_BASE
0xcf00 0194	AVSFIFO_PART1_LIMIT
0xcf00 0198	AVSFIFO_PART1_FLUSH
0xcf00 019c	AVSFIFO_PART2_BASE
0xcf00 01a0	AVSFIFO_PART2_LIMIT
0xcf00 01a4	AVSFIFO_PART2_FLUSH
0xcf00 01a8	AVSFIFO_PART3_BASE
0xcf00 01ac	AVSFIFO_PART3_LIMIT
0xcf00 01b0	AVSFIFO_PART3_FLUSH
0xcf00 01fc	AVSFIFO_STAT

Table 72: AVS Media FIFO Memory Map

6.7.2 AVSFIFO_PARTn_BASE

Address - 0xcf00 0184

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED				BASE											
Reset:	0				0											
	RW				RW											

Name	Bits	Reset	Dir	Description
Base	11:0	0	RW	The lowest RAM address at which this partition can store data.

The BASE register is both readable and writeable.

6.7.3 AVSFIFO_PART0_LIMIT

Address - 0xcf00 0188

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED				LIMIT											
Reset:	0				0											
	RW				RW											

Name	Bits	Reset	Dir	Description
Limit	11:0	0	RW	The highest RAM address at which this partition can store data

The LIMIT register is both readable and writeable.

6.7.4 AVSFIFO_PART0_FLUSH

Address - 0xcf00 018c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	W															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED															
Reset:	0															
	W															

Name	Bits	Reset	Dir	Description
Flush	31:0	0	W	This register does not store data.

The FLUSH register is write-only.

6.7.5 AVSFIFO_STAT

Address - 0xcf00 01fc

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	R															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARTITION7 OVER FLOW	PARTITION7 UNDER FLOW	PARTITION6 OVER FLOW	PARTITION6 UNDER FLOW	PARTITION5 OVER FLOW	PARTITION5 UNDER FLOW	PARTITION4 OVER FLOW	PARTITION4 UNDER FLOW	PARTITION3 OVER FLOW	PARTITION3 UNDER FLOW	PARTITION2 OVER FLOW	PARTITION2 UNDER FLOW	PARTITION1 OVER FLOW	PARTITION1 UNDER FLOW	PARTITION0 OVER FLOW	PARTITION0 UNDER FLOW
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Name	Bits	Reset	Dir	Description
Partition 7 overflow	15	0	R	Partition 7 overflow
Partition 7 underflow	14	0	R	Partition 7 underflow
Partition 6 overflow	13	0	R	Partition 6 overflow
Partition 6 underflow	12	0	R	Partition 6 underflow
Partition 5 overflow	11	0	R	Partition 5 overflow
Partition 5 underflow	10	0	R	Partition 5 underflow
Partition 4 overflow	9	0	R	Partition 4 overflow
Partition 4 underflow	8	0	R	Partition 4 underflow
Partition 3 overflow	7	0	R	Partition 3 overflow
Partition 3 underflow	6	0	R	Partition 3 underflow
Partition 2 overflow	5	0	R	Partition 2 overflow
Partition 2 underflow	4	0	R	Partition 2 underflow
Partition 1 overflow	3	0	R	Partition 1 overflow
Partition 1 underflow	2	0	R	Partition 1 underflow
Partition 0 overflow	1	0	R	Partition 0 overflow
Partition 0 underflow	0	0	R	Partition 0 underflow

The MFIFO_STATUS register is read-only and cleared on read.

6.8 AVS MIDI Interface

The AVS MIDI interface consist of one receive buffer handling MIDI data from all 4 Isoc. Receivers, and two transmit buffers, one for each Isoc. Transmitter.

6.8.1 Module Configuration

Address	Register
0xcf00 01e4	AVSMIDI_STAT
0xcf00 01e8	AVSMIDI_CTRL
0xcf00 01ec	AVSMIDI_RX
0xcf00 01f0	AVSMIDI_TX0
0xcf00 01f4	AVSMIDI_TX1

Table 73: AVS MIDI Memory Map

6.8.2 AVSMIDI_STAT

Address - 0xcf00 01e4

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED					RX_QUADS_IN_BUF					BUF_FULL_RX	BUF_EMPTY_RX	BUF_FULL_TX1	BUF_EMPTY_TX1	BUF_FULL_TX0	BUF_EMPTY_TX0
Reset:	0					0					0	0	0	0	0	0
	RW					RW					RW	RW	RW	RW	RW	RW

Name	Bits	Reset	Dir	Description
rx_quads_in_buf	10:6	0	RW	current number of quadlets in the Rx buffer
buf_full_rx	5	0	RW	Rx buffer is full
buf_empty_rx	4	0	RW	Rx buffer is empty
buf_full_tx1	3	0	RW	Tx1 buffer is full
buf_empty_tx1	2	0	RW	Tx1 buffer is empty
buf_full_tx0	1	0	RW	Tx0 buffer is full
buf_empty_tx0	0	0	RW	Tx0 buffer is empty

6.8.3 AVSMIDI_CTRL

Address - 0xcf00 01e8

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RESERVED															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED													RX_IRQ_EN	TX1_IRQ_EN	TX0_IRQ_EN
Reset:	0													0	0	0
	RW													RW	RW	RW

Name	Bits	Reset	Dir	Description
rx_irq_en	2	0	RW	Rx interrupt enable
tx1_irq_en	1	0	RW	Tx1 interrupt enable
tx0_irq_en	0	0	RW	Tx0 interrupt enable

The CTRL register is both readable and writable.

6.8.4 AVSMIDI_RX

Address - 0xcf00 01ec

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIDI_QUADLET_RX															
Reset:	0															
	R															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MIDI_QUADLET_RX															
Reset:	0															
	R															

Name	Bits	Reset	Dir	Description
MIDI quadlet data	31:0	0	R	MIDI quadlet data with the following format: [31:29] MIDI port mapping [28:27] source MIDI machine number (AVS Rx0-3) [26] '0' [25:24] counter (number of valid bytes) [23:16] MIDI byte 1 [15:8] MIDI byte 2 [7:0] MIDI byte 3 This format is similar to the one defined in IEC 61883-6.

The RX register is read-only.

6.8.5 AVSMIDI_TXn

Address - 0xcf00 01f0

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIDI_QUADLET_TX															
Reset:	0															
	W															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MIDI_QUADLET_TX															
Reset:	0															
	W															

Name	Bits	Reset	Dir	Description
MIDI quadlet data	31:0	0	W	MIDI quadlet data with the following format: [31:29] MIDI port mapping [28:26] '000' [25:24] counter [23:16] MIDI byte 1 [15:8] MIDI byte 2 [7:0] MIDI byte 3 This format is similar to the one defined in IEC 61883-6.

The TXn register is write-only.

6.9 AVS General

6.9.1 Module Configuration

Address	Register
0xc800 0000	PDB_INT (AVC_CTRL)

Table 74: AVS General Memory Map

6.9.2 PDB_INT (AVC_CTRL)

Address - 0xc800 0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset:	0															
	RW															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													IRX_RUN	ITX_RUN	AVS_RUN
Reset:	0													0	0	0
	RW													RW	RW	RW

Name	Bits	Reset	Dir	Description
irx_run	2	0	RW	Activate the AVS isochronous receive interface.
itx_run	1	0	RW	Activate the AVS isochronous transmit interface.
avs_run	0	0	RW	Activate the AVS.

Chapter 7 Crystal Oscillator

TCD22XX, like most digital chips, contains an on-board oscillator. The ARM7 RISC is clocked by this oscillator, as well as the other internal functions (DICE Router, start up state machines, etc.). The on-chip oscillator itself is not really an oscillator, but is an amplifier suitable for being used as the feedback amplifier in an oscillator circuit with off-chip components (crystal or ceramic resonator, resistors and capacitors). The figure below shows the typical connections to TCD22XX.

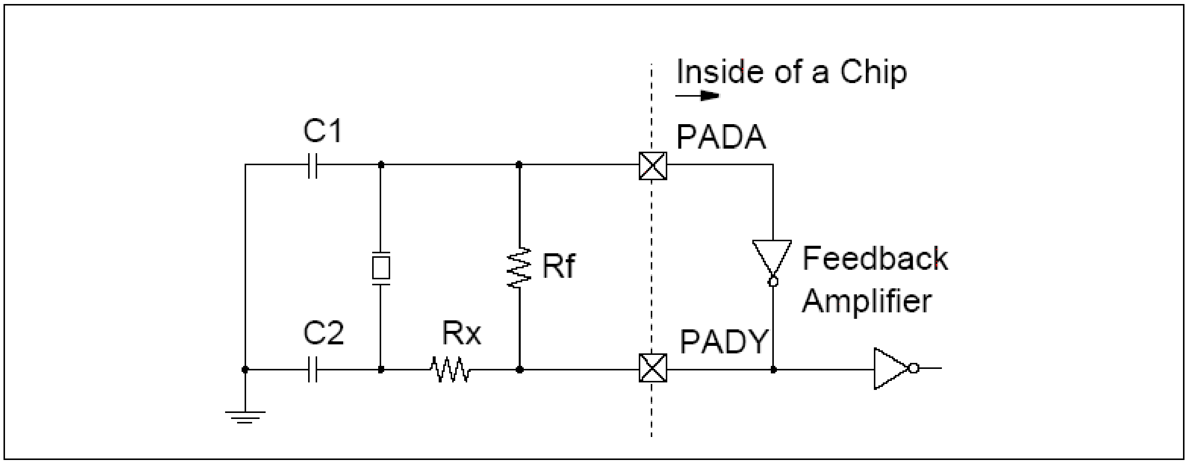


Figure 47: On-Chip oscillator typical connections

The external components commonly used for the oscillator circuit are a positive reactance (normal crystal oscillator), two capacitors, C1 and C2, and two resistors, Rf and Rx.

7.1.1 Crystal Specifications

Specifications for an appropriate crystal are not very critical. Any fundamental mode crystal of medium or better quality can be used. Crystal resistance affects start-up time and steady state amplitude but can be compensated by the choice of C1 and C2, however, the lower the crystal resistance, the better. A discussion of external R and C components follows below.

7.1.1.1 Oscillation Frequency

The oscillation frequency is mainly determined by the crystal. The on-chip oscillator has little effect on the frequency. The influence of the on-chip oscillator on frequency results from its input and output (pin-to-ground) capacitances which parallel C1 and C2, and the PADA-to- PADY (pin-to-pin) capacitance which parallels the crystal. The input and pin-topin capacitances are about 7pF each.

7.1.1.2 C1 and C2 Selection

Optimal values for C1 and C2 depend on whether a quartz crystal or ceramic resonator is used, and on application-specific requirements for start-up time and frequency tolerance. Start-up time is sometimes more critical in microcontroller systems than frequency stability because of various reset and initialization requirements. Accuracy of the oscillator frequency is less commonly critical, as when the oscillator is being used as a time base. As a general rule, fast start-up and stable frequency tend to pull the oscillator design in opposite directions. Considerations of both start-up time and frequency stability over temperature suggest that C1 and C2 should be about equal and at least 15pF (but they don’t have to be either). Increasing the value of these capacitors above 40pF or 50pF improves frequency stability, but also increases the start-up time. If the capacitors are too large (several hundred pF), the oscillator won’t start up at all.

7.1.1.3 Rf and Rx Selection

A large Rf (1M&) holds the on-chip oscillator (a CMOS inverter) in its linear region allowing it to oscillate. The inverter has a fairly low output resistance which destabilizes the oscillator circuit. Rx of several k& is added to the feedback network, as shown in the Figure, to stabilize the oscillator circuit. At higher oscillator frequencies, a 20pF or 30pF capacitor is sometimes used in place of Rx to compensate for the internal propagation delay.

7.1.1.4 PCB CONSIDERATIONS

Noise glitches arising at PADA or PADY pins at the wrong time can cause a miscount in the internal clock-generating circuitry. These kinds of glitches can be produced through capacitive coupling between the oscillator components and PCB traces carrying digital signals with fast rise and fall times. For this reason, the oscillator components should be mounted close to the chip and have short, direct traces to the PADA, PADY, and VSS pins. If possible, use dedicated VDD and VSS pins for the on-chip oscillator. In addition, surrounding oscillator components with “quiet” traces (VDD and VSS) will alleviate capacitive coupling to signals having fast edges. To minimize inductive coupling, the PCB layout should minimize lead, wire, and trace lengths for oscillator components.

Chapter 8 Electrical Characteristics

8.1 DC Characteristics

1.8v Core supply measured at 1.8v and ambient temperature of 20 deg.

Condition	**min**	**typ**	**Max**	**Comment**
nReset = 0v		20 mA		System Reset
Power Down		2 mA		Prepared to wake on LinkOn
ARM, no audio		97 mA		Audio subsystem not started
Normal Operation		100 mA	180mA	96KHz AES and 1394 Audio

3.3v Core supply measured at 3.3v and ambient temperature of 20 deg.

Condition	**Min**	**typ**	**Max**	**Comment**
nReset = 0v		-		System Reset
Power Down		-		Prepared to wake on LinkOn
ARM, no audio		-		Audio subsystem not started
Normal Operation		115 mA	200 mA	96KHz AES and 1394 Audio

The 3.3v supply consumption will depend on the actual loading of the outputs of the chip, these numbers are for a typical application such as the evaluation board with code executing from SDRAM.

8.1.1 3.3V DC Characteristics

Symbol	Parameter		Condition	Min	Typ.	Max	Unit
V_{IH}	High level input voltage LVCMOS interface			2.0			V
V_{IL}	Low level input voltage LVCMOS interface					0.8	
V_T	Switching threshold				1.4		
V_{T+}	Schmitt trigger, positive-going threshold		CMOS			2.0	
V_{T-}	Schmitt trigger, negative-going threshold		CMOS	0.8			
I_{IH}	High level input current	Input buffer	$V_{IN} = V_{DD}$	-10		10	µA
		Input buffer with pull-down		10	33	60	
I_{IL}	Low level input current	Input buffer	$V_{IN} = V_{SS}$	-10		10	
		Input buffer with pull-down		-60	-33	-10	
V_{OH}	High level output voltage	Type B1 to B24	I_{OH} = -1µA	V_{DD} – 0.05			V
		Type B1	I_{OH} = -1mA	2.4			
		Type B2	I_{OH} = -2mA				
		Type B4	I_{OH} = -4mA				
		Type B8	I_{OH} = -8mA				
		Type B12	I_{OH} = -12mA				
		Type B16	I_{OH} = -16mA				
		Type B20	I_{OH} = -20mA				
		Type B24	I_{OH} = -24mA				
V_{OL}	Low level output voltage	Type B1 to B24	I_{OH} = 1µA			0.05	
		Type B1	I_{OH} = 1mA			0.4	
		Type B2	I_{OH} = 2mA				
		Type B4	I_{OH} = 4mA				
		Type B8	I_{OH} = 8mA				
		Type B12	I_{OH} = 12mA				
		Type B16	I_{OH} = 16mA				

Symbol	Parameter		Condition	Min	Typ.	Max	Unit
		Type B20	I_{OH} = 20mA				
		Type B24	I_{OH} = 24mA				
I_{OZ}	Tri-state output leakage current		V_{OUT} = V_{DD} or V_{SS}	-10		10	µA
I_{DD}	Quiescent supply current					100	
C_{IN}	Input capacitance		Any input and bi-directional buffers			4	pF
C_{OUT}	Output capacitance		Any output buffer				

8.1.2 Absolute Maximum Ratings

Symbol	Parameter	Rating			Unit
			Min	Max	
V_{DD}	DC supply voltage	1.8V V_{DD}	-0.5	2.7	V
		3.3V V_{DD}	-0.5	4.8	
V_{IN}	DC input voltage	3.3V input buffer	-0.5	4.8	
		3.3V interface/ 5V tolerant input buffer	-0.5	6.5	
V_{OUT}	DC output voltage	3.3V output buffer	-0.5	4.8	
		3.3V interface/ 5V tolerant output buffer	-0.5	6.5	
I_{IO}	Input/Output current	± 20			mA
T_A	Storage temperature	-65 to 150			°C

8.1.3 Recommended Operating Conditions

Symbol	Parameter	Rating			Unit
			Min	Max	
V_{DD}	DC supply voltage for internal (=V_{DDIN})	1.8V V_{DD}	1.65	1.95	V
	DC supply voltage for I/O block (=V_{DDIO})	3.3V V_{DD}	3.0	3.6	
	DC supply voltage for analog core (=V_{DDA})	1.8V V_{DD}	1.8 – 5%	1.8 + 5%	
V_{IN}	DC input voltage	3.3V input buffer	-0.3	V_{DCIO} + 0.3	
		3.3V interface/ 5V tolerant input buffer	-0.3	5.5	
V_{OUT}	DC output voltage	3.3V output buffer	-0.3	V_{DCIO} + 0.3	
		3.3V interface/ 5V tolerant output buffer	-0.3	5.5	
T_A	Commercial temperature range			0 to 70	°C
	Industrial temperature range			-40 to 85	

8.2 PLL Characteristics

8.2.1 Recommended Operating Conditions

Characteristics	Symbol	Min	Typ	Max	Unit
Supply voltage differential	AVDD18D/ AVDD18A	−0.1	–	0.1 V	V
Operating temperature	Topr	−40	–	85	°C

8.2.2 DC Electrical Characteristics

Characteristics	Symbol	Min	Typ	Max	Unit
Operating voltage	AVDD18D/ AVDD18A	1.65	1.8	1.95	V
Digital input voltage high	IIH	0.7VDD	–	–	V
Digital input voltage low	IIL	–	–	0.3VDD	V
Dynamic current	IDD	–	–	3	mA
Power down current	IPD	–	–	220	uA

8.2.3 AC Electrical Characteristics

Characteristics		Symbol	Min	Typ	Max	Unit
Input frequency		FIN	4	–	50	MHz
Output clock frequency		FOUT	20	–	300	MHz
VCO output frequency		FVCO	160	–	400	MHz
Input clock duty cycle		TID	40	–	60	%
Output clock duty cycle		TOD	45	–	55	%
Locking time		TLT	–	–	150	us
Cycle to cycle jitter	20M ~100MHz	TJCC	–300	–	300	ps
	100M ~ 200MHz	TJCC	–200	–	200	ps
	200M ~ 300MHz	TJCC	–120	–	120	ps

Chapter 9 Thermal Ratings

Operating Temperature

	MIN	TYP	MAX	UNIT
Operating ambient temperature, TCD22xx	0		70	**°C**
Operating ambient temperature, TCD22xx-E	-40		85	**°C**

Thermal Characteristics

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
128 QFP	Board mounted, no air flow		46.4		**°C** /W
144 LQFP	Board mounted, no air flow		45.3		**°C** /W

Absolute Maximum Ratings over Operating Temperature Ranges

Supply voltage range AV_{dd} (1.8V)	-0.5 V to 2.7 V
V_{dd} (3.3V)	-0.5 V to 4.8 V
PLL_V_{dd} (1.8V)	-0.5 V to 2.7 V
Continuous total power dissipation	See Dissipation Rating Table
Operating free–air temperature T_A – (TCD22xx)	0 **°C** to 70 **°C**
Operating free–air temperature T_A – (TCD22xx-E)	-40 **°C** to 85 **°C**
Storage temperature range T_{stg} -	-65 **°C** to 150 **°C**

Exposure to absolute–maximum–rated conditions for extended periods affects device reliability. Stresses beyond those listed under absolute maximum ratings cause permanent damage to the device.

Appendix 1. Memory Map and Register summary

A.1 Memory map

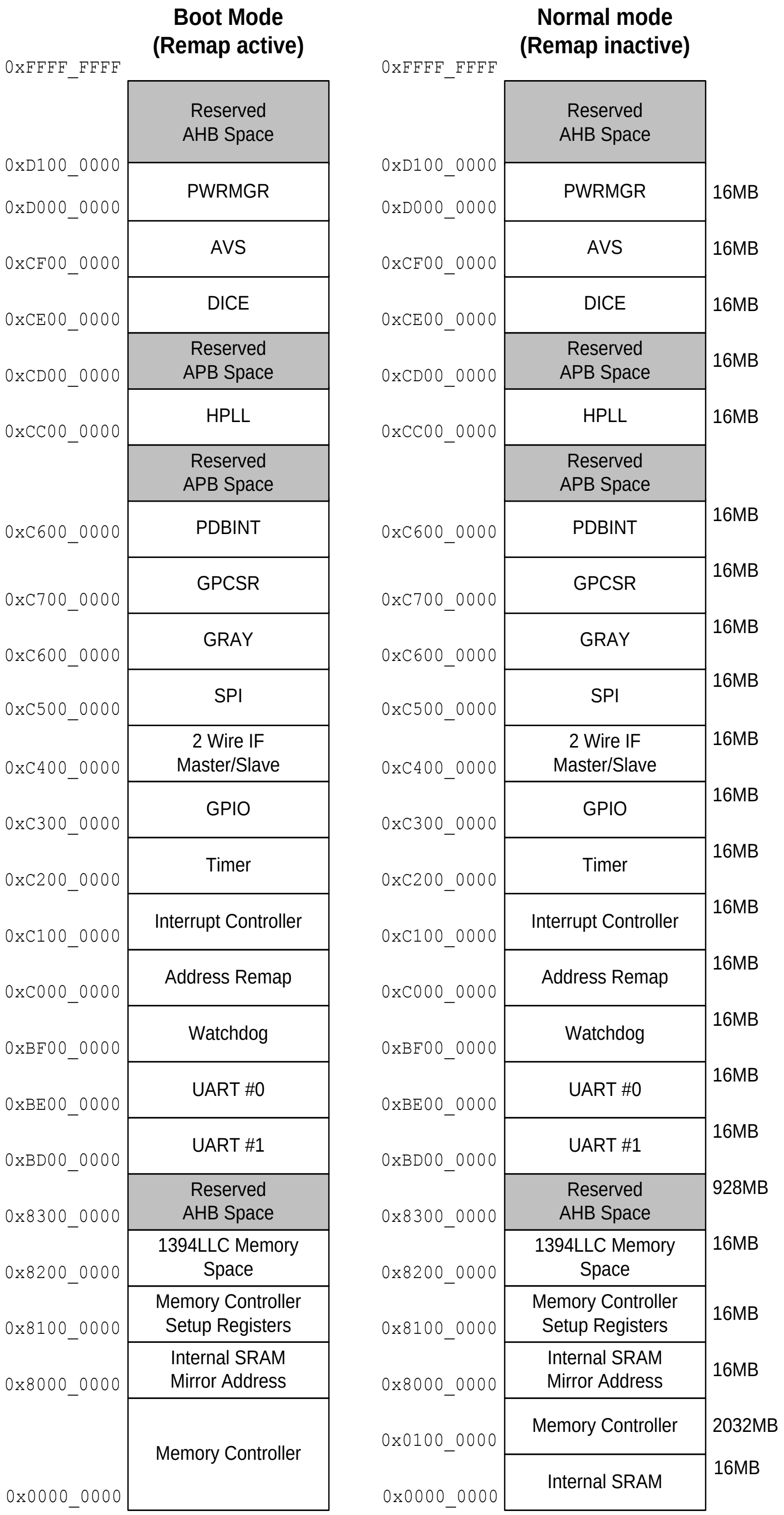


Figure 48. Global Memory Map (allocated Address Space)

A.2 DICE JR Register Summary

A.2.1 ARM Peripherals

EBI (External Bus Interface) Memory Map	See Chapter 4 ARM Peripherals
Address	Register
0x8100 0000	EBI_SCONR
0x8100 0004	EBI_STMG0R
0x8100 0008	EBI_STMG1R
0x8100 000c	EBI_SCTLR
0x8100 0010	EBI_SREFR
0x8100 0014	EBI_SCSLR0
0x8100 0018	EBI_SCSLR1
0x8100 001c	EBI_SCSLR2
0x8100 0020	EBI_SCSLR3
0x8100 0024	EBI_SCSLR4
0x8100 0028	EBI_SCSLR5
0x8100 002c	EBI_SCSLR6
0x8100 0030	EBI_SCSLR7
0x8100 0054	EBI_SMSKR0
0x8100 0058	EBI_SMSKR1
0x8100 005c	EBI_SMSKR2
0x8100 0060	EBI_SMSKR3
0x8100 0064	EBI_SMSKR4
0x8100 0068	EBI_SMSKR5
0x8100 006c	EBI_SMSKR6
0x8100 0070	EBI_SMSKR7
0x8100 0074	EBI_CSALIAS0
0x8100 0078	EBI_CSALIAS1
0x8100 0084	EBI_CSREMAP0
0x8100 0088	EBI_CSREMAP1
0x8100 0094	EBI_SMTMGR_SET0
0x8100 0098	EBI_SMTMGR_SET1
0x8100 009c	EBI_SMTMGR_SET2
0x8100 00a0	EBI_FLASH_TRPDR
0x8100 00a4	EBI_SMCTLR

1394 LLC Memory Map	See Chapter 4 – ARM Peripherals
Address	Register
0x8200 0000	VERSION_REG_DP
0x8200 0004	ND_ID_REG_DP
0x8200 0008	LNK_CTRL_REG_DP
0x8200 000c	LCSR_REG_DP
0x8200 0010	CY_TMR_REG_DP
0x8200 0014	ATFIFO_STAT_REG_DP
0x8200 0018	ITFIFO_STAT_REG_DP
0x8200 001c	ARFIFO_STAT_REG_DP
0x8200 0020	IRFIFO_STAT_REG_DP
0x8200 0024	ISOC_RX_ENB_REG_1_DP
0x8200 0028	ISOC_RX_ENB_REG_2_DP
0x8200 002c	ISO_TX_STAT_REG_DP
0x8200 0030	ASY_TX_STAT_REG_DP
0x8200 0044	PHY_CTRL_REG_DP
0x8200 0048	INTERRUPT_REG_SET_DP
0x8200 004c	INTERRUPT_REG_CLEAR_DP
0x8200 0050	INTR_MASK_REG_SET_DP
0x8200 0054	INTR_MASK_REG_CLEAR_DP
0x8200 0058	DIAG_REG_DP
0x8200 005c	BUS_STAT_REG_DP
0x8200 0060	ASY_TX_FIFO_SPACE_REG_DP
0x8200 0064	ASY_RX_FIFO_QLETS_REG_DP
0x8200 0068	ISO_TX_FIFO_SPACE_REG_DP
0x8200 006c	ISO_RX_FIFO_QLETS_REG_DP
0x8200 0070	ISO_DATA_PATH_REG_DP
0x8200 0074	ASY_TX_FIRST_REG_DP
0x8200 0078	ASY_CONTINUE_REG_DP
0x8200 007c	ASY_CONTINUE_UPDATE_REG_DP
0x8200 0080	ASY_TX_FIFO_DEPTH_REG_DP
0x8200 0084	ASY_RX_FIFO_REG_DP
0x8200 0088	ASY_RX_FIFO_DEPTH_REG_DP
0x8200 008c	ISO_TX_FIRST_REG_DP
0x8200 0090	ISO_CONTINUE_REG_DP
0x8200 0094	ISO_CONTINUE_UPDATE_REG_DP
0x8200 0098	ISO_TX_FIFO_DEPTH_REG_DP
0x8200 009c	ISO_RX_FIFO_REG_DP
0x8200 00a0	ISO_RX_FIFO_DEPTH_REG_DP
0x8200 00a4	HST_ACC_ERR_REG_DP
0x8200 00a8	RET_CT_REG_DP

1394 LLC Memory Map	See Chapter 4 – ARM Peripherals
Address	Register
0x8200 00ac	DIG_FSM_STAT_REG
0x8200 00b0	ISO_TX_ENB_REG_1_DP
0x8200 00b4	ISO_TX_ENB_REG_2_DP
0x8200 00b8	ISO_HDR_REG_DP
0x8200 00bc	LPS_REG_DP
0x8200 00c0	PING_REG_DP
0x8200 00c4	ISOC_EXPC_CHAN_REG1
0x8200 00c8	ISOC_EXPC_CHAN_REG2
0x8200 00cc	DUP_EXPC_STAT_REG
0x8200 00d0	ASYN_RX_ENB_REG_1_DP
0x8200 00d4	ASYN_RX_ENB_REG_2_DP

UART #1 Memory map	See Chapter 4 – Arm Peripherals
Address	Register
0xbd00 0000	UART#1 RBR, THR, DLL[a]
0xbd00 0004	UART#1 IER, DLH[a]
0xbd00 0008	UART#1 IIR, FCR
0xbd00 000c	UART#1 LCR
0xbd00 0010	UART#1 MCR
0xbd00 0014	UART#1 LSR
0xbd00 0018	UART#1 MSR
0xbd00 001c	UART#1 SCR

UART #0 Memory map	See Chapter 4 – Arm Peripherals
Address	Register
0xbe00 0000	UART#0 RBR, THR, DLL
0xbe00 0004	UART#0 IER, DLH
0xbe00 0008	UART#0 IIR, FCR
0xbe00 000c	UART#0 LCR
0xbe00 0010	UART#0 MCR
0xbe00 0014	UART#0 LSR
0xbe00 0018	UART#0 MSR
0xbe00 001c	UART#0 SCR

Watchdog Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xbf00 0000	WD_RESET_EN
0xbf00 0004	WD_INT

Watchdog Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xbf00 0008	WD_PRESCALE_LOAD
0xbf00 000c	WD_PRESCALE_CNT
0xbf00 0010	WD_COUNT

Address remap memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc000 0004	DICE Family ID Register
0xc000 0008	Remap Register

Interrupt Controller Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc100 0000	INTCTRL_ENABLE
0xc100 0008	INTCTRL_MASK
0xc100 0010	INTCTRL_FORCE
0xc100 0018	INTCTRL_RAW
0xc100 0020	INTCTRL_STAT
0xc100 0028	INTCTRL_MASKSTAT
0xc100 0030	INTCTRL_FINALSTAT
0xc100 0038	INTCTRL_INTVECTOR
0xc100 0040	INTCTRL_VECTOR0
0xc100 0048	INTCTRL_VECTOR1
0xc100 0050	INTCTRL_VECTOR2
0xc100 0058	INTCTRL_VECTOR3
0xc100 0060	INTCTRL_VECTOR4
0xc100 0068	INTCTRL_VECTOR5
0xc100 0070	INTCTRL_VECTOR6
0xc100 0078	INTCTRL_VECTOR7
0xc100 0080	INTCTRL_VECTOR8
0xc100 0088	INTCTRL_VECTOR9
0xc100 0090	INTCTRL_VECTOR10
0xc100 0098	INTCTRL_VECTOR11
0xc100 00a0	INTCTRL_VECTOR12
0xc100 00a8	INTCTRL_VECTOR13
0xc100 00b0	INTCTRL_VECTOR14
0xc100 00b8	INTCTRL_VECTOR15
0xc100 00c0	INTCTRL_FIQ_ENABLE
0xc100 00c4	INTCTRL_FIQ_MASK
0xc100 00c8	INTCTRL_FIQ_FORCE

Interrupt Controller Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc100 00cc	INTCTRL_FIQ_RAW
0xc100 00d0	INTCTRL_FIQ_STAT
0xc100 00d4	INTCTRL_FIQ_FINALSTAT
0xc100 00d8	INTCTRL_SYSTEM_PRIORITY_LEVEL

Dual Timer Memory Map	See Chapter 4 – Arm Peripherals
Address Range	Register
0xc200 0000 to 0xc200 0010	Timer 1 Registers
0xc200 0014 to 0xc200 0024	Timer 2 Registers
0xc200 00a0 to 0xc200 00a4	Timer System Registers

GPIO Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc300 0000	GPIO_DR
0xc300 0004	GPIO_DDR
0xc300 0030	GPIO_INTEN
0xc300 0034	GPIO_INTMSK
0xc300 0038	GPIO_INTSENSE
0xc300 003c	GPIO_INTPOL
0xc300 0040	GPIO_INTSTAT
0xc300 0044	GPIO_RAWINTSTAT
0xc300 0048	GPIO_DEBOUNCE
0xc300 004c	GPIO_EOI
0xc300 0050	GPIO_EXT
0x3c00 0060	GPIO_SYNC

I2C Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc400 0000	IC_CON
0xc400 0004	IC_TAR
0xc400 0008	IC_SAR
0xc400 000c	IC_HS_MAR
0xc400 0010	IC_DATA_COMMAND
0xc400 0014	IC_SS_HCNT
0xc400 0018	IC_SS_LCNT
0xc400 001c	IC_FS_HCNT
0xc400 0020	IC_FS_LCNT
0xc400 0024	IC_HS_HCNT

I2C Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc400 0028	IC_HS_LCNT
0xc400 002c	IC_INTR_STAT
0xc400 0030	IC_INTR_MASK
0xc400 0034	IC_RAW_INTR_STAT
0xc400 0038	IC_RX_TL
0xc400 003c	IC_TX_TL
0xc400 0040	IC_CLR_INTR
0xc400 0044	IC_CLR_RX_UNDER
0xc400 0048	IC_CLR_RX_OVER
0xc400 004c	IC_CLR_TX_OVER
0xc400 0050	IC_CLR_RD_REQ
0xc400 0054	IC_CLR_TX_ABRT
0xc400 0058	IC_CLR_RX_DONE
0xc400 005c	IC_CLR_ACTIVITY
0xc400 0060	IC_CLR_STOP_DET
0xc400 0064	IC_CLR_START_DET
0xc400 0068	IC_CLR_GEN_CALL
0xc400 006c	IC_ENABLE
0xc400 0070	IC_STATUS
0xc400 0074	IC_TXFLR
0xc400 0078	IC_RXFLR
0xc400 007c	IC_SRESET
0xc400 0080	IC_TX_ABRT_SOURCE

SPI Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc500 0000	SPI Control register
0xC500 0004	SPI Status Register
0xC500 0008	SPI Interrupt Mask Register
0xC500 0010	SPI Data Register
0xC500 0014	SPI Baud Rate Register

GRAY Encoder Memory Map	See Chapter 4 – Arm Peripherals
Address	Register
0xc600 0000	GRAY_STAT
0xc600 0004	GRAY_CTRL
0xc600 0008	GRAY_CNT

GPCSR (General Purpose CSR) Memory Map	See Chapter 4 ARM Peripherals
Address	Register
0xc700 0000	GPCSR_SYSTEM
0xC700 0004	GPCSR_AUDIO_SELECT
0xC700 0008	GPCSR_GPIO_SELECT
0xC700 0014	GPCSR_CHIP_ID
0xC700 0024	GPCSR_IRQ_SEL0_5
0xC700 0028	GPCSR_IRQ_SEL6_11
0xC700 002c	GPCSR_IRQ_SEL12_17
0xC700 0030	GPCSR_IRQ_SEL18
0xC700 0034	GPCSR_FIQ_SEL0_5
0xC700 0038	GPCSR_FIQ_SEL6_7

A.2.2 DICE

Jet^{TM} PLL Memory map	See Chapter 5 - DICE
Address	Register
0xcc00 0000	PLL1_CAF_ENABLE
0xcc00 0004	PLL1_CAF_SELECT
0xcc00 0008	PLL1_COAST
0xcc00 0018	PLL1_REF_SEL
0xcc00 001c	PLL1_REF_EDG
0xcc00 0028	PLL1_RDIV
0xcc00 002c	PLL1_THROTTLE
0xcc00 0058	PLL1_U_THRESHOLD
0xcc00 0060	PLL1_BW_FLOOR
0xcc00 0064	PLL1_BW_CEILING
0xcc00 0068	PLL1_SHP_FIX
0xcc00 006c	PLL1_SHP_VAR
0xcc00 0070	PLL1_MAX_SLW_FIX
0xcc00 0074	PLL1_MAX_SLW_VAR
0xcc00 0078	PLL1_DCNT_LIN
0xcc00 007c	PLL1_DCNT_EXP
0xcc00 0088	PLL1_LOOSE_THR
0xcc00 0098	PLL1_MIN_PER
0xcc00 009c	PLL1_MAX_PER
0xcc00 00b0	PLL1_NDIV_F
0xcc00 00b4	PLL1_NDIV_E

Jet^{TM} PLL Memory map	See Chapter 5 - DICE
Address	Register
0xcc00 00b8	PLL1_NDIV_B
0xcc00 00bc	PLL1_BYP_F
0xcc00 00c0	PLL1_PHASE_LAG
0xcc00 00c8	PLL1_FRACT_RES
0xcc00 00d0	PLL1_BURST_LEN
0xcc00 00d8	PLL1_GPO_EN
0xcc00 00dc	PLL1_GPO_1
0xcc00 00e0	PLL1_GPO_2
0xcc00 00e4	PLL1_GPO_3
0xcc00 00f0	PLL1_X1X2_MODE
0xcc000100	PLL1_CHAIN_I
0xcc000104	PLL1_SINK_I
0xcc000108	PLL1_ANCHOR_I
0xcc00010c	PLL1_IANCHOR_VAL
0xcc000110	PLL1_UNBND_I
0xcc000118	PLL1_IDET
0xcc000120	PLL1_IDIV_C
0xcc000124	PLL1_IDIV_F
0xcc000128	PLL1_IDIV_S
0xcc000130	PLL1_INV_CDI
0xcc000134	PLL1_HBL_CDI
0xcc000144	PLL1_SINK_E
0xcc000148	PLL1_ANCHOR_E
0xcc00014c	PLL1_E_ANC_VAL
0xcc000150	PLL1_UNBIND_E
0xcc000158	PLL1_EDET_X1
0xcc00015c	PLL1_EDET_X2
0xcc000160	PLL1_EDIV_C
0xcc000164	PLL1_EDIV_F
0xcc000168	PLL1_EDIV_S
0xcc000170	PLL1_INV_CDE
0xcc000174	PLL1_HBL_CDE
0xcc000180	PLL1_DIVIDE_CJ
0xcc000184	PLL1_INVERT_CJ
0xcc000280	PLL1_FAMILY_ID
0xcc000284	PLL1_FORM_ID
0xcc000288	PLL1_REVISION_ID

Jet^{TM} PLL Memory map	See Chapter 5 - DICE
Address	Register
0xcc00028c	PLL1_INSTANCE_ID
0xcc0002b8	PLL1_MTR_SELECT
0xcc0002bc	PLL1_MTR_EDGES
0xcc0002c0	PLL1_RES_EX
0xcc0002c4	PLL1_PUNC_MP
0xcc0002cc	PLL1_MTR_PERIOD
0xcc0002d0	PLL1_GREATEST_MP
0xcc0002d4	PLL1_GREATEST_MP_$
0xcc0002d8	PLL1_SMALLEST_MP
0xcc0002dc	PLL1_SMALLEST_MP_$
0xcc000300	PLL1_TICK_RATE
0xcc000304	PLL1_TURN_RATE
0xcc000308	PLL1_MAIN_STATUS
0xcc00030c	PLL1_MAIN_STATUS_$
0xcc000320	PLL1_DETECT_R
0xcc000324	PLL1_DETECT_F
0xcc000328	PLL1_STICKY_BITS
0xcc00032c	PLL1_STICKY_BITS_$
0xcc000350	PLL1_IRQ_ENABLES
0xcc00038c	PLL1_NCO_PERIOD
0xcc000390	PLL1_GREATEST_NP
0xcc000394	PLL1_GREATEST_NP_$
0xcc000398	PLL1_SMALLEST_NP
0xcc00039c	PLL1_SMALLEST_NP_$
0xcc0003d8	PLL1_GPI
0xcc0003e0	PLL1_CONFIG_AC
0xcc0003f0	PLL1_SHUTDOWN_M
0xcc0003f4	PLL1_SHUTDOWN_I
0xcc0001f8	PLL1_SHUTDOWN_E

Router Memory Map	See Chapter 5 - DICE
Address	Register
0xce00 0000	ROUTER_CTRL
0xce00 0400	ROUTER_ENTRY0
0xce00 0404	ROUTER_ENTRY1
:	:

Router Memory Map	See Chapter 5 - DICE
Address	Register
0xce00 07fc	ROUTER_ENTRY127

Clock Controller Memory Map	See Chapter 5 - DICE
Address	Register
0xce01 0000	SYNC_CTRL
0xce01 0004	DOMAIN_CTRL
0xce01 0008	EXTCLK_CTRL
0xce01 000c	BLK_CTRL
0xce01 0010	REFEVENT_CTRL
0xce01 0014	SRCNT_CTRL
0xce01 0018	SRCNT_MODE
0xce01 001c	Reserved
0xce01 0020	Reserved
0xce01 0024	AES_VCO_SETUP
0xce01 0028	Reserved
0xce01 002c	Reserved
0xce01 0030	Reserved
0xce01 0034	PRESCALER
0xce01 0038	Reserved
0xce01 003c	HPLL_REF
0xce01 0040	SRCNT1
0xce01 0044	SRCNT2
0xce01 0048	SR_MAX_CNT1
0xce01 004c	SR_MAX_CNT2

AES Receiver Memory Map	See Chapter 5 - DICE
Address	Register
0xce02 0000	CTRL
0xce02 0004	STAT_ALL
0xce02 0008	STAT_RX0
0xce02 000c	STAT_RX1
0xce02 0010	STAT_RX2
0xce02 0014	STAT_RX3
0xce02 0018	V_BIT
0xce02 0040	PLL_PULSE_WIDTH
0xce02 0044	FORCE_VCO
0xce02 0048	VCO_MIN_LSB
0xce02 004c	VCO_MIN_MSB

AES Receiver Memory Map	See Chapter 5 - DICE
Address	Register
0xce02 0080	CHSTAT_0_BYTE0
0xce02 0084	CHSTAT_0_BYTE1
0xce02 0088	CHSTAT_0_BYTE2
0xce02 008c	CHSTAT_0_BYTE3
0xce02 0090 - 0xce02 009c	CHSTAT_1_BYTE0-3
0xce02 00a0 - 0xce02 00ac	CHSTAT_2_BYTE0-3
0xce02 00b0 - 0xce02 00bc	CHSTAT_3_BYTE0-3
0xce02 00c0 - 0xce02 00cc	CHSTAT_4_BYTE0-3
0xce02 00d0 - 0xce02 00dc	CHSTAT_5_BYTE0-3
0xce02 00e0 - 0xce02 00ec	CHSTAT_6_BYTE0-3
0xce02 00f0 - 0xce02 00fc	CHSTAT_7_BYTE0-3
0xce02 0100 – 0xce02 015c	CHSTAT_FULL_BYTE0-23

AES Transmitter Memory map	See Chapter 5 - DICE
Address	Register
0xce03 0000	MODE_SEL
0xce03 0004	CBL_SEL
0xce03 0008	CS_SEL0
0xce03 000c	CS_SEL1
0xce03 0010	CS_SEL2
0xce03 0014	MUTE
0xce03 0018	V_BIT
0xce03 0040	USR_SEL0
0xce03 0044	USR_SEL1
0xce03 0048	USR_SEL2
0xce03 004c	USR_SEL3
0xce03 0080	CHSTAT_0_BYTE0
0xce03 0084	CHSTAT_0_BYTE1
0xce03 0088	CHSTAT_0_BYTE2
0xce03 008c	CHSTAT_0_BYTE3
0xce03 0090 - 0xce03 009c	CHSTAT_1_BYTE0-3
0xce03 00a0 - 0xce03 00ac	CHSTAT_2_BYTE0-3
0xce03 00b0 - 0xce03 00bc	CHSTAT_3_BYTE0-3
0xce03 00c0 - 0xce03 00cc	CHSTAT_4_BYTE0-3
0xce03 00d0 - 0xce03 00dc	CHSTAT_5_BYTE0-3
0xce03 00e0 - 0xce03 00ec	CHSTAT_6_BYTE0-3
0xce03 00f0 - 0xce03 00fc	CHSTAT_7_BYTE0-3
0xce03 0100 – 0xce03 015c	CHSTAT_FULL_BYTE0-23

ADAT Receiver Memory Map	See Chapter 5 - DICE
Address	Register
0xce04 0000	ADATRX0
0xce04 0004	ADATRX1

ADAT Transmitter	See Chapter 5 - DICE
Address	Register
0xce05 0000	ADATTX_CTRL1
0xce05 0004	ADATTX0_MUTE
0xce05 0008	ADATTX1_MUTE

Audio Mixer Memory Map	See Chapter 5 - DICE
Address	Register
0xce06 0000	MIXER_CTRL
0xce06 0004	MIXER_OVL
0xce06 0008	MIXER_NUMOFCH
0xce06 0800	MIXER_COEFF RAM Coeff 0 for channel 0
0xce06 0804	MIXER_COEFF RAM Coeff 1 for channel 0
0xce06 0808	MIXER_COEFF RAM Coeff 2 for channel 0
…	
0xce06 0c78	MIXER_COEFF RAM Coeff 16 for channel 15
0xce06 0c7c	MIXER_COEFF RAM Coeff 17 for channel 15

I^{N}S Receivers Memory Map	See Chapter 5 - DICE
Address	Register
0xce08 0000	INS0_RX0_SETUP
0xce08 0020	INS0_RX1_SETUP
0xce08 0040	INS0_RX2_SETUP
0xce08 0060	INS0_RX3_SETUP
0xce0a 0000	INS1_RX0_SETUP
0xce0a 0020	INS1_RX1_SETUP
0xce0a 0040	INS1_RX2_SETUP
0xce0a 0060	INS1_RX3_SETUP

I^{N}S Transmitters Memory Map	See Chapter 5 - DICE
Address	Register
0xce09 0000	INS0_TX0_SETUP
0xce09 0020	INS0_TX1_SETUP
0xce09 0040	INS0_TX2_SETUP
0xce09 0060	INS0_TX3_SETUP

I^{N}S Transmitters Memory Map	See Chapter 5 - DICE
Address	Register
0xce09 0080	INS0_CLKP_SETUP
0xce09 0fe0	INS0_MUTE
0xce0b 0000	INS1_TX0_SETUP
0xce0b 0020	INS1_TX1_SETUP
0xce0b 0040	INS1_TX2_SETUP
0xce0b 0060	INS1_TX3_SETUP
0xce0b 0080	INS1_CLKP_SETUP
0xce0b 0fe0	INS1_MUTE

ARM Audio Transeiver Memory Map	See Chapter 5 - DICE
Address	Register
0xce16 0000 – 0xce16 0080	ARMAUDIO_BUF
0xce16 0100	ARMAUDIO_CTRL

A.2.3 AVS

AVS Audio Receivers Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 0000	ARX1_CFG0
0xcf00 0004	ARX1_CFG1
0xcf00 0008	ARX1_QSEL0
0xcf00 000c	ARX1_QSEL1
0xcf00 0010	ARX1_QSEL2
0xcf00 0014	ARX1_QSEL3
0xcf00 0018	ARX1_QSEL4
0xcf00 001c	ARX1_PHDR
0xcf00 0020	ARX1_CIP0
0xcf00 0024	ARX1_CIP1
0xcf00 0028	ARX1_ADO_CFG
0xcf00 002c	ARX1_ADO_MIDI
0xcf00 0030	ARX2_CFG0
0xcf00 0034	ARX2_CFG1
0xcf00 0038	ARX2_QSEL0
0xcf00 003c	ARX2_QSEL1
0xcf00 0040	ARX2_QSEL2
0xcf00 0044	ARX2_QSEL3
0xcf00 0048	ARX2_QSEL4
0xcf00 004c	ARX1_PHDR

0xcf00 0050	ARX1_CIP0
0xcf00 0054	ARX1_CIP1
0xcf00 0058	ARX2_ADO_CFG
0xcf00 005c	ARX2_ADO_MIDI

AVS Audio Transmitters Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 00c0	ATX1_CFG
0xcf00 00c4	ATX1_TSTAMP
0xcf00 00c8	ATX1_PHDR
0xcf00 00cc	ATX1_CIP0
0xcf00 00d0	ATX1_CIP1
0xcf00 00d4	ATX1_ADI_CFG
0xcf00 00d8	ATX1_ADI_MIDI
0xcf00 00dc	ATX2_CFG
0xcf00 00e0	ATX2_TSTAMP
0xcf00 00e4	ATX2_PHDR
0xcf00 00e8	ATX2_CIP0
0xcf00 00ec	ATX2_CIP1
0xcf00 00f0	ATX2_ADI_CFG
0xcf00 00f4	ATX2_ADI_MIDI

AVS ITP (Internal Time Processor) Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 01f8	ITP_CFG

AVS Audio Transmitter Format Handler Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 02c0	FMT_TXDI1_CFG0
0xcf00 02c4	FMT_TXDI1_CFG1
0xcf00 02c8	FMT_TXDI1_CFG2
0xcf00 02cc	FMT_TXDI1_CFG3
0xcf00 02d0	FMT_TXDI1_CFG4
0xcf00 02d4	FMT_TXDI1_CFG5
0xcf00 02d8	FMT_TXDI1_CFG6
0xcf00 02dc	FMT_TXDI1_CSBLOCK_BYTEn
0xcf00 02f4	FMT_TXDI1_CHANNELn_CS/LABEL
0xcf00 0340	FMT_TXDI2_CFG0
0xcf00 0344	FMT_TXDI2_CFG1
0xcf00 0348	FMT_TXDI2_CFG2

0xcf00 034c	FMT_TXDI2_CFG3
0xcf00 0350	FMT_TXDI2_CFG4
0xcf00 0344	FMT_TXDI2_CFG5
0xcf00 0348	FMT_TXDI2_CFG6
0xcf00 034c	FMT_TXDI2_CSBLOCK_BYTEn
0xcf00 0374	FMT_TXDI2_CHANNELn_CS/LABEL

AVS Audio Receiver Formap Hadler Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 0200	FORMAT_RXDI1_CFG
0xcf00 0204	FORMAT_RXDI1_LABELn
0xcf00 0214	FORMAT_RXDI1_CSBLOCKn
0xcf00 0230	FORMAT_RXDI2_CFG
0xcf00 0234	FORMAT_RXDI2_LABELn
0xcf00 0244	FORMAT_RXDI2_CSBLOCKn

AVS Interrupt Controller Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 013c	AVSI_INT0_STATUS
0xcf00 0140	AVSI_INT0_MASK
0xcf00 0144	AVSI_INT1_STATUS
0xcf00 0148	AVSI_INT1_MASK

AVS Media FIFO Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 0184	AVSFIFO_PART0_BASE
0xcf00 0188	AVSFIFO_PART0_LIMIT
0xcf00 018c	AVSFIFO_PART0_FLUSH
0xcf00 0190	AVSFIFO_PART1_BASE
0xcf00 0194	AVSFIFO_PART1_LIMIT
0xcf00 0198	AVSFIFO_PART1_FLUSH
0xcf00 019c	AVSFIFO_PART2_BASE
0xcf00 01a0	AVSFIFO_PART2_LIMIT
0xcf00 01a4	AVSFIFO_PART2_FLUSH
0xcf00 01a8	AVSFIFO_PART3_BASE
0xcf00 01ac	AVSFIFO_PART3_LIMIT
0xcf00 01b0	AVSFIFO_PART3_FLUSH
0xcf00 01fc	AVSFIFO_STAT

AVS MIDI Interface Memory Map	See Chapter 6 - AVS
Address	Register
0xcf00 01e4	AVSMIDI_STAT
0xcf00 01e8	AVSMIDI_CTRL
0xcf00 01ec	AVSMIDI_RX
0xcf00 01f0	AVSMIDI_TX0
0xcf00 01f4	AVSMIDI_TX1

AVS General Memory Map	See Chapter 6 - AVS
Address	Register
0xc800 0000	PDB_INT (AVC_CTRL)

Appendix 2. Revision history

Revision	Made By	Notes
0.01	M. Lave	Created based on DICE II
0.03	M. Lave	Corrected Power pin errors.
0.04	M. Lave	Final pinout
0.05	L.Sherbak	Minor fixes in pinout
0.06	L.Sherbak	Updates in AVS, ADAT, CC, Router, adding InS & Mixer
0.07	M. Lave	Fixed DICE JR/Mini TCD22xx confusion, inserted clock controller block diagram. Changed mixer description. Removed references to two PLL, router instances. Removed irrelevant soldering information.
0.08	L. Sherbak	Minor text fixes to InS text
0.09	L.Sherbak	Adding SPI spec
0.10	L.Sherbak	Fixing GPIO5 select description in GPCSR
0.11	L.Sherbak	Fixing ADAT RX register description
0.12	M.Lave	Fixing minor discrepancies between register descriptions and register graphics
0.13	L.Sherbak	Fixing discrepancies in GPIO tables
0.14	L.Sherbak	Adding Address Remap block description Adding Register Summary
0.15	L.Sherbak	Fixing bugs in ARM memory controller description & InS
0.16	M. Lave	Added spec for Industrial versions.
1.00	M. Lave	SPI figure updated, missing reference. Moved this table to Appendix.