Digital Interface Communications Engine

Users Guide

Revision 2.3

April 16, 2007

Revision History

Revision	Made By	Notes
1.3	B. Moses	First version published by Wavefront
1.4	B. Moses	Misc edits
1.5	A. Glover	• Modified AVS register description tables • Fixed various AVS mistakes • Added signal description tables to all appropriate sections • Fixed reset values for AES, DSAI, and Clock controller • Output drive/open drain issue fixed in main signal description table • Indication of input, output, bi-directional, for shared pins fixed in main signal description table
1.6	B. Moses	• Added DSAI RX and TX timing diagrams • Added JET PLL section • Added soldering profile
1.7	A. Glover	• Changed references to the D2A interface, to the I2S interface in GPCSR chapter • Description and diagram of Clock Doubler added to Clock Controller chapter • Added diagrams for functional descriptions, and register descriptions to the Watchdog chapter. The Watchdog bug is not described. • Description of the QSEL registers added to the AVS chapter
1.8	A. Glover	• Added paragraph to the AES Tx section describing block sync configuration and application • Removed register field from the DSAI TX and DSAI RX sections that selects between 8 and 16 channel mode • Added tables to the I2S chapter to illustrate I2S channel configuration for different Receivers and Transmitters
1.9	A. Glover	• Added "Interfacing to Non-Memory Devices with Ready Pin" section to the Memory Controller (EBI) chapter • Added "DC Characteristics" and "PLL Characteristics" sections to the Electrical Characteristics chapter

1.91	A. Glover	• Corrected IC_CON register in I2C section, corrected reset values throughout I2C section • Corrected PHDR and CIP header register in AVS section • Added ball composition to soldering section • Modified I2S section text for further clarity. Added a new diagram to illustrate clock and data flow. Changed register descriptions and naming conventions • Added to memory map description • Added descriptions to the I2C Chapter • Changed TDIF register names • Changed various references to system 49.152MHz clock to refer to "ARM system clock (typically 49.152MHz)" and to "pclk" where necessary.
2.0	J. Meyer	• New document format
2.1	B. Moses	• Many corrections
2.2	B. Karr	• Changes for TCAT publishing
2.3	B. Karr	• Revised timing diagrams for DSAI

Table of Contents

Figures

Tables

Chapter 1 About DICE II

DICEII comes in two versions:

- **DICEII-STD** Limited version not including DTCP compliant encryption/decryption functionality (5C).

- **DICEII-CP** Full version including DTCP compliant encryption/decryption functionality (5C). This version will only be available to customers having DTLA Adopter status.

Each device is available in a **LEAD FREE** 272 PBGA package.

1.1 Introduction

The DICE II chip 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 64 input channels and 32 output channels distributed on several isochronous channels.

1.2 Block Diagram

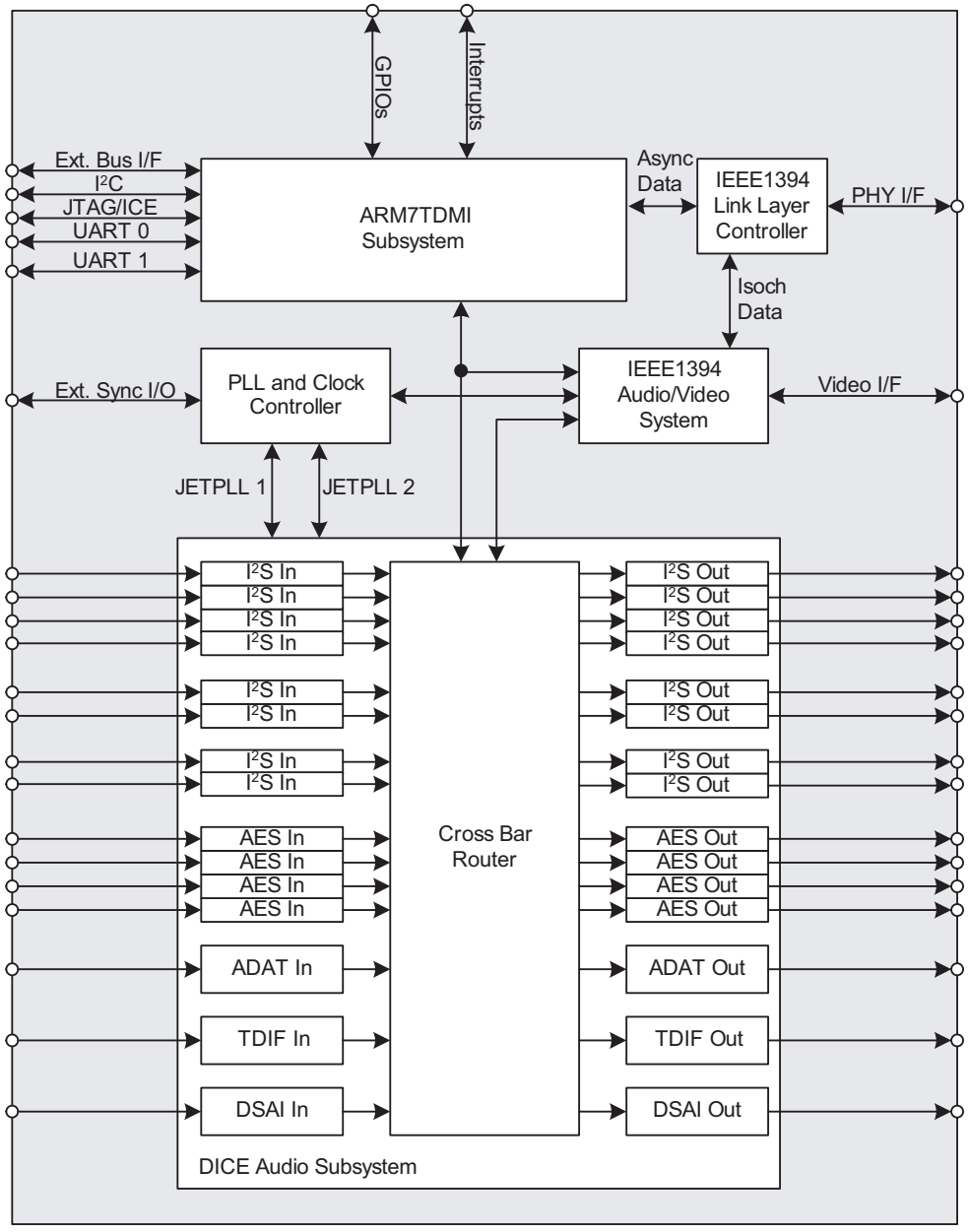


Figure 1.1: DICE II Block Diagram

1.3 Chip Features

CPU core
- Full 32-bit ARM7TDMI RISC processor
- 32-bit internal bus
- 16-bit Thumb mode
- 32 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

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 FIFO's
- Supports all standard RS232 Rates
- Supports MIDI rate

General Purpose Input Output (GPIO)
- 16 individual ports
- Each port configurable as input or output
- Each port configurable for level or edge sensitive interrupts
- Configurable deglitching logic for each port

Quad 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)
- Two individual Sample rate domains
- One JETTM PLL per domain
- One Cross-bar router per domain
- AES Receiver, 8 channels (4 ch @ 192KHz)
- AES Transmitter, 8 channels (4 ch @ 192KHz)
- I2S Receiver, 16 channels (12 ch @ 192KHz)
- I2S Transmitter, 16 channels (12 ch @ 192KHz)
- ADAT Receiver, 8 channels (4 ch @ 96KHz)
- ADAT Transmitter, 8 channels (4 ch @ 96KHz)
- Digital Serial Audio Interface Receiver, 32 channels (16 ch @192KHz)
- Digital Serial Audio Interface Transmitter, 32 channels (16 ch @192KHz)
- ARM Audio Receiver/Transmitter, 8 channels (4 ch @ 192KHz)
- IEC 61883-6 Isoc. Receiver, 64 channels (32 ch @ 192KHz)
- IEC 61883-6 Isoc. Transmitter, 32 channels (16 ch @ 192KHz)

IEEE 1394 Audio Video System (AVS)
- Configured as either input or output
- 8 bit interface
- Supports gated clock
- Supports DV, MPEG2 and raw Isoc.
- Configurable smoothing engine.

Power and operating voltage
- 1800 mW maximum, 900 mW typical (TBD)
- 3.3 volts - I/O
- 1.8 volts – core

1.4 Package

Both types of DICE II (STD and CP) are available in a LEAD FREE 272 Plastic Ball Grid Array (PBGA) with the following dimensions:

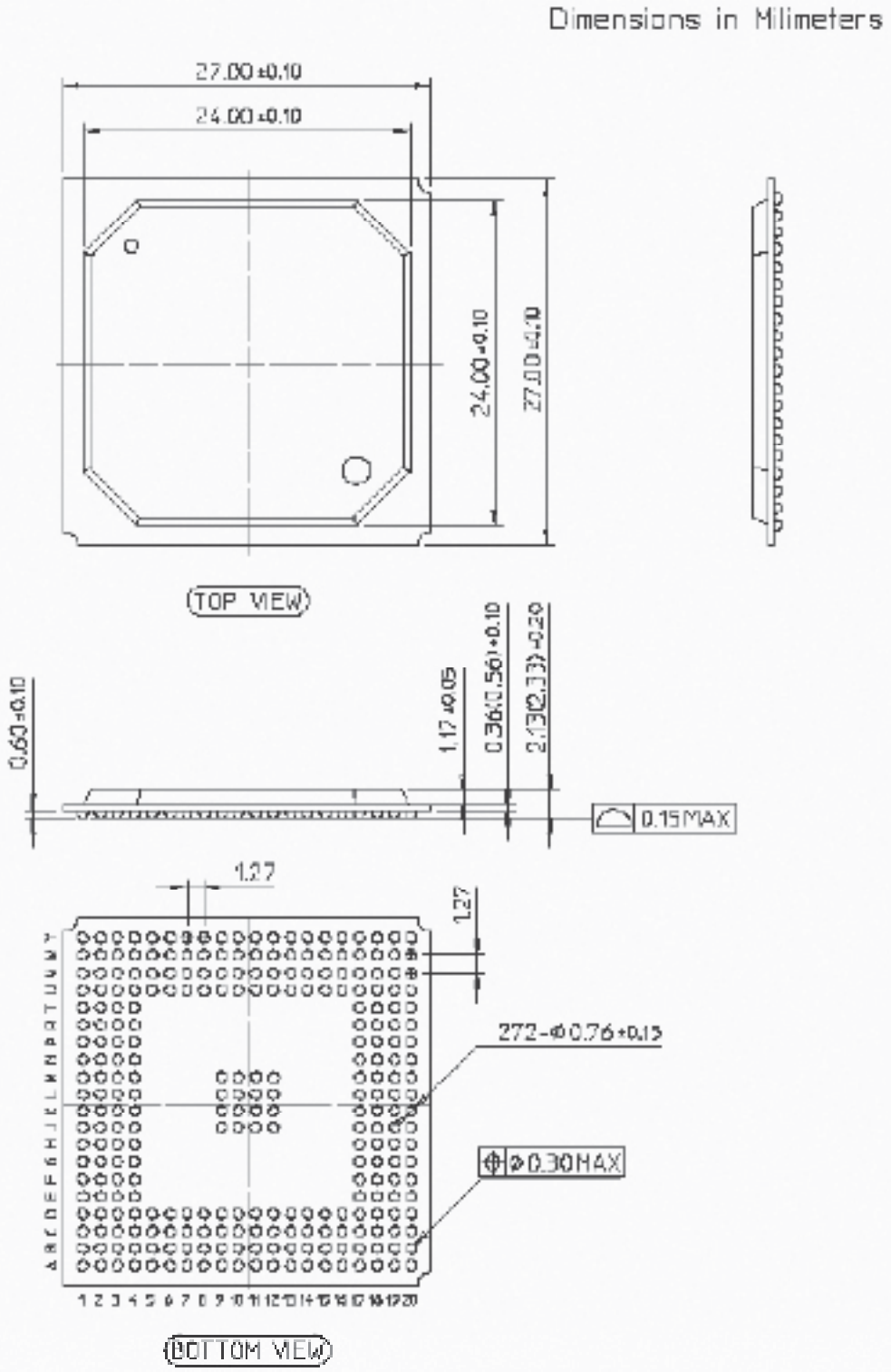


Figure 1.2: BGA272 dimensions

1.5 Pinout

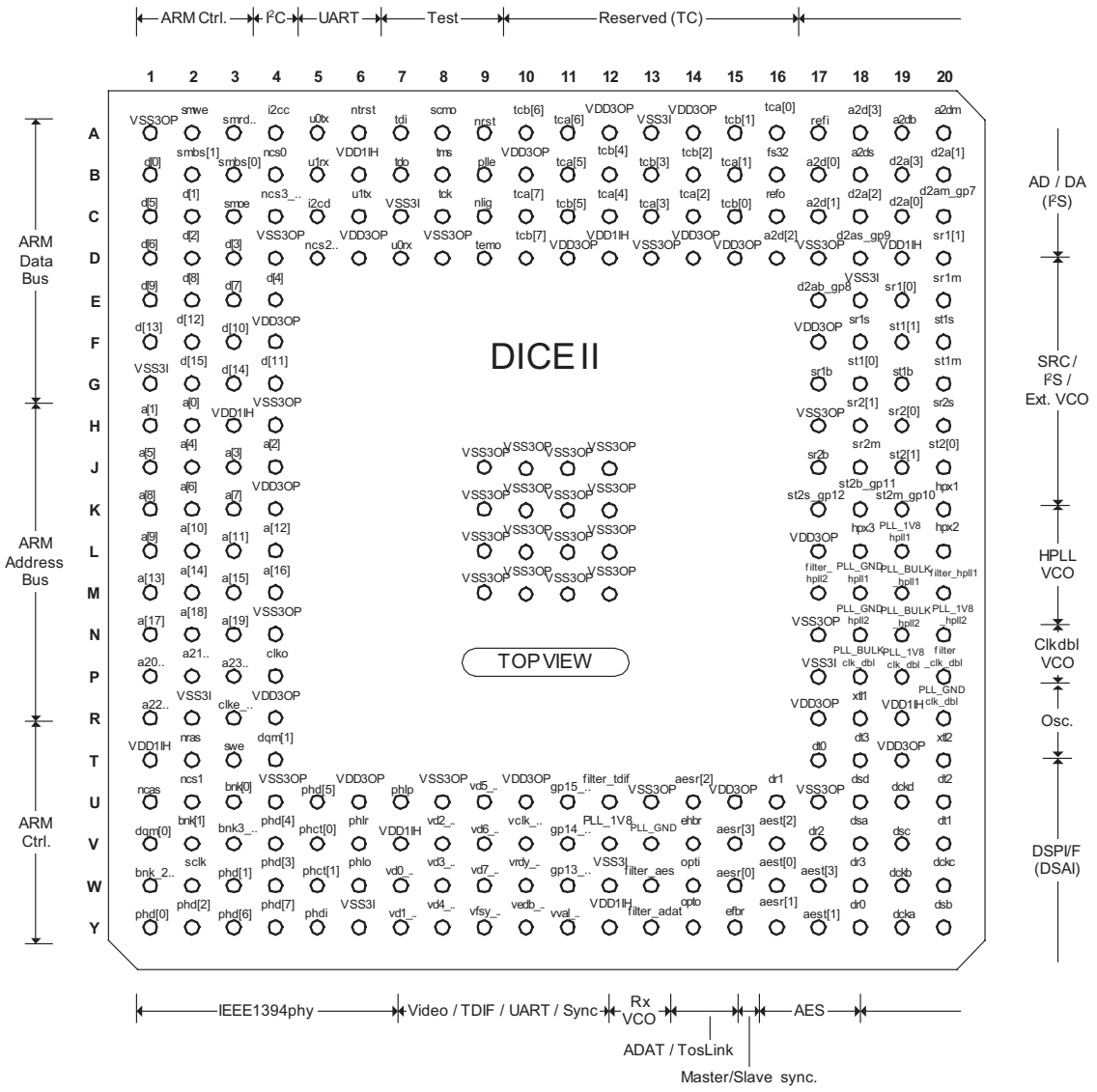


Figure 1.3: PBGA Pinout

1.6 Signal Description

The following table lists each I/O signal for the DICEII. Note that the DICEII uses 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.

1.6.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
DATA BUS				
D0	B1	I/O (S_1)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU_2,$5V_3$)
D1	C2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D2	D2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D3	D3	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D4	E4	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D5	C1	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D6	D1	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D7	E3	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D8	E2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D9	E1	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D10	F3	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	PBGA Pin	I/O	Drive (mA)	Description
D11	G4	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D12	F2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D13	F1	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D14	G3	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D15	G2	I/O (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
ADDRESS BUS				
A0	H2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A1	H1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A2	J4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A3	J3 (shared[4])	O4	8	Address Bus. Shared address pins for SDRAM and Static memory.
A4	J2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A5	J1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A6	K2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A7	K3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A8	K1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A9	L1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A10	L2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A11	L3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A12	L4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.

4 All shared pins are bi-directional even though the particular function described might not be bi-directional.

Signal	PBGA Pin	I/O	Drive (mA)	Description
A13	M1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A14	M2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A15	M3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A16	M4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A17	N1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A18	N2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A19	N3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A20	P1 (shared)	O4	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
A21	P2 (shared)	O4	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
A22	R1 (shared)	O4	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
A23	P3 (shared)	O4	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
CHIP SELECTS				
CS0*	B4	O	4	Shared SDRAM and Static Memory Chip Selects
CS1*	U2	O	4	Shared SDRAM and Static Memory Chip Selects
CS2*	D5 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects
CS3*	C4 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects
CS4*	P3 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS5*	R1 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS6*	P2 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS7*	P1 (shared)	O4	6	Shared SDRAM and Static Memory Chip Selects (5V)

Signal	PBGA Pin	I/O	Drive (mA)	Description
GREY CODE ROTARY ENCODER				
EN1_A	P1 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN1_B	P2 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_A	R1 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_B	P3 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN3_A	W1(shared)	I (S)	6	Rotary Encoder Input (5V)
EN3_B	V3 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN4_A	D5 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN4_B	C4 (shared)	I (S)	6	Rotary Encoder Input (5V)
GENERAL PURPOSE I/O				
GPIO1	R3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO2	W1 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO3	V3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO4	A3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO5	D5 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO6	C4 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO7	C20 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO8	E17 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO9	D18 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO10	K19 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO11	K18 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO12	K17 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO13	W11 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO14	V11 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO15	U11 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO16	P3 (shared)	I/O (S)	6	General Purpose I/O (5V)
RAM CLOCK				
CLKO	P4	O	8	SDRAM Interface AHB Bus Clock (Z_5)

5 Z indicates that the output is Z-stateable

Signal	PBGA Pin	I/O	Drive (mA)	Description
SDRAM DEDICATED SIGNALS				
CLKE	R3 (shared)	O	6	SDRAM Interface Clock Enable
RAS*	T2	O	8	SDRAM Interface Row Address Strobe
CAS*	U1	O	8	SDRAM Interface Column Address Strobe
SDRAM_WE	T3	O	8	SDRAM Interface Write Enable
SDRAM_DQM0	V1	O	8	SDRAM Interface Lower byte mask
SDRAM_DQM1	T4	O	8	SDRAM Interface Upper byte mask
SDRAM_BNK0	U3	O	8	SDRAM Interface Bank Address
SDRAM_BNK1	V2	O	8	SDRAM Interface Bank Address
SDRAM_BNK2	W1 (shared)	O4 (S)	6	SDRAM Interface Bank Address (5V)
SDRAM_BNK3	V3 (shared)	O4 (S)	6	SDRAM Interface Bank Address (5V)
PHY INTERFACE				
SCLK	W2	I (S)	-	49.152MHz PHY Clock input
PHD0	Y1	I/O (S)	8	PHY tristatable data line bit 0
PHD1	W3	I/O (S)	8	PHY tristatable data line bit 1
PHD2	Y2	I/O (S)	8	PHY tristatable data line bit 2
PHD3	W4	I/O (S)	8	PHY tristatable data line bit 3
PHD4	V4	I/O (S)	8	PHY tristatable data line bit 4
PHD5	U5	I/O (S)	8	PHY tristatable data line bit 5
PHD6	Y3	I/O (S)	8	PHY tristatable data line bit 6
PHD7	Y4	I/O (S)	8	PHY tristatable data line bit 7
PHCT0	V5	I/O (S)	8	PHY tristatable control line bit 0
PHCT1	W5	I/O (S)	8	PHY tristatable control line bit 1
PHDI	Y5	I (S)	-	A high indicates isolation barrier is not present (PU, 5V)
PHLR	V6	O	8	Serial request output from S-LINK (Z)
PHLP	U7	O	4	Link power status. Pulsing if isol. barrier present
PHLO	W6	I (S)	-	Link on indication from PHY. Pulsing when asserted (PU, 5V)

Signal	PBGA Pin	I/O	Drive (mA)	Description
VIDEO INTERFACE				
VD0	W7 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 0 (5V)
VD1	Y7 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 1 (5V)
VD2	V8 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 2 (5V)
VD3	W8 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 3 (5V)
VD4	Y8 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 4 (5V)
VD5	U9 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 5 (5V)
VD6	V9 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 6 (5V)
VD7	W9 (shared)	I/O (S)	6	Video Interface – Data Byte Bit 7 (5V)
VFSYNC	Y9 (shared)	I/O (S)	6	Video Interface – Video Sync Signal (5V)
VRDY	W10 (shared)	I/O (S)	6	Video Interface – Video Ready Signal (5V)
VCLK	V10 (shared)	I/O (S)	6	Video Interface – Video Clock (5V)
VEND_DB	Y10 (shared)	I/O (S)	6	Video Interface – End of Data Block (5V)
VVALID	Y11 (shared)	I/O (S)	6	Video Interface – Video Data Valid (5V)
BLOCK SYNC				
BLKS	W11 (shared)	I/O (S)	6	Block Sync Input/Output Signal (5V)
WORD CLOCK				
WCKI	V11	I (S)	6	Word Clock In (5V)
WCKO	U11	O	6	Word Clock Out
OPTICAL INTERFACE				
OPTO	Y14	O	8	Optical Audio Out
OPTI	W14	I	-	Optical Audio In (5V)
EXTERNAL SAMPLE CLOCKS				
EXT_FBR	Y15	I/O (S)	6	External 1fs base rate clock (5V)
EXT_512BR	V14	I/O (S)	6	External 512 x base rate clock (5V)
AES INTERFACE				
AES_RX0	W15	I	-	AES3 Receiver Ch0/1 (5V)
AES_RX1	Y16	I	-	AES3 Receiver Ch2/3 (5V)
AES_RX2	U14	I	-	AES3 Receiver Ch4/5 (5V)
AES_RX3	V15	I	-	AES3 Receiver Ch6/7 (5V)
AES_TX0	W16	O	2	AES3 Transmitter Ch0/1
AES_TX1	Y17	O	2	AES3 Transmitter Ch2/3
AES_TX2	V16	O	2	AES3 Transmitter Ch4/5
AES_TX3	W17	O	2	AES3 Transmitter Ch6/7

Signal	PBGA Pin	I/O	Drive (mA)	Description
DSAI INTERFACE				
DSAI_RX0	Y18	I	-	DSAI Receiver 0 data line (5V)
DSAI_RX1	U16	I	-	DSAI Receiver 1 data line (5V)
DSAI_RX2	V17	I	-	DSAI Receiver 2 data line (5V)
DSAI_RX3	W18	I	-	DSAI Receiver 3 data line (5V)
DSAI_CKA	Y19	I/O (S)	8	DSAI Clock A
DSAI_SYNCA	V18	I/O (S)	8	DSAI Sync A
DSAI_CKB	W19	I/O (S)	8	DSAI clock B
DSAI_SYNCB	Y20	I/O (S)	8	DSAI Sync B
DSAI_CKC	W20	I/O (S)	8	DSAI Clock C
DSAI_SYNCC	V19	I/O (S)	8	DSAI Sync C
DSAI_CKD	U19	I/O (S)	8	DSAI Clock D
DSAI_SYNCD	U18	I/O (S)	8	DSAI Sync D
DSAI_TX0	T17	O	4	DSAI Transmitter 0 data line
DSAI_TX1	V20	O	4	DSAI Transmitter 1 data line
DSAI_TX2	U20	O	4	DSAI Transmitter 2 data line
DSAI_TX3	T18	O	4	DSAI Transmitter 3 data line
XTAL				
XTAL2	T20	O	-	XTAL for clock doubler/power manager/ LLC
XTAL1	R18	I	-	XTAL for clock doubler/power manager/ LLC
EXTERNAL VCO CONNECTIONS				
HPX3	L18	I/O (S)	8	GPIO
HPX2	L20	O	8	GPO (Z)
HPX1	K20	O	8	GPO (Z)

Signal	PBGA Pin	I/O	Drive (mA)	Description
I2S INTERFACE				
I2S_TX2_MCK	K19 (shared)	O4	8	I2S Transmitter 2 Master Clock
I2S_TX2_BICK	K18 (shared)	O4	8	I2S Transmitter 2 Bit Clock
I2S_TX2_LRCLK	K17 (shared)	O4	8	I2S Transmitter 2 Left/Right Clock
I2S_TX2_D0	J20	O	2	I2S Transmitter 2 Data Ch.0/1
I2S_TX2_D1	J19	O	2	I2S Transmitter 2 Data Ch.2/3
I2S_RX2_MCK	J18	O	8	I2S Receiver 2 Master Clock
I2S_RX2_BICK	J17	O	8	I2S Receiver 2 Bit Clock
I2S_RX2_LRCK	H20	O	8	I2S Receiver 2 Left/Right Clock
I2S_RX2_D0	H19	I	-	I2S Receiver 2 Data (Ch. 0/1) (5V)
I2S_RX2_D1	H18	I	-	I2S Receiver 2 Data (Ch. 2/3) (5V)
I2S_TX1_MCK	G20	O	8	I2S Transmitter 1 Master Clock
I2S_TX1_BICK	G19	O	8	I2S Transmitter 1 Bit Clock
I2S_TX1_LRCK	F20	O	8	I2S Transmitter 1 Left/Right Clock
I2S_TX1_D0	G18	O	2	I2S Transmitter 1 Data Ch.0/1
I2S_TX1_D1	F19	O	2	I2S Transmitter 1 Data Ch.2/3
I2S_RX1_MCK	E20	O	8	I2S Receiver 1 Master Clock
I2S_RX1_BICK	G17	O	8	I2S Receiver 1 Bit Clock
I2S_RX1_LRCK	F18	O	8	I2S Receiver 1 Left/Right Clock
I2S_RX1_D0	E19	I	-	I2S Receiver 1 Data (Ch. 0/1) (5V)
I2S_RX1_D1	D20	I	-	I2S Receiver 2 Data (Ch. 2/3) (5V)
I2S_TX0_MCK	C20 (shared)	O4	8	I2S Transmitter 0 Master Clock
I2S_TX0_BICK	E17 (shared)	O4	8	I2S Transmitter 0 Bit Clock
I2S_TX0_LRCK	D18 (shared)	O4	8	I2S Transmitter 0 Left/Right Clock
I2S_TX0_D0	C19	O	2	I2S Transmitter 0 Data Ch.0/1
I2S_TX0_D1	B20	O	2	I2S Transmitter 0 Data Ch.2/3
I2S_TX0_D2	C18	O	2	I2S Transmitter 0 Data Ch.4/5
I2S_TX0_D3	B19	O	2	I2S Transmitter 0 Data Ch.6/7
I2S_RX0_MCK	A20	O	8	I2S Receiver 0 Master Clock
I2S_RX0_BICK	A19	O	8	I2S Receiver 0 Bit Clock
I2S_RX0_LRCK	B18	O	8	I2S Receiver 0 Left/Right Clock
I2S_RX0_D0	B17	I	-	I2S Receiver 0 Data (Ch. 0/1) (5V)
I2S_RX0_D1	C17	I	-	I2S Receiver 0 Data (Ch. 2/3) (5V)
I2S_RX0_D2	D16	I	-	I2S Receiver 0 Data (Ch. 4/5) (5V)
I2S_RX0_D3	A18	I	-	I2S Receiver 0 Data (Ch. 6/7) (5V)

Signal	PBGA Pin	I/O	Drive (mA)	Description
RESERVED				
REFI	A17	I	-	Test pin. Connect to 10Kohm pullup resistor.
REFO	C16	O	-	Test pin
FS32	B16	O	-	Test pin
TCA[0]	A16	O	-	Test pin
TCB[0]	C15	O	-	Test pin
TCA[1]	B15	O	-	Test pin
TCB[1]	A15	O	-	Test pin
TCA[2]	C14	O	-	Test pin
TCB[2]	B14	O	-	Test pin
TCA[3]	C13	O	-	Test pin
TCB[3]	B13	O	-	Test pin
TCA[4]	C12	O	-	Test pin
TCB[4]	B12	O	-	Test pin
TCA[5]	B11	O	-	Test pin
TCB[5]	C11	O	-	Test pin
TCA[6]	A11	O	-	Test pin
TCB[6]	A10	O	-	Test pin
TCA[7]	C10	O	-	Test pin
TCB[7]	D10	O	-	Test pin
RESET				
RESET*	A9	I (S)	-	Reset – active low (PU, 5V)
PLL				
PLLE	B9	I	-	PLL Enable (5V)
NLIG	C9	I	-	Ignore PLL no-lock before releasing reset, active high (5V)
TEST				
TEMO	D9	I	-	Test mode pin (PD_6, 5V), internal pulldown
SCMO	A8	I	-	Scan mode select: HI – boundary scan, LO - debug (PD, 5V)

6 PD indicates that internal Pull-Down resistor is present on pad

Signal	PBGA Pin	I/O	Drive (mA)	Description
JTAG INTERFACE				
TMS	B8	I	-	JTAG -Test mode select (PU, 5V)
TCK	C8	I	-	JTAG -Test clock (5V)
TDI	A7	I	-	JTAG -Test Data In (PU, 5V)
TDO	B7	O	4	JTAG -Test Data Out (Z, 5V)
TRST*	A6	I	-	JTAG –Test Reset (active low) (PD, 5V)
I2C INTERFACE				
I2C_CLK	A4	I/O (S)	6	I2C Clock (OD[7], 5V)
I2C_DATA	C5	I/O (S)	6	I2C Data (OD, 5V)
SRAM INTERFACE				
SRAM_READY	A3 (Shared)	I4 (S)	6	SRAM ready (read enable)
SRAM_BS[0]	B3	O	4	SRAM lower byte select
SRAM_BS[1]	B2	O	4	SRAM upper byte select
SRAM_WE*	A2	O	8	SRAM write enable
SRAM_OE*	C3	O	8	SRAM output enable
TDIF SIGNALS				
TDF_I0	W7 (shared)	I4 (S)	6	TDIF audio data input 1 (5V)
TDF_I1	Y7 (shared)	I4 (S)	6	TDIF audio data input 2 (5V)
TDF_I2	V8 (shared)	I4 (S)	6	TDIF audio data input 3 (5V)
TDF_I3	W8 (shared)	I4 (S)	6	TDIF audio data input 4 (5V)
TDF_ILR	Y8 (shared)	I4 (S)	6	TDIF left right clock input (5V)
TDF_IFS0	U9 (shared)	I4 (S)	6	TDIF sample rate 0 input (5V)
TDF_IFS1	V9 (shared)	I4 (S)	6	TDIF sample rate 1 input (5V)
TDF_IEM	W9 (shared)	I4 (S)	6	TDIF emphasis input (5V)
TDF_O0	Y9 (shared)	O4	6	TDIF audio data output 1
TDF_O1	W10 (shared)	O4	6	TDIF audio data output 2
TDF_O2	V10 (shared)	O4	6	TDIF audio data output 3
TDF_O3	Y10 (shared)	O4	6	TDIF audio data output 4
TDF_OLR	Y11 (shared)	O4	6	TDIF left right clock output
TDF_OFS0	W11 (shared)	O4	6	TDIF sample rate 0 output
TDF_OFS1	V11 (shared)	O4	6	TDIF sample rate 1 output
TDF_OEM	U11 (shared)	O4	6	TDIF emphasis output

7 OD indicates Open Drain pad type. External Pull-Up resistor required

Signal	PBGA Pin	I/O	Drive (mA)	Description
UART SIGNALS				
U0_CTS	W7 (shared)	I4 (S)	6	Clear To Send UART Status input; active low (5V)
U0_DSR	Y7 (shared)	I4 (S)	6	Data Set Ready UART Status input; active-low (5V)
U0_DCD	V8 (shared)	I4 (S)	6	Data Carrier Detect UART Status input; active-low (5V)
U0_RI	W8 (shared)	I4 (S)	6	Ring Indicator UART Status input; active low (5V)
U1_CTS	Y8 (shared)	I4 (S)	6	Clear To Send UART Status input; active low (5V)
U1_DSR	U9 (shared)	I4 (S)	6	Data Set Ready UART Status input; active-low (5V)
U1_DCD	V9 (shared)	I4 (S)	6	Data Carrier Detect UART Status input; active-low (5V)
U1_RI	W9 (shared)	I4 (S)	6	Ring Indicator UART Status input; activelow (5V)
U0_DTS	Y11 (shared)	O4	6	UART Control Data Terminal Ready output; active-low
U0_RTS	W11 (shared)	O4	6	UART Control Request To Send output; active-low
U0_OUT1	V11 (shared)	O4	6	UART Control Programmable output 1 output; active-low
U0_OUT2	U11 (shared)	O4	6	UART Control Programmable output 2 output; active-low
U1_DTS	Y9 (shared)	O4	6	UART Control Data Terminal Ready output; active-low
U1_RTS	W10 (shared)	O4	6	UART Control Request To Send output; active-low
U1_OUT1	V10 (shared)	O4	6	UART Control Programmable output 1 output; active-low
U1_OUT2	Y10 (shared)	O4	6	UART Control Programmable output 2 output; active-low
UART0_TX	A5	O	4	Serial output; active-high
UART0_RX	D7	I	-	Serial input; active-high (5V)
UART1_TX	C6	O	4	Serial output; active-high
UART1_RX	B5	I	-	Serial input; active-high (5V)
512FS BASE RATE CLOCKS				
HFS1	U9	I (S)	6	512 x base rate 96kHz, XTAL 24.576MHz
HFS2	V9	I (S)	6	512 x base rate 88kHz, XTAL 22.5792MHz

Signal	PBGA Pin	I/O	Drive (mA)	Description
FILTERS				
FILTER_TDIF	U12	A	-	TDIF Receiver VCO filter component connection
FILTER_ADAT	Y13	A	-	ADAT Receiver filter component connection
FILTER_AES	W13	A	-	AES Receiver filter component connection
FILTER_CLK_DBL	P20	A	-	Clock Doubler VCO filter component connection
FILTER_HPLL2	M17	A	-	JETPLL filter component connection
FILTER_HPLL1	M20	A	-	JETPLL filter component connection
PLL 1.8V				
PLL_1V8 (AES, ADAT, TDIF)	V12	P	-	PLL 1.8 V
PLL_1V8 (CLK_ DBL)	P19	P	-	PLL 1.8 V
PLL_1V8 (HPLL2)	N20	P	-	PLL 1.8 V
PLL_1V8 (HPLL1)	L19	P	-	PLL 1.8 V
PLL BULK BIAS				
PLL_BULK (CLK_ DBL)	P18	P	-	PLL Bulk Bias
PLL_BULK (HPLL2)	N19	P	-	PLL Bulk Bias
PLL_BULK (HPLL1)	M19	P	-	PLL Bulk Bias
PLL GROUND				
PLL_GND (AES, ADAT, TDIF)	V13	P	-	PLL Ground
PLL_GND (CLK_ DBL)	R20	P	-	PLL Ground
PLL_GND (HPLL2)	N18	P	-	PLL Ground
PLL_GND (HPLL1)	M18	P	-	PLL Ground

Signal	PBGA Pin	I/O	Drive (mA)	Description
CORE 1.8V				
VDD1IH	H3	P	-	Core 1.8 V
VDD1IH	T1	P	-	Core 1.8 V
VDD1IH	V7	P	-	Core 1.8 V
VDD1IH	Y12	P	-	Core 1.8 V
VDD1IH	R19	P	-	Core 1.8 V
VDD1IH	D19	P	-	Core 1.8 V
VDD1IH	D12	P	-	Core 1.8 V
VDD1IH	B6	P	-	Core 1.8 V
I/O 3.3V				
VDD3OP (VDD power ring)	F4	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	K4	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	R4	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	U6	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	U10	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	U15	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	R17	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	L17	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	F17	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	D15	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	D11	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	A12	P		I/O 3.3 V
VDD3OP (VDD power ring)	A14	P		I/O 3.3 V
VDD3OP (VDD power ring)	B10	P		I/O 3.3 V
VDD3OP (VDD power ring)	D14	P		I/O 3.3 V

Signal	PBGA Pin	I/O	Drive (mA)	Description
VDD3OP (VDD power ring)	D6	P	-	I/O 3.3 V
VDD3OP (VDD power ring)	T19	P	-	I/O 3.3 V
CORE GROUND				
VSS3I	G1	P	-	Core Gound
VSS3I	R2	P	-	Core Gound
VSS3I	Y6	P	-	Core Gound
VSS3I	W12	P	-	Core Gound
VSS3I	P17	P	-	Core Gound
VSS3I	E18	P	-	Core Gound
VSS3I	A13	P	-	Core Gound
VSS3I	C7	P	-	Core Gound
I/O GROUND				
VSS3OP (VSS power ring)	A1	P	-	I/O Ground
VSS3OP (VSS power ring)	D4	P	-	I/O Ground
VSS3OP (VSS power ring)	H4	P	-	I/O Ground
VSS3OP (VSS power ring)	N4	P	-	I/O Ground
VSS3OP (VSS power ring)	U4	P	-	I/O Ground
VSS3OP (VSS power ring)	U8	P	-	I/O Ground
VSS3OP (VSS power ring)	U13	P	-	I/O Ground
VSS3OP (VSS power ring)	U17	P	-	I/O Ground
VSS3OP (VSS power ring)	N17	P	-	I/O Ground
VSS3OP (VSS power ring)	H17	P	-	I/O Ground
VSS3OP (VSS power ring)	D17	P	-	I/O Ground
VSS3OP (VSS power ring)	D13	P	-	I/O Ground
VSS3OP (VSS power ring)	D8	P	-	I/O Ground

Signal	PBGA Pin	I/O	Drive (mA)	Description
VSS3OP (VSS power ring)	J9	P	-	I/O Ground
VSS3OP (VSS power ring)	J10	P	-	I/O Ground
VSS3OP (VSS power ring)	J11	P	-	I/O Ground
VSS3OP (VSS power ring)	J12	P	-	I/O Ground
VSS3OP (VSS power ring)	K9	P	-	I/O Ground
VSS3OP (VSS power ring)	K10	P	-	I/O Ground
VSS3OP (VSS power ring)	K11	P	-	I/O Ground
VSS3OP (VSS power ring)	K12	P	-	I/O Ground
VSS3OP (VSS power ring)	L9	P	-	I/O Ground
VSS3OP (VSS power ring)	L10	P	-	I/O Ground
VSS3OP (VSS power ring)	L11	P	-	I/O Ground
VSS3OP (VSS power ring)	L12	P	-	I/O Ground
VSS3OP (VSS power ring)	M9	P	-	I/O Ground
VSS3OP (VSS power ring)	M10	P	-	I/O Ground
VSS3OP (VSS power ring)	M11	P	-	I/O Ground
VSS3OP (VSS power ring)	M12	P	-	I/O Ground

Table 1.1: Signal Descriptions

1.6.2 MULTI-FUNCTION PINS

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

PBGA	Function 1	Function 2	Function 3	Function 4
P1	A20 (O)	CS7 (O)	EN1_A (I)	
P2	A21 (O)	CS6* (O)	EN1_B (I)	
R1	A22 (O)	CS5* (O)	EN2_A (I)	
P3	A23 (O)	CS4* (O)	EN2_B (I)	GPIO16 (I/O)
A3	A3 (O)	SRAM_READY (I)		
R3	CLKE (O)	GPIO1 (I/O)		
W1	SDRAM_BNK2 (O)	GPIO2 (I/O)	EN3_A (I)	
V3	SDRAM_BNK3 (O)	GPIO3 (I/O)	EN3_B (I)	
W7	VD0 (I/O)	TDF_I0 (I)	U0_CTS (I)	
Y7	VD1 (I/O)	TDF_I1 (I)	U0_DSR (I)	
V8	VD2 (I/O)	TDF_I2 (I)	U0_DCD (I)	
W8	VD3 (I/O)	TDF_I3 (I)	U0_RI (I)	
Y8	VD4 (I/O)	TDF_ILR (I)	U1_CTS (I)	
U9	VD5 (I/O)	TDF_IFS (I)	U1_DSR (I)	HFS1 (I)
V9	VD6 (I/O)	TDF_IFS1 (I)	U1_DCD (I)	HFS2 (I)
W9	VD7 (I/O)	TDF_IEM (I)	U1_RI (I)	
Y9	VFSYNC (I/O)	TDF_O0 (O)	U1_DTS (O)	
W10	VRDY (I/O)	TDF_O1 (O)	U1_RTS (O)	
V10	VCLK (I/O)	TDF_O2 (O)	U1_OUT1 (O)	
Y10	VEND_DB (I/O)	TDF_O3 (O)	U1_OUT2 (O)	
Y11	VVALID (I/O)	TDF_OLR (O)	U0_DTS (O)	
W11	GPIO13 (I/O)	TDF_OFS0 (O)	U0_RTS (O)	BLKS
V11	GPIO14 (I/O)	TDF_OFS1 (O)	U0_OUT1 (O)	WCKI (I)
U11	GPIO15 (I/O)	TDF_OEM (O)	U0_OUT2 (O)	WCKO (O)
K19	I2S_TX2_MCK (O)	GPIO10 (I/O)		
K18	I2S_TX2_BICK (O)	GPIO11 (I/O)		
K17	I2S_TX2_LRCLK (O)	GPIO12 (I/O)		
C20	I2S_TX0_MCK (O)	GPIO7 (I/O)		
E17	I2S_TX0_BICK (O)	GPIO8 (I/O)		
D18	I2S_TX0_LRCK (O)	GPIO9 (I/O)		
A3	SRAM_READY (I)	GPIO4 (I/O)		
D5	CS2* (O)	GPIO5 (I/O)	EN4_A (I)	
C4	CS3* (O)	GPIO6 (I/O)	EN4_B (I)	

Table 1.2: Shared Pins

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 realised 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 and 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 optimised for speed or for code density by switching between THUMB and ARM execution as appropriate.

Chapter 3 Memory Map

3.1 Memory Map

This section shows 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 32KB; however, 16MB of address space is allocated to it. The address bits 15-24 are ignored when internal SRAM is accessed.

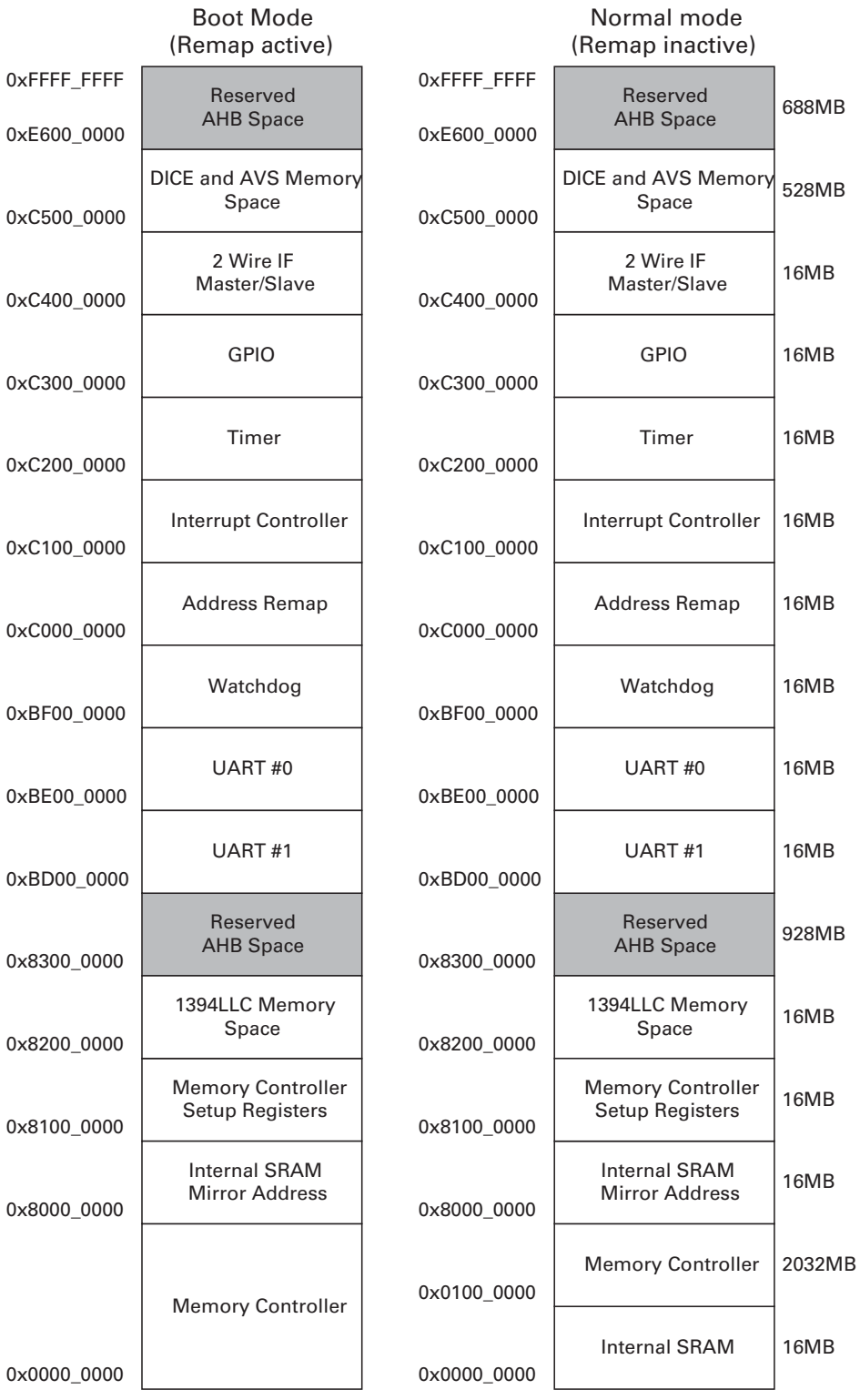


Figure 3.1: Global Memory Map (allocated address space)

Chapter 4 ARM Peripherals

4.1 General Purpose Control and Status Registers

The GPCSR controls various modes and pin mappings in the DICE II. Due to the high integration some pins share several functions. The GPCSR selects the mapping of those pins.

4.1.1 MODULE CONFIGURATION

Address	Register
0xc700 0000	GPCSR_SYSTEM
0xC700 0004	GPCSR_IO_SELECT0
0xC700 0008	GPCSR_DSAI_SELECT
0xC700 000c	GPCSR_DSAI_CLOCK_INV
0xC700 0010	GPCSR_VIDEO_SELECT
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 4.1: GPCSR Memory Map

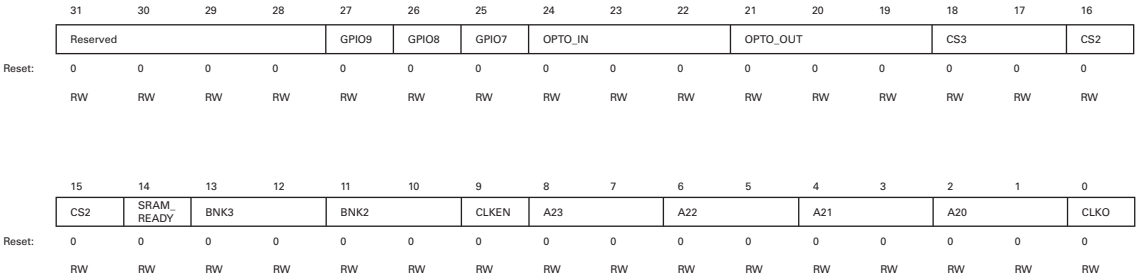
4.1.2 GRCSR_SYSTEM

Address – 0xc700 0000

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

4.1.3 GPCSR_IO_SELECT0

Address – 0xc700 0004

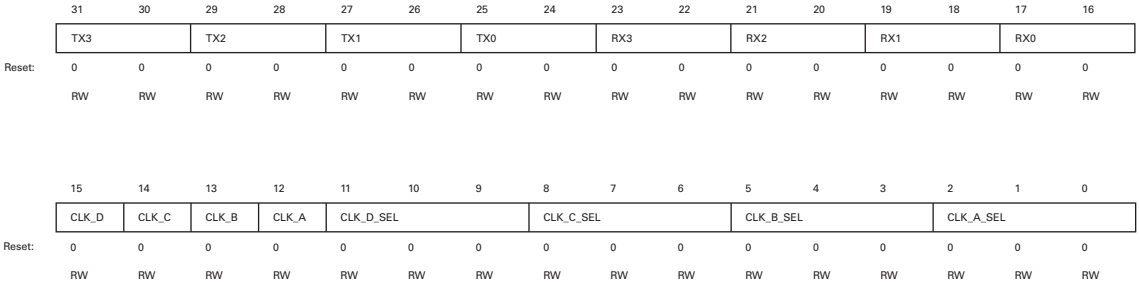


Name	Bit	Reset	Dir	Description
GPIO9	27	0	RW	Selects between the GPIO9 and the I2S_LR functionality. 0: GPIO9 1: I2S_LR Note: When set for GPIO the D to A interface can use the A to D clocks as long as they are set-up to be synchronous.
GPIO8	26	0	RW	Selects between the GPIO8 and the I2S_BICK functionality. 0: GPIO8 1: I2S_BICK Note: When set for GPIO the D to A interface can use the A to D clocks as long as they are set-up to be synchronous.
GPIO7	25	0	RW	Selects between the GPIO7 and the I2S_MCK functionality. 0: GPIO7 1: I2S_MCK Note: When set for GPIO the D to A interface can use the A to D clocks as long as they are set-up to be synchronous.
OPTO_IN	24:22	000	RW	Selects the destination for the optical input. 000: To ADAT Rx 001: To AES0 Rx 010: To AES1 Rx 011: To AES2 Rx 100: To AES3 Rx
OPTO_OUT	21:19	000	RW	Selects the source for the optical output. 000: To ADAT Tx 001: To AES0 Tx 010: To AES1 Tx 011: To AES2 Tx 1xx: To AES3 Tx
CS3	18:17	00	RW	Selects the function for the CS3 Pin. 00: ENC_4B Input (rotary encoder) 01: GPIO6 10: CS3 11: Reserved
CS2	16:15	00	RW	Selects the function for the CS2 Pin. 00: ENC_4A Input (rotary encoder) 01: GPIO5 10: CS2 11: Reserved

Name	Bit	Reset	Dir	Description
SRAM_READY	14	0	RW	Selects the function for the SRAM_READY Pin. 0: GPIO4 1: SRAM_READY
BNK3	13:12	00	RW	Selects the function of the BNK3 pin. 00: ENC_3B Input (rotary encoder) 01: GPIO3 10: BNK3 11: Reserved
BNK2	11:10	00	RW	Selects the function of the BNK2 pin. 00: ENC_3A Input (rotary encoder) 01: GPIO2 10: BNK2 11: Reserved
CLKEN	9	0	RW	Selects the function of the CLK_EN pin. 0: GPIO1 1: CLK_EN
A23	8:7	00	RW	Selects the function of the A23 pin. 00: ENC_2B Input (rotary encoder) 01: CS4 10: A23 11: GPIO16
A22	6:5	00	RW	Selects the function of the A23 pin. 00: ENC_2A Input (rotary encoder) 01: CS5 10: A22 11: Reserved
A21	4:3	00	RW	Selects the function of the A21 pin. 00: ENC_1B Input (rotary encoder) 01: CS6 10: A21 11: Reserved
A20	2:1	00	RW	Selects the function of the A20 pin. 00: ENC_1A Input (rotary encoder) 01: CS7 10: A20 11: Reserved
CLKO	0	0	RW	Disables the CLK_OUT pin. This clock is used by SDRAM, but can be disabled to lower EMI and power consumption. 0: CLK_OUT enabled 1: CLK_OUT disabled

4.1.4 GPCSR_DSAI_SELECT

Address – 0xc700 0008



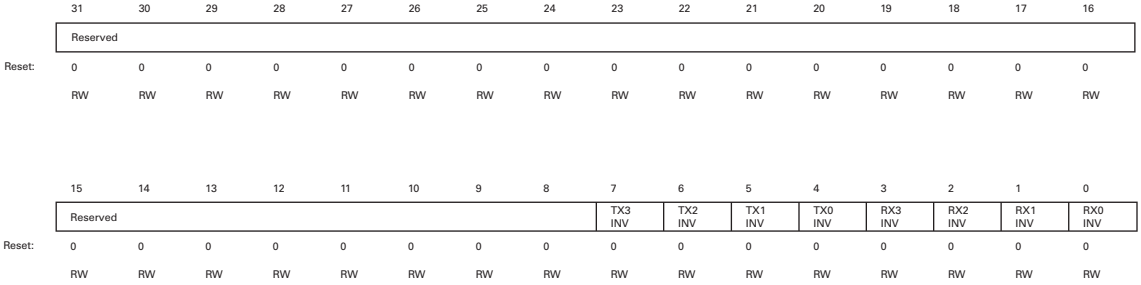
Name	Bit	Reset	Dir	Description
TX3	31:30	00	RW	Selects the clock source for DSAITX3. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
TX2	29:28	00	RW	Selects the clock source for DSAITX2. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
TX1	27:26	00	RW	Selects the clock source for DSAITX1. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
TX0	25:24	00	RW	Selects the clock source for DSAITX0. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
RX3	23:22	00	RW	Selects the clock source for DSAI RX3. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
RX2	21:20	00	RW	Selects the clock source for DSAI RX2. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
RX1	19:18	00	RW	Selects the clock source for DSAI RX1. 00: CLK A 01: CLK B 10: CLK C 11: CLK D

Name	Bit	Reset	Dir	Description
RX0	17:16	00	RW	Selects the clock source for DSAI RX0. 00: CLK A 01: CLK B 10: CLK C 11: CLK D
CLK_D	15	0	RW	Selects whether the CLK_D (CLK and SYNC) signals are outputs or inputs. 0: Input 1: Output
CLK_C	14	0	RW	Selects whether the CLK_C (CLK and SYNC) signals are outputs or inputs. 0: Input 1: Output
CLK_B	13	0	RW	Selects whether the CLK_B (CLK and SYNC) signals are outputs or inputs. 0: Input 1: Output
CLK_A	12	0	RW	Selects whether the CLK_A (CLK and SYNC) signals are outputs or inputs. 0: Input 1: Output
CLK_D_SEL	11:9	000	RW	In case of the CLK_D being an output, this register selects the source for the clock and sync signals. 000: RX0 001: RX1 010: RX2 011: RX3 100:TX0 101:TX1 110:TX2 111:TX3
CLK_C_SEL	8:6	000	RW	In case of the CLK_C being an output, this register selects the source for the clock and sync signals. 000: RX0 001: RX1 010: RX2 011: RX3 100:TX0 101:TX1 110:TX2 111:TX3
CLK_B_SEL	5:3	000	RW	In case of the CLK_B being an output, this register selects the source for the clock and sync signals. 000: RX0 001: RX1 010: RX2 011: RX3 100:TX0 101:TX1 110:TX2 111:TX3

Name	Bit	Reset	Dir	Description
CLK_A_SEL	2:0	000	RW	In case of the CLK_A being an output, this register selects the source for the clock and sync signals. 000: RX0 001: RX1 010: RX2 011: RX3 100: TX0 101: TX1 110: TX2 111: TX3

4.1.5 GPCSR_DSAI_CLOCK_INV

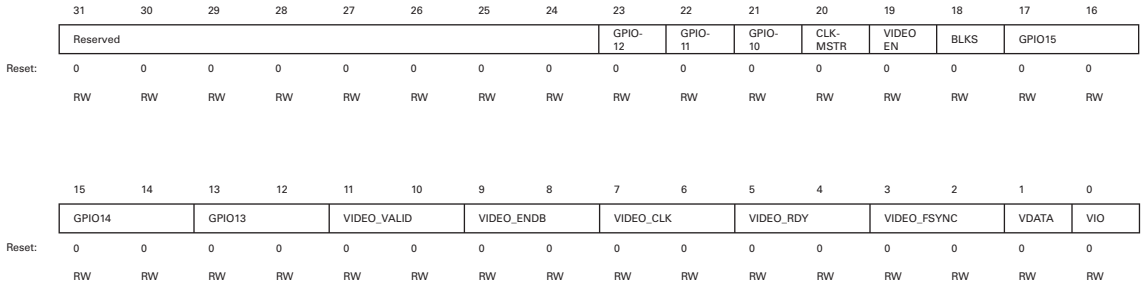
Address – 0xc700 000C



Name	Bit	Reset	Dir	Description
TX3 INV	7	0	RW	Inverts the clock input to DSAITX3. 0: Non inverted 1: Inverted
TX2 INV	6	0	RW	Inverts the clock input to DSAITX2. 0: Non inverted 1: Inverted
TX1 INV	5	0	RW	Inverts the clock input to DSAITX1. 0: Non inverted 1: Inverted
TX0 INV	4	0	RW	Inverts the clock input to DSAITX0. 0: Non inverted 1: Inverted
RX3 INV	3	0	RW	Inverts the clock input to DSAI RX3. 0: Non inverted 1: Inverted
RX2 INV	2	0	RW	Inverts the clock input to DSAI RX2. 0: Non inverted 1: Inverted
RX1 INV	1	0	RW	Inverts the clock input to DSAI RX1. 0: Non inverted 1: Inverted
RX0 INV	0	0	RW	Inverts the clock input to DSAI RX0. 0: Non inverted 1: Inverted

4.1.6 GPCSR_VIDEO_SELECT

Address – 0xc700 0010



Name	Bit	Reset	Dir	Description
GPIO12	23	0	RW	Selects the function of the SRC2_TX_LR / GPIO12 pin. 0: GPIO12 1: SRC2_TX_LR Note: The SRC2_TX can still be used, utilizing the clocks from one of the other I2S interfaces as long as they are set-up to be synchronous.
GPIO11	22	0	RW	Selects the function of the SRC2_TX_BICK / GPIO11 pin. 0: GPIO11 1: SRC2_TX_BICK Note: The SRC2_TX can still be used, utilizing the clocks from one of the other I2S interfaces as long as they are set-up to be synchronous.
GPIO10	21	0	RW	Selects the function of the SRC2_TX_MCK / GPIO10 pin. 0: GPIO10 1: SRC2_TX_MCK Note: The SRC2_TX can still be used, utilizing the clocks from one of the other I2S interfaces as long as they are set-up to be synchronous.
CLK_MSTR	20	0	RW	Selects the direction of the Master / Slave interface on the EXT_FBR and EXT_512FB pins. 0: Slave (inputs) 1: Master (outputs)
VIDEO_EN	19	0	RW	Global output enable for the VIDEO port. Enables/disables all Video pins in the video port, even if they are configured for another function. 0: Disabled 1: Enabled
BLKS	18	0	RW	Selects the direction of the Block Sync pin. 0: Input 1: Output
GPIO15	17:16	00	RW	Selects the alternative function for the GPIO15 pin. 00: GPIO15 01: TDIF_OUT_EMPH 10: UART0_OUT2 11: WCLK_OUT

Name	Bit	Reset	Dir	Description
GPIO14	15:14	00	RW	Selects the alternative function for the GPIO14 pin. 00: GPIO14 01: TDIF_OUT_FS1 10: UART0_OUT1 11: WCLK_IN
GPIO13	13:12	00	RW	Selects the alternative function for the GPIO13 pin. 00: GPIO13 01: TDIF_OUT_FS0 10: UART0_RTS 11: BLKS
VIDEO_VALID	11:10	00	RW	Selects the alternative function for the VIDEO_VALID pin. 00: VIDEO_VALID 01: TDIF_OUT_LR 10: UART0_DTR 11: Reserved
VIDEO_ENDB	9:8	00	RW	Selects the alternative function for the VIDEO_ENDB pin. 00: VIDEO_ENDB 01: TDIF_OUT_3 10: UART1_OUT2 11: Reserved
VIDEO_CLK	7:6	00	RW	Selects the alternative function for the VIDEO_CLK pin. 00: VIDEO_CLK 01: TDIF_OUT_2 10: UART1_OUT1 11: Reserved
VIDEO_RDY	5:4	00	RW	Selects the alternative function for the VIDEO_RDY pin. 00: VIDEO_RDY 01: TDIF_OUT_1 10: UART1_RTS 11: Reserved
VIDEO_FSYNC	3:2	00	RW	Selects the alternative function for the VIDEO_FSYNC pin. 00: VIDEO_FSYNC 01: TDIF_OUT_0 10: UART1_DTR 11: Reserved
VDATA	1	0	RW	Selects between the video data port and other function 0: Other function (all inputs) 1: Video data
VIO	0	0	RW	Selects whether the video port is an input or output port. 0: Input 1: Output

4.1.7 CONSIDERATIONS FOR CHIP SELECTS

Chip select 2 to 7 (CS2-CS7) 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.

4.1.8 CONSIDERATIONS FOR A20 TO A23

Address pins A20 to A23 are by default programmed to have an alternative function as an input. If the device connected to CS0 (the boot device) is using those address lines a pull-down should be added to make sure they are interpreted as '0' 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 DICE II. 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	Reserved
00001	WatchDog
00010	1394Link_on
00011	1394Link
00100	Gray
00101	GPIO
00110	Timer
00111	UART0
01000	UART1
01001	I2C
01010	AVS_IRQ0
01011	AVS_IRQ1
01100	AVS_IRQ2
01101	ARM_AUDIO_OVERFLOW
01110	ARM_AUDIO_IRQ
01111	JETPLL0
10000	JETPLL1
10001	POWER_MGR
10010	MIDI
10011 – 11111	Reserved

Table 4.2: Physical Interrupt Sources

IRQ_SEL0_5 – 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

IRQ_SEL6_11 – 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

IRQ_SEL12_17 – 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

IRQ_SEL18 – 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

FIQ_SEL0_5 – 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

FIQ_SEL6_7 – 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
- 23 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.
- Support for 8 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.

The memory map illustrated in Table 4.5 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 8 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	PBGA Pin	I/O	Drive (mA)	Description
DATA BUS				
D0	B1	IO (S_1)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU_2,$5V_3$)
D1	C2	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D2	D2	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D3	D3	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D4	E4	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D5	C1	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D6	D1	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D7	E3	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D8	E2	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D9	E1	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D10	F3	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D11	G4	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D12	F2	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D13	F1	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D14	G3	IO (S)	6	Data Bus. Shared read/write data for external SDRAM and Static memory. (PU,5V)
D15	G2	IO (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	PBGA Pin	I/O	Drive (mA)	Description
ADDRESS BUS				
A0	H2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A1	H1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A2	J4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A3	J3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A4	J2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A5	J1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A6	K2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A7	K3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A8	K1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A9	L1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A10	L2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A11	L3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A12	L4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A13	M1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A14	M2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A15	M3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A16	M4	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A17	N1	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A18	N2	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A19	N3	O	8	Address Bus. Shared address pins for SDRAM and Static memory.
A20	P1	O	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)

Signal	PBGA Pin	I/O	Drive (mA)	Description
A21	P2	O	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
A22	R1	O	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
A23	P3	O	6	Address Bus. Shared address pins for SDRAM and Static memory. (5V)
CHIP SELECTS				
CS0*	B4	O	4	Shared SDRAM and Static Memory Chip Selects
CS1*	U2	O	4	Shared SDRAM and Static Memory Chip Selects
CS2*	D5 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
CS3*	C4 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects
CS4*	P3 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS5*	R1 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS6*	P2 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects (5V)
CS7*	P1 (shared)	O	6	Shared SDRAM and Static Memory Chip Selects (5V)
RAM CLOCK				
clko	P4	O	8	SDRAM Interface AHB Bus Clock (Z_4)
SDRAM DEDICATED SIGNALS				
clke	R3 (shared)	O	6	SDRAM Interface Clock Enable
ras*	T2	O	8	SDRAM Interface Row Address Strobe
cas*	U1	O	8	SDRAM Interface Column Address Strobe
sDRAM_we	T3	O	8	SDRAM Interface Write Enable
sDRAM_dqm0	V1	O	8	SDRAM Interface Lower byte mask
sDRAM_dqm1	T4	O	8	SDRAM Interface Upper byte mask
sDRAM_bnk0	U3	O	8	SDRAM Interface Bank Address
sDRAM_bnk1	V2	O	8	SDRAM Interface Bank Address
sDRAM_bnk2	W1 (shared)	O (S)	6	SDRAM Interface Bank Address (5V)
SDRAM_bnk3	V3 (shared)	O (S)	6	SDRAM Interface Bank Address (5V)

4 Z indicates that the output is Z-stateable

Signal	PBGA Pin	I/O	Drive (mA)	Description
SRAM INTERFACE				
SRAM_READY	A3	I	6	SRAM ready (read enable)
sRAm_bs[0]	B3	O	4	SRAM lower byte select
sRAm_bs[1]	B2	O	4	SRAM upper byte select
sRAm_we*	A2	O	8	SRAM write enable
sRAm_oe*	C3	O	8	SRAM output enable

Table 4.3: Signal Description

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_IO_SELECT0 – 0xc700 0004. 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 below. Note that the upper 4 CS signals and the upper 4 Address signals share the same pins, so two GPCSR settings are shown.
Using upper 4 Address signals - GPCSR_IO_SELECT0 – 0xc700 0004 = 0x0005 6b54
Using upper 4 CS signals - GPCSR_IO_SELECT0 – 0xc700 0004 = 0x0005 6aaa

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

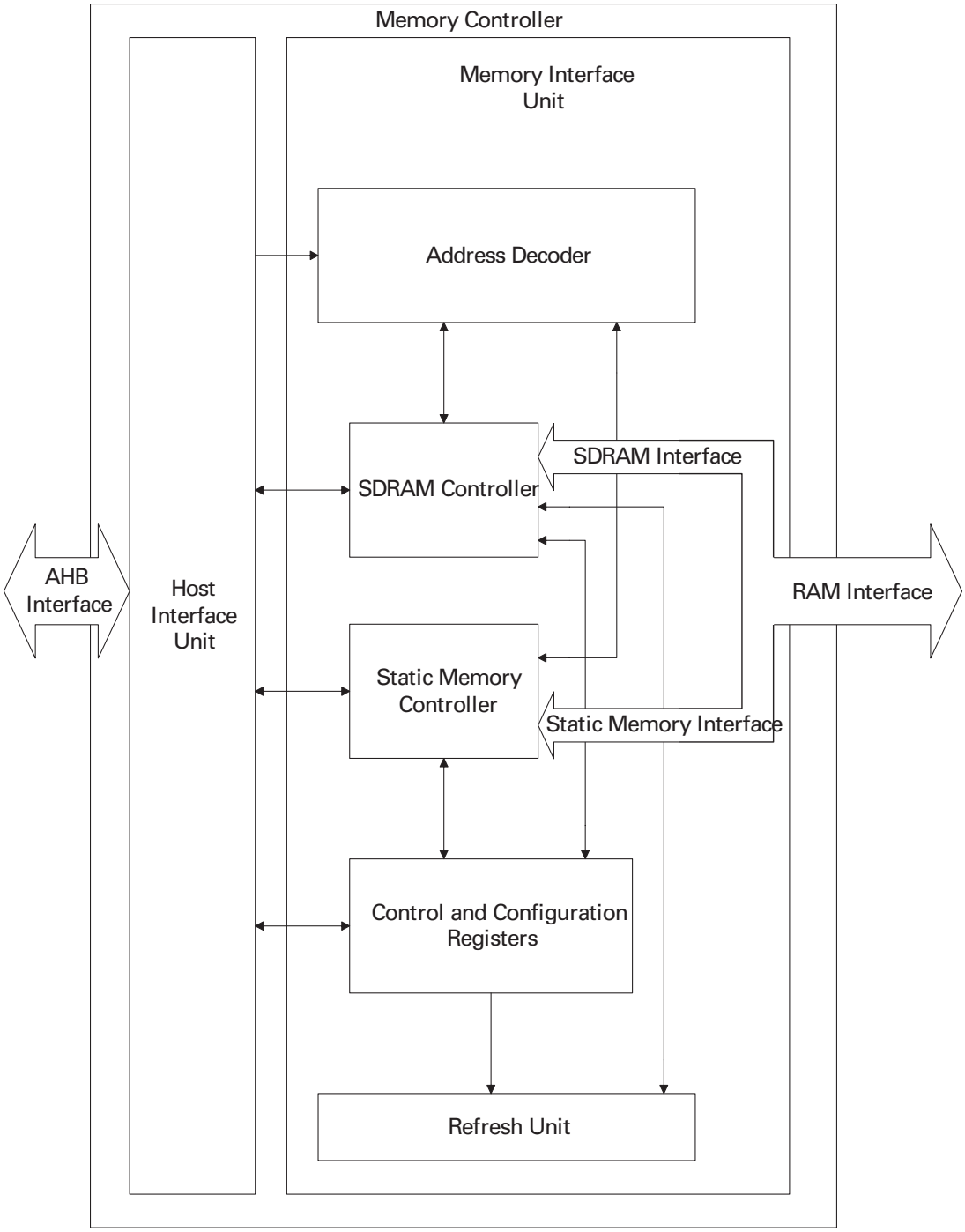


Figure 4.1: Internal Functional Diagram

4.2.6 CONSIDERATIONS FOR CHIP SELECTS

Chip select 2 to 7 (CS2-CS7) 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 CONSIDERATIONS FOR A20 TO A23

Address pins A20 to A23 are default programmed to have an alternative function as an input. If the device connected to CS0 (the boot device) are using those address lines a pull-down should be added to make sure they are interpreted as '0' during boot.

4.2.8 MODULE CONFIGURATION

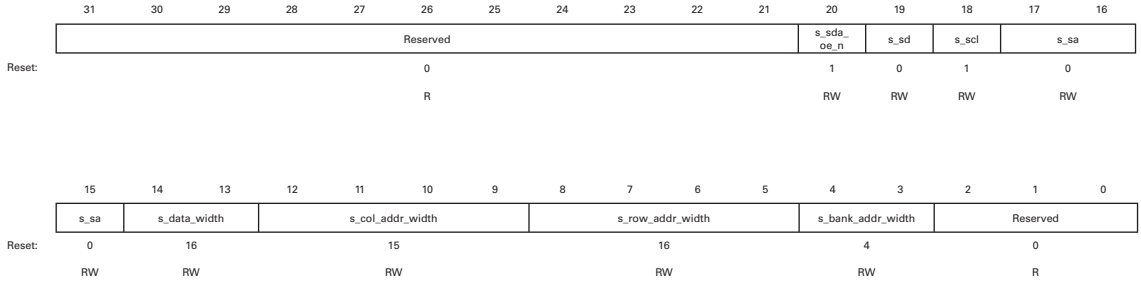
Address	Register
0x8100 0000	EBI_SCONR
0x8100 0004	EBI_STMG0R
0x8100 0008	EBI_STMG1R
0x8100 000c	EBI_SCTLR
0x8100 0010	EBI_SREFR
0x8100 0014	EBI_SCSLR0
0x8100 0018	EBI_SCSLR1
0x8100 001c	EBI_SCSLR2
0x8100 0020	EBI_SCSLR3
0x8100 0024	EBI_SCSLR4
0x8100 0028	EBI_SCSLR5
0x8100 002c	EBI_SCSLR6
0x8100 0030	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 4.4: EBI Module Description

4.2.9 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.



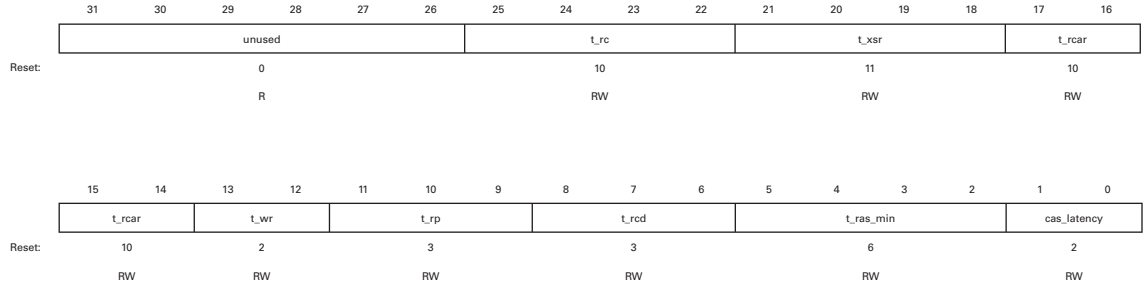
Name	Bit	Reset	Dir	Description
s_sda_oe_n	20	1	RW	Output enable for bi-directional data pin for I2C serial presence detect (SPD) logic 1 – corresponds to data written into bit 19 0 – programs for data reads
s_sd	19	0	RW	Bi-directional data for serial presence detect (SPD) logic; data written into bit 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.

4.2.10 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.

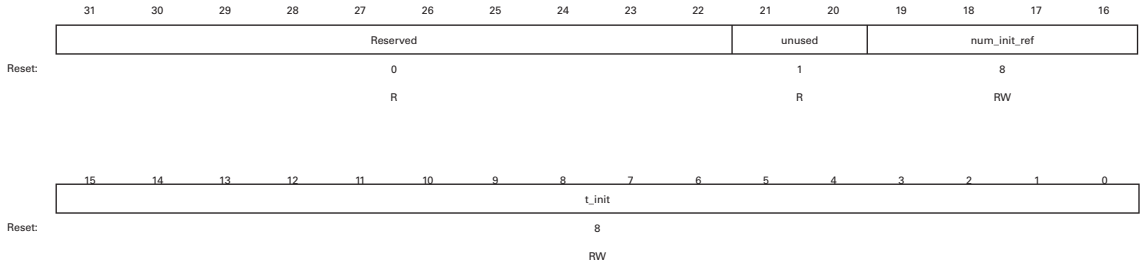


Name	Bit	Reset	Dir	Description
t_rc	25:22	10	RW	Active-to-active command period; values of 0-15 correspond to t_rc of 1-16 clocks.
t_xsr	21:18	11	RW	Exit self-refresh to active or auto-refresh command time; minimum time controller should wait after taking SDRAM out of self-refresh mode before issuing 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.
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

4.2.11 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.

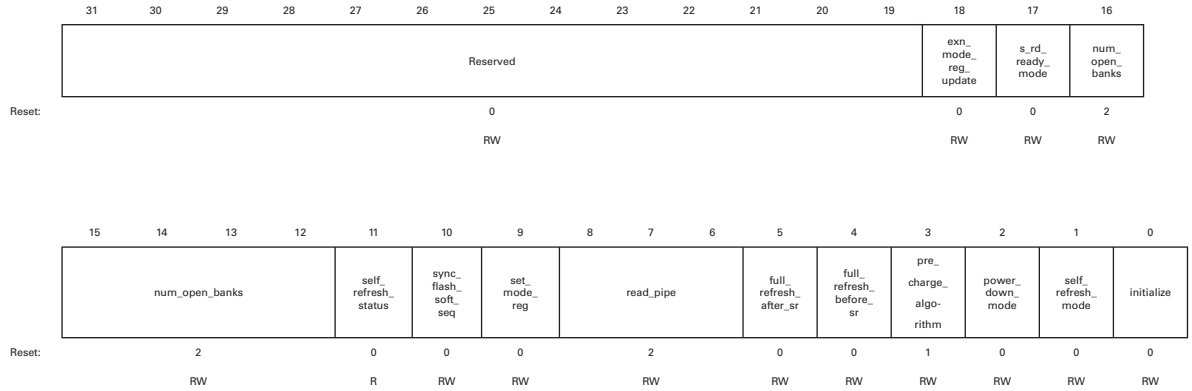


Name	Bit	Reset	Dir	Description
num_init_ref	19:16	8	RW	Number of auto-refreshes during initialization; values 0-15 correspond to 1-16 auto-refreshes
t_init	15:0	8	RW	Number of clock cycles to hold SDRAM inputs stable after power up, before issuing any commands

4.2.12 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.



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 selfrefresh 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 for SyncFlash operations: 1 – Software Command Sequence (SCS) 0 – Hardware Command Sequence (HCS)
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
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.

4.2.13 SREFR – SDRAM REFRESH INTERVAL REGISTER

Address - 0x8100 0010

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

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

4.2.14 SCSLR (0-7) – CHIP SELECT REGISTERS

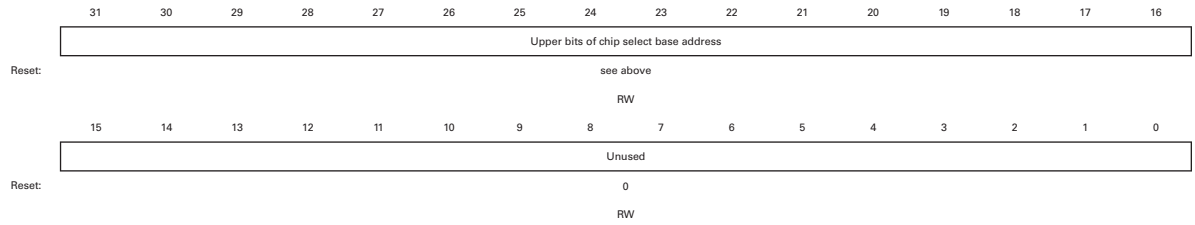
Address - 0x8100 0014 - 0x8100 0030
The memory controller has eight chip selects and a 32-bit AHB address width. There are eight chip select registers, each one holding the base address value that corresponds to its chip select.

The memory controller uses mask registers that specify the size of the memory connected to each chip select. 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 4.5 below is the default after reset. You can later change a default chip select address 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.

0x7FF0_0000	ROM (CS_7)
0x7FE0_0000	ROM (CS_6)
0x7FD0_0000	ROM (CS_5)
0x7FC0_0000	ROM (CS_4)
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 4.5: Default Chip Select Memory Map

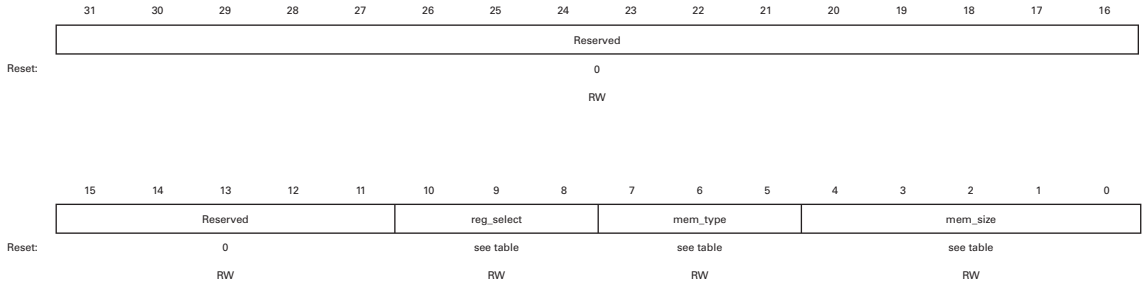


Name	Bit	Reset	Dir	Description
Chip Select Address	31:16	see above	RW	The address of the selected one of eight memory chips

4.2.15 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.



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

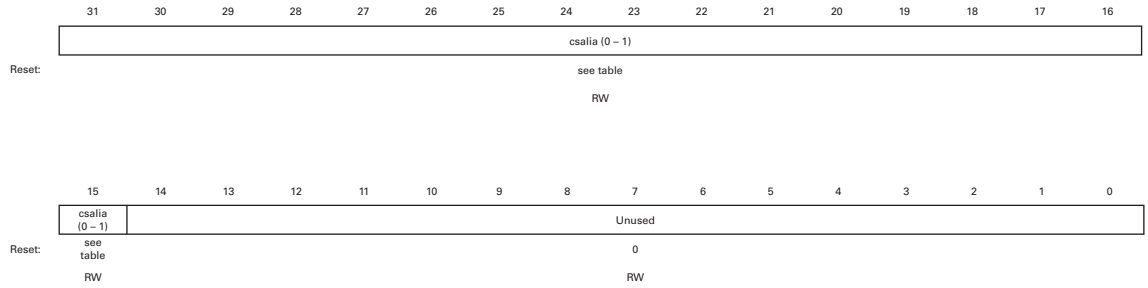
Table 4.6: Default Chip Select Memory Attributes

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

4.2.16 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.



Chip Select	Alias Address
0	0x0
1	0x7f900000

Table 4.7: Alias Addresses

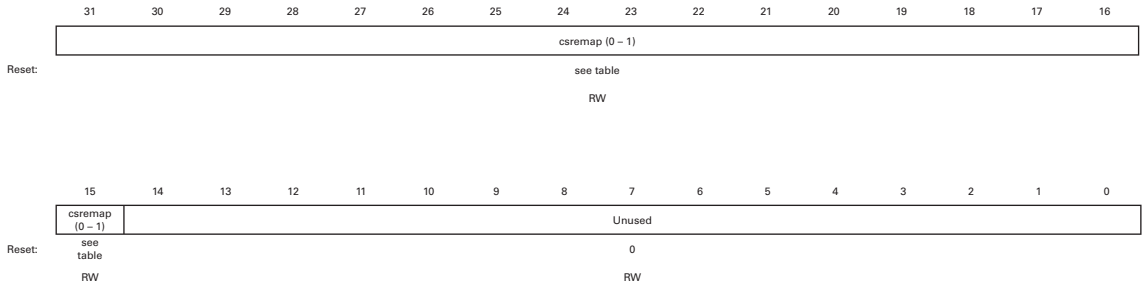
Name	Bit	Reset	Dir	Description
csalias (0 – 1)	31:15	see table	RW	Aliasing register bits for the chip select. Compared with corresponding AHB address to generate the chip select. The number of bits compared depends on size of memory selected by chip select (specified in mask register). 64 KB – bits 31:15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

4.2.17 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.



Chip Select	Remap Address
0	0x0
1	0x7f900000

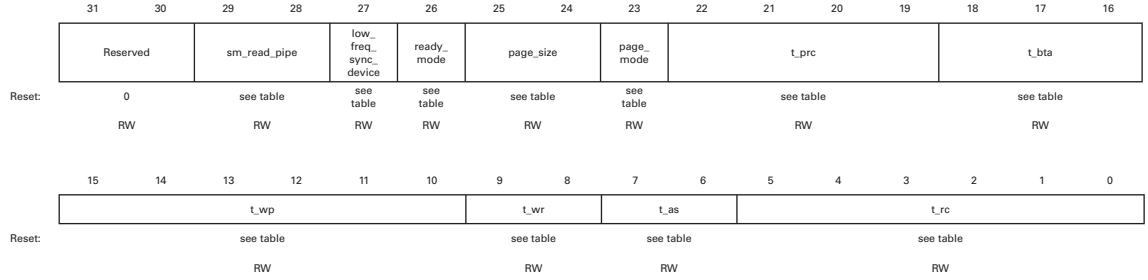
Table 4.8: Remap Addresses

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

4.2.18 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.



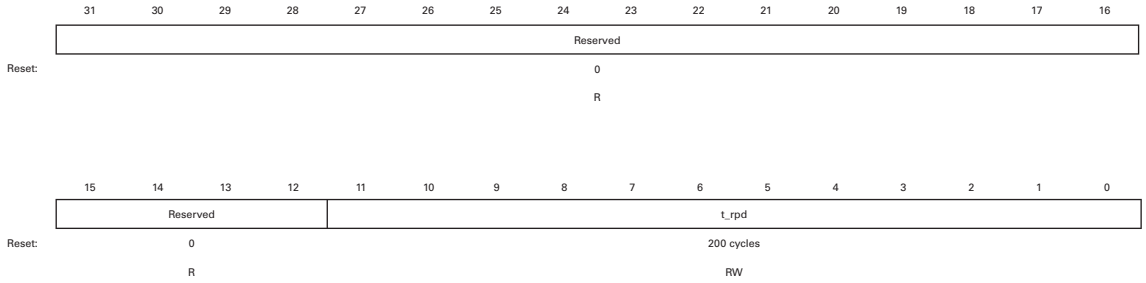
Set	sm_read_ pipe	low_freq_ sync_device	ready_ mode	page_ size	page_ mode	t_prc	t_bta	t_wp	t_wr	t_as	t_rc
0	1	0	0	4	0	1	1	2	0	1	2
1	1	0	0	4	0	16	4	20	3	1	28
2	1	0	0	4	0	15	4	20	3	3	63

Table 4.9: 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
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.

4.2.19 FLASH_TRPDR – FLASH TIMING REGISTER

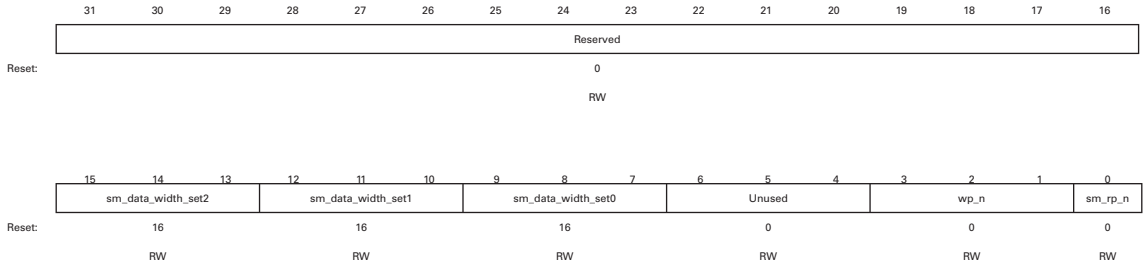
Address - 0x8100 00A0



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.

4.2.20 SMCTLR – STATIC MEMORY CONTROL REGISTER

Address - 0x8100 00A4



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

4.2.21 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.

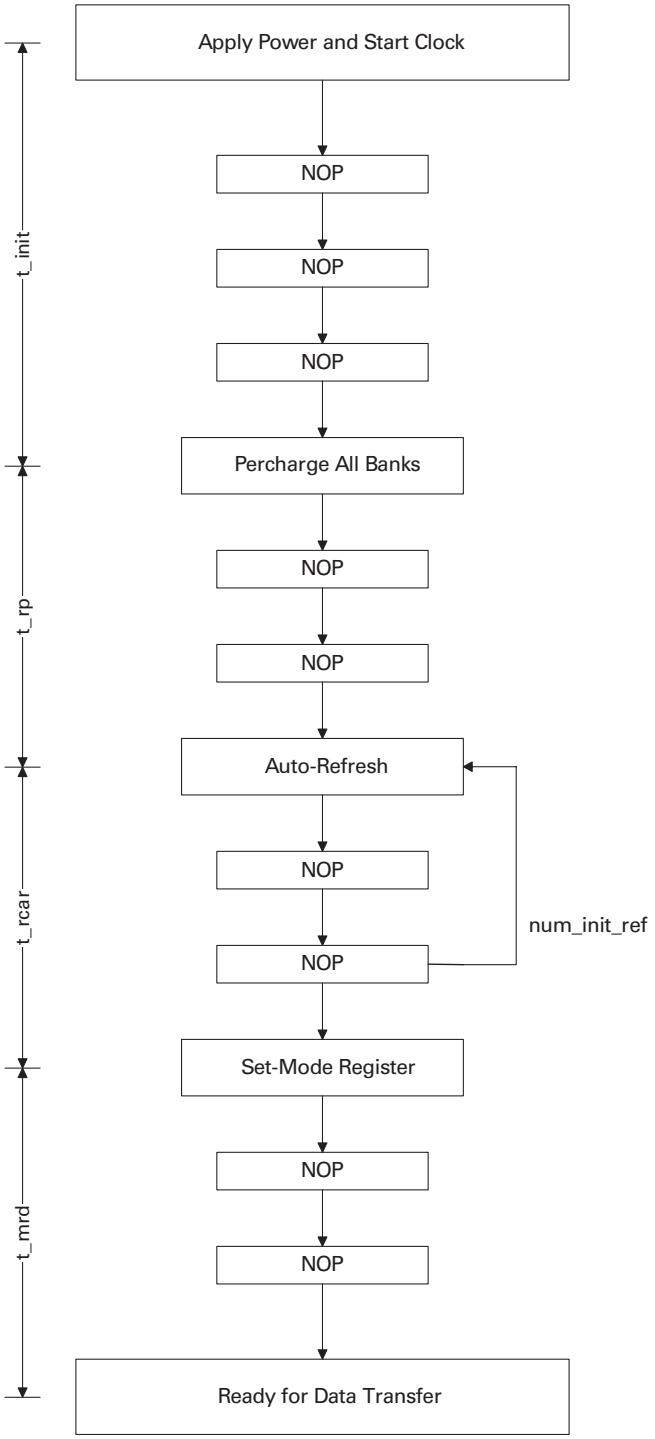


Figure 4.2: SDRAM Power ON Initialization Diagram

4.2.22 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 4.10: Read/Write Timing Delays

4.2.23 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.

4.2.24 SDRAM REFRESH

4.2.24.1 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 4.11: 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.

4.2.24.2 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.

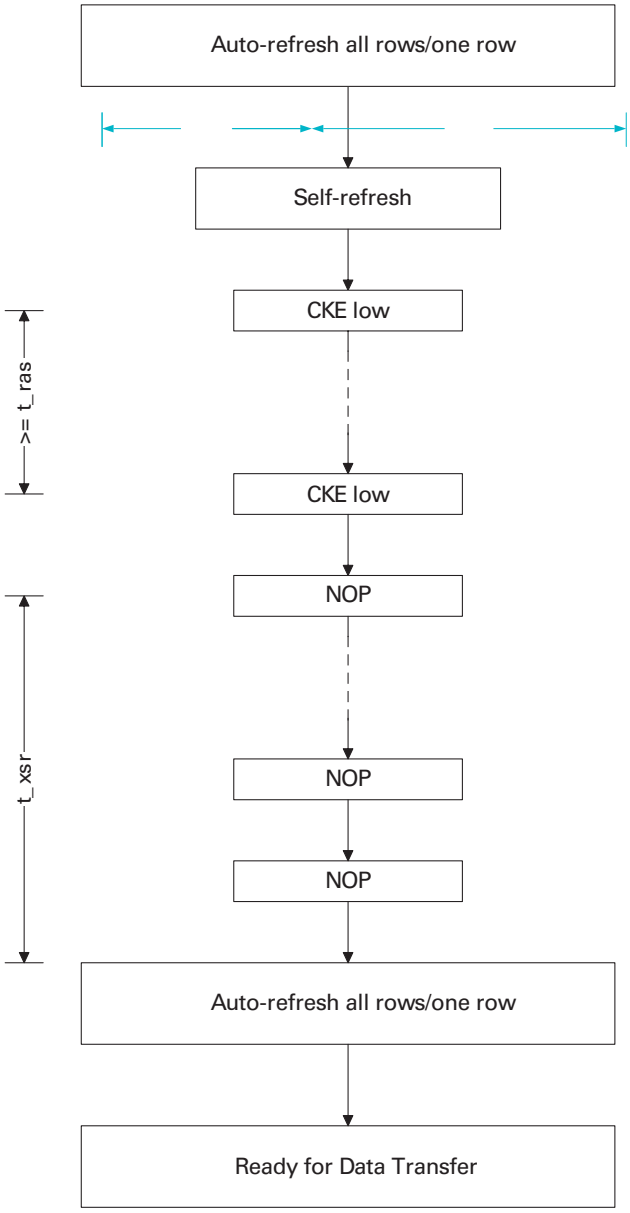


Figure 4.3: Auto-Refresh Diagram

4.2.25 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; it should be 0 to bring the SDRAM out of power-down mode.

- Use the power-down input pin; it 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.

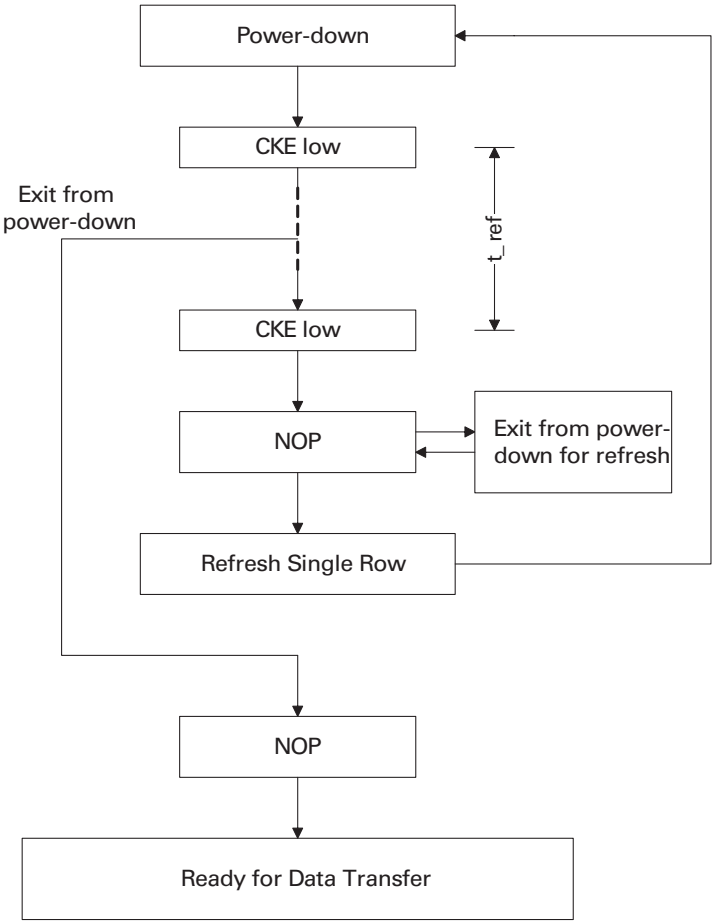


Figure 4.4: Power DOWN Diagram

4.2.26 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 select 0 and chip select 1.

When aliasing is enabled, the chip select becomes active when the AHB address matches either the chip select base address or the chip select alias address. Similarly, 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 4.12: Chip Select Decoding

4.2.27 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 4.13: SDRAM Read/Write Timing Parameters

4.2.27.1 SDRAM PAGE-HIT SINGLE WRITE (HBURST = SINGLE)

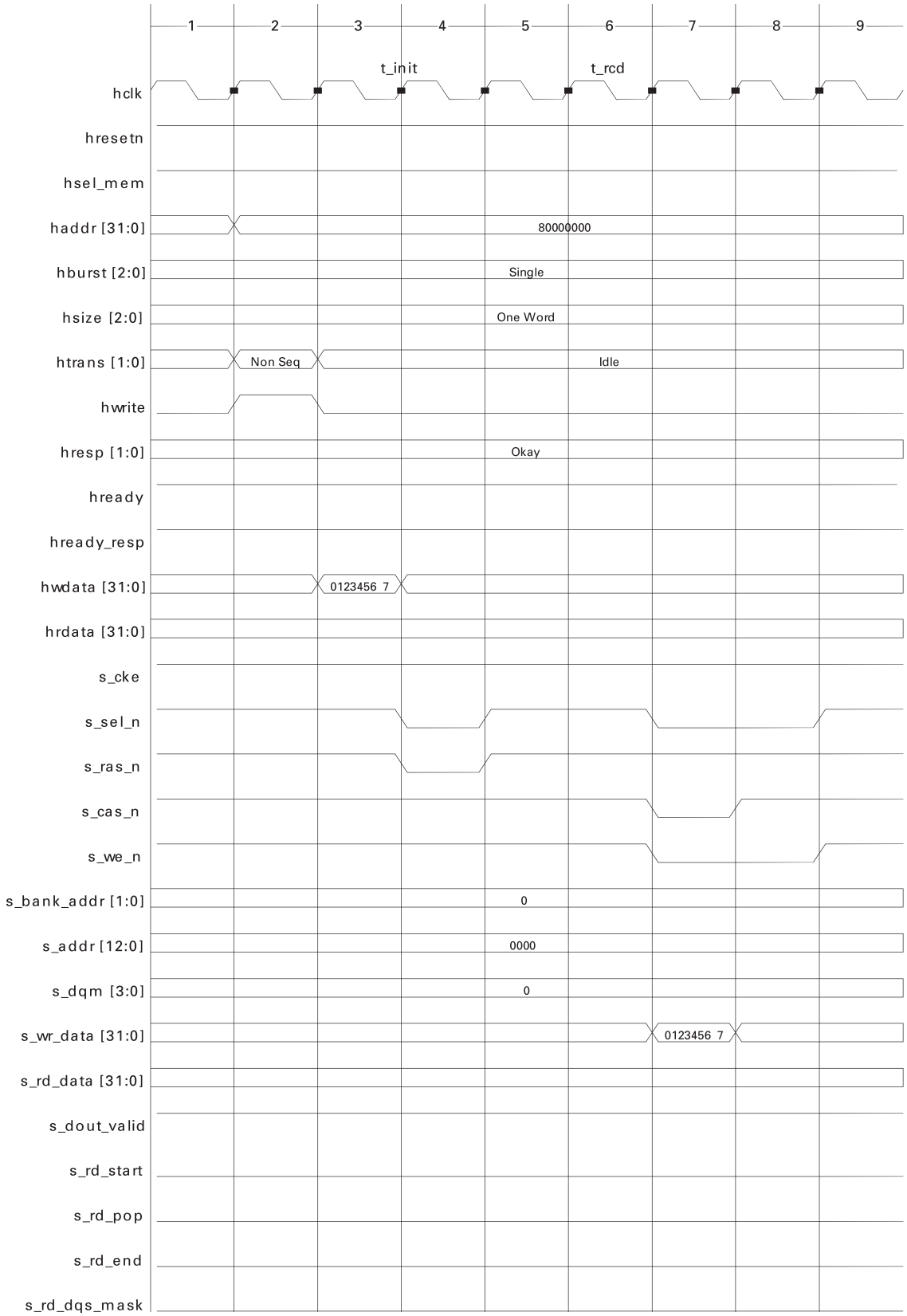


Figure 4.5: SDRAM Page-Hit Single Write

4.2.27.2 SDRAM PAGE-MISS SINGLE WRITE (HBURST = SINGLE)

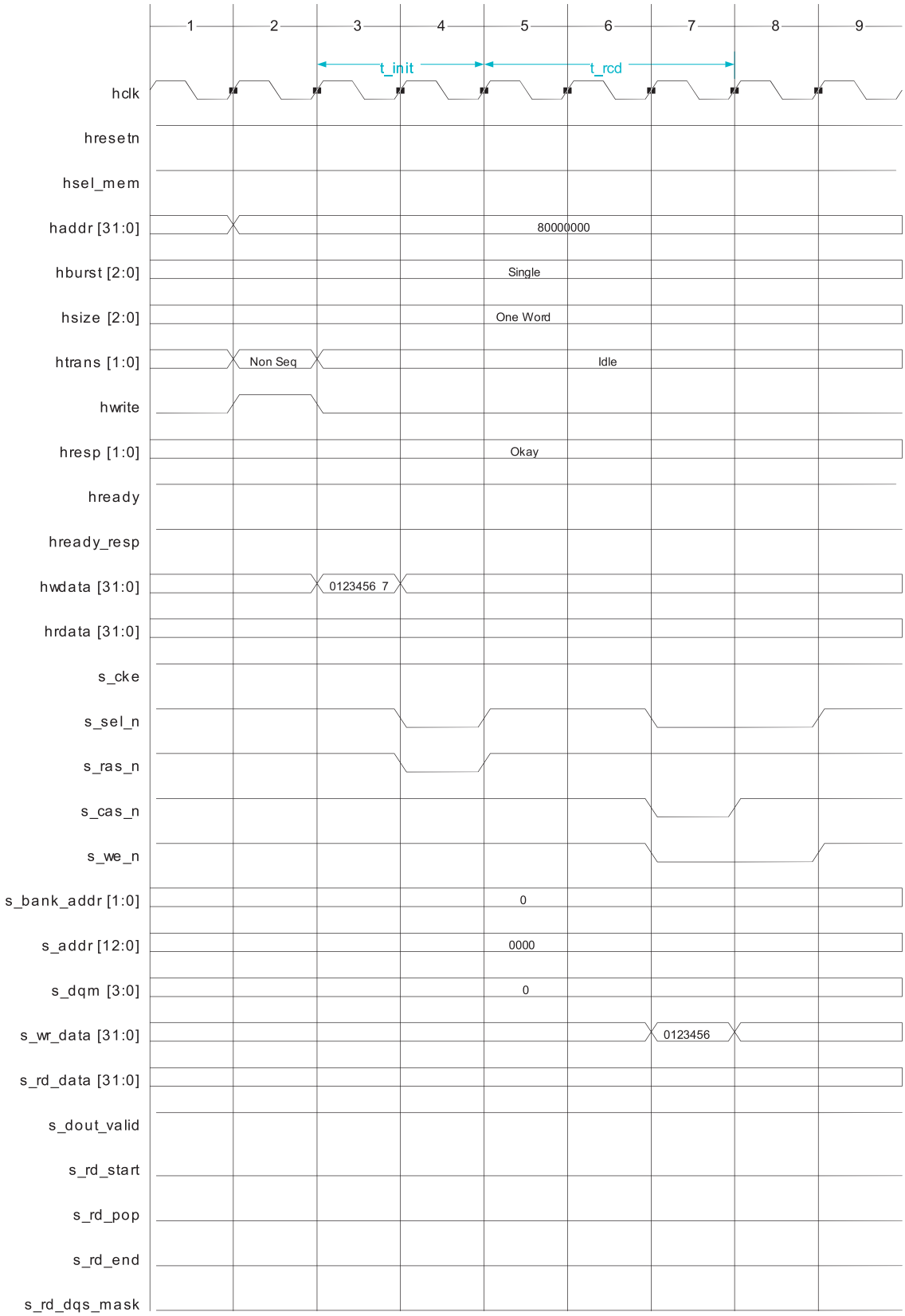


Figure 4.6: SDRAM Page-Miss Single Write

4.2.27.3 SDRAM PAGE-HIT BURST WRITE (HBURST = INCR 8)

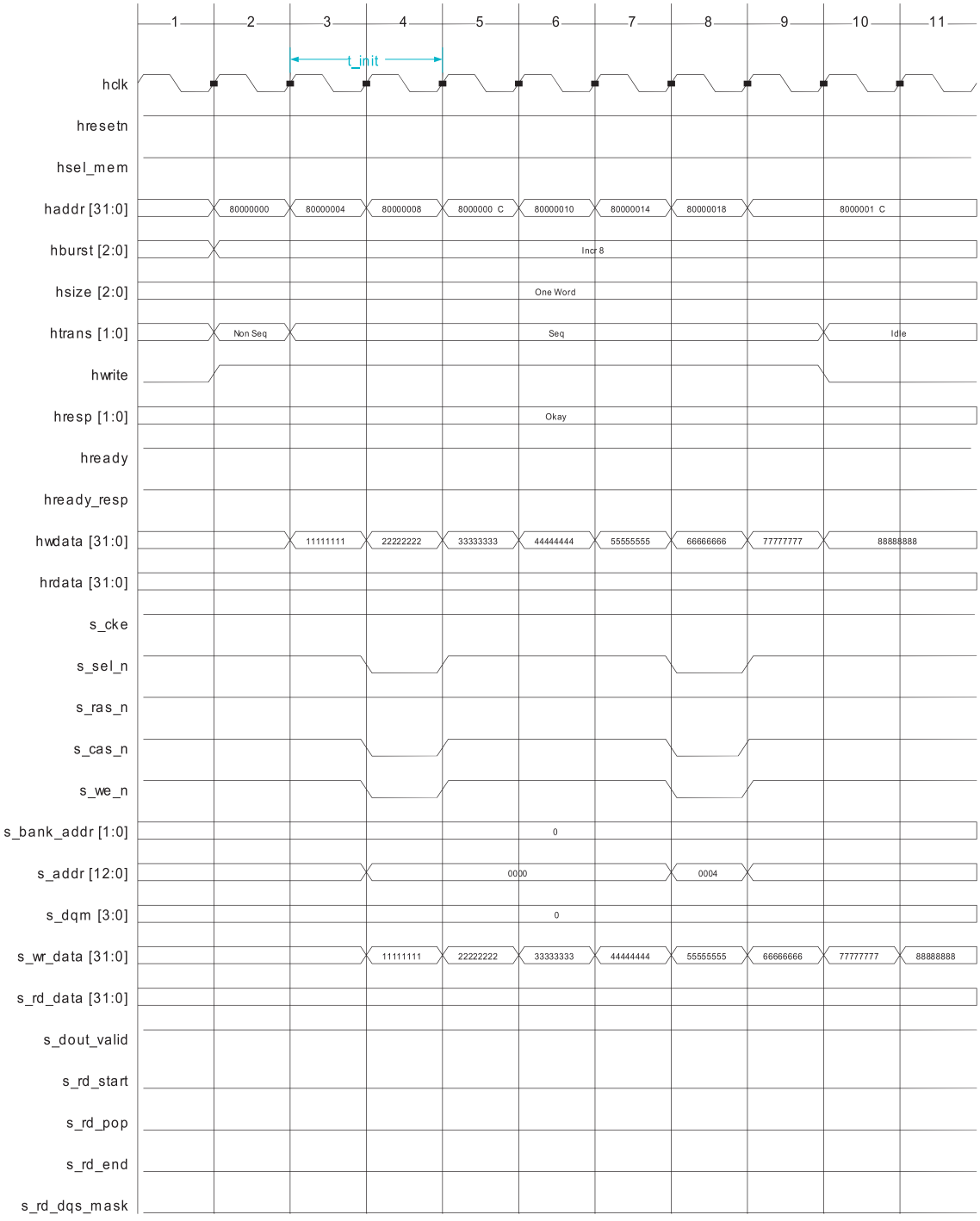


Figure 4.7: SDRAM Page-Hit Burst Write

4.2.27.4 SDRAM PAGE-HIT SINGLE READ (HBURST = SINGLE)

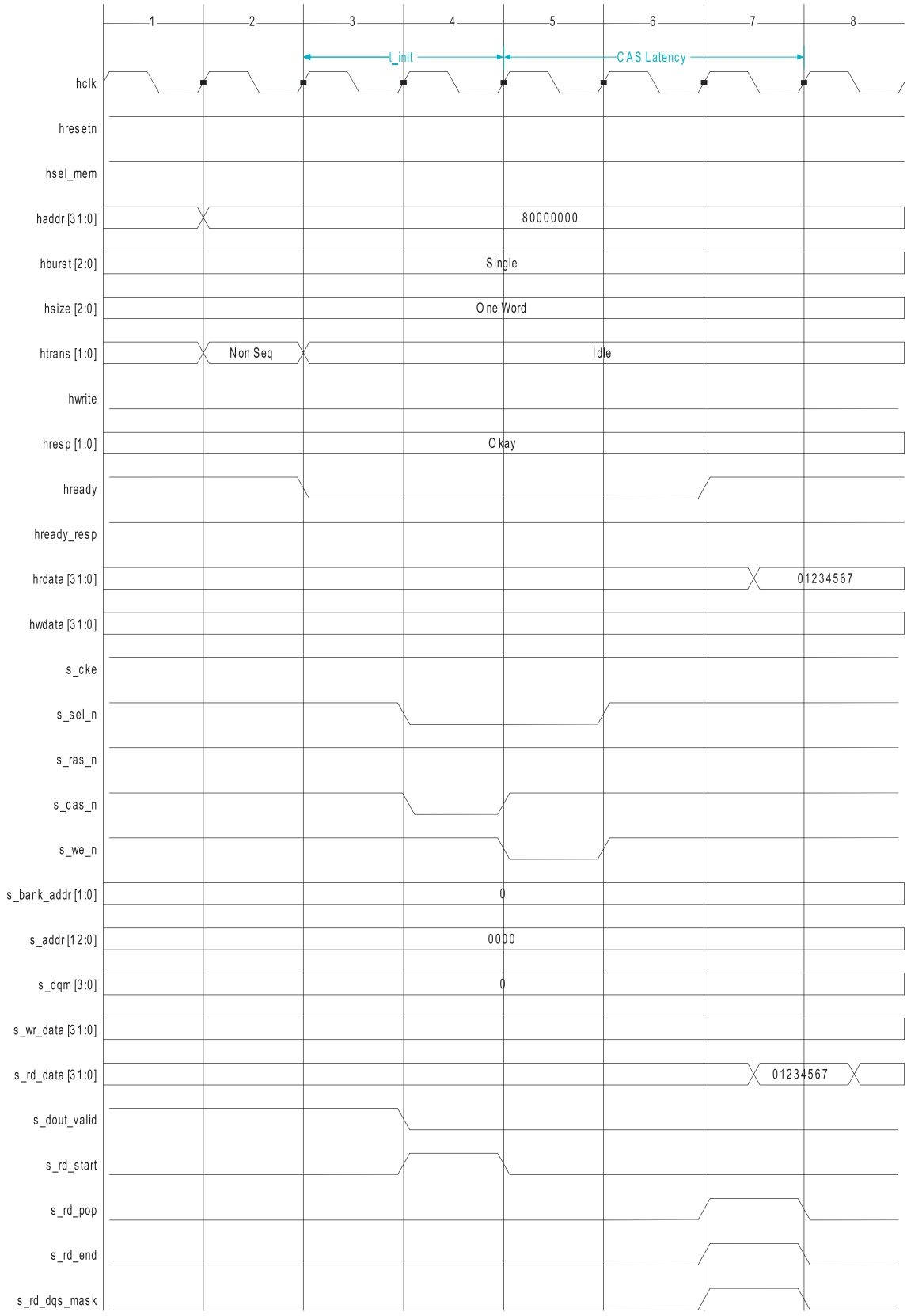


Figure 4.8: SDRAM Page-Hit Single Read

4.2.27.5 SDRAM PAGE-MISS SINGLE READ (HBURST = SINGLE)

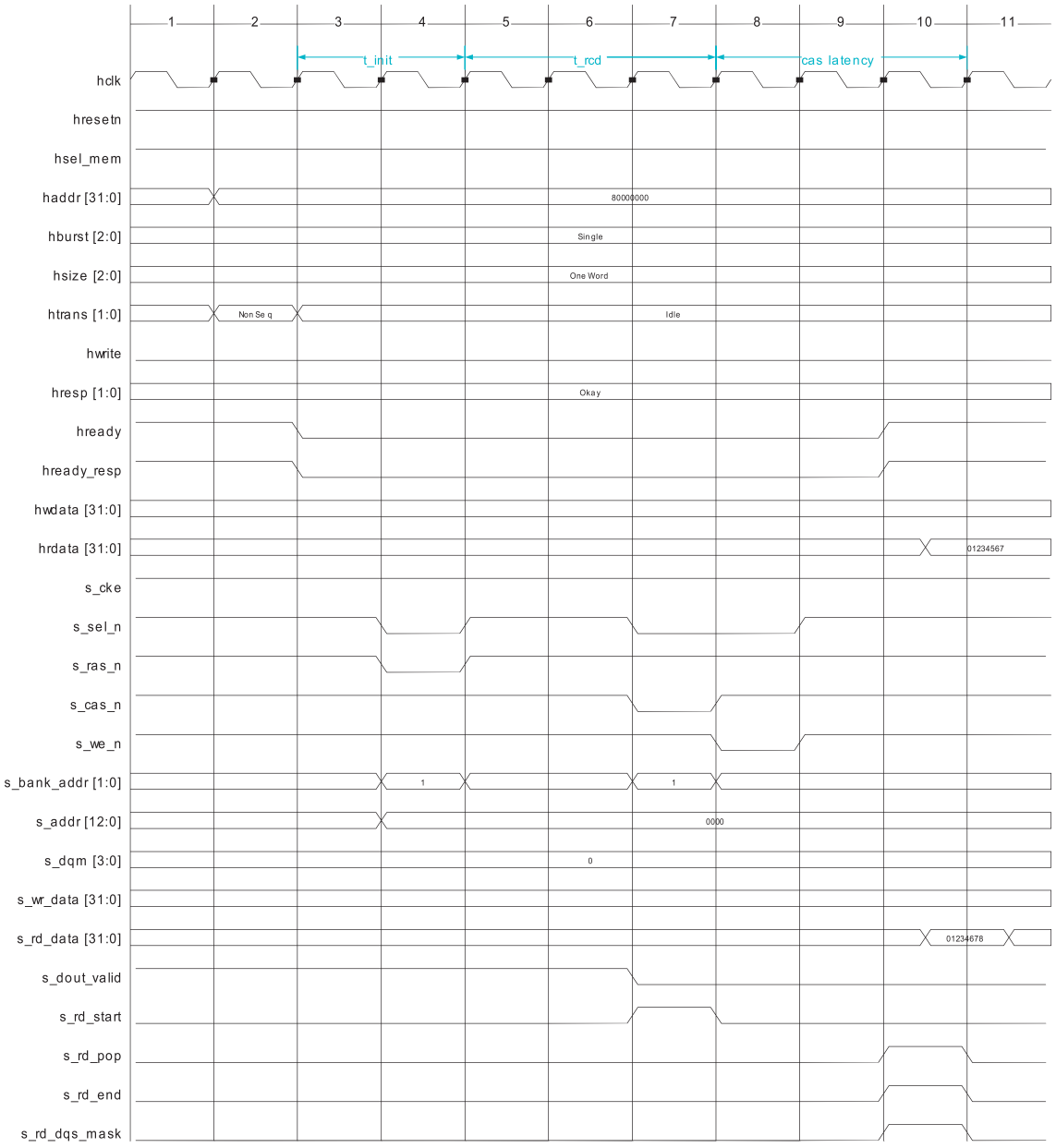


Figure 4.9: SDRAM Page-Miss Single Read

4.2.27.6 SDRAM PAGE-HIT BURST READ (HBURST = INCR 8)

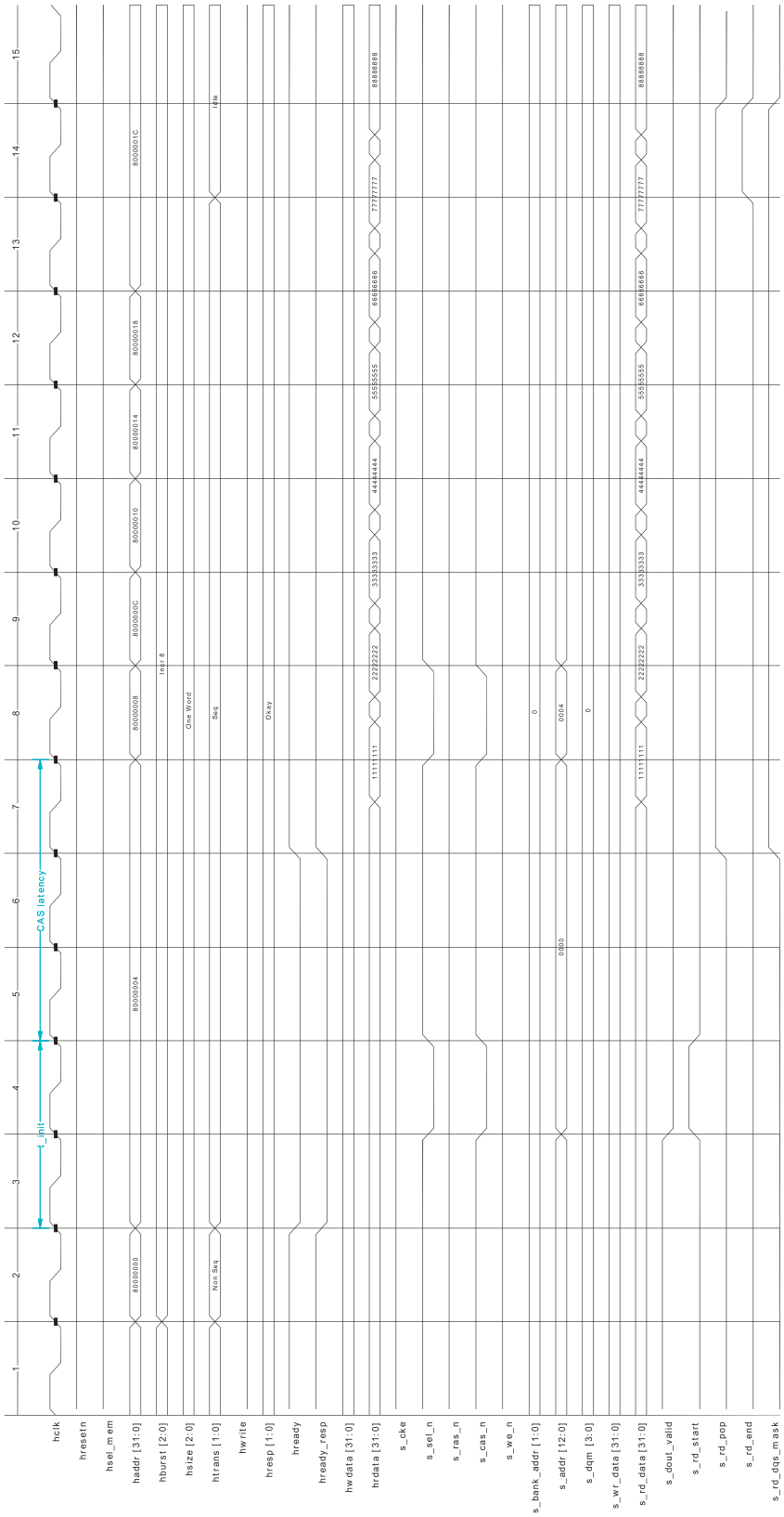


Figure 4.10: SDRAM Page-Hit Burst Read

4.2.28 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.

4.2.29 STATIC MEMORY CHIP SELECTION

The memory controller supports up to eight chip selects. The Address Mask Registers (SMSKRn) and the Chip Select Registers (SCSLRn) 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.

4.2.30 FLASH MEMORY

4.2.30.1 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.

4.2.30.2 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.

4.2.30.3 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).

4.2.31 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 4.14: Static Memory Read/Write Timing Parameters

4.2.31.1 READ TIMING OF SRAM, FLASH AND ROM

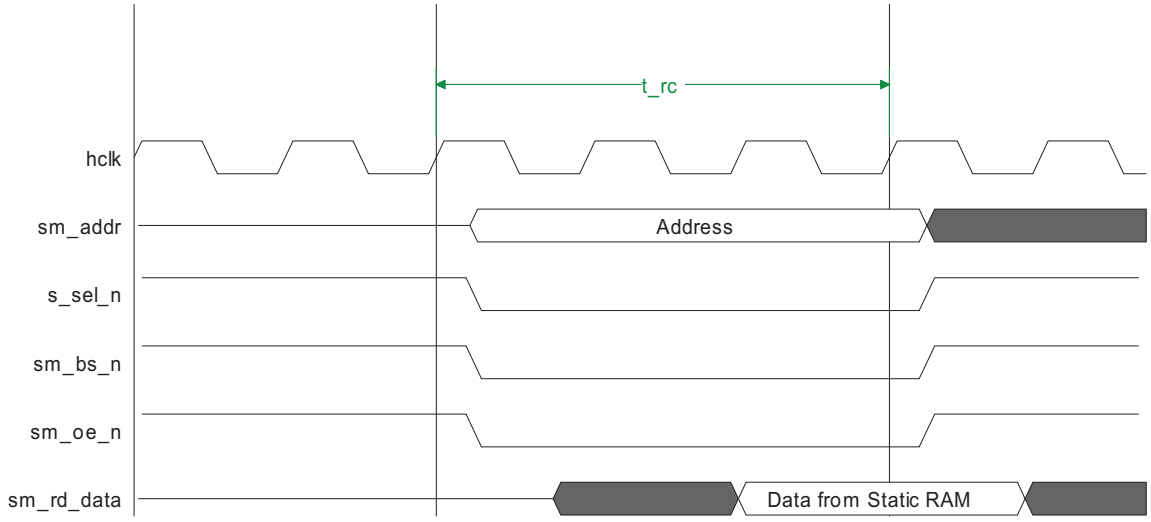


Figure 4.11: Static Read Timing

4.2.31.2 PAGE READ TIMING OF FLASH AND ROM

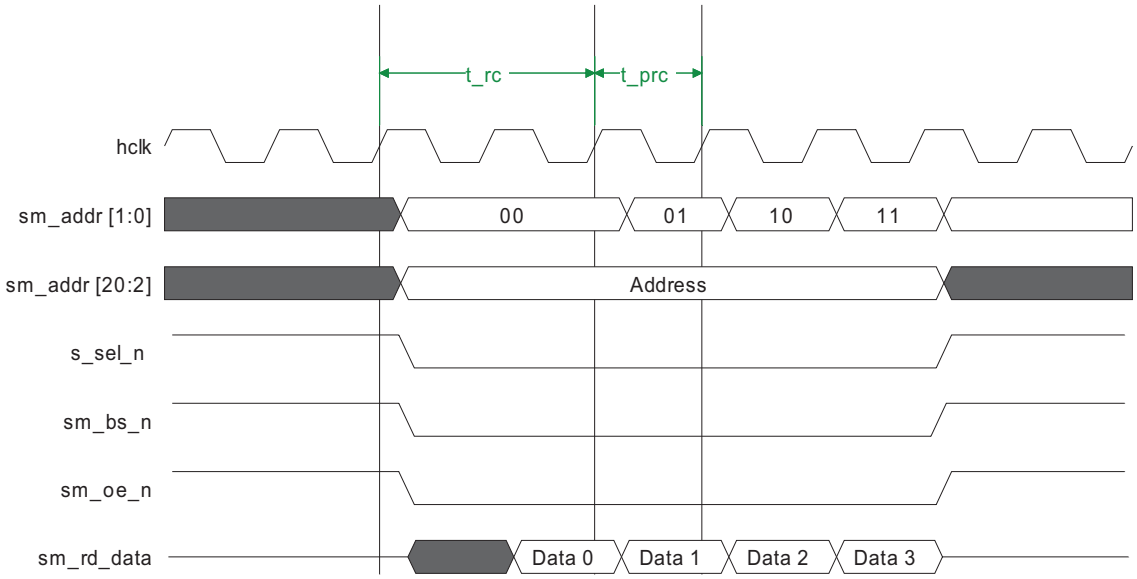


Figure 4.12: FLASH and ROM Page Read Timing

4.2.31.3 WRITE TIMING OF SRAM AND FLASH

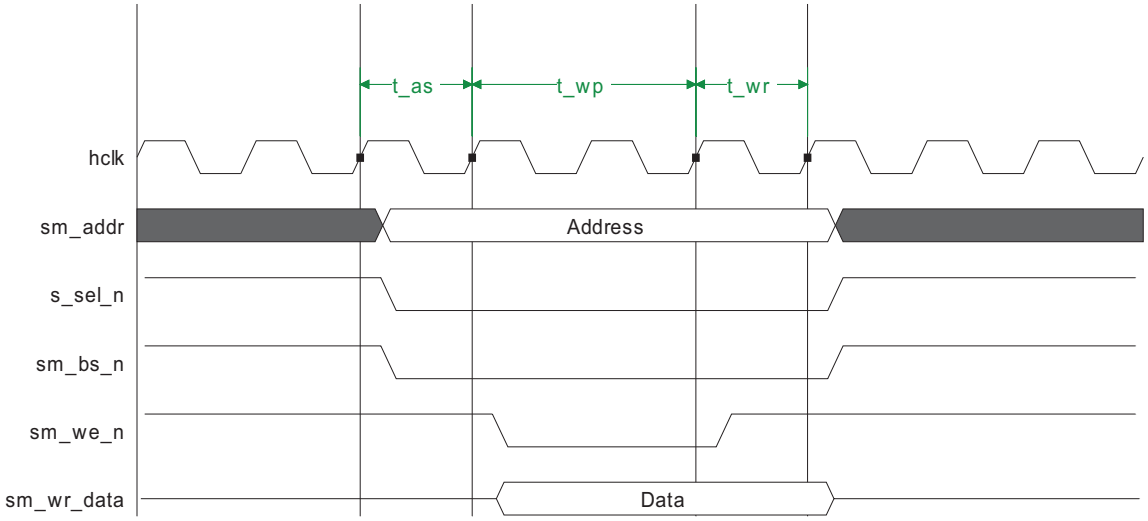


Figure 4.13: SRAM and FLASH Write Timing

4.2.31.4 EXTERNAL MEMORY DATA BUS TURNAROUND TIMING

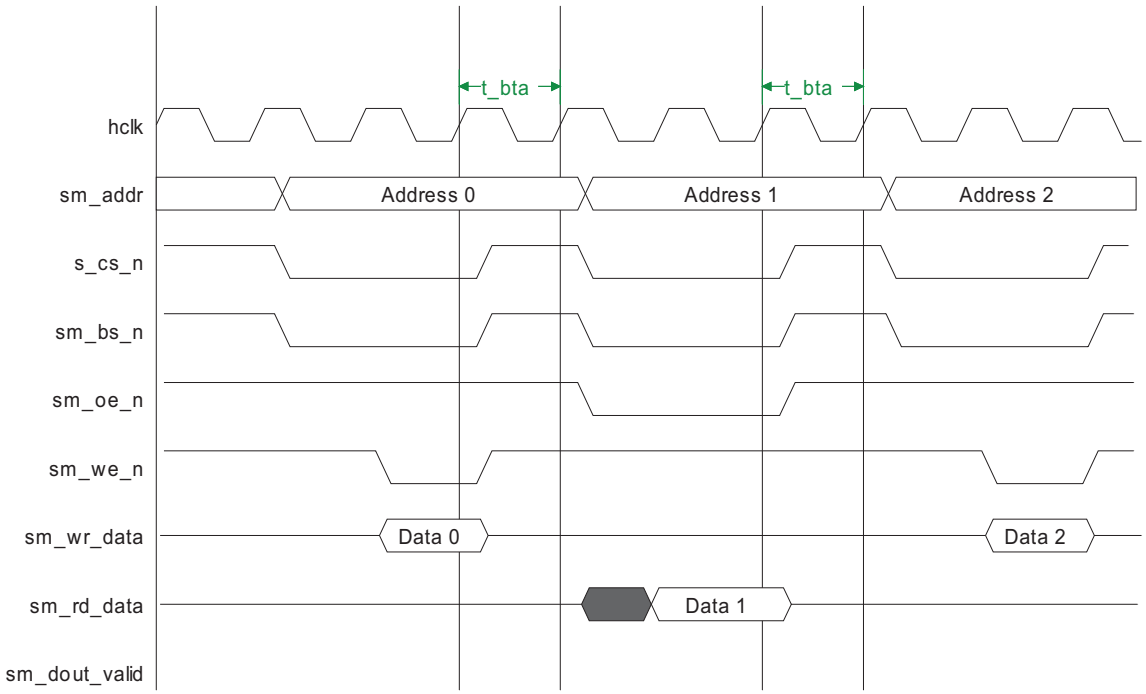


Figure 4.14: Turnaround Timing

4.2.31.5 FIRST READ AFTER RESET/POWER-DOWN

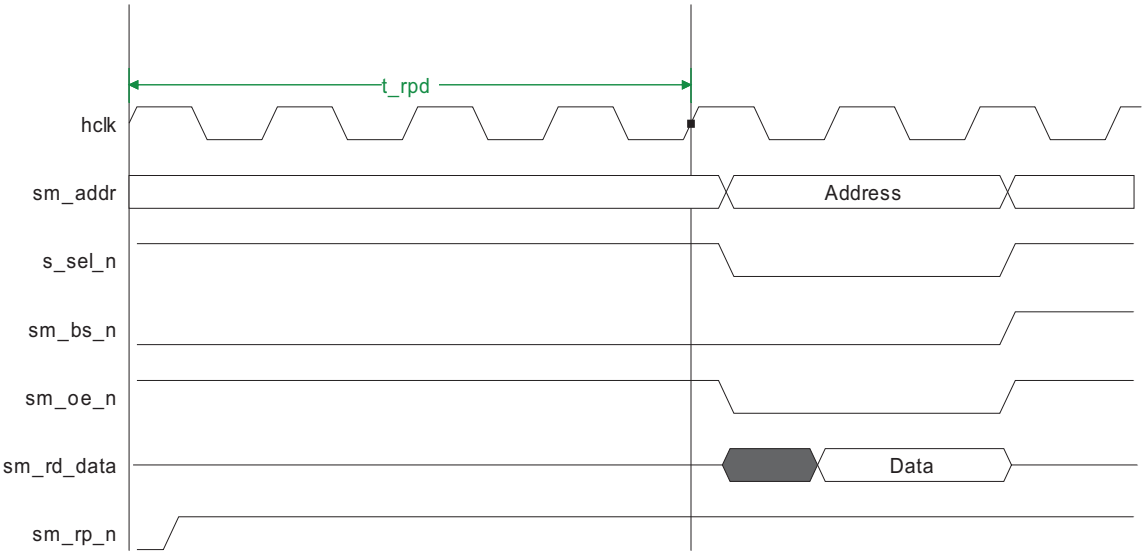


Figure 4.15: Reset/Power-Down Timing

4.2.32 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 DICEII. Note that the SRAM_READY pin is a multi-function pin, so the SRAM_READY bit of the GPCSR_IO_SELECT0 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.

4.2.32.1 I/O INTERFACE BETWEEN NON-MEMORY DEVICE AND DICEII

The following are conditions for interfacing Non-Memory device pins to the DICEII Memory Controller pins:

- Address pins – Connect the address pins to the DICEII 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 DICEII data bus pins.

4.2.32.2 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.

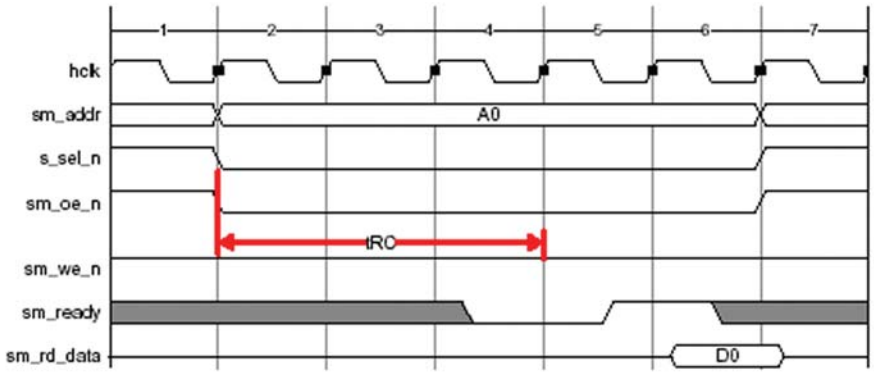


FIgure 4.16: Timing Diagram of Read Access

4.2.32.3 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.

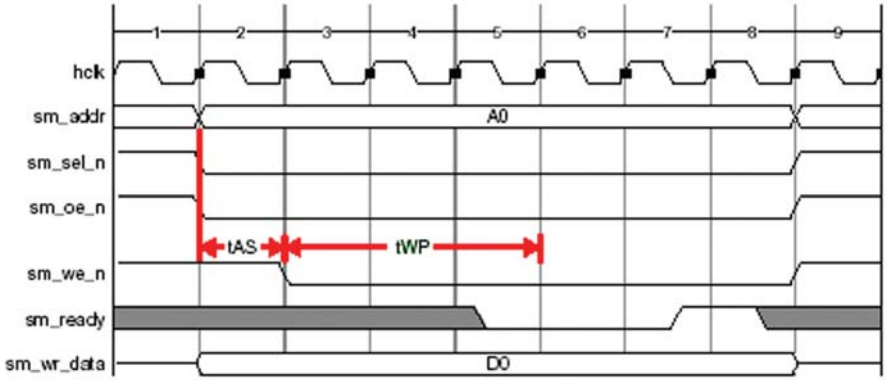


Figure 4.17: Write Access of the Device with Ready Signal

(Note: this diagram has an error - sm_oe_n remains HI during a write cycle. This will be fixed in the next revisionof this document.)

4.2.32.4 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 I²C

4.3.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
i2c_cLK	A4	I/O (S)	6	I2C Clock (OD[12], 5V)
i2c_dATA	C5	I/O (S)	6	I2C Data (OD, 5V)

Table 4.15: I²C 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 DICE II is a fully compliant with Philips I2C definitions. The following is a list of general features of the DICE II 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

4.3.3 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 an acknowledge (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.

4.3.4 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.

4.3.5 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.

4.3.6 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.

4.3.7 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.

4.3.8 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 whichthe I2C Interface responds.
3. Write to the IC_CON register to specify whether 10-bit addressing is supported andwhether 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 assuing 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 another 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 requests 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.

4.3.9 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

4.3.10 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.

4.3.11 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 4.16: I²C Register Memory Map

4.3.12 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.

4.3.13 IC_CON REGISTER – I2C CONTROL

Address – 0xC400 0000

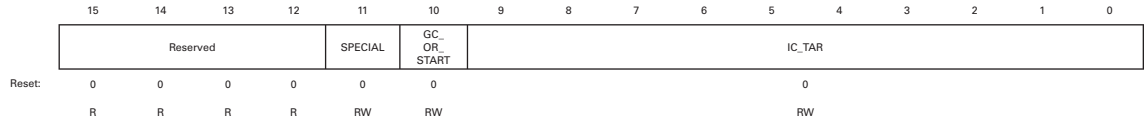
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved									IC_ SLAVE _DIS ABLE	IC_RE START _EN	IC_10BIT ADDR_ MASTER	IC_ 10BIT ADDR_ SLAVE	SPEED		IC_ MASTER_ MODE
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	3		1
	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW		RW

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

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
IC_10BITADDR_SLAVE	3	1	RW	When acting as a slave, this blt 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

4.3.14 IC_TAR REGISTER – I2C TARGET ADDRESS

Address – 0xC400 0004

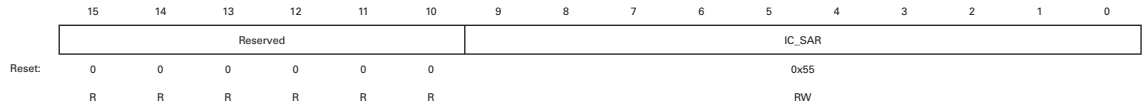


This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

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
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.

4.3.15 IC_SAR REGISTER – I2C SLAVE ADDRESS

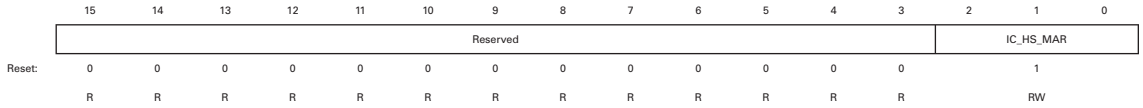
Address – 0xC400 0008



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.

4.3.16 IC_HS_MAR REGISTER – I2C MASTER MODE CODE ADDRESS

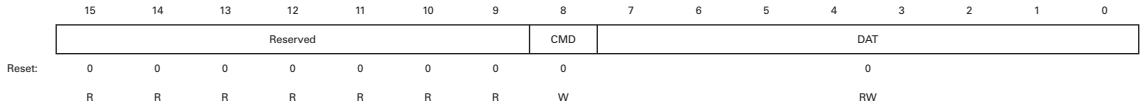
Address – 0xC400 000C



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.

4.3.17 IC_DATA_CMD REGISTER – I2C RX/TX DATA BUFFER AND COMMAND

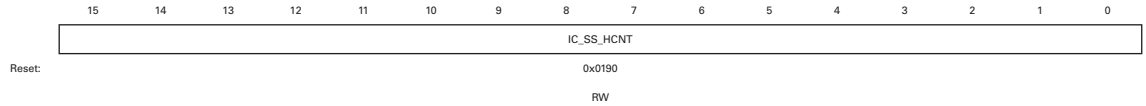
Address – 0xC400 0010



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 be 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.

4.3.18 IC_SS_HCNT REGISTER –STANDARD SPEED IC_CLK HIGH COUNT

Address – 0xC400 0014



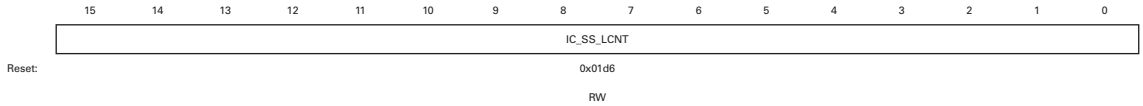
Name	Bit	Reset	Dir	Description
IC_SS_HCNT	15:0	0x0190	RW	The IC_SS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for STANDARD speed. It is not used for HIGH SPEED mode but must be set correctly since 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 4.17. 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

Table 4.17: I²C Standard Speed High Period Count Calculations

4.3.19 IC_SS_LCNT REGISTER – STANDARD SPEED IC_CLK LOW COUNT

Address – 0xC400 0018



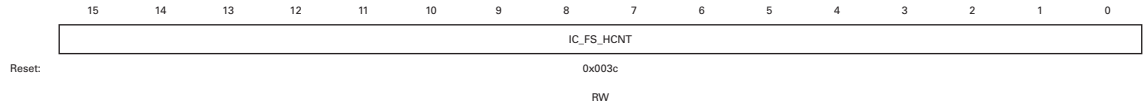
Name	Bit	Reset	Dir	Description
IC_SS_LCNT	15:0	0x01d6	RW	The IC_SS_LCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for STANDARD speed. It is not used for HIGH SPEED mode but must be set correctly since 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 4.18. 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 LowTime (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

Table 4.18: I²C Standard Speed Low Period Count Calculations

4.3.20 IC_FS_HCNT REGISTER – FAST SPEED IC_CLK HIGH COUNT

Address – 0xC400 001C



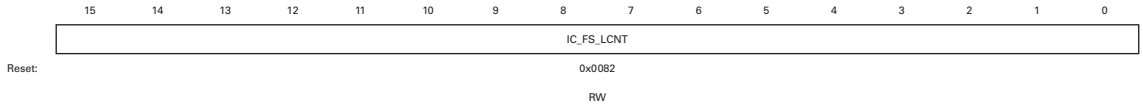
Name	Bit	Reset	Dir	Description
IC_FS_HCNT	15:0	0x003c	RW	The IC_FS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for FAST SPEED. It is not used for HIGH SPEED mode but must be set correctly since 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 4.19. 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

Table 4.19: I²C Fast Speed High Period Count Calculations

4.3.21 IC_FS_LCNT REGISTER – FAST SPEED IC_CLK LOW COUNT

Address – 0xC400 0020



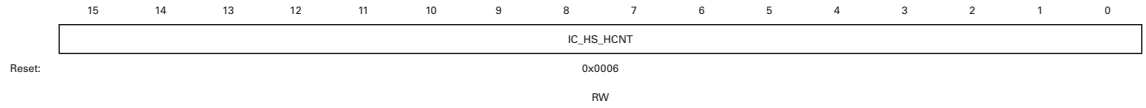
Name	Bit	Reset	Dir	Description
IC_FS_LCNT	15:0	0x0082	RW	The IC_FS_LCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for FAST SPEED. It is not used for HIGH SPEED mode but must be set correctly since 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 4.20. 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

Table 4.20: I²C Fast Speed Low Period Count Calculations

4.3.22 IC_HS_HCNT REGISTER – HIGH SPEED IC_CLK HIGH COUNT

Address – 0xC400 0024



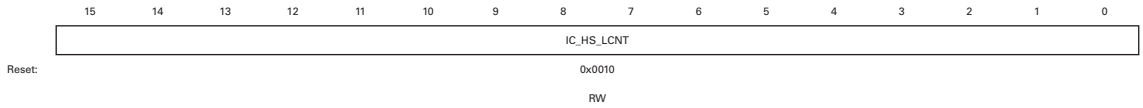
Name	Bit	Reset	Dir	Description
IC_HS_HCNT	15:0	0x0006	RW	The IC_HS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock high period count for HIGH SPEED. Sample I2C HIGH SPEED high period count calculations are shown in Table 4.21. 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

Table 4.21: I²C High Speed High Period Count Calculations

4.3.23 IC_HS_LCNT REGISTER – HIGH SPEED IC_CLK LOW COUNT

Address – 0xC400 0028



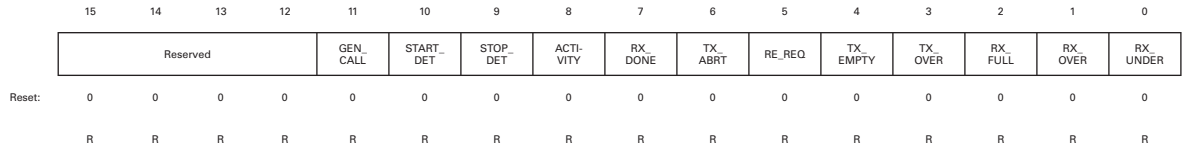
Name	Bit	Reset	Dir	Description
IC_HS_LCNT	15:0	0x0010	RW	The IC_HS_HCNT Register must be set before any I2C bus transaction can take place to insure proper I/O timing. This register sets the SCL clock low period count for HIGH SPEED. Sample I2C HIGH SPEED low period count calculations are shown in Table 4.22. 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

Table 4.22: I²C High Speed Low Period Count Calculations

4.3.24 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.



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.

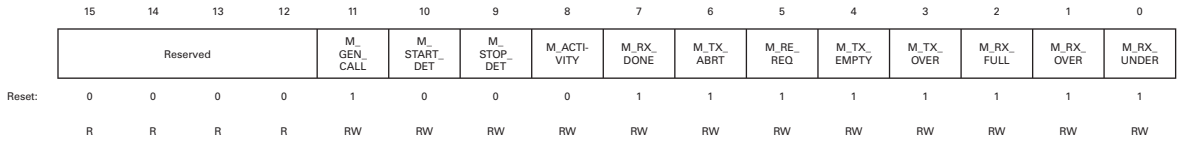
Name	Bit	Reset	Dir	Description
TX_ABRT	6	0	R	In general this bit will be set to logic 1 when the I2C module acting as a master is unable to complete a command that the processor has sent. The conditions which set TX_ABRT are: · no slave acknowledges after the address is sent · the addressed slave does 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.
RX_FULL	2	0	R	Set when the receive buffer goes above the RX_TL threshold in Register IC_RX_TL. Automatically cleared by hardware when buffer level goes below the threshold.

Name	Bit	Reset	Dir	Description
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.

4.3.25 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.



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

4.3.26 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

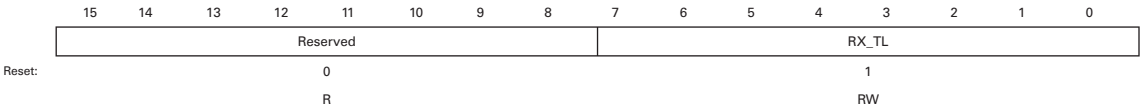
Name	Bit	Reset	Dir	Description
R_GEN_CALL	11	0	R	Indicates that a general call request was received. The I2C module will store the received data in the RX buffer.
R_START_DET	10	0	R	Indicates whether a start condition has occurred on the I2C interface
R_STOP_DET	9	0	R	Indicates whether a stop condition 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.
R_RX_FULL	2	0	R	Set when the receive buffer goes above the RX_TL threshold in Register IC_RX_TL. Automatically cleared by hardware when buffer level goes below the threshold.

Name	Bit	Reset	Dir	Description
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.

4.3.27 IC_RX_TL REGISTER – I2C RX THRESHOLD LEVEL

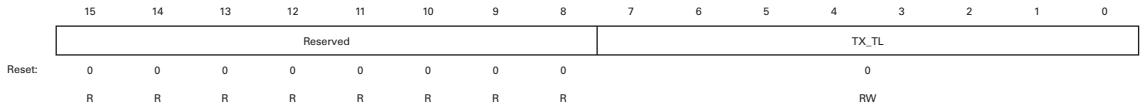
Address – 0xC400 0038



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.

4.3.28 IC_TX_TL REGISTER – I2C TX THRESHOLD LEVEL

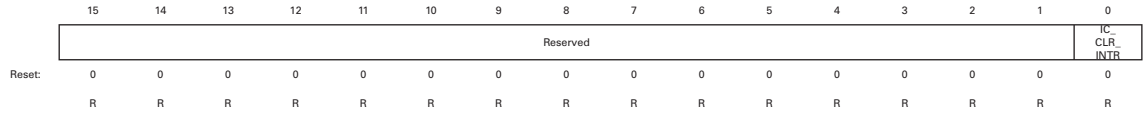
Address – 0xC400 003C



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.

4.3.29 IC_CLR_INTR REGISTER – CLEAR ALL INTERRUPTS

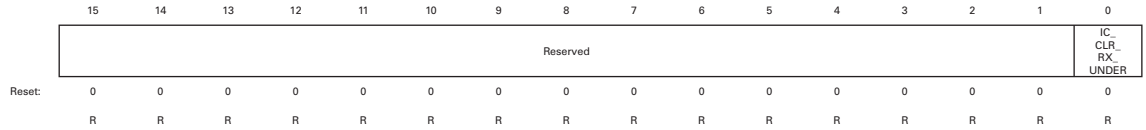
Address – 0xC400 0040



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

4.3.30 IC_CLR_ RX_UNDER REGISTER – CLEAR RX_UNDER INTERRUPT

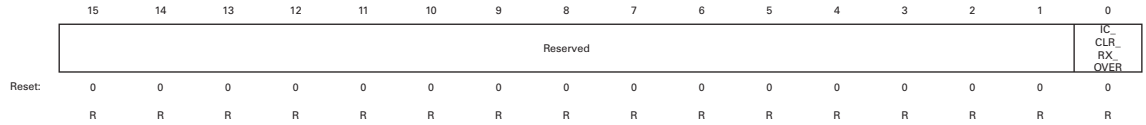
Address – 0xC400 0044



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

4.3.31 IC_CLR_RX_OVER REGISTER – CLEAR RX_OVER INTERRUPT

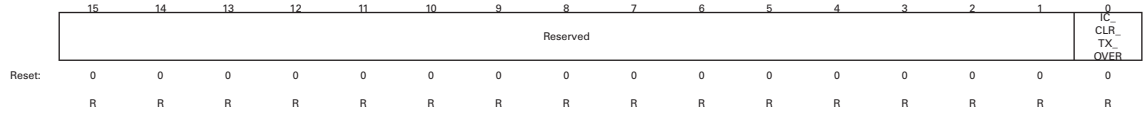
Address – 0xC400 0048



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

4.3.32 IC_CLR_TX_OVER REGISTER – CLEAR TX_OVER INTERRUPT

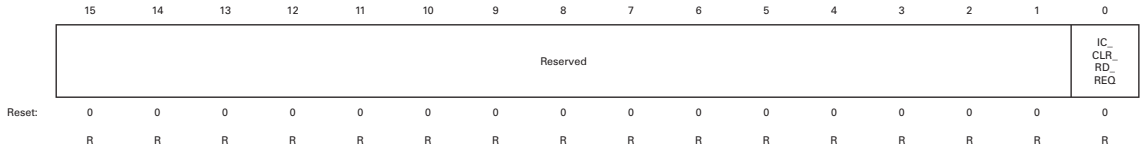
Address – 0xC400 004C



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

4.3.33 IC_CLR_RD_REQ REGISTER – CLEAR RD_REQ INTERRUPT

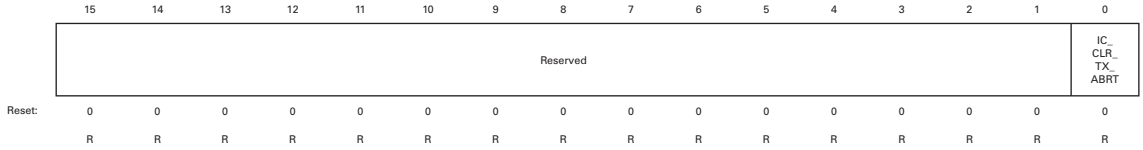
Address – 0xC400 0050



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

4.3.34 IC_CLR_TX_ABRT REGISTER – CLEAR TX_ABRT INTERRUPT

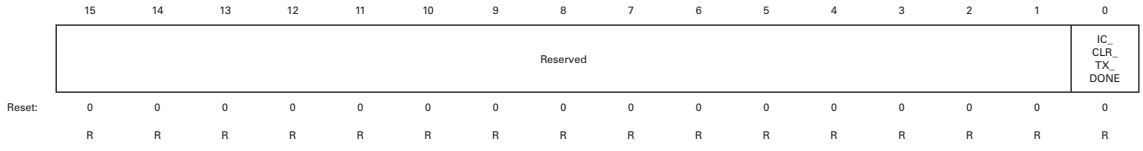
Address – 0xC400 0054



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

4.3.35 IC_CLR_RX_DONE REGISTER – CLEAR RX_DONE INTERRUPT

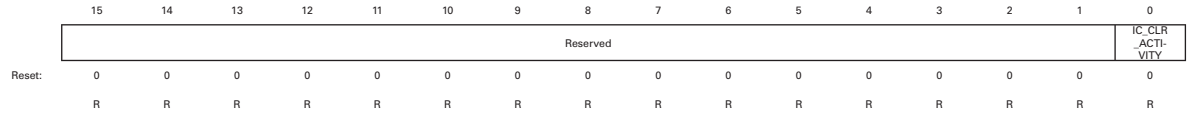
Address – 0xC400 0058



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

4.3.36 IC_CLR_ACTIVITY REGISTER – CLEAR ACTIVITY INTERRUPTS

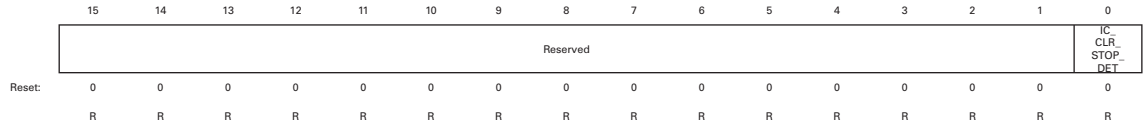
Address – 0xC400 005C



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

4.3.37 IC_CLR_STOP_DET REGISTER – CLEAR STOP_DET INTERRUPTS

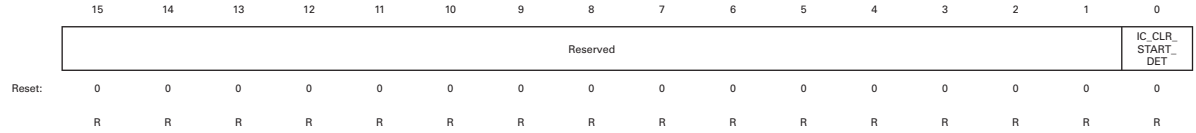
Address – 0xC400 0060



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

4.3.38 IC_CLR_START_DET REGISTER – CLEAR START_DET INTERRUPT

Address – 0xC400 0064



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

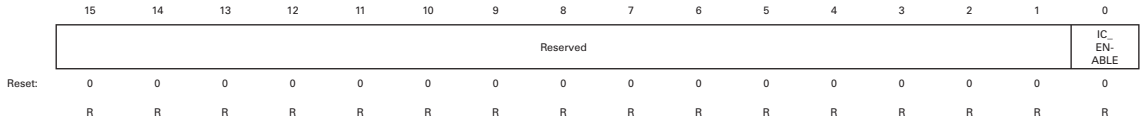
4.3.39 IC_CLR_GEN_CALL REGISTER – CLEAR GENERAL CALL INTERRUPT

Address – 0xC400 0068

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

4.3.40 IC_ENABLE REGISTER – I2C ENABLE

Address – 0xC400 006C



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. 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..

4.3.41 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.

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
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.**

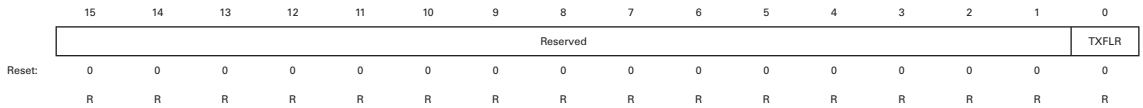
4.3.42 IC_TXFLR REGISTER – I2C TRANSMIT FIFO LEVEL REGISTER

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.

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

4.3.43 IC_RXFLR REGISTER – I2C RECEIVE FIFO LEVEL REGISTER

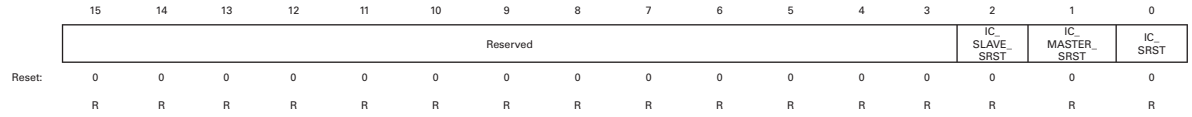
Address – 0xC400 0078
This register contains the number of valid data entries in the receive FIFO buffer. It is cleared when the I2C is disabled or whenever there is a transmit abort. It increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.



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

4.3.44 IC_SRESET REGISTER – I2C SOFT RESET REGISTER

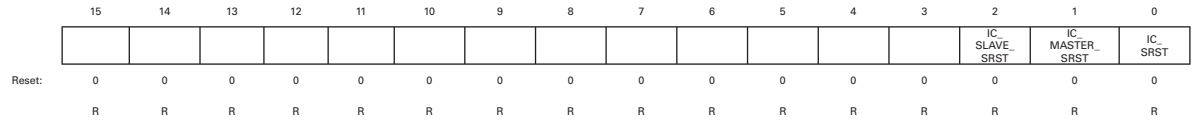
Address – 0xC400 007c
This register is used to issue a soft reset to the master and/or the slave state machines. Reading this register does not clear it; but is automatically cleared by hardware.



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
IC_SRST	0	0	W	Issues a soft reset to both the master and slave state machines. 1 = perform the reset

4.3.45 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.



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.
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 when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).
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.

4.4 UART

4.4.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
U0_CTS	W7 (shared)	I	6	Clear To Send UART Status; activelow
U0_DSR	Y7 (shared)	I	6	Data Set Ready UART Status; active-low
U0_DCD	V8 (shared)	I	6	Data Carrier Detect UART Status; active-low
U0_RI	W8 (shared)	I	6	Ring Indicator UART Status; activelow
U1_CTS	Y8 (shared)	I	6	Clear To Send UART Status; activelow
U1_DSR	U9 (shared)	I	6	Data Set Ready UART Status; active-low
U1_DCD	V9 (shared)	I	6	Data Carrier Detect UART Status; active-low
U1_RI	W9 (shared)	I	6	Ring Indicator UART Status; activelow
U0_DTS	Y11 (shared)	O	6	UART Control Data Terminal Ready; active-low
U0_RTS	W11 (shared)	O	6	UART Control Request To Send; active-low
U0_OUT1	V11 (shared)	O	6	UART Control Programmable output 1; active-low
U0_OUT2	U11 (shared)	O	6	UART Control Programmable output 2; active-low
U1_DTS	Y9 (shared)	O	6	UART Control Data Terminal Ready; active-low
U1_RTS	W10 (shared)	O	6	UART Control Request To Send; active-low
U1_OUT1	V10 (shared)	O	6	UART Control Programmable output 1; active-low
U1_OUT2	Y10 (shared)	O	6	UART Control Programmable output 2; active-low
uART0_tx	A5	O	4	Serial output; active-high
uART0_rx	D7	I	-	Serial input; active-high
uART1_tx	C6	O	4	Serial output; active-high
uART1_rx	B5	I	-	Serial input; active-high

The function of these pins is software configurable via the GPCSR module, specifically GPCSR_VIDEO_SELECT – 0xc700 0010. Refer to the GPCSR module documentation for more information.
To set the shared pins to function as UART pins, the register mentioned above should be set as shown below.
GPCSR_VIDEO_SELECT – 0xc700 0010 = 0x0002 aaa8
Note that this register only configures the UART output pins. For inputs, the pin will contain whatever signal has been connected to it.

4.4.2 FEATURES

The UART in the DICE II 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.

4.4.3 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 4.18.

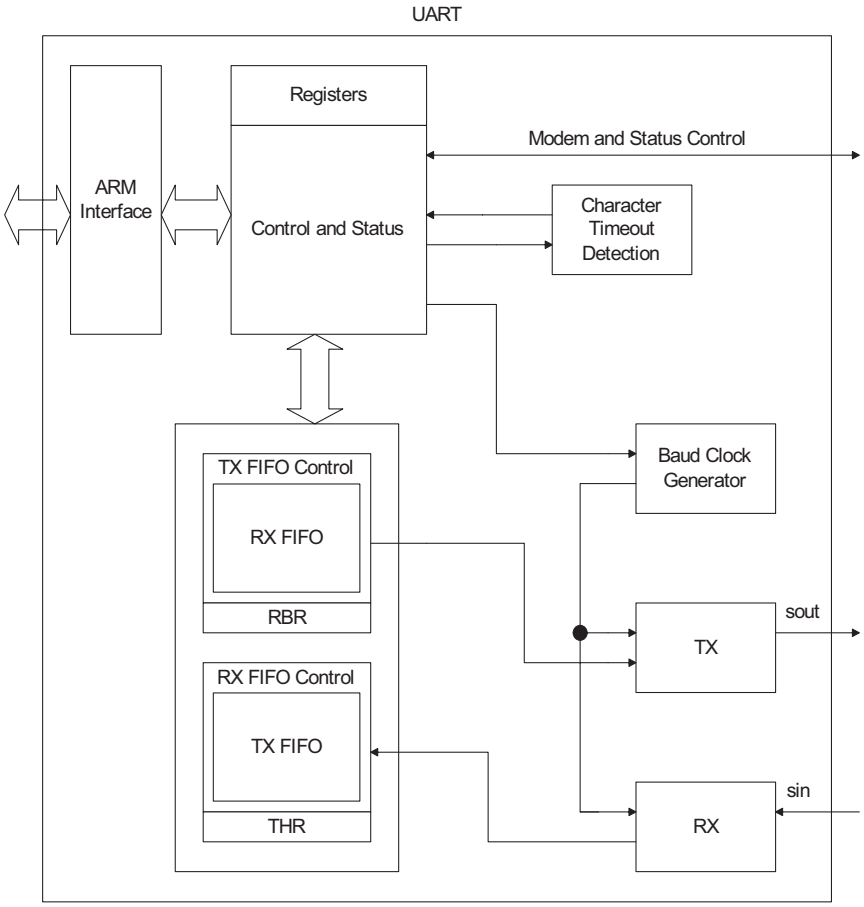


Figure 4.18: UART Generalized Functional Diagram

4.4.4 APB INTERFACE

Standard AMBA 2.0 compliant APB interface.

4.4.5 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.

4.4.6 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.

4.4.7 CHARACTER TIMEOUT DETECTION

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

4.4.8 BAUD CLOCK GENERATOR

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

4.4.9 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.

4.4.10 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.

4.4.11 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 4.19 shows the frame formats supported.

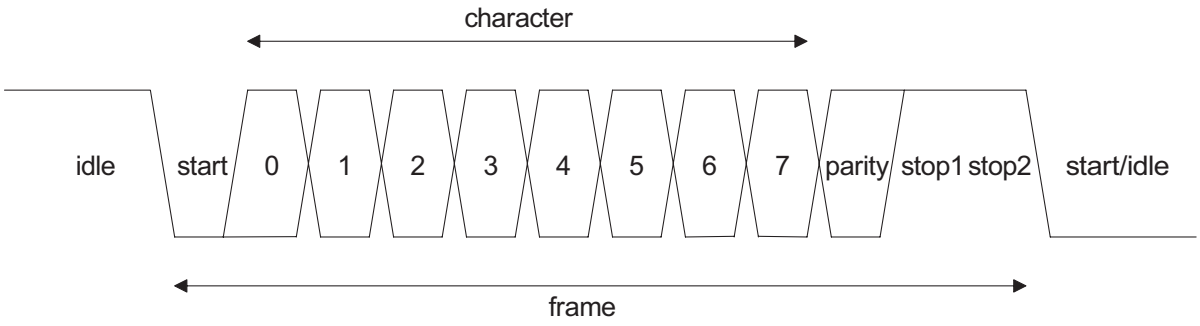


Figure 4.19: Serial Frame Format (using 8 data bit configuration)

4.4.12 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 4.23: UART Memory Map

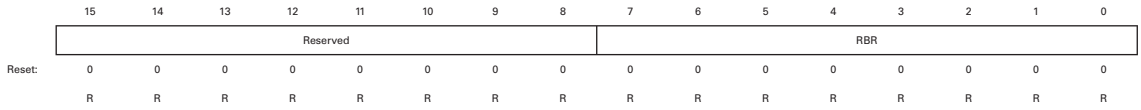
a Bit 7 of the Line Control Register (LCR) enables reading and writing of the Divisor Latch Registers (DLL, DLH).

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.

4.4.13 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.

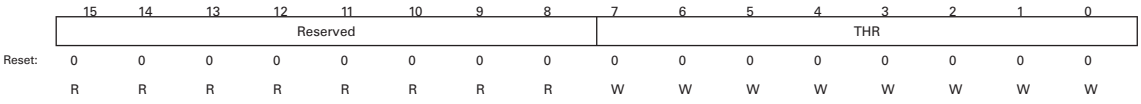


Name	Bit	Reset	Dir	Description
Receive Buffer Register (RBR)	7:0	0	R	Contains data character from serial input port.

4.4.14 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.



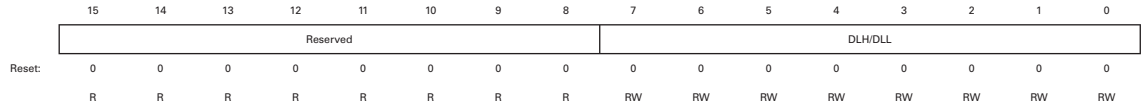
Name	Bit	Reset	Dir	Description
Transmit Holding Register (THR)	7:0	0	W	Contains data byte for serial transmission.

4.4.15 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)

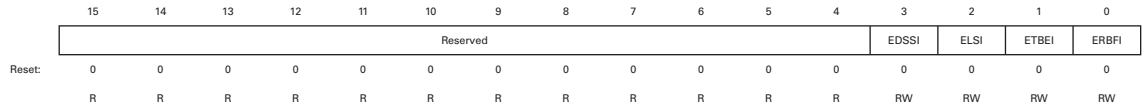


Name	Bit	Reset	Dir	Description
Divisor Latch High (High Byte)	7:0	0	RW	High byte Divisor Latch Register
Divisor Latch Low (Low Byte)	7:0	0	RW	Low byte Divisor Latch Register

4.4.16 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.

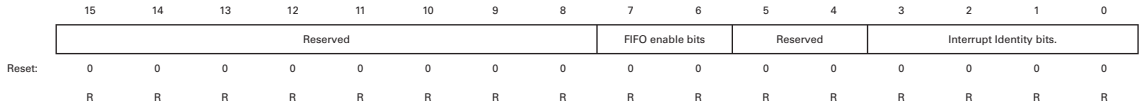


Name	Bit	Reset	Dir	Description
EDSSI	3	0	RW	Enable the Modem Status Interrupt
ELSI	2	0	RW	Enable the Receiver Line Status Interrupt
ETBEI	1	0	RW	Enable the Transmitter Holding Register Empty Interrupt
ERBFI	0	0	RW	Enable the Received Data Available Interrupt

4.4.17 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 4.24 below.



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 4.24 for details

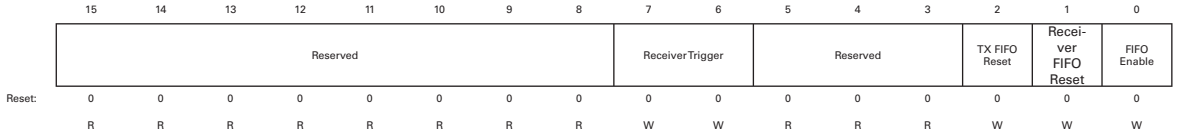
IIR bits				Interrupt Set and Reset Functions			
3	2	1	0	Priority	Type	Source	Reset and Control
0	0	0	1	-	-	-	-
0	1	1	0	first	Receiver line status	Overrun/parity/ framing errors or break interrupt	Reading the line status register
0	1	0	0	second	Received data available	Receiver data available or read 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 4.24: Interrupt Control Functions

4.4.18 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.

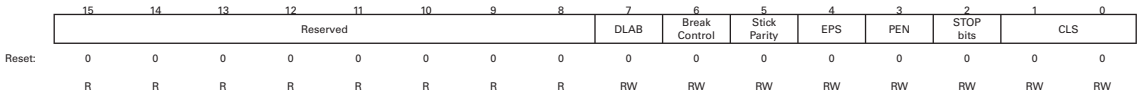


Name	Bit	Reset	Dir	Description
Receiver Trigger (RT)	7:6	0	W	Sets the trigger level in the receiver FIFO for the Enable Received Data Available Interrupt (ERBFI) 00 = 1 character in FIFO, 01 = 4 characters in FIFO 10 = 8 characters in FIFO, 11 = 14 characters in FIFO
Transmitter FIFO Reset	2	0	W	Resets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

4.4.19 LINE CONTROL REGISTER (LCR)

Address – 0xBD00 000C

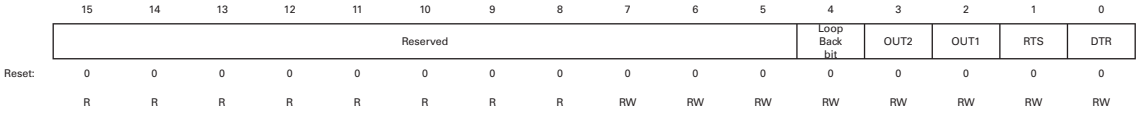
The Line Control Register controls the format of the data that is serially transmitted and received by the UART.



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.
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

4.4.20 MODEM CONTROL REGISTER (MCR)

Address – 0xBD00 0010

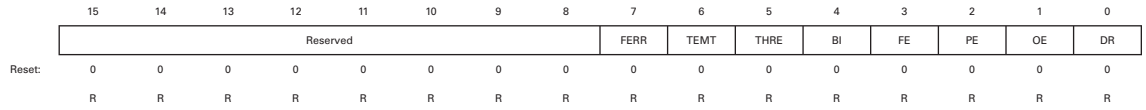


Name	Bit	Reset	Dir	Description
Loop Back bit	4	0	RW	This feature is used for diagnostic purposes. When set, data on the sout line is held HIGH, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem 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

4.4.21 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.

4.4.22 MODEM STATUS REGISTER (MSR)

Address – 0xBD00 0018

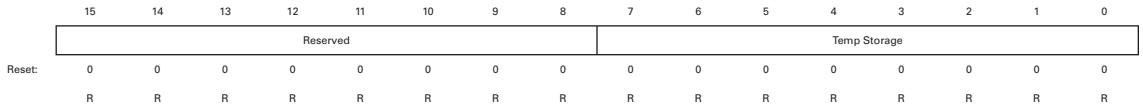
The Modem Status Register contains the current status of the modem control input lines and if they changed. DCTS (bit 0), DDSR (bit 1) and DDCD (bit 3) bits record whether the modem control lines (cts_n, dsr_n and dcd_n) have changed since the last time the CPU read the MSR. The CTS, DSR, RI and DCD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

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).
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).

4.4.23 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.



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

4.4.24 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 4.25 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 4.25: Determining Baud Rate

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.

4.5 GPIO

4.5.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
GPIO1	R3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO2	W1 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO3	V3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO4	A3 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO5	D5 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO6	C4 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO7	C20 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO8	E17 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO9	D18 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO10	K19 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO11	K18 (shared)	I/O (S)	8	General Purpose I/O (5V)
GPIO12	K17 (shared)	I/O (S)	8	General Purpose I/O (5V)
gpIO13	W11 (shared)	I/O (S)	6	General Purpose I/O (5V)
gpIO14	V11 (shared)	I/O (S)	6	General Purpose I/O (5V)
gpIO15	U11 (shared)	I/O (S)	6	General Purpose I/O (5V)
GPIO16	P3 (shared)	I/O (S)	6	General Purpose I/O (5V)

Table 4.26: 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 registers GPCSR_IO_SELECT0 – 0xc700 0004 and GPCSR_VIDEO_SELECT – 0xc700 0010. Refer to the GPCSR module documentation for more information.
To set the shared pins to function as GPIO pins, the registers mentioned above should be set as shown below.
GPCSR_IO_SELECT0 0xc700 0004 = 0x0004 9580
GPCSR_VIDEO_SELECT 0xc700 0010 = 0x0000 0000
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.

4.5.2 MODULE CONFIGURATION

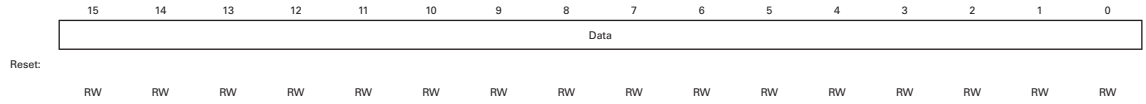
Address	Register
0xc300 0000	GPIO_DR
0xc300 0004	GPIO_DDR
0xc300 0030	GPIO_INTEN
0xc300 0034	GPIO_INTMSK
0xc300 0038	GPIO_INTSENSE
0xc300 003c	GPIO_INTPOL
0xc300 0040	GPIO_INTSTAT
0xc300 0044	GPIO_RAWINTSTAT
0xc300 0048	GPIO_DEBOUNCE
0xc300 004c	GPIO_EOI
0xc300 0050	GPIO_EXT
0x3c00 0060	GPIO_SYNC

Table 4.27: GPIO Memory Map

Note that all programmable registers are actually 32 bits wide. However, the upper 16 bits of all the registers are Reserved. Therefore, only the lower 16 bits of each register is shown in this document.

4.5.3 GPIO_DR DATA REGISTER

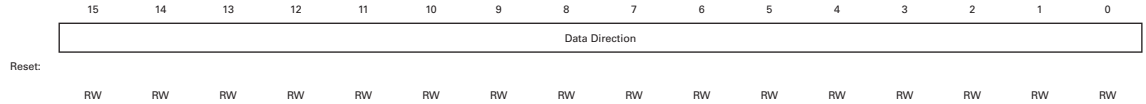
Address – 0Xc300 0000



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.

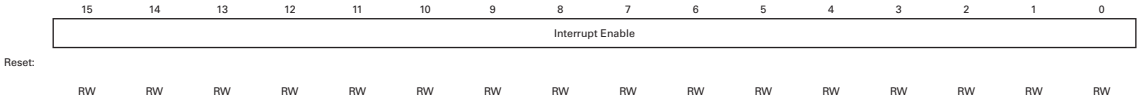
4.5.4 GPIO_DDR DATA DIRECTION REGISTER

Address – 0Xc300 0004



4.5.5 GPIO_INTEN INTERRUPT ENABLE REGISTER

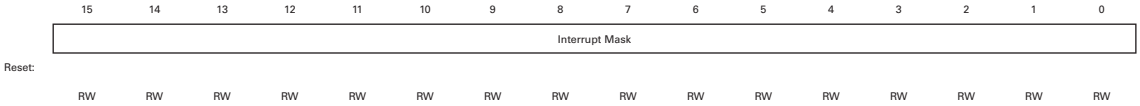
Address – 0Xc300 0030
This register is available only if GPIO port is configured to generate interrupts.



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

4.5.6 GPIO_INTMASK INTERRUPT MASK REGISTER

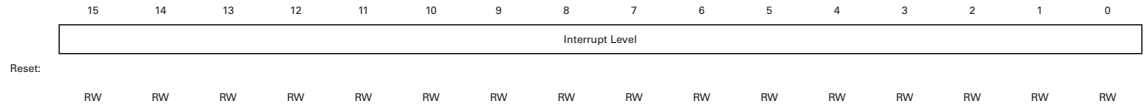
Address – 0Xc300 0034
This register is available only if GPIO port is configured to generate interrupts.



Name	Bit	Reset	Dir	Description
Interrupt Mask	15:0		RW	Controls whether an interrupt can create an interrupt for the interruptcontroller 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

4.5.7 GPIO_INTSENSE INTERRUPT LEVEL REGISTER

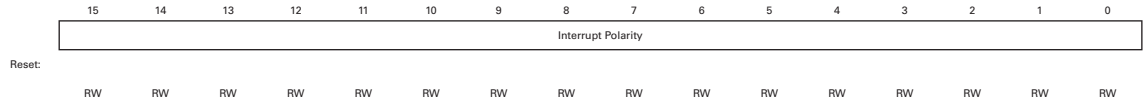
Address – 0Xc300 0038
This register is available only if GPIO port is configured to generate interrupts.



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

4.5.8 GPIO_INTPOL INTERRUPT POLARITY REGISTER

Address – 0Xc300 003c
This register is available only if GPIO port is configured to generate interrupts.

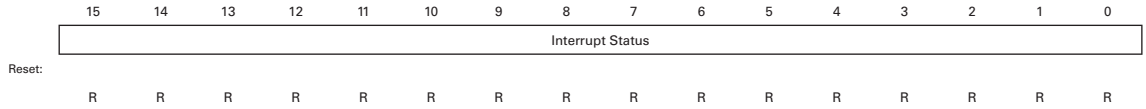


Name	Bit	Reset	Dir	Description
Interrupt Polarity	15:0		RW	Controls the polarity of edge or level sensitivity that can occur on input. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high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

4.5.9 GPIO_INTSTAT INTERRUPT STATUS REGISTER

Address – 0Xc300 0040

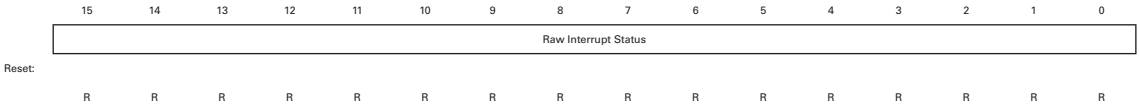
This register is available only if GPIO port is configured to generate interrupts.



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

4.5.10 GPIO_RAWINTSTAT RAW INTERRUPT STATUS (PREMASKING) REGISTER

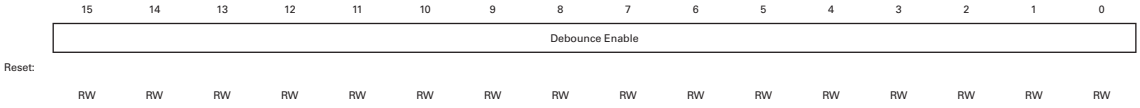
Address – 0Xc300 0044



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

4.5.11 GPIO_DEBOUNCE DEBOUNCE ENABLE REGISTER

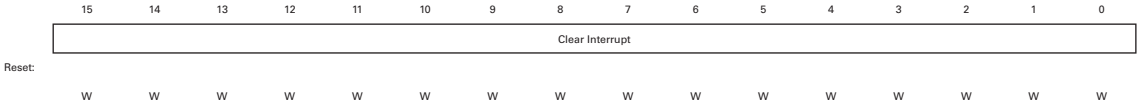
Address – 0Xc300 0048



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

4.5.12 GPIO_EOI CLEAR INTERRUPT REGISTER

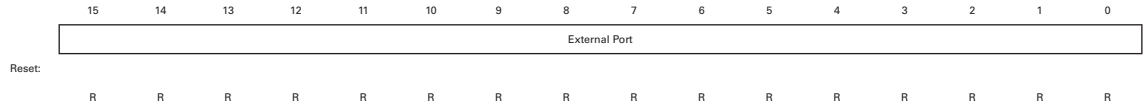
Address – 0Xc300 004c



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

4.5.13 GPIO_EXT EXTERNAL PORT REGISTER

Address – 0Xc300 0050



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.

4.5.14 GPIO_SYNC LEVEL SENSITIVE SYNCHRONIZATION ENABLE REGISTER

Address – 0Xc300 0060

Name	Bit	Reset	Dir	Description
Synchronization Level	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

4.6 1394 Link

4.6.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive(mA)	Description
sclk	W2	I (S)	-	49.152MHz PHY Clock
phd0	Y1	I/O (S)	8	PHY tristatable data line bit 0
phd1	W3	I/O (S)	8	PHY tristatable data line bit 1
phd2	Y2	I/O (S)	8	PHY tristatable data line bit 2
phd3	W4	I/O (S)	8	PHY tristatable data line bit 3
phd4	V4	I/O (S)	8	PHY tristatable data line bit 4
phd5	U5	I/O (S)	8	PHY tristatable data line bit 5
phd6	Y3	I/O (S)	8	PHY tristatable data line bit 6
phd7	Y4	I/O (S)	8	PHY tristatable data line bit 7
phct0	V5	I/O (S)	8	PHY tristatable control line bit 0
phct1	W5	I/O (S)	8	PHY tristatable control line bit 1
phdi	Y5	I (S)	-	A high indicates isolation barrier is not present (PU, 5V)
phlr	V6	O	8	Serial request output from S-LINK (Z)
phlp	U7	O	4	Link power status. Pulsing if isol. barrier present
phlo	W6	I (S)	-	Link on indication from PHY. Pulsing when asserted (PU, 5V)

Table 4.28: 1394 Link Signal Description

4.6.2 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
0x8200 0030	ASY_TX_STAT_REG_DP

Address	Register
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 4.29: 1394 LLC Memory Map

4.7 GRAY, Rotary Encoder Interface

This module can decode the input from 4 rotary encoders. Each interface consists of 2 pins, A and B.

4.7.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
EN1_A	P1 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN1_B	P2 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_A	R1 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN2_B	P3 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN3_A	W1(shared)	I (S)	6	Rotary Encoder Input (5V)
EN3_B	V3 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN4_A	D5 (shared)	I (S)	6	Rotary Encoder Input (5V)
EN4_B	C4 (shared)	I (S)	6	Rotary Encoder Input (5V)

Table 4.30: GRAY, Rotary Encoder Interface Signal Description

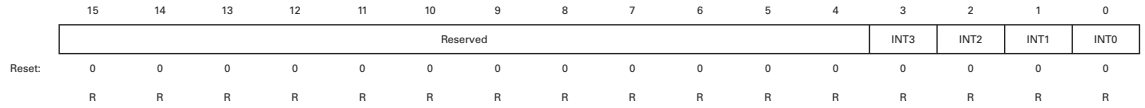
4.7.2 MODULE CONFIGURATION

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

Table 4.31: GRAY, Rotary Encoder Interface Memory Map

4.7.3 GRAY_STAT

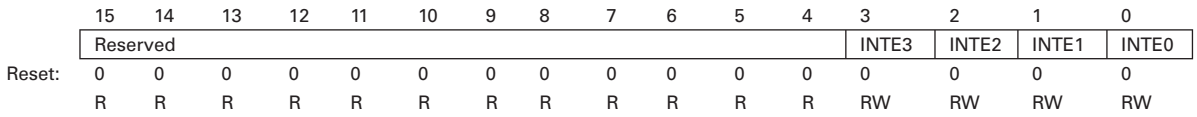
Address – 0Xc600 0000



Name	Bit	Reset	Dir	Description
INT3	3	0	R	This bit indicates the changed status of counter 3. This bit will be set whenever the counter is changed and cleared when the GRAY_CNT register is read.
INT2	2	0	R	This bit indicates the changed status of counter 2. This bit will be set whenever the counter is changed and cleared when the GRAY_CNT register is read.
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.

4.7.4 GRAY_CTRL

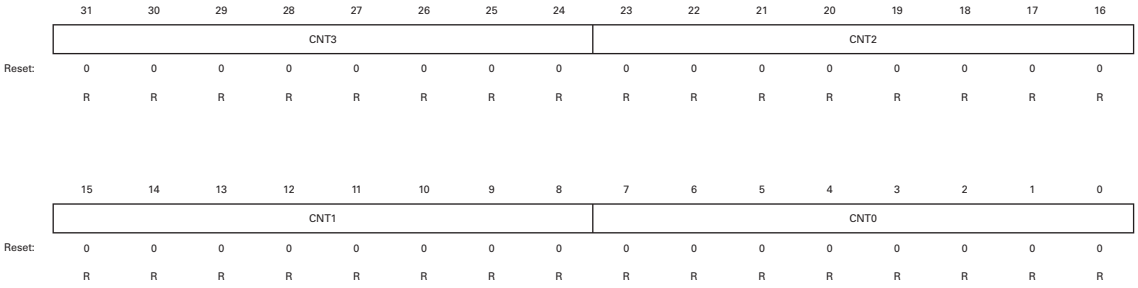
Address – 0Xc600 0004



Name	Bit	Reset	Dir	Description
INTE3	3	0	R	This bit enables the interrupt on change for counter 3.
INTE2	2	0	R	This bit enables the interrupt on change for counter 2.
INTE1	1	0	R	This bit enables the interrupt on change for counter 1.
INTE0	0	0	R	This bit enables the interrupt on change for counter 0.

4.7.5 GRAY_CNT

Address – 0Xc600 0008



Name	Bit	Reset	Dir	Description
CNT3	31:24	0	R	Count indicating amount of change since last read. The value range is -128 to 127 with saturation logic.
CNT2	23:16	0	R	Count indicating amount of change since last read. The value range is -128 to 127 with saturation logic.
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.

4.8 Interrupt Controller

4.8.1 FEATURES

AIC supports the following features:

- 32 IRQ normal interrupt sources
- 8 FIQ fast interrupt sources
- Vectored interrupts

- Software interrupts
- Priority filtering
- Masking

4.8.2 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 4.20 shows a block diagram of the AIC.

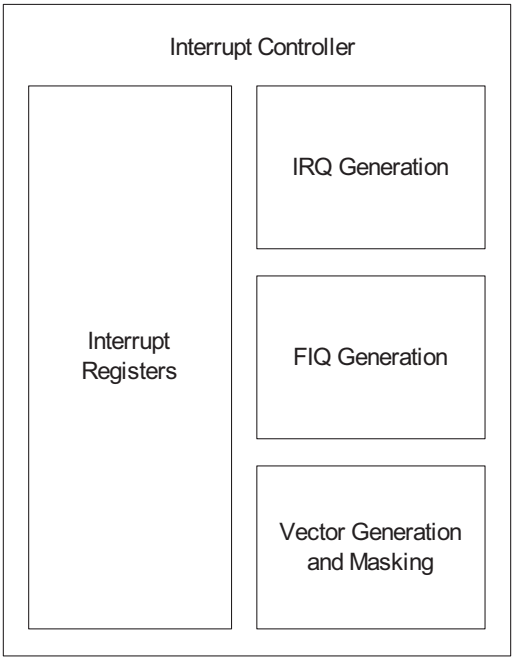


Figure 4.20: Block Diagram of AIC

4.8.3 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 4.21 and described in the following sections.

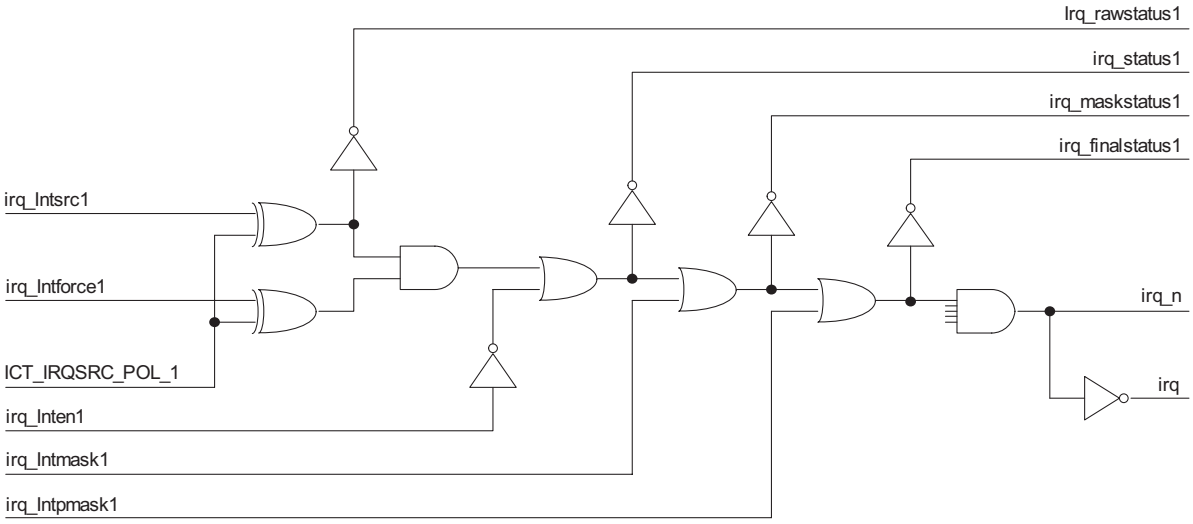


Figure 4.21: IRQ Internal Diagram (Interrupt 1)

4.8.4 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

4.8.5 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.

4.8.6 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.

4.8.7 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.

4.8.8 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.

4.8.9 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. See Figure 4.22 for further detail.

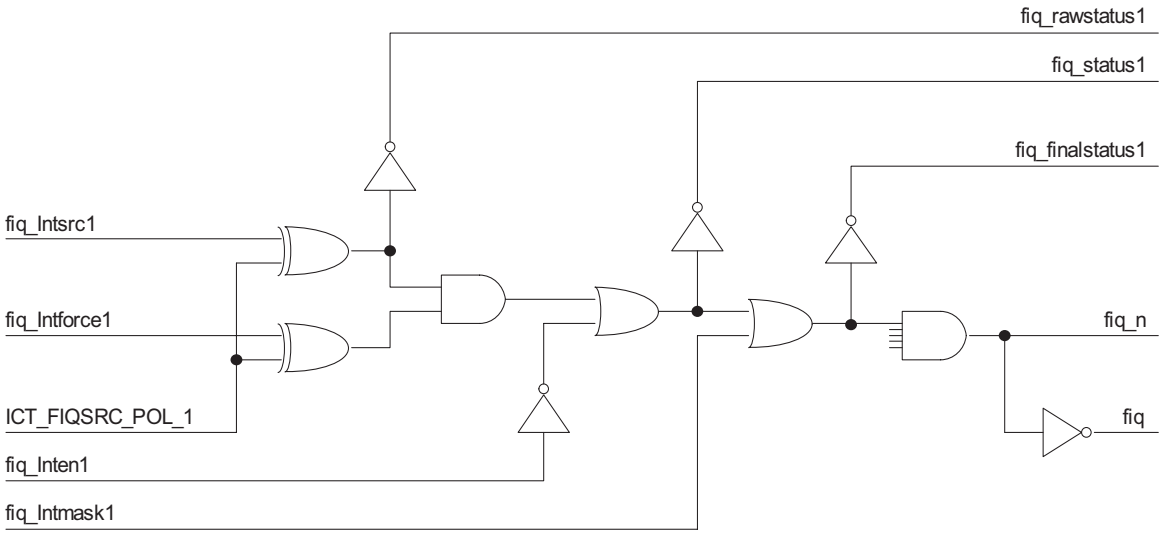


Figure 4.22: FIQ Internal Diagram (Interrupt 1)

4.8.10 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

4.8.11 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).

4.8.12 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.

4.8.13 MODULE CONFIGURATION

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
0xc100 00cc	INTCTRL_FIQ_RAW
0xc100 00d0	INTCTRL_FIQ_STAT
0xc100 00d4	INTCTRL_FIQ_FINALSTAT
0xc100 00d8	INTCTRL_SYSTEM_PRIORITY_LEVEL

Table 4.32: Interrupt Controller Memory Map

4.8.14 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.

4.8.15 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.

4.8.16 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.

4.8.17 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.

4.8.18 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.

4.8.19 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.

4.8.20 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.

4.8.21 INTCTRL_INTVECTOR

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

4.8.22 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.

4.8.23 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.

4.8.24 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.

4.8.25 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.

4.8.26 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.

4.8.27 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.

4.8.28 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.

4.8.29 NTCTRL_SYSTEM_PRIORITY_LEVEL

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

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 4.33 Interrupt Priority Levels

4.9 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 functionallity 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.

4.9.1 FUNCTIONAL DESCRIPTION

The watchdog internal modules are illustrated in **Error! Reference source not found.**. 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.

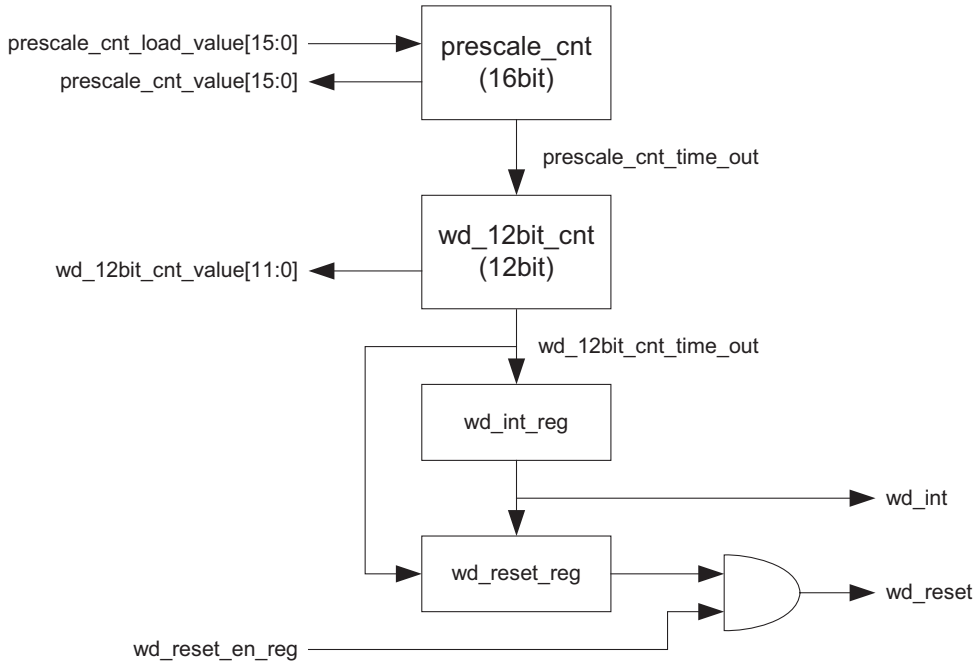


Figure 4.23: Basic illustration of the watchdog

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 Figure 4.24.

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, which sets **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.

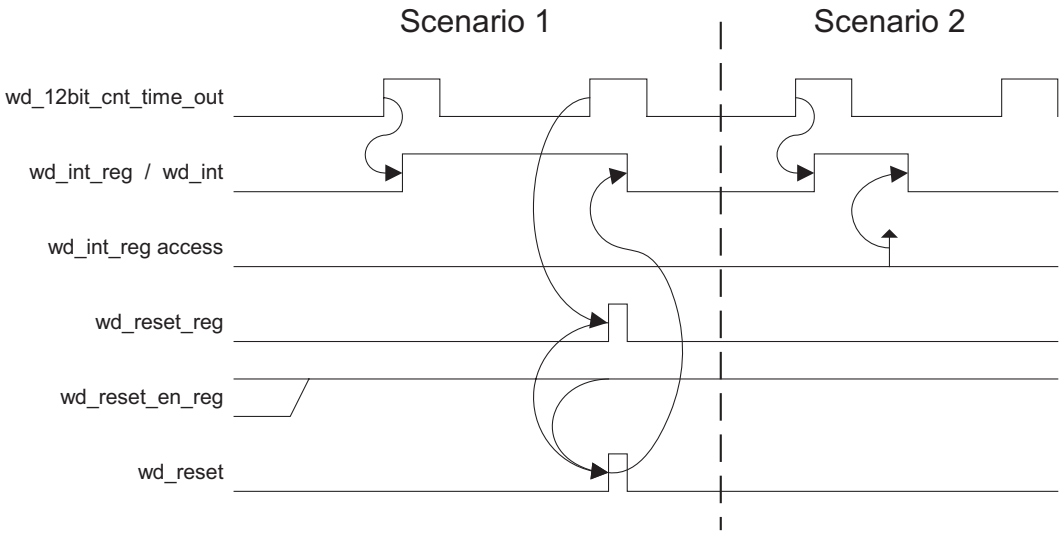


Figure 4.24 Wave form illustrating update of wd_int and wd_reset.
The wd_int_reg_access is a virtual signal illustrating that the wd_int_reg gets cleared.

4.9.2 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 4.34: Watch Dog Memory Map

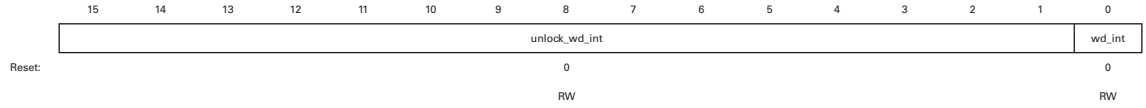
4.9.3 WD_RESET_EN

0xbf00 0000

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.

4.9.4 WD_INT

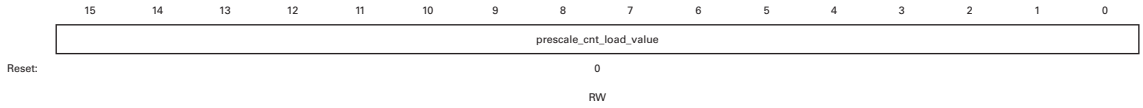
0xbf00 0004



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.

4.9.5 WD_PRESCALE_LOAD

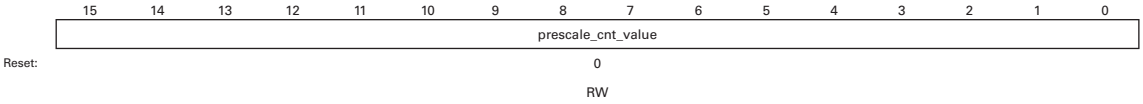
0xbf00 0008



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

4.9.6 WD_PRESCALE_CNT

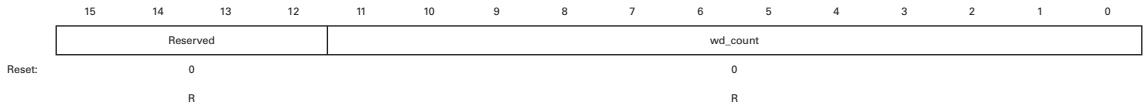
0xbf00 000c



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

4.9.7 WD_COUNT

0xbf00 0010



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

4.10 Dual Timer

4.10.1 INTRODUCTION

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

4.10.2 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

4.10.3 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 II.

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 II Users Guide.

4.10.4 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.

4.10.5 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 4.34 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:

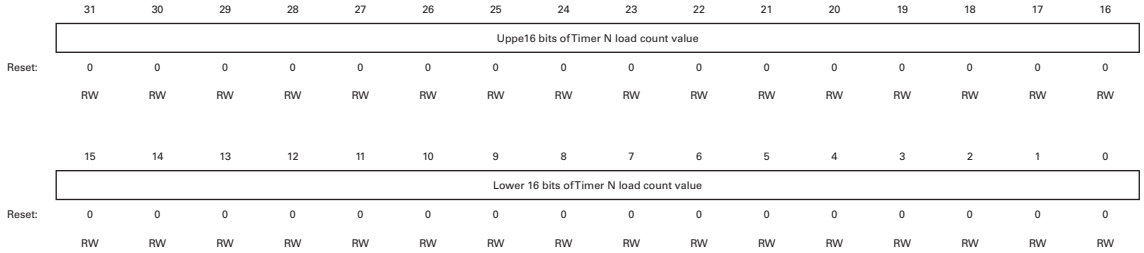
Address Range	Function
0xc200 0000 to 0xc200 0010	Timer 1 Registers
0xc200 0014 to 0xc200 0024	Timer 2 Registers
0xc200 00a0 to 0xc200 00a4	Timer System Registers

Table 4.35: Timer Memory Map

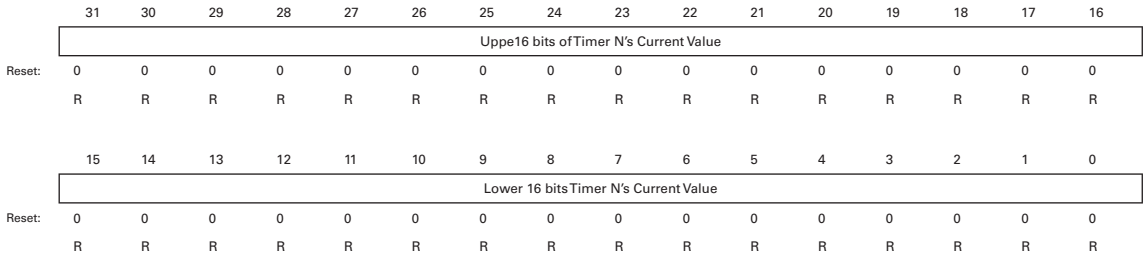
4.10.6 TIMER REGISTERS

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.

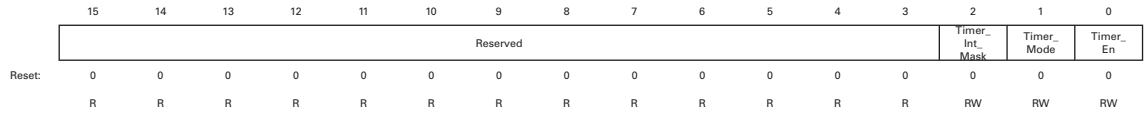
4.10.6.1 TIMERNLOADCOUNT



4.10.6.2 TIMERNCURRENTVALUE

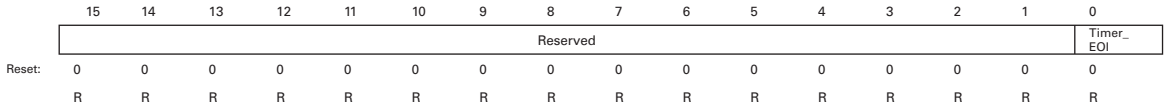


4.10.6.3 TIMERNCONTROL



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

4.10.6.4 TIMER_N_EOI



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

4.10.6.5 TIMERNINTSTATUS

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.

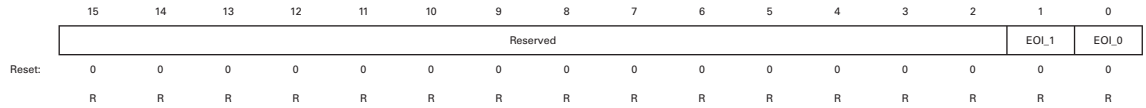
4.10.7 TIMER SYSTEM REGISTERS

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.

4.10.7.1 TIMERSINTSTATUS

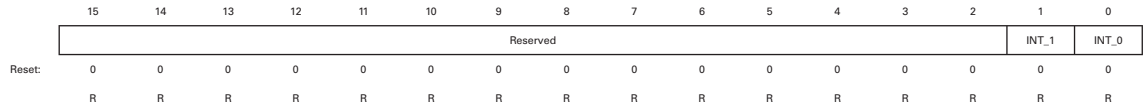
Field	Function	Description
Timer0_Int_Status RO	Timer 0's interrupt status after masking	This bit contains the interrupt status for Timer 0 after masking. Reading from this register does not clear the interrupt from Timer 0.
Timer1_Int_Status RO	Timer 1's interrupt status after masking	This bit contains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

4.10.7.2 TIMERSEOI



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

4.10.7.3 TIMERSRAWINTSTATUS



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.

4.10.8 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.

Chapter 5 DICE

5.1 ROUTER

5.1.1 BACKGROUND

The DICE II supports a tremendous amount of I/O – a total of 144 audio input channels and 112 audio output channels, over its various I2S, ADAT, TDIF, DSAI, ARM, and IEEE 1394 AVS ports. Moreover, DICE II supports two independent clock domains, each of which can be synchronized to any of the audio inputs. In order to manage all of these inputs and outputs, DICE II contains two Cross Bar Routers – one for each clock domain. These routers allow any audio input to be connected to any audio output in each of the two clock domains.

5.1.2 THEORY OF OPERATION

5.1.2.1 REFERENCING RECEIVE AND TRANSMIT PORTS

The various DICE II receive and transmit ports are referenced using two fields – BLOCK ID and CHANNEL NUMBER. The BLOCK ID is simply the name of the port being referenced – I2S, ADAT, DSAI, etc. The CHANNEL NUMBER is the individual audio channel (or "sequence" in IEC 61883-6 parlance) being referenced. Thus, connections in the DICE II router are specified using the following convention:

[SOURCE BLOCK ID:CHAN] -connected to- [DESTINATION BLOCK ID:CHAN]

5.1.2.2 ROUTER TABLES

A look-up table (Router Table) specifies all of the router connections in a configuration. Since DICE II provides an independent router for each of its two clock domains, there are actually two Router Tables. Each Router Table has up to 256 rows and 4 columns. A row in the table represents one of the possible 256 router connections in that clock domain. The four columns specify the SOURCE BLOCK:CHAN and DESTINATION BLOCK:CHAN fields for that row's router connection. Since there are "only" 144 possible input channels, and 112 output channels, across all of the various I/O and AVS ports, the maximum number of allowed router connections (256) exceeds the number of possible physical connections in and out of the chip. Therefore, the router is more than capable of supporting every conceivable permutation of input/output connections.

5.1.2.3 ROUTER OPERATION

During each sample period, audio samples are moved from Receive (Source) ports to Transmit (Destination) ports as specified by the connections (rows) in the two Router Tables. A simplified block diagram of this process is shown in Figure 1. Functions related to the reading of input samples are colored in yellow. Functions related to the writing of output samples are colored in blue. At the beginning of a new sample period, each router reads the first row from its Router Table. The SOURCE BLOCK ID and CHAN fields are parsed and used to read an audio sample from an input (Receive) port. The DESTINATION BLOCK ID and CHAN fields are used to determine where that sample should be written (one of the Transmit ports). Then, the next row is read from each Router Table and a sample from the specified Receive port is transferred to the specified Transmit port. The entire Router Table is iterated within one sample period. Then, when the next sample period begins, the whole process repeats, starting at the first Router Table row again. The two independent routers perform these operations simultaneously, asynchronous to each other. Output samples are clocked through double-buffered latches to ensure that they are transmitted with zero phase error with respect to each other.

The ROUTER CONTROL register contains an error bit indicating if the router was not able to complete the routing entries within one sample period. This bit was for debugging purposes only and can be ignored by the application developer because the router will always be able to handle the maximum number of connections.

5.1.3 ROUTER BLOCK DIAGRAM

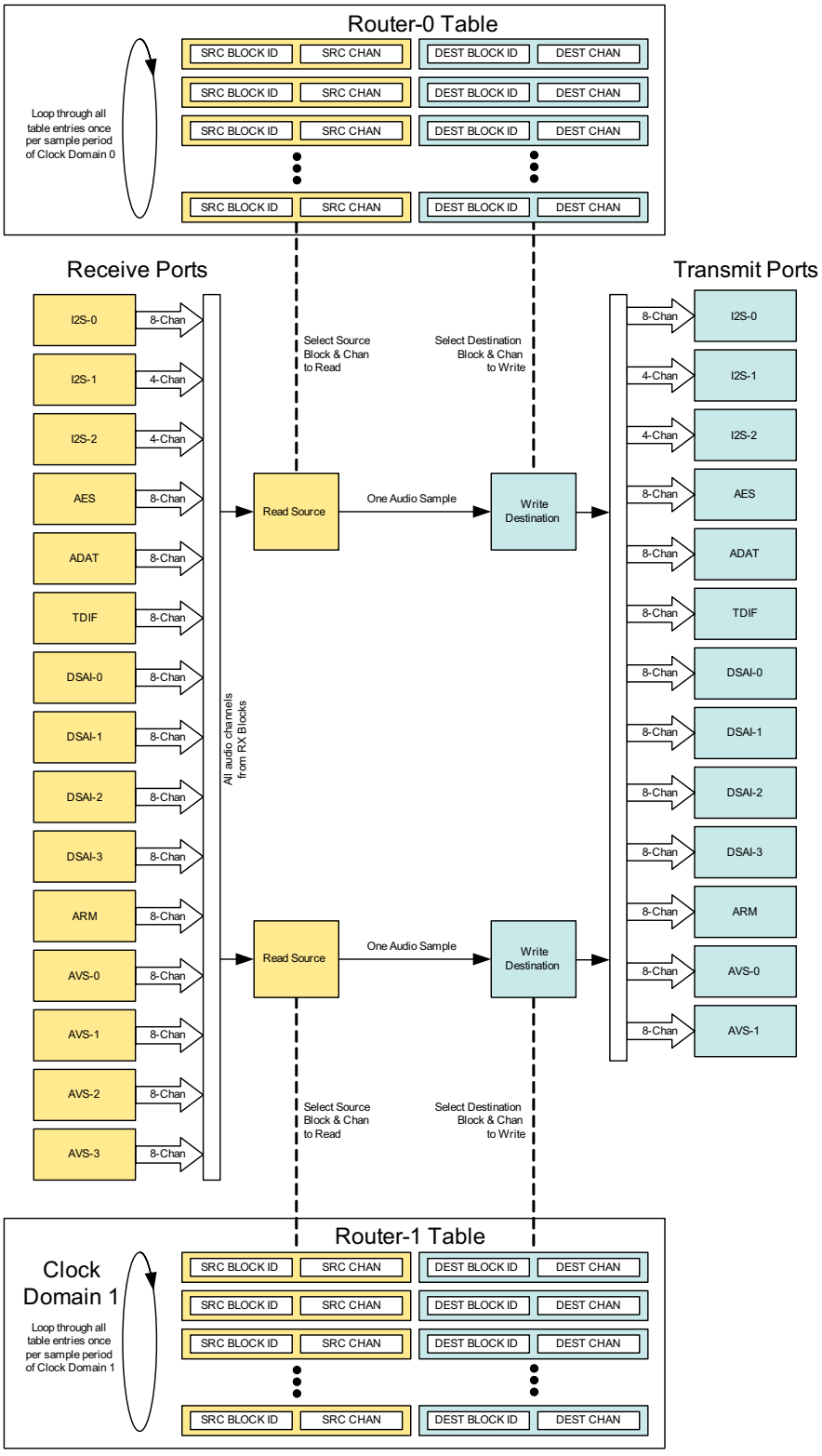


Figure 5.1: Block Diagram of DICE II Routers

5.1.4 ROUTER CONFIGURATION REGISTERS

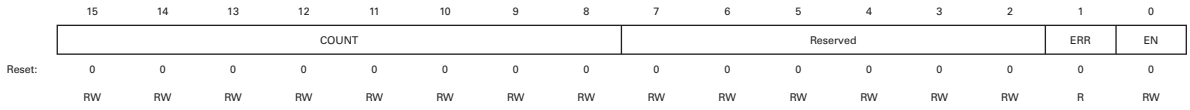
As described in the previous section, each of the two routers is configured using a Router Table. The registers involved in configuring the SOURCE BLOCK ID, SOURCE CHAN, DESTINATION BLOCK ID, and DESTINATION CHAN fields in the Router Tables are shown in the following tables.

Address	Register
0xce00 0000	ROUTER0_CTRL
0xce00 0400	ROUTER0_ENTRY0
0xce00 0404	ROUTER0_ENTRY1
:	:
0xce00 07fc	ROUTER0_ENTRY255
0xce00 0800	ROUTER1_CTRL
0xce00 0c00	ROUTER1_ENTRY0
0xce00 0c04	ROUTER1_ENTRY1
:	:
0xce00 0ffc	ROUTER1_ENTRY255

Table 5.1: Router Memory Map

5.1.5 ROUTER0_CTRL

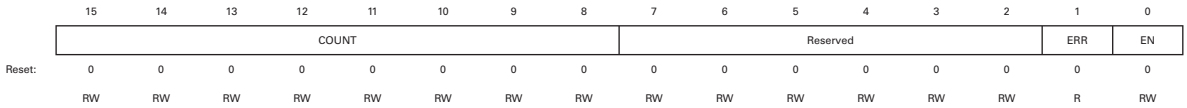
0xce00 0000



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.
ERR	7	0	R	This read-only bit indicates that the router was not able to complete the routing within on cycle. It can be ignored since this error condition can never occur.
EN	0	0	RW	This bit enables this router.

5.1.6 ROUTER1_CTRL

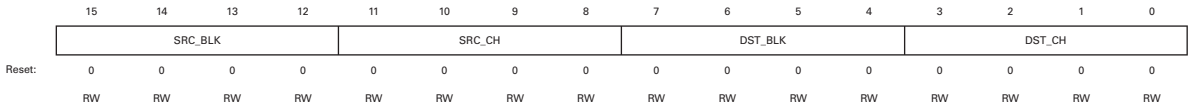
0xce00 0800



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.
ERR	7	0	R	This read-only bit indicates that the router was not able to complete the routing within on cycle. It can be ignored since this error condition can never occur.
EN	0	0	RW	This bit enables this router.

5.1.7 ROUTER0_ENTRYM

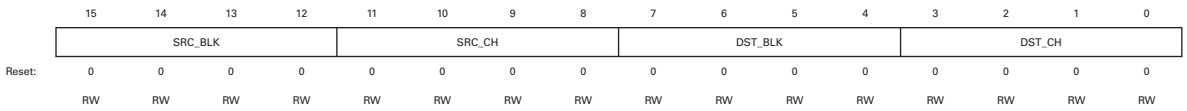
0xce00 0400 - 0xce00 07fc



Name	Bit	Reset	Dir	Description
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.

5.1.8 ROUTER1_ENTRYM

0xce00 0c00 - 0xce00 0ffc



Name	Bit	Reset	Dir	Description
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.

5.1.9 SOURCE BLOCK ID'S

ID	Block	Channels
0	AES	8
1	ADAT	8/4
2	TDIF	8
3	DSAI-0	8
4	DSAI-1	8
5	DSAI-2	8
6	DSAI-3	8
7	I²S-0	8
8	I²S-1	4
9	I²S-2	4
10	ARM	8
11	AVS-1	16
12	AVS-2	16
13	AVS-3	16
14	AVS-4	16
15	Reserved	N/A

Table 5.2 Source Block ID Codes

5.1.10 DESTINATION BLOCK ID'S

ID	Block	Channels
0	AES	8
1	ADAT	8/4
2	TDIF	8
3	DSAI-0	8
4	DSAI-1	8
5	DSAI-2	8
6	DSAI-3	8
7	I²S-0	8
8	I²S-1	4
9	I²S-2	4
10	ARM	8
11	AVS-1	16
12	AVS-2	16
13	Reserved	N/A
14	Reserved	N/A
15	Reserved	N/A

Table 5.3 Destination Block ID Codes

5.1.11 AVS TRANSMIT EXCEPTION

In general, the DICE II Router allows any audio input channel to be connected to any audio output channel. There is one exception to this rule – routing of audio channels to the two AVS Transmitters. Due to a minor bug in the router logic, as the router processes a table entry with an AVS transmitter as destination, it overwrites the AVS Transmitter channel corresponding to the SOURCE CHAN with random data before writing the correct data to the specified AVS DESTINATION CHAN. The bug does not occur when the SOURCE CHAN and DESTINATION CHAN are equal because the AVS SOURCE/DESTINATION CHAN will be written with correct data immediately after the random data, and the double-buffered transmit ports never output the temporary random value. In addition, when digital audio is routed in blocks the bug does not occur if channel position within the block does not change.

Example A:

AES Ch0-7 -> AVS-1 Ch 0-7.

In this case, each of the source and destination channels is equal, so the bug does not occur (the random data written to each AVS-1 Ch is immediately overwritten by "good" data).

Example B:

AES Ch4 -> AVS-1 Ch 2

This will correctly transfer the data from AES Ch4 to AVS-1 Ch2, but it will also write random data into AVS-1 Ch4. This can be corrected if AVS-1 Ch4 is written with correct data in a later routing entry
(e.g. ADAT Ch 4 -> AVS-1 Ch 4). But if AVS-1 Ch 4 is not updated with valid data after the AES Ch4 -> AVS-1 Ch2 entry, it will remain updated with this random data value.

The AVS receive and transmit blocks each support 16 audio channels, whereas all of the I/O ports support 4 or 8 channels. This would suggest that it is impossible to route audio to the upper 8 channels of the AVS Transmitters since the SOURCE CHAN and DESTINATION CHAN must be equal. However, the I/O receiver blocks mirror their data values around 8, for a total of 16 "base" and "mirrored" source channels. The table below shows how the receiver channels are mirrored:

Receive Ports	Physical Channels	Mirrored to Channels
I2S-0	0-7	8-15
I2S-1	0-3	8-11
I2S-2	0-3	8-11
AES	0-7	8-15
ADAT	0-7 (0-3 in SMUX mode)	8-15 (8-11 in SMUX mode)
TDIF	0-7	8-15
ARM	0-7	8-15
DSAI-0	0-7	8-15
DSAI-1	0-7	8-15
DSAI-2	0-7	8-15
DSAI-3	0-7	8-15
AVS-1	0-15	Na
AVS-2	0-15	Na
AVS-3	0-15	Na
AVS-4	0-15	Na

The router entries can use the "mirrored" values instead of the "base" values to reference sources. Let's rework the above examples.

Example A':

AES Ch 8-15 -> AVS-1 Ch 8-15.

Due to the mirroring of AES channels, AES Ch 8-15 is the same as AES Ch 0-7. The 8 AES audio channels will be properly routed to the upper 8 channels of the AVS-1 Transmitter because the source and channel numbers are identical.

Example B':

AES Ch12 -> AVS-1 Ch 2

Due to the mirroring of AES channels, AES Ch12 is the same as AES Ch4. The router bug will cause corruption of AVS-1 Ch 12, before AVS-1 Ch2 is written with proper data. AVS-1 Ch4 will remain "untouched" by the bug because it is not referenced. AVS-1 Ch 12 can be written with correct data in a later router entry, or left "corrupted" if the application does not require 12 channels of streaming via AVS-1.

The firmware implements a workaround by placing limited restrictions on the routing. Using mirrored source channels, the lower eight channels of any AVS transmitter can be routed freely and the upper eight are restricted to a continuous source block.

5.1.12 WORK-AROUND USING UNUSED TRANSMIT/RECEIVE I/O PORTS

If the above method of routing audio "1:1" to the AVS Transmit ports is too restrictive, a different method can be employed using unused I/O Transmit/Receive ports. Since full cross bar routing is allowed to all of the other (non-AVS) Transmit ports, an unused Transmit port can be connected to an unused Receive port (e.g. DSAI-3 Tx D1 to DSAI-2 RX D1). Audio sources can be mapped arbitrarily to this I/O port, fed back into the Receive port via the PCB, and then mapped 1:1 to the desired AVS Transmit channel. The below example illustrates this work around:

Desired routing:

Receive (Input) Channel	Routed to	AVS-0 Transmit Channel
AES CHAN 0	->	AVS-0 CHAN 6
AES CHAN 1	->	AVS-0 CHAN 4
AES CHAN 2	->	AVS-0 CHAN 3
AES CHAN 3	->	AVS-0 CHAN 2
AES CHAN 4	->	AVS-0 CHAN 7
AES CHAN 5	->	AVS-0 CHAN 1
AES CHAN 6	->	AVS-0 CHAN 5
AES CHAN 7	->	AVS-0 CHAN 0

We can use an unused DSAI port (say, DSAI-3) to perform the routing:

Receive Channel	Routed to	Transmit Channel
AES CHAN 0	->	DSAI-3 CHAN 6
AES CHAN 1	->	DSAI-3 CHAN 4
AES CHAN 2	->	DSAI-3 CHAN 3
AES CHAN 3	->	DSAI-3 CHAN 2
AES CHAN 4	->	DSAI-3 CHAN 7
AES CHAN 5	->	DSAI-3 CHAN 1
AES CHAN 6	->	DSAI-3 CHAN 5
AES CHAN 7	->	DSAI CHAN 0
DSAI-3 CHAN 0	->	AVS-0 CHAN 0
DSAI-3 CHAN 1	->	AVS-0 CHAN 1
DSAI-3 CHAN 2	->	AVS-0 CHAN 2
DSAI-3 CHAN 3	->	AVS-0 CHAN 3
DSAI-3 CHAN 4	->	AVS-0 CHAN 4
DSAI-3 CHAN 5	->	AVS-0 CHAN 5
DSAI-3 CHAN 6	->	AVS-0 CHAN 6
DSAI-3 CHAN 7	->	AVS-0 CHAN 7

5.1.13 ROUTING AT 192KHZ

When the system is running at "quad" sample rates (176.4KHz and 192KHz), two neighbor channels inside the DICE II are used for each channel. This effectively halves the number of channels on most interfaces at the quad rates (with the exception of I2S-1 and I2S-2). The router must be configured to connect each pair of "neighbor" channels from source to destination, to properly route the complete data stream. The firmware DAL API (see below) hides this complexity and takes care of routing two channels for each channel specified in the

API calls.

5.1.14 DICE FIRMWARE ABSTRACTION LAYER (DAL) APIS

The firmware DAL API hides the nuances of the router, as shown below:
The destination and source are bit fields containing the module id and the channel id, plus a type. The type can be a "1-Block", a "2-Block", a "4-Block" or an "8-Block". The type has two functions. First, it provides an easy way to route a block of channels in one call and second it provides a way to hide the complexity of the AVS transmitter exceptions. The only restriction on the routing call is that the specified source type must be compatible with the basic destination type. In this context, "compatible" means "greater or equal to". For example, a 1-Block can be connected to any type because every type is greater or equal to a 1-Block. A 4-Block can be connected to another 4-Block or an 8-Block, but not to a 1-Block or a 2-Block.

Further explanation is found in the file: dalRouting.h. This file also contains defines for sources and destinations. In general, all destination blocks except the AVS transmitter have 1-block as the basic type for all the channels. The AVS has eight 1-Blocks and one 8-Block for the upper channels. This means that the lower eight channels can be routed freely and the upper eight must be matched with an 8-block.

Examples:

Route all ADAT channels to all AES channels:

```
dalSetRoute (0, TX_AES_CH0_7, RX_ADAT_CH0_7);
```

Route 2 ADAT channels and 2 AES channels to AVS1:

```
dalSetRoute (0, TX_AVS1_CH0, RX_ADAT_CH3);
dalSetRoute (0, TX_AVS1_CH1, RX_ADAT_CH1);
dalSetRoute (0, TX_AVS1_CH2, RX_AES_CH7);
dalSetRoute (0, TX_AVS1_CH3, RX_AES_CH1);
```

Route AES Ch 0 and 1 to all 4 ADAT pairs (same source to several destinations):

```
dalSetRoute (0, TX_ADAT_CH0, RX_AES_CH0);
dalSetRoute (0, TX_ADAT_CH1, RX_AES_CH1);
dalSetRoute (0, TX_ADAT_CH2, RX_AES_CH0);
dalSetRoute (0, TX_ADAT_CH3, RX_AES_CH1);
dalSetRoute (0, TX_ADAT_CH4, RX_AES_CH0);
dalSetRoute (0, TX_ADAT_CH5, RX_AES_CH1);
dalSetRoute (0, TX_ADAT_CH6, RX_AES_CH0);
dalSetRoute (0, TX_ADAT_CH7, RX_AES_CH1);
```

Route all AES channels to the upper 8 channels of AVS1:

dalSetRoute (0,TX_AVS1_CH8_15, RX_AES_CH0_7);

This is not legal for the upper AVS channels:

dalSetRoute (0,TX_AVS1_CH10, RX_ADAT_CH3);

This will result in an error return from the call.

For a more precise description please refer to dalRouting.h and the Firmware SDK documentation.

5.1.15 MUTING AN OUTPUT

When the DAL is created all the destinations belonging to that interface are set to listen to the muted source. In order to mute an output the muted source can be selected in the router call at any time.

5.2 CLOCK CONTROLLER

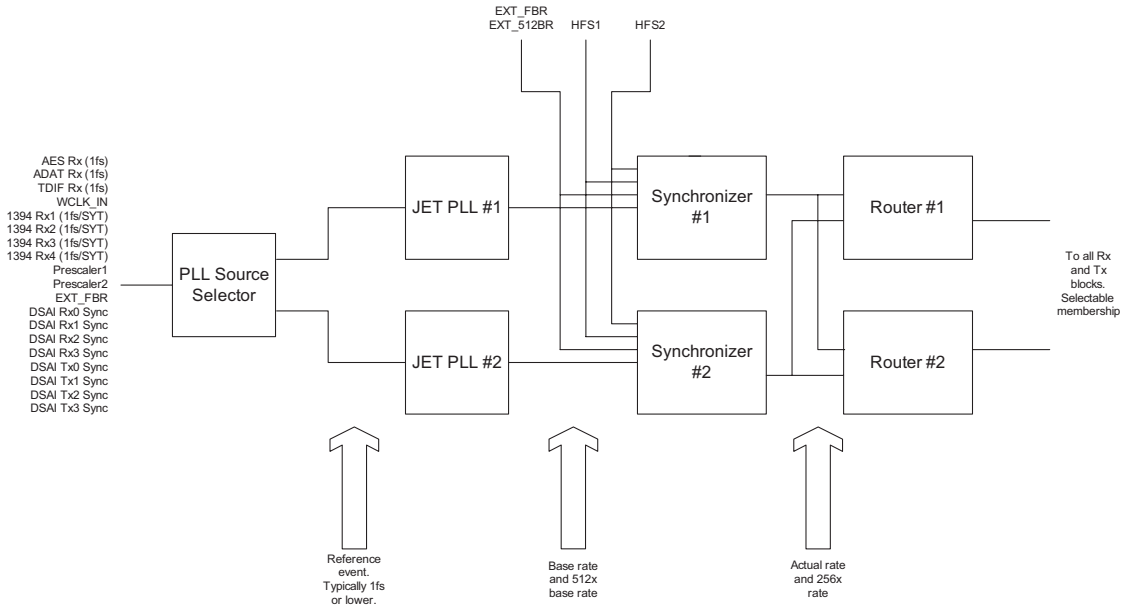


Figure 5.2: Clock Controller Block Diagram

The clock controller contains the logic to handle selection of clock sources, clock domain memberships, blocksync 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 DICE II.

The clock system consists of three types of blocks. The DICEII contains two instances of each block. The three types are:

- Router
- Synchronizer
- JET™ PLL

Each router gets its clocks from a synchronizer. To operate correctly, each router requires a 1fs and a 256fs clock.

Each synchronizer takes as input a base rate clock and a 512 x base rate clock, generates a pair of base rate and 256 x base rate clocks, and a pair of base rate x 2 and 256 x base rate x 2 clocks which each router can select from. In most cases the synchronizer will get its input from one of the JET™ PLL's, but in certain cases the synchronizer will slave to other sources such as the external slave interface or the fixed crystal inputs for precise internal rate generation.

The JET™ PLL takes any reference input and generates a base rate and 512 x base rate clocks. The base rate can be programmed to have a fractional relationship to the incoming reference.

The clock controller also contains two 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 muxes 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 xtal1 and xtal2 (typically 25.000MHz at xtal1/xtal2 doubled to 50.000MHz). Figure 27 illustrates how it is connected inside DICE II. As seen in the figure both sclk and the clock doubler output can be selected as input for both of the clock domains (JET™ PLL and system). The flexibility provided by the design allows convenient support of both 1394 applications and non-1394 applications, without sacrificing JET™ PLL performance.

In typical 1394 applications 'clk_out_hpll' is fed by the clock doubler, and 'clk_out_system' from the sclk (PHY clock) input, because the 1394 LLC and PHY must be 'synchronous'. Since the sclk has been generated within the PHY device using "unknown" PLL technology the quality of this clock cannot be guaranteed. For this reason, the JET™ PLL uses the output of the clock doubler, since we are in "full control" of the quality of this clock. Furthermore, we can choose a crystal frequency that is "out of sync" with the normal audio sample rates (recommended 25.000 MHz) which avoids beating and improves JET™ PLL performance. For 1394 applications we must power-up/power-down on a request sent over 1394. When powered down, the sclk from the PHY is disabled. To detect a wake-up both the Power Manager and the 1394 LLC require a "second clock". This clock is the direct input from 'xtl1'.

In typical non-1394 applications both clocks (JET™ PLL and system) are fed by the clock doubler and the sclk input is unused. This requires only one external clock source. Selection between the 2 clocks via firmware may be forced if desired.

"clk_out_hpll' is dedicated for the JET™ PLL block only, which means that the rest (including the prescalers within the Clock Controller) are fed with 'clock_out_system'.

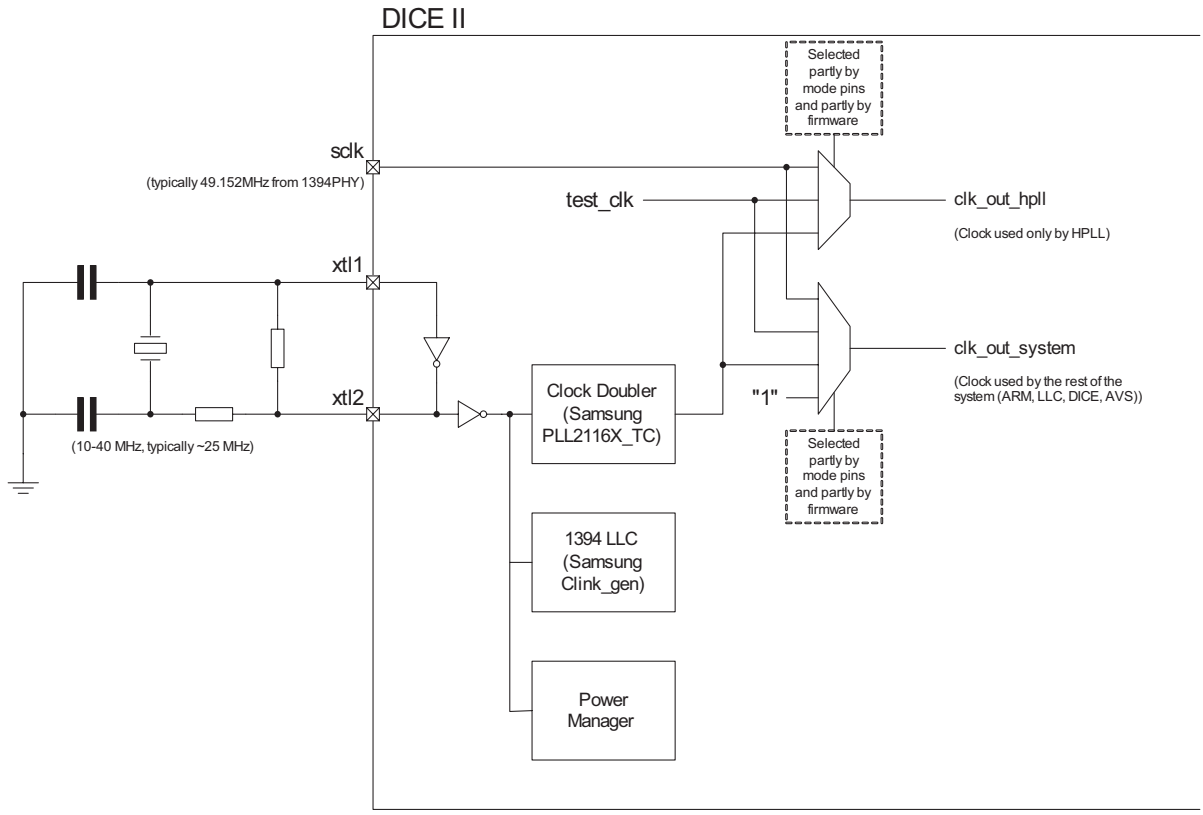


Figure 5.3: Clock Doubler Block Diagram

5.2.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
EXT_FBR	Y15	I/O (S)	6	External 1fs base rate clock (5V)
EXT_512BR	V14	I/O (S)	6	External 512 x base rate clock (5V)
HFS1	U9	I (S)	6	512 x base rate 96kHz, XTAL 24.576MHz
HFS2	V9	I (S)	6	512 x base rate 88kHz, XTAL 22.5792MHz
XTAL2	T20	O	-	XTAL - clock doubler/power manager/LLC
XTAL1	R18	I	-	XTAL - clock doubler/power manager/LLC
PLLE	B9	I	-	PLL Enable, HI=clock doubler (crystal input), LO=SCLK
NLIG	C9	I	-	No Lock Ignore - Connect to 1Kohm pullup.
WCKI	V11	I (S)	6	Word Clock In (5V)
WCKO	U11	O	6	Word Clock Out

Table 5.4: Clock Controller Signal Description

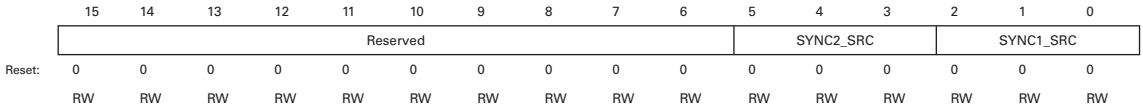
5.2.2 MODULE CONFIGURATION

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	RX_DOMAIN
0xce01 0020	TX_DOMAIN
0xce01 0024	AES_VCO_SETUP
0xce01 0028	ADAT_VCO_SETUP
0xce01 002c	TDIF_VCO_SETUP
0xce01 0030	SMUX
0xce01 0034	PRESCALER1
0xce01 0038	PRESCALER2
0xce01 003c	HPLL_REF
0xce01 0040	SRCNT1
0xce01 0044	SRCNT2
0xce01 0048	SR_MAX_CNT1
0xce01 004c	SR_MAX_CNT2

Table 5.5: Clock Controller Memory Map

5.2.3 SYNC_CTRL

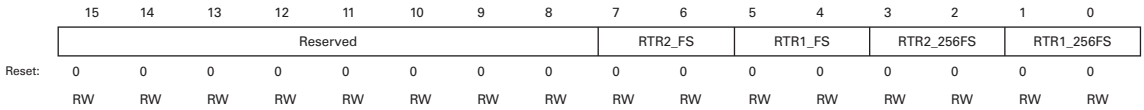
0xce01 0000



Name	Bit	Reset	Dir	Description
SYNC1_SRC	2:0	0	RW	Selects the clock source for Synchronizer 1 000: 22.5792MHz clock (must be mounted on EXT_512FS1) 001: 24.576MHz clock (must be mounted on EXT_512FS2) 010: Slave Inputs (EXT_FBR and EXT_512FB) 011: PLL1 xxx: Reserved for internal use
SYNC2_SRC	5:3	0	RW	Selects the clock source for Synchronizer 2 000: 22.5792MHz clock (must be mounted on EXT_512FS1) 001: 24.576MHz clock (must be mounted on EXT_512FS2) 010: Slave Inputs (EXT_FBR and EXT_512FB) 011: PLL2 xxx: Reserved for internal use

5.2.4 DOMAIN_CTRL

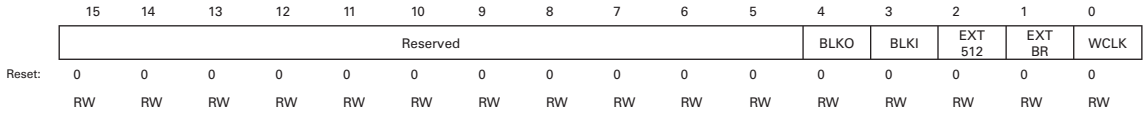
0xce01 0004



Name	Bit	Reset	Dir	Description
RTR2_FS	7:6	0	RW	Selects the FS source for Router2 00: base rate from Synchronizer2 01: 2 x base rate from Synchronizer2 10: base rate from Synchronizer1 11: 2 x base rate from Synchronizer1
RTR1_FS	5:4	0	RW	Selects the FS source for Router1 00: base rate from Synchronizer1 01: 2 x base rate from Synchronizer1 10: base rate from Synchronizer2 11: 2 x base rate from Synchronizer2
RTR2_256FS	3:2	0	RW	Selects the 256xFS source for Router2, Should be equal to RTR2_FS 00: 256 x base rate from Synchronizer2 01: 512 x base rate from Synchronizer2 10: 256 x base rate from Synchronizer1 11: 512 x base rate from Synchronizer1
RTR1_256FS	1:0	0	RW	Selects the 256xFS source for Router1, Should be equal to RTR1_FS 00: 256 x base rate from Synchronizer1 01: 512 x base rate from Synchronizer1 10: 256 x base rate from Synchronizer2 11: 512 x base rate from Synchronizer2

5.2.5 EXTCLK_CTRL

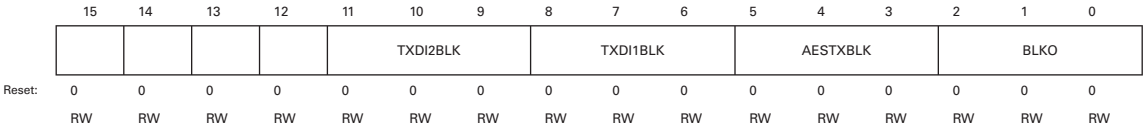
0xce01 0008



Name	Bit	Reset	Dir	Description
EXT512	2	0	RW	Selects the source for the external master mode output for 512x base rate (EXT_512FB) 0: Synchronizer1 1: Synchronizer2
EXTBR	1	0	RW	Selects the source for the external master mode output for base rate. (EXT_FBR), should equal EXT512. 0: Synchronizer1 1: Synchronizer2
WCLK	0	0	RW	Selects the source for the word clock output (WCLK_OUT) 0: Router1, 1fs 1: Router2, 1fs

5.2.6 BLK_CTRL

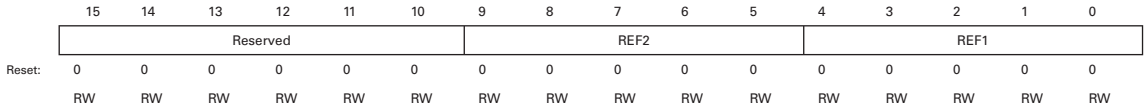
0xce01 000c



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: Blocksync from AVS ARx3 100: Blocksync from AVS ARx4 101: Blocksync from AESTx Blocksync Generator 111: Blocksync from AVS ATx1 110: Blocksync from External Blocksync input
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: Blocksync from AVS ARx3 100: Blocksync from AVS ARx4 101: Blocksync from AESTx Blocksync Generator 110: Blocksync from External Blocksync input 111: Blocksync from AVS ATx2
AESTXBLK	5:3	000	RW	Selects the block sync source for the AESTx. 000: Blocksync from AES Rx 001: Blocksync from AVS ARx1 010: Blocksync from AVS ARx2 011: Blocksync from AVS ARx3 100: Blocksync from AVS ARx4 101: Blocksync from External Blocksync input 110: Blocksync from AVS ATx1 111: Blocksync from AVS ATx2 Note: In order to use the block sync pin as an input the BLKS bit in the GPCSR_VIDEO_SELECT register should be set to '0'.
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: Blocksync from AVS ARx3 100: Blocksync from AVS ARx4 101: Blocksync from AESTX Blocksync Generator 110: Blocksync from AVS ATx1 111: Blocksync from AVS ATx2 Note: In order to drive the block sync pin as an output the BLKS bit in the GPCSR_VIDEO_SELECT register should be set to '1'.

5.2.7 REFEVENT_CTRL

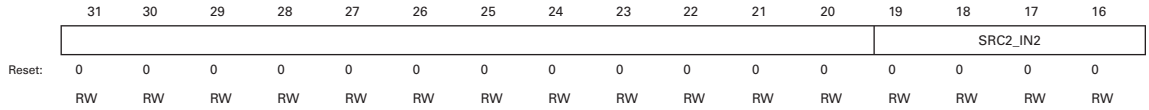
0xce01 0010



Name	Bit	Reset	Dir	Description
REF2	9:5	0	RW	Selects the ref. event source for PLL2. 00000: AES Rx (1fs) 00001: ADAT Rx (1fs) 00010: TDIF Rx (1fs) 00011: WCLK_IN 00100: 1394 Rx1 (1fs/SYT_INTERVAL) 00101: 1394 Rx2 (1fs/SYT_INTERVAL) 00110: 1394 Rx3 (1fs/SYT_INTERVAL) 00111: 1394 Rx4 (1fs/SYT_INTERVAL) 01000: Prescaler2 01001: EXT_FBR pin 01010: DSAI RX0 Sync In 01011: DSAI RX1 Sync In 01100: DSAI RX2 Sync In 01101: DSAI RX3 Sync In 01110: DSAITX0 Sync In 01111: DSAITX1 Sync In 10000: DSAITX2 Sync In 10001: DSAITX3 Sync In
REF1	4:0	0	RW	Selects the ref. event source for PLL1. 00000: AES Rx (1fs) 00001: ADAT Rx (1fs) 00010: TDIF Rx (1fs) 00011: WCLK_IN 00100: 1394 Rx1 (1fs/SYT_INTERVAL) 00101: 1394 Rx2 (1fs/SYT_INTERVAL) 00110: 1394 Rx3 (1fs/SYT_INTERVAL) 00111: 1394 Rx4 (1fs/SYT_INTERVAL) 01000: Prescaler1 01001: EXT_FBR pin 01010: DSAI RX0 Sync In 01011: DSAI RX1 Sync In 01100: DSAI RX2 Sync In 01101: DSAI RX3 Sync In 01110: DSAITX0 Sync In 01111: DSAITX1 Sync In 10000: DSAITX2 Sync In 10001: DSAITX3 Sync In

5.2.8 SRCNT_CTRL

0xce01 0014



	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	

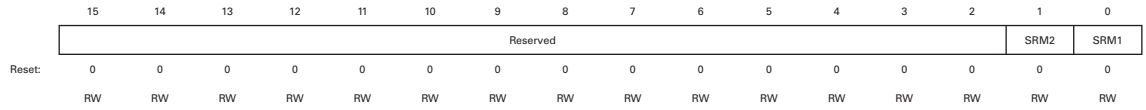
Name	Bit	Reset	Dir	Description
SRC2_IN2	19:15	0	RW	Selects source 2 for sample rate counter 2 See table below.
SRC2_IN1	14:10	0	RW	Selects source 1 for sample rate counter 2 See table below.
SRC1_IN2	9:5	0	RW	Selects source 2 for sample rate counter 1 See table below.
SRC1_IN1	4:0	0	RW	Selects source 1 for sample rate counter 1 See table below.

Source Select Value	Source
00000	AES Rx (1fs)
00001	ADAT Rx (1fs)
00010	TDIF Rx (1fs)
00011	WCLK_IN
00100	1394 Rx1 (1fs/SYT_INTERVAL)
00101	1394 Rx2 (1fs/SYT_INTERVAL)
00110	1394 Rx3 (1fs/SYT_INTERVAL)
00111	1394 Rx4 (1fs/SYT_INTERVAL)
01000	Router 1 1fs
01001	Router 2 1fs
01010	EXT_FBR pin
01011	DSAI RX0 Sync In
01100	DSAI RX1 Sync In
01101	DSAI RX2 Sync In
01110	DSAI RX3 Sync In
01111	DSAITX0 Sync In
10000	DSAITX1 Sync In
10001	DSAITX2 Sync In
10010	DSAITX3 Sync In
10011	AES Rx0 (1fs)
10100	AES Rx1 (1fs)
10101	AES Rx2 (1fs)
10110	AES Rx3 (1fs)

Table 5.6: Sample Rate Counter Input Selection

5.2.9 SRCNT_MODE

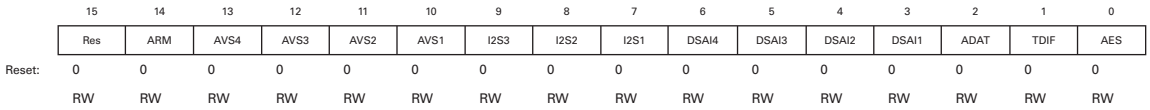
0xce01 0018



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 RX_DOMAIN

0xce01 001C



Name	Bit	Reset	Dir	Description
ARM	14	0	RW	Selects the ARM Audio interface router membership 0: Router 2 1: Router 1 Note: ARM Rx and Tx use same domain.
AVS4	13	0	RW	Selects the AVS4 interface router membership 0: Router 2 1: Router 1
AVS3	12	0	RW	Selects the AVS3 audio interface router membership 0: Router 2 1: Router 1
AVS2	11	0	RW	Selects the AVS2 Audio interface router membership 0: Router 2 1: Router 1
AVS1	10	0	RW	Selects the AVS1 Audio interface router membership 0: Router 2 1: Router 1
I2S3	9	0	RW	Selects the I²S 3 Audio interface router membership 0: Router 2 1: Router 1
I2S2	8	0	RW	Selects the I²S 2 Audio interface router membership 0: Router 2 1: Router 1
I2S1	7	0	RW	Selects the I²S 1 Audio interface router membership 0: Router 2 1: Router 1
DSAI4	6	0	RW	Selects the DSAI 4 Audio interface router membership 0: Router 2 1: Router 1
DSAI3	5	0	RW	Selects the DSAI 3 Audio interface router membership 0: Router 2 1: Router 1
DSAI2	4	0	RW	Selects the DSAI 2 Audio interface router membership 0: Router 2 1: Router 1
DSAI1	3	0	RW	Selects the DSAI 1 Audio interface router membership 0: Router 2 1: Router 1
ADAT	2	0	RW	Selects the ADAT Audio interface router membership 0: Router 2 1: Router 1
TDIF	1	0	RW	Selects the TDIF Audio interface router membership 0: Router 2 1: Router 1
AES	0	0	RW	Selects the AES Audio interface router membership 0: Router 2 1: Router 1

5.2.11 TX_DOMAIN

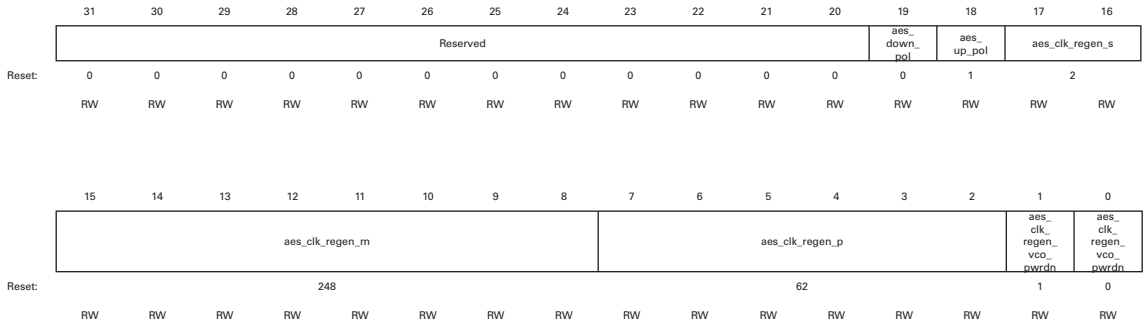
0xce01 0020

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				AVS2	AVS1	I2S3	I2S2	I2S1	DSAI4	DSAI3	DSAI2	DSAI1	ADAT	TDIF	AES
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
AVS2	11	0	RW	Selects the AVS2 Audio interface router membership 0: Router 2 1: Router 1
AVS1	10	0	RW	Selects the AVS1 Audio interface router membership 0: Router 2 1: Router 1
I2S3	9	0	RW	Selects the I²S 3 Audio interface router membership 0: Router 2 1: Router 1
I2S2	8	0	RW	Selects the I²S 2 Audio interface router membership 0: Router 2 1: Router 1
I2S1	7	0	RW	Selects the I²S 1 Audio interface router membership 0: Router 2 1: Router 1
DSAI4	6	0	RW	Selects the DSAI 4 Audio interface router membership 0: Router 2 1: Router 1
DSAI3	5	0	RW	Selects the DSAI 3 Audio interface router membership 0: Router 2 1: Router 1
DSAI2	4	0	RW	Selects the DSAI 2 Audio interface router membership 0: Router 2 1: Router 1
DSAI1	3	0	RW	Selects the DSAI 1 Audio interface router membership 0: Router 2 1: Router 1
ADAT	2	0	RW	Selects the ADAT Audio interface router membership 0: Router 2 1: Router 1
TDIF	1	0	RW	Selects the TDIF Audio interface router membership 0: Router 2 1: Router 1
AES	0	0	RW	Selects the AES Audio interface router membership 0: Router 2 1: Router 1

5.2.12 AES_VCO_SETUP

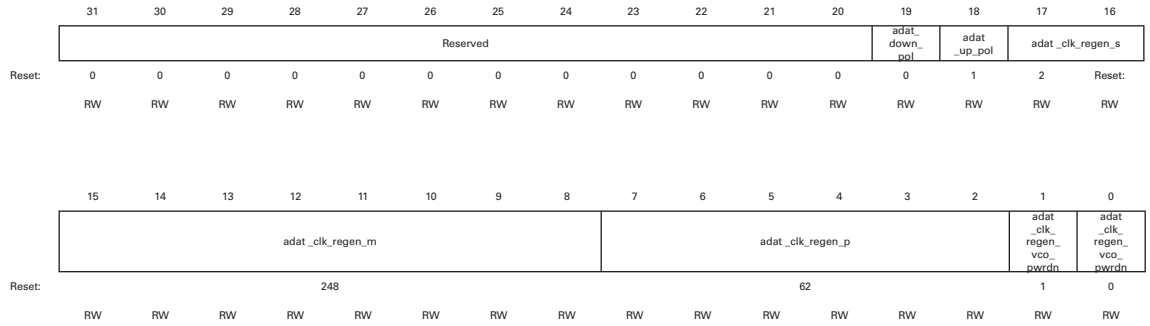
0xce01 0024



Name	Bit	Reset	Dir	Description
aes_down_pol	19	0	RW	Select polarity of DOWN signal to onchip 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 onchip 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
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

5.2.13 ADAT_VCO_SETUP

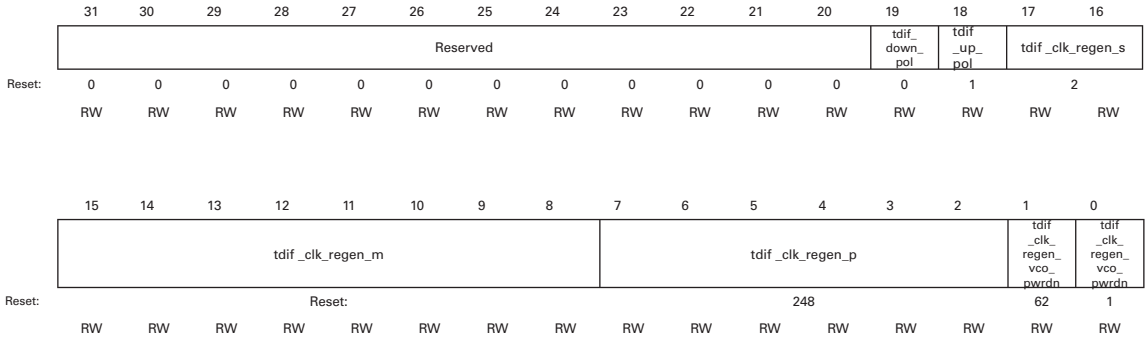
0xce01 0028



Name	Bit	Reset	Dir	Description
adat_down_pol	19	0	RW	Select polarity of DOWN signal to onchip ADAT VCO 0: Down signal to ADAT VCO is active low 1: Down signal to ADAT VCO is active high
adat _up_pol	18	1	RW	Select polarity of UP signal to onchip ADAT VCO 0: Up signal to ADAT VCO is active low 1: Up signal to ADAT VCO is active high
adat _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
adat _clk_regen_m	15:8	248	RW	Set the value for the Main - Divider (M) Range is 1 to 248
adat _clk_regen_p	7:2	62	RW	Set the value for the Pre - Divider (P) Range is 1 to 62
adat _clk_regen_vco_pwrdn	1	1	RW	Disable/Enable the internal VCO for the ADAT Receiver 0: Enable internal ADAT Receiver VCO 1: Disable internal ADAT Receiver VCO
adat _clk_regen_vco_ext_ clk_sel	0	0	RW	Selects source of VCO clock for ADAT Receiver either int. or ext. 0: From internal VCO / Clock 1: From external ADAT Receiver VCO

5.2.14 TDIF_VCO_SETUP

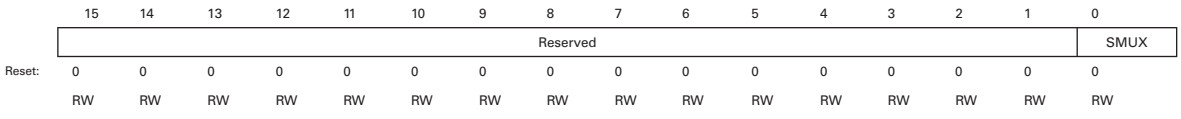
0xce01 002C



Name	Bit	Reset	Dir	Description
tdif_down_pol	19	0	RW	Select polarity of DOWN signal to onchip tdif VCO 0: Down signal to TDIF VCO is active low 1: Down signal to TDIF VCO is active high
tdif _up_pol	18	1	RW	Select polarity of UP signal to onchip tdif VCO 0: Up signal to TDIF VCO is active low 1: Up signal to TDIF VCO is active high
tdif _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
tdif _clk_regen_m	15:8	248	RW	Set the value for the Main - Divider (M) Range is 1 to 248
tdif _clk_regen_p	7:2	62	RW	Set the value for the Pre - Divider (P) Range is 1 to 62
tdif _clk_regen_vco_pwrdn	1	1	RW	Disable/Enable the internal VCO for the tdif Receiver 0: Enable internal TDIF Receiver VCO 1: Disable internal TDIF Receiver VCO
tdif _clk_regen_vco_ext_clk_ sel	0	0	RW	Selects source of VCO clock for tdif Receiver either int. or ext. 0: From internal VCO / Clock 1: From external TDIF Receiver VCO

5.2.15 SMUX

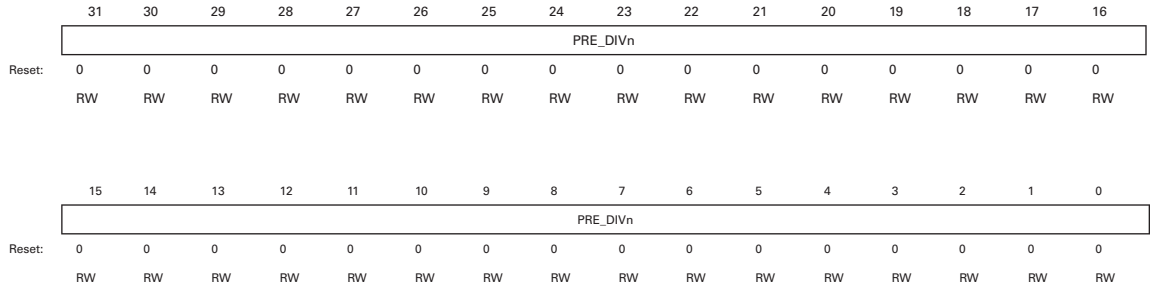
0xce01 0030



Name	Bit	Reset	Dir	Description
SMUX	0	0	RW	Select Base Rate Sync for S-MUX ADAT 0: Sync to router base rate 1: Sync to ADAT Rx

5.2.16 PRESCALERN

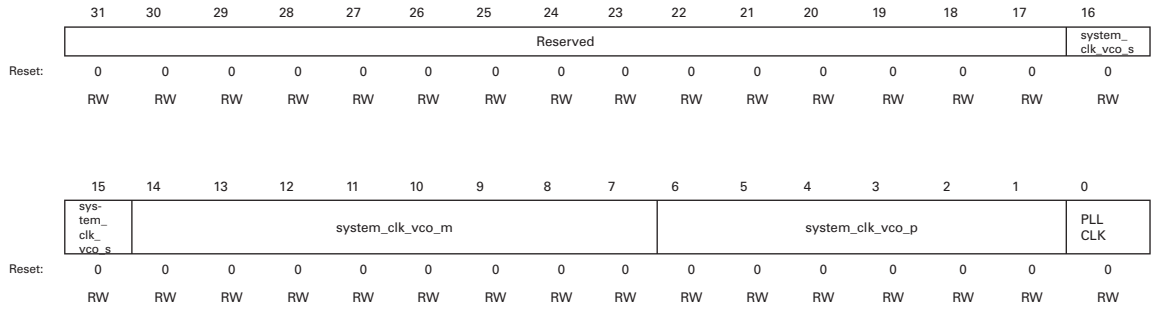
0xce01 0034 – 0xce010038



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

5.2.17 HPLL_REF

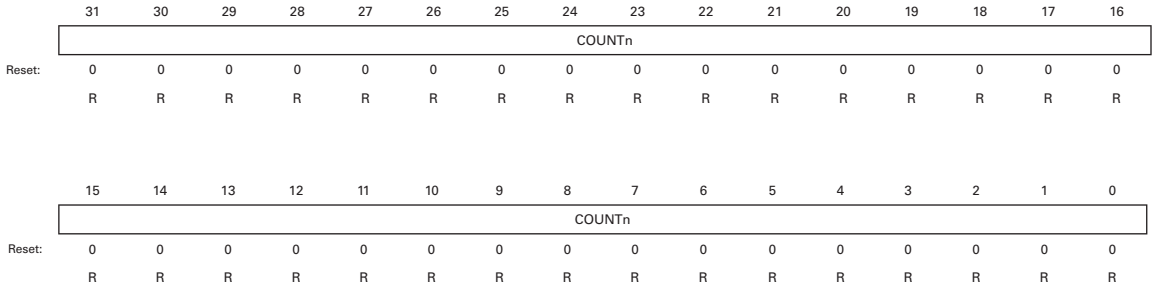
0xce01 003c



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)

5.2.18 SRCNTN

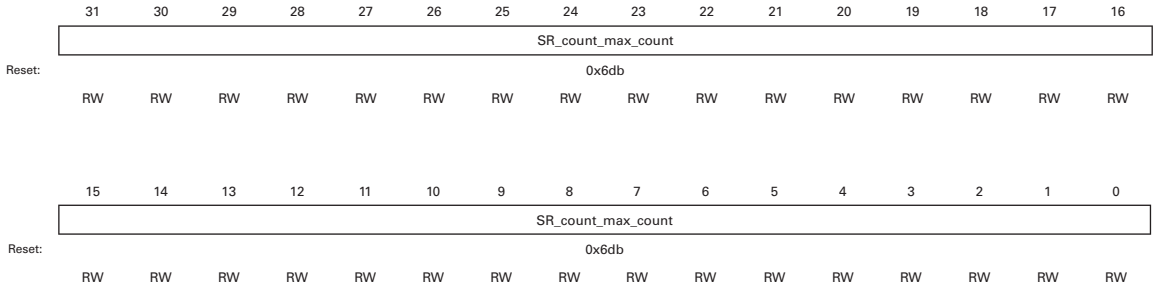
0xce01 0040



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.

5.2.19 SR_MAX_CNTN

0xce01 0048 - 0xce01 004c



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 PLLS

5.3.1 JET PLL BACKGROUND

The two Jitter Elimination Technologies (JET) PLLs on the DICE II chip feature state-of-the-art jitter rejection abilities and extremely low intrinsic jitter levels. The PLLs are ideally suited for the clock and data boundaries between any analog or digital source and destination.

Like all phase-locked loops, JET PLLs 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 tradeoff 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 PLLs sidestep the above-mentioned tradeoff. They incorporate 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 more-effectively suppressed.

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

JET PLLs contain 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 PLLs can also phase-lock an associated frame sync to the reference.

JET PLLs 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 DICE II.

The JET PLLs also allow 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 PLLs have 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 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 PLLs 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 technologies in are covered by several patents.

5.3.2 BLOCK DIAGRAM

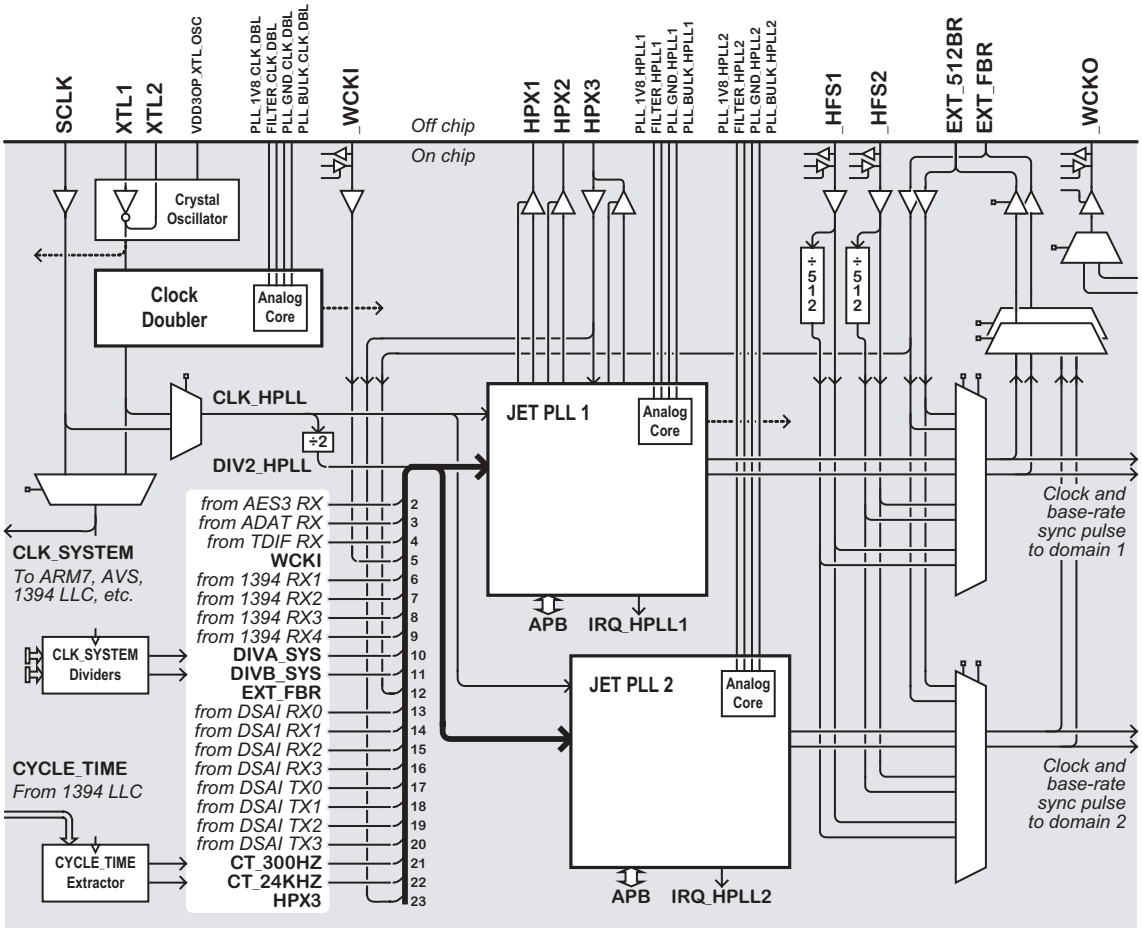


Figure 5.4: Detailed JET PLL Block Diagram

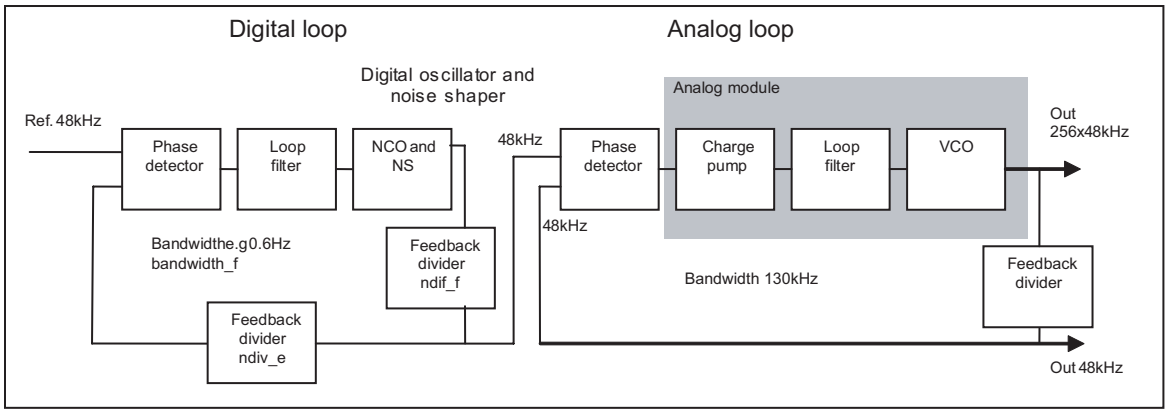


Figure 5.5: Simple block diagram of JET including location of dividers

5.3.3 BASIC REGISTERS

The JET PLL is instantiated two times allowing independent operation on each of the two clock domains on the DICE II chip. In this way it is possible to have jitter rejection for both domains, running e.g. 44.1 kHz in domain 1 and 48 or 96 kHz in domain 2.

The JET PLL is handled by control registers and status registers.

5.3.4 CONTROL REGISTERS

In the control registers the JET 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.

The optional external VCO2 clock HPX-pins on the chip can be used as General Purpose Outputs.

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').
gpo_grant	W	'hD8	'd0	3	Configures pins X3, X2, X1 as general-purpose outputs.
x1_gpo	W	'hDC	'd0	2	If pin X1 is a GPO, sets it to Z/H/L.
x2_gpo	W	'hE0	'd0	2	If pin X2 is a GPO, sets it to Z/H/L.
X3_gpo	W	'hE4	'd0	2	If pin X3 is a GPO, sets it to Z/H/L.

Table 5.7: Basic JET Registers

5.3.5 STATUS REGISTERS

At address …308hex the JET PLL is presenting its main status register. The 2 least significant bits are presented here:

bit	Register/field name	Brief description
1	unlocked	Triggered if phase offset wraps or exceeds 'u_threshold'.
0	ref_flawed	Triggered by reference discontinuities. Hi when auto coasting.

Table 5.8: JET PLL Status Registers

The register read value "unlocked" going high means that the JET 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 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 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 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 II the prescalers (outside the JET 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		Prescaler	ndiv_e divider	Actual FS	Deviation
(Hz)		(32bit)	(12bit)	(Hz)	(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

Table 5.9: Internal sampling rates generated with the JET PLL.

* When generating internal rates with great precision the register bandwidth_c should be set to 2dec. This will ensure stability in the JET 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 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

Table 5.10:Jitter rejection bandwidth set in the JET.

The -60dB frequency is at approx 13 times the -3dB frequency. The roll off is 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 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

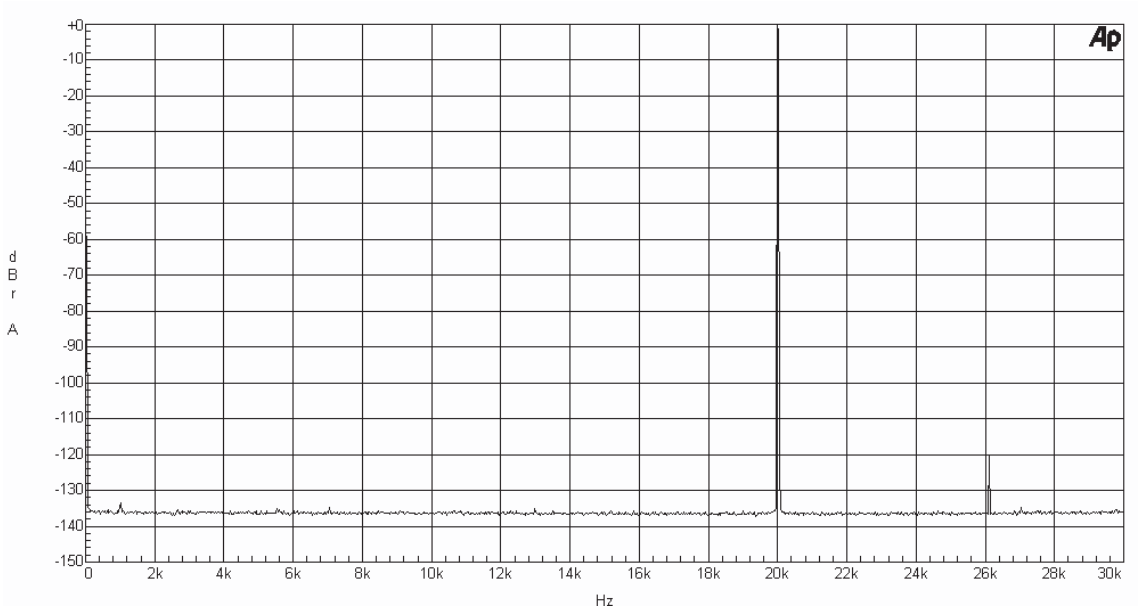


Figure 5.6: Resulting spectrum when converting a 20 kHz audio tone. The spectrum looks the same with and without incoming jitter being removed.

5.3.9 JET GENERAL PURPOSE OUTPUTS

The HPX-pins (1, 2 and 3) on the DICE II can be used as general purpose outputs (GPO's). The 3 pins are granted to be GPO's by setting the gpo_grant register. When set to 1hex the HPX1 pin is GPO, set to 2hex HPX2 is GPO and set to 4hex HPX3 is GPO. Set to 7hex all three HPX-pins are set to GPO's.

The actual value of HPX1 is set by setting x1_gpo. 0hex = high Z, 1hex = high and 2hex = low.

5.3.9.1 JET GENERAL PURPOSE INPUTS

One of the HPX pins (HPX3) can be used as a general purpose input (GPI). The status of this input can be read at the address ...3D8hex (one bit width.)

Register	r/w	Addr hex	Width bits	Brief description
GPI	R	'h3D8	1	For reading HPX3 as general purpose input.

Table 5.11: HPX3 General Purpose Input Register

5.3.10 EXTERNAL CIRCUIT (JET PLL, AES, ADAT, TDIF AND CLK_DBL).

DICE II has 6 analog PLL's requiring 3 external components each to set the analog PLL loop filter values. COG / NPO types are recommended for the two capacitors. X7R types and 10% may do well in certain cases. A concern here is that pronounced mechanical capacitance changes have been observed on several non COG/NPO capacitor types.

The actual analog PLL loop filter circuit for each PLL is C2//(C1+R1), a series circuit between the chip filter pin and GND. See below figure. Minimum wiring length as well as allowing no other current to be drawn in the filter gnd loops is highly recommended.

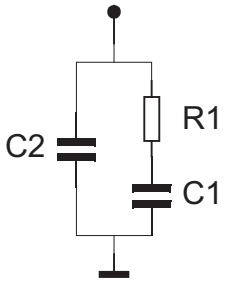


Figure 5.7: External filter diagram.

	JET PLL's	Clock doubler	ADAT	AES	TDIF
C1	10nF	470pF	10nF	10nF	10nF
C2	100pF	10pF	10pF	10pF	470pF
R1	1kohm	4.7kohm	8.2kohm	1kohm	10kohm

Table 5.12: Recommended PLL Values
Recommended values for the different PLL loop filters on the DICE II chip.

5.3.11 JET PLL REGISTERS

The JET PLL registers are categorized as Basic or Advanced. Application firmware typically reads and writes the values of the basic registers. Firmware provided by Wavefront in the DICE II 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.

5.3.11.1 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.

register / field name	r, w *	Adr. hex	Value decimal	Width bits	brief description
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	Configures pins X3, X2, X1 as general-purpose outputs.
x1_gpo	W	'h37	0	2	If pin X1 is a GPO, sets it to Z/H/L.
x2_gpo	W	'h38	0	2	If pin X2 is a GPO, sets it to Z/H/L.
x3_gpo	W	'h39	0	2	If pin X3 is a GPO, sets it to Z/H/L.
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.

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

5.3.11.2 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.

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

5.3.11.3 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.

5.3.11.4 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

Address	Register
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
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

Address	Register
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
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_$

Address	Register
0xcc0003d8	PLL1_GPI
0xcc0003e0	PLL1_CONFIG_AC
0xcc0003f0	PLL1_SHUTDOWN_M
0xcc0003f4	PLL1_SHUTDOWN_I
0xcc0001f8	PLL1_SHUTDOWN_E
0xcd00 0000	PLL2_CAF_ENABLE
0xcd00 0004	PLL2_CAF_SELECT
0xcd00 0008	PLL2_COAST
0xcd00 0018	PLL2_REF_SEL
0xcd00 001c	PLL2_REF_EDG
0xcd00 0028	PLL2_RDIV
0xcd00 002c	PLL2_THROTTLE
0xcd00 0058	PLL2_U_THRESHOLD
0xcd00 0060	PLL2_BW_FLOOR
0xcd00 0064	PLL2_BW_CEILING
0xcd00 0068	PLL2_SHP_FIX
0xcd00 006c	PLL2_SHP_VAR
0xcd00 0070	PLL2_MAX_SLW_FIX
0xcd00 0074	PLL2_MAX_SLW_VAR
0xcd00 0078	PLL2_DCNT_LIN
0xcd00 007c	PLL2_DCNT_EXP
0xcd00 0088	PLL2_LOOSE_THR
0xcd00 0098	PLL2_MIN_PER
0xcd00 009c	PLL2_MAX_PER
0xcd00 00b0	PLL2_NDIV_F
0xcd00 00b4	PLL2_NDIV_E
0xcd00 00b8	PLL2_NDIV_B
0xcd00 00bc	PLL2_BYP_F
0xcd00 00c0	PLL2_PHASE_LAG
0xcd00 00c8	PLL2_FRACT_RES
0xcd00 00d0	PLL2_BURST_LEN
0xcd00 00d8	PLL2_GPO_EN
0xcd00 00dc	PLL2_GPO_1
0xcd00 00e0	PLL2_GPO_2
0xcd00 00e4	PLL2_GPO_3
0xcd00 00f0	PLL2_X1X2_MODE
0xcd000100	PLL2_CHAIN_I

Address	Register
0xcd000104	PLL2_SINK_I
0xcd000108	PLL2_ANCHOR_I
0xcd00010c	PLL2_IANCHOR_VAL
0xcd000110	PLL2_UNBND_I
0xcd000118	PLL2_IDET
0xcd000120	PLL2_IDIV_C
0xcd000124	PLL2_IDIV_F
0xcd000128	PLL2_IDIV_S
0xcd000130	PLL2_INV_CDI
0xcd000134	PLL2_HBL_CDI
0xcd000144	PLL2_SINK_E
0xcd000148	PLL2_ANCHOR_E
0xcd00014c	PLL2_E_ANC_VAL
0xcd000150	PLL2_UNBIND_E
0xcd000158	PLL2_EDET_X1
0xcd00015c	PLL2_EDET_X2
0xcd000160	PLL2_EDIV_C
0xcd000164	PLL2_EDIV_F
0xcd000168	PLL2_EDIV_S
0xcd000170	PLL2_INV_CDE
0xcd000174	PLL2_HBL_CDE
0xcd000180	PLL2_DIVIDE_CJ
0xcd000184	PLL2_INVERT_CJ
0xcd000280	PLL2_FAMILY_ID
0xcd000284	PLL2_FORM_ID
0xcd000288	PLL2_REVISION_ID
0xcd00028c	PLL2_INSTANCE_ID
0xcd0002b8	PLL2_MTR_SELECT
0xcd0002bc	PLL2_MTR_EDGES
0xcd0002c0	PLL2_RES_EX
0xcd0002c4	PLL2_PUNC_MP
0xcd0002cc	PLL2_MTR_PERIOD
0xcd0002d0	PLL2_GREATEST_MP
0xcd0002d4	PLL2_GREATEST_MP_$
0xcd0002d8	PLL2_SMALLEST_MP
0xcd0002dc	PLL2_SMALLEST_MP_$
0xcd000300	PLL2_TICK_RATE
0xcd000304	PLL2_TURN_RATE

Address	Register
0xcd000308	PLL2_MAIN_STATUS
0xcd00030c	PLL2_MAIN_STATUS_$
0xcd000320	PLL2_DETECT_R
0xcd000324	PLL2_DETECT_F
0xcd000328	PLL2_STICKY_BITS
0xcd00032c	PLL2_STICKY_BITS_$
0xcd000350	PLL2_IRQ_ENABLES
0xcd00038c	PLL2_NCO_PERIOD
0xcd000390	PLL2_GREATEST_NP
0xcd000394	PLL2_GREATEST_NP_$
0xcd000398	PLL2_SMALLEST_NP
0xcd00039c	PLL2_SMALLEST_NP_$
0xcd0003d8	PLL2_GPI
0xcd0003e0	PLL2_CONFIG_AC
0xcd0003f0	PLL2_SHUTDOWN_M
0xcd0003f4	PLL2_SHUTDOWN_I
0xcd0001f8	PLL2_SHUTDOWN_E

Table 5.13: JET™ PLL Memory Map

5.4 AES RECEIVERS

The DICE 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

Signal	PBGA Pin	I/O	Drive (mA)	Description
AES_RX0	W15	I	-	AES3 Receiver Ch0/1 (5V)
AES_RX1	Y16	I	-	AES3 Receiver Ch2/3 (5V)
AES_RX2	U14	I	-	AES3 Receiver Ch4/5 (5V)
AES_RX3	V15	I	-	AES3 Receiver Ch6/7 (5V)

Table 5.13: AES Receiver Signal Description

5.4.2 MODULE CONFIGURATION

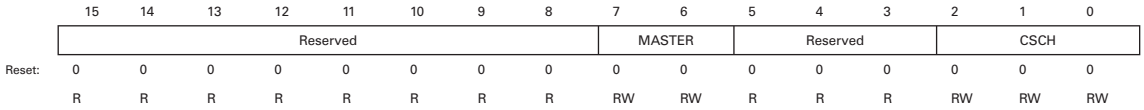
Address	Register
0xce02 0000	CTRL
0xce02 0004	STAT_ALL
0xce02 0008	STAT_RX0
0xce02 000c	STAT_RX1
0xce02 0010	STAT_RX2
0xce02 0014	STAT_RX3
0xce02 0018	V_BIT
0xce02 0040	PLL_PULSE_WIDTH
0xce02 0044	FORCE_VCO
0xce02 0048	VCO_MIN_LSB
0xce02 004c	VCO_MIN_MSB
0xce02 0080	CHSTAT_0_BYTE0
0xce02 0084	CHSTAT_0_BYTE1
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
0xce02 00f0 - 0xce02 00fc	CHSTAT_7_BYTE0-3
0xce02 0100 – 0xce02 015c	CHSTAT_FULL_BYTE0-23

Table 5.14: 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

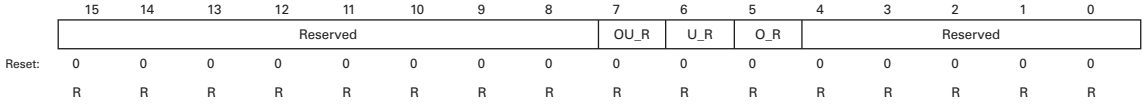
0xce02 0000



Name	Bit	Reset	Dir	Description
MASTER	7:6	0	RW	Selects the master receiver.
CSCH	2:0	0	RW	Selects the channel to receive full Channel Status from.

5.4.4 STAT_ALL

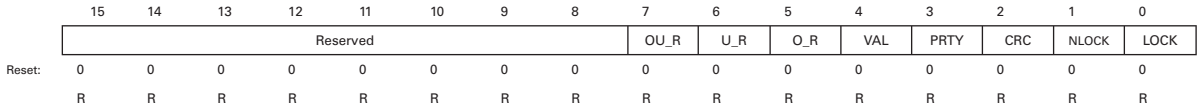
0xce02 0004



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 have occured.
U_R	6	0	R	Indicates resampling which 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

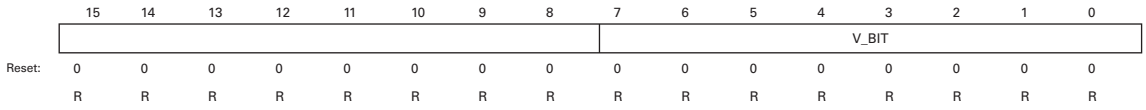
0xce02 0008 - 0xce02 0014



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 have occured.
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

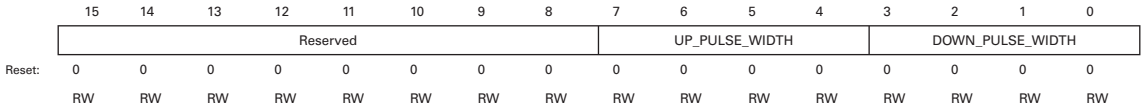
0xce02 0018



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

5.4.7 PLL_PULSE_WIDTH

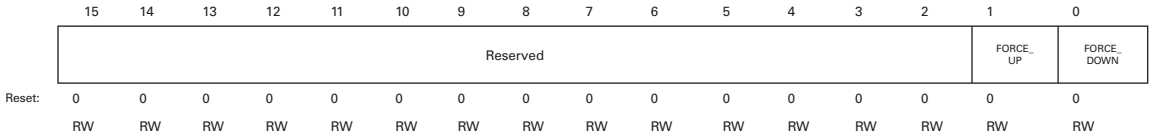
0xce02 0040



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

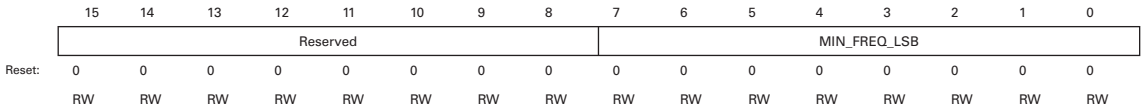
0xce02 0044



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

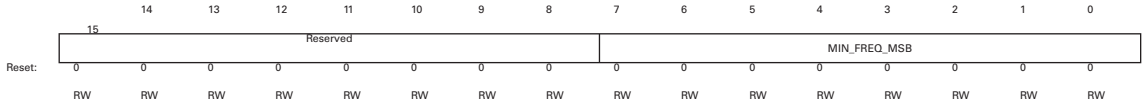
0xce02 0048



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

0xce02 004c



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 II 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

Signal	PBGA Pin	I/O	Drive (mA)	Description
AES_TX0	W16	O	2	AES3 Transmitter Ch0/1
AES_TX1	Y17	O	2	AES3 Transmitter Ch2/3
AES_TX2	V16	O	2	AES3 Transmitter Ch4/5
AES_TX3	W17	O	2	AES3 Transmitter Ch6/7

Table 5.15: AES Transmitter Signal Description

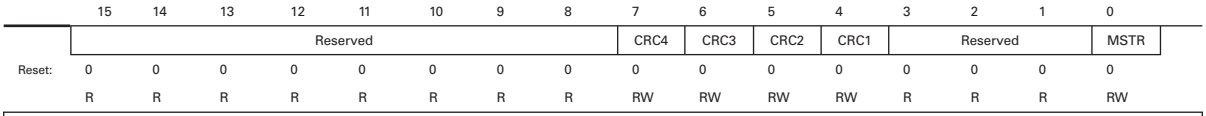
5.5.2 MODULE CONFIGURATION

Address	Register
0xce03 0000	MODE_SEL
0xce03 0004	CBL_SEL
0xce03 0008	CS_SEL1
0xce03 000c	CS_SEL2
0xce03 0010	CS_SEL3
0xce03 0014	MUTE
0xce03 0018	V_BIT
0xce03 0040	USR_SEL1
0xce03 0044	USR_SEL2
0xce03 0048	USR_SEL3
0xce03 004c	USR_SEL4
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

Table 5.16: AES Transmitter Memory Map

5.5.3 MODE_SEL

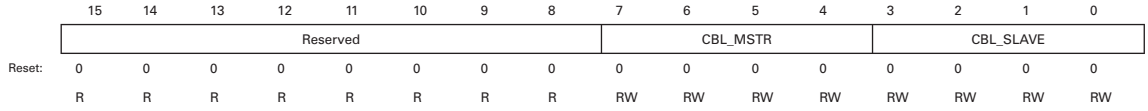
0xce03 0000



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
MSTR	0	0	RW	Selects the transmitter mode. Refer to section ???. 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

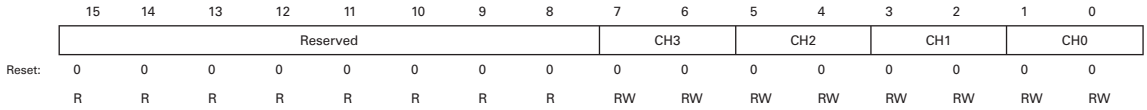
0xce03 0004



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_SEL1

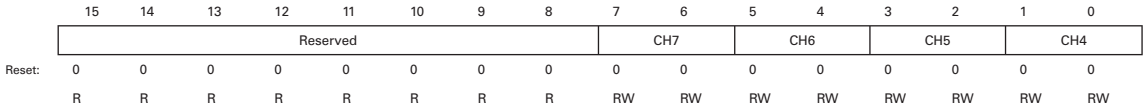
0xce03 0008



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
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_SEL2

0xce03 000c

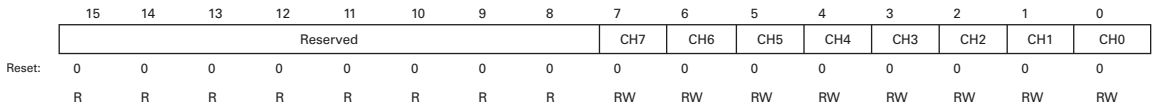


Name	Bit	Reset	Dir	Description
CH7	7:6	0	RW	Selects the Channel Status source in slave mode. 00: From bit 26 in audio channel 7 01: From bit 26 in audio channel selected by CBL_SLAVE 10: From Memory mapped CS registers defined below. 11: reserved
CH6	5:4	0	RW	Selects the Channel Status source in slave 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_SEL3

0xce03 0010

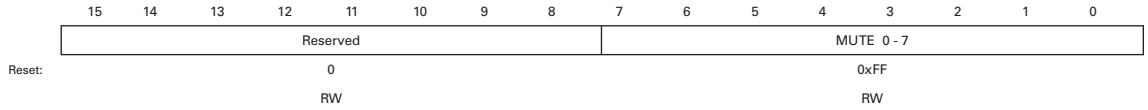
When the memory mapped Channel Status mode is selected for a channel, the first 4 channel status bytes (bit 0-31) can come from either a common memory mapped file or an individual per channel file. The remaining 20 bytes (bit 32-192) are always defined by the common file.



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

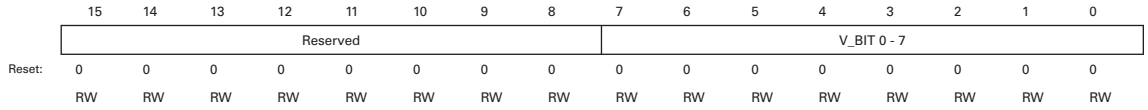
0xce03 0014



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

0xce03 0018

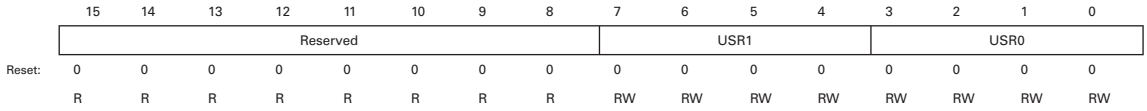


Name	Bit	Reset	Dir	Description
V_BIT 0 - 7	7:0	0	RW	One bit for each channel selects whether the V bit should indicate valid audio (V=0) or invalid audio (V=1). Default configuration after reset is to indicate valid audio. 0 = Audio valid. 1 = Audio invalid.

5.5.10 USR_SEL1

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.

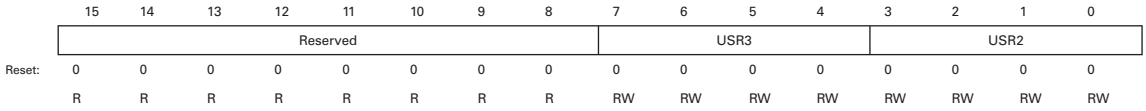


Name	Bit	Reset	Dir	Description
USR1	7:4	0	RW	4 bits for each channel selects the USER bit source. Slave mode: 0xxx = USER bit from Audio register file (same as audio data). 1xxx = USER bit set to 0 Master mode: 0nnn = USER bit from AES Receiver channel nnn (nnn = "000" => Rx channel 0, ..., nnn = "111" => 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_SEL2

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.



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_SEL3

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.

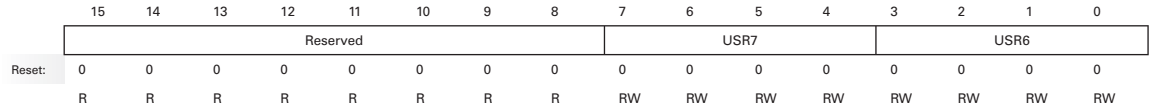
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								USR5				USR4			
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

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_SEL4

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.



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

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

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

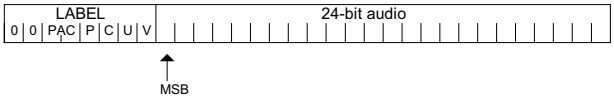
In order to have full control of the extra data send in an AES sub-frame, two basic modes are provided. The Parity bit and the Left/Right sub-frame bits are always calculated by the transmitter.

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 I²S RECEIVER

The DICE II contains three I2S receiver modules, one 8 channel module and 2 4-channel modules. These are not only I2S compliant, but highly configurable serial audio interface receivers 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. Each of the three modules has individual clock signals and can be assigned to either of the two clock domains. Each module has one set of clock lines including MCK, BICK and LRCK, and 2 or 4 data lines each communicating 2 channels of audio.

All 3 modules can receive data at 192KHz sample rates. Since the highest sample rate that can be used internally (system) is 96KHz, when running at 192KHz rate the 8 channel interface becomes a 4 channel interface. Due to the design of the 4-channel modules, when running at 192KHz they will remain 4-channel interfaces, for a total of 12 channels at 192KHz. 192KHz samples from the 4 channels are multiplexed into consecutive pairs in the 96KHz system buffers per the S-Mux standard. The diagram below illustrates the router channel configuration for 48/96KHz and 192KHz operation. 192KHz operation is selected by setting a bit in the I²S receiver setup registers.

I2S Dataline	D0		D1		D2		D3	
I2S Channel	0	1	2	3	4	5	6	7
Router Channel	1	2	3	4	5	6	7	8
Audio Sequence	m	n	o	p	q	r	s	t

Table 5.17: I2S Rx 1, 96k router channel configuration
96k router channel configuration for I2S Receiver 1 (8-channel)

I2S Dataline	D0				D1			
I2S Channel	0		1		2		3	
Router Channel	1	2	3	4	5	6	7	8
Audio Sequence	m	m+1	n	n+1	o	o+1	p	p+1

Table 5.18: I2S Rx 1, 192k router channel configuration)
192k router channel configuration for I2S Receiver 1 (8-channel)

I2S Dataline	D0		D1		x		x	
I2S Channel	0	1	2	3	x	x	x	x
Router Channel	1	2	3	4	5	6	7	8
Audio Sequence	m	n	o	p	x	x	x	x

Table 5.19: I2S Rx 2, 96k router channel configuration
96k router channel configuration for I2S Receiver 2 or 3 (4-channel interface)

I2S Dataline	D0				D1			
I2S Channel	0		1		2		3	
Router Channel	1	2	3	4	5	6	7	8
Audio Sequence	m	m+1	n	n+1	o	o+1	p	p+1

Table 5.20: I2S Rx 2 or 3, 192k router channel configuration
192k router channel configuration for I2S Receiver 2 or 3 (4-channel interface)

5.6.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
I2S_RX2_MCK	J18	O	8	I2S Receiver 2 Master Clock
I2S_RX2_BICK	J17	O	8	I2S Receiver 2 Bit Clock
I2S_RX2_LRCK	H20	O	8	I2S Receiver 2 Left/Right Clock
I2S_RX2_D0	H19	I	-	I2S Receiver 2 Data (Ch. 0/1) (5V)
I2S_RX2_D1	H18	I	-	I2S Receiver 2 Data (Ch. 2/3) (5V)
I2S_RX1_MCK	E20	O	8	I2S Receiver 1 Master Clock
I2S_RX1_BICK	G17	O	8	I2S Receiver 1 Bit Clock
I2S_RX1_LRCK	F18	O	8	I2S Receiver 1 Left/Right Clock
I2S_RX1_D0	E19	I	-	I2S Receiver 1 Data (Ch. 0/1) (5V)
I2S_RX1_D1	D20	I	-	I2S Receiver 2 Data (Ch. 2/3) (5V)
I2S_RX0_MCK	A20	O	8	I2S Receiver 0 Master Clock
I2S_RX0_BICK	A19	O	8	I2S Receiver 0 Bit Clock
I2S_RX0_LRCK	B18	O	8	I2S Receiver 0 Left/Right Clock
I2S_RX0_D0	B17	I	-	I2S Receiver 0 Data (Ch. 0/1) (5V)
I2S_RX0_D1	C17	I	-	I2S Receiver 0 Data (Ch. 2/3) (5V)
I2S_RX0_D2	D16	I	-	I2S Receiver 0 Data (Ch. 4/5) (5V)
I2S_RX0_D3	A18	I	-	I2S Receiver 0 Data (Ch. 6/7) (5V)

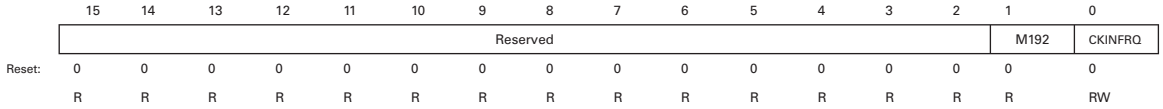
Table 5.21: I²S Receiver Signal Description

5.6.2 MODULE CONFIGURATION

Address	Register
0xce10 0000	I2S_RX1_MODE
0xce10 0004	I2S_RX1_CH1_CTRL
0xce10 0008	I2S_RX1_CH2_CTRL
0xce10 000c	I2S_RX1_CH3_CTRL
0xce10 0010	I2S_RX1_CH4_CTRL
0xce12 0000	I2S_RX2_MODE
0xce12 0004	I2S_RX2_CH1_CTRL
0xce12 0008	I2S_RX2_CH2_CTRL
0xce14 0000	I2S_RX3_MODE
0xce14 0004	I2S_RX3_CH1_CTRL
0xce14 0008	I2S_RX3_CH2_CTRL

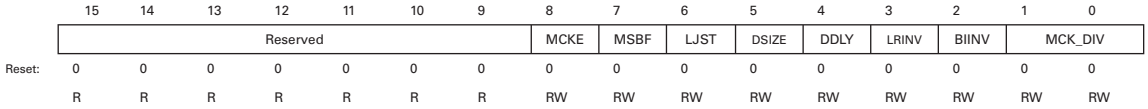
Table 5.22: I²S Receiver Memory Map

5.6.3 I2S_RXN_MODE



Name	Bit	Reset	Dir	Description
CKINFRQ	0	0	RW	This bit selects the frequency of the clock input to the I²S Receiver block from the clock controller. 0: 256fs (256 x the sampling frequency) 1: 512fs (512 x the sampling frequency)
M192	1	0	RW	This bit selects the sample rate mode for the interface. 0: Normal mode, 8 (4) channels of base rate or 2 x base rate 1: Double mode, 4 (2) channels of 4 x base rate. Note: In Double mode the channels are packed in pairs at half the sample rate from the router point of view.

5.6.4 I2S_RXN_CHM_CTRL



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							MCKE	MSBF	LJST	DSIZE	DDLY	LRINV	BIINV	MCK_DIV	
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
MCKE	8	0	RW	This bit enables the MCK signal. On some of the receivers the MCK pin has alternate functions. (see GPCSR for details) 0: MCK disabled 1: MCK Enabled Note: This bit is only valid in the CH0 register. All channels in one receiver share the clock pins.
MSBF	7	0	RW	This bit selects the ordering of the bits in the data frame. 0: LSB is sent first 1: MSB is sent first
LJST	6	0	RW	This bit selects the justification of the bits in the data frame when 24 bit data size is selected. 0: Right justified (padded LSB's) 1: Left justified (padded MSB's)
DSIZE	5	0	RW	This bit selects the data size. 0: 24 bit data 1: 32 bit data
DDLY	4	0	RW	This bit selects delayed data to comply with I²S. 0: No Delay, first bit aligned with LRCK edge. 1: One bit delay after LRCK edge.
LRINV	3	0	RW	This bit selects the phase of the LRCK pin. On some of the receivers the LRCK pin has alternate functions. (see GPCSR for details) 0: LRCK is high for left channel data, low for right channel data. 1: LRCK is low for left channel data, high for right channel data. Note: This bit is only valid in the CH0 register. All channels in one receiver share the clock pins.
BIINV	2	0	RW	This bit selects the phase of the BICK pin. On some of the receivers the LRCK pin has alternate functions. (see GPCSR for details) 0: The frame starts with a negative edge. 1: The frame starts with a positive edge. Note: This bit is only valid in the CH0 register. All channels in one receiver share the clock pins.
MCK_DIV	1:0	0	RW	This bit selects the frequency of the MCK pin. . On some of the receivers the LRCK pin has alternate functions. (see GPCSR for details) 00: 256fs (128fs in Double mode) 01: 128fs (64fs in Double mode) 10: 64fs (32 fs in Double mode) 11: Reserved Note: This bit is only valid in the CH0 register. All channels in one receiver share the clock pins.

5.6.5 MODES OF OPERATION

Several options are provided to comply with different serial audio formats.

MCK_DIV [1:0]

The receiver can be set up to output a master clock that differs in frequency from the router master clock. Binary combinations on the two pins on the input to the receiver will generate divided versions of the input master clock as follows:

0 = Use router 256fs

1 = Divide router 256fs by 2.

2 = Divide router 256fs by 4.

BIINV

BICK is the 64fs bit clock output from the receiver. The receiver can be set up so that this clock has either a rising or falling edge corresponding to the transition on the output word sync (LRCK_OUT) as follows:

0 = Bit clock falling edge on falling LRCK_OUT

1 = Bit clock rising edge on falling LRCK_OUT

LRINV

The receiver can be set up to output a word sync (LRCK_OUT) that is either in phase or 180° out of phase (inverted) with respect to the internal word sync.

0 = LRCK_OUT in phase with internal word sync. (High during left sub frame)

1 = LRCK_OUT 180° out of phase with internal word sync. (low during left sub frame)

DDLY

The receiver can be programmed to comply with devices that have a one serial bit clock delay after a transition on the word clock, that is, transmitters that send the MSB/LSB of the next word one clock period after a transition on LRCK_OUT (for example I^2S compliant devices). The DDLY input is programmed as follows:

0 = No delay (LRCK_OUT aligned with MSB/LSB of current sample)

1 = 1 serial bit word offset (MSB of next sample transmitted 1 BICK period after transition on LRCK_OUT)

DSIZE

The receiver has the possibility to handle either 24-bit or 32-bit data. This is important with respect to justification (left/right) of the data, since left or right justification can only be possible in 24-bit mode. This pin selects either 24-but or 32-bit operation as follows:

0 = 24-bit operation

1 = 32-bit operation

LJST

Since the receiver always outputs a sample in parallel with LSB in the lowest bit position it must know whether data received is left or right justified, so that the data can be shifted to comply with the output data format. When operating in 24-bit mode, this pin is programmed so that the receiver knows whether or not to shift data in the output register before outputting it to the DICE II router. The value driven on this pin is ignored when operating in 32-bit mode. The pin is programmed as follows:

0 = Right justified 24-bit data

1 = Left justified 24-bit data

MSB

The receiver always presents the received data in parallel, 32 bits with MSB first. In order to present the data MSB first (MSB in the highest bit position), it has to know whether the transmitter device transmits MSB or LSB first. To do this we program the input pin MSB_FIRST as follows:

0 = LSB first

1 = MSB first

The following two sections illustrate the receiver operating in two different modes:

5.6.6 NORMAL OPERATION

Figure 2 below illustrates the serial audio receiver operating in a normal or 'standard' mode, similar to crystal ADC's and many DSP serial ports. In this mode, we have a word length of 64 bits, ie. 32 bits per channel, MSB is transmitted first, left justified, and data is transmitted on the falling edge and latched on the rising edge of the serial bit clock.

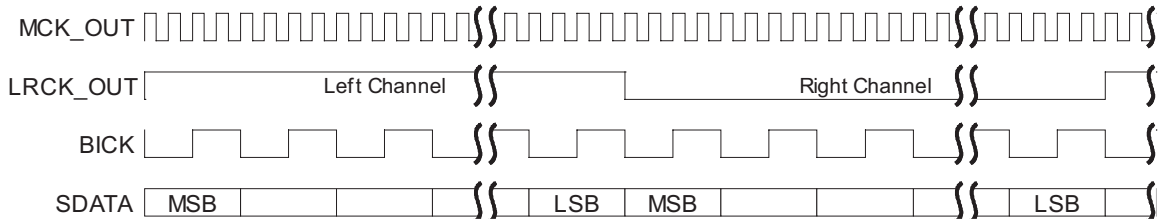


Figure 5.8: Serial audio receiver in master mode - normal operation

To achieve the above mode of operation we must program the following pins on the input to the receiver:

- MSB transmission beginning on the transition on LRCK_OUT, i.e. DDLY = 0.
- Receive MSB transmitted first, i.e. MSB = 1.
- Each bit transmitted beginning on the falling edge of the bit clock and clocked in on the rising edge of the bit clock, i.e. BIINV = 0.
- Left Channel transmitted during LRCK_OUT = 1, i.e. LRINV = 0.

5.6.7 I²S 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 (I²S), a serial link especially for digital audio.

The bus has three lines:

- Continuous serial clock SCK (BICK)
- Word Select WS (LRCK)
- Serial data SD (SDATA)

The most distinguishing feature of the I²S 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, whereas the position of the LSB depends on the word length. If the receiver is sent fewer bits than it's word length, the missing bits are set to zero internally. If the receiver is sent more bits than it's word length, the missing bits are ignored.

Our receiver can be programmed as a fully compliant I²S receiver and is designed to operate as a master on the I²S bus, which means that it must output the Word Select (through the LRCK_OUT pin) clock and the continuous serial clock (output BICK) to the I²S transmitter operating in slave mode.

Figure 5.9 illustrates I²S operation with a 32 bit word length.

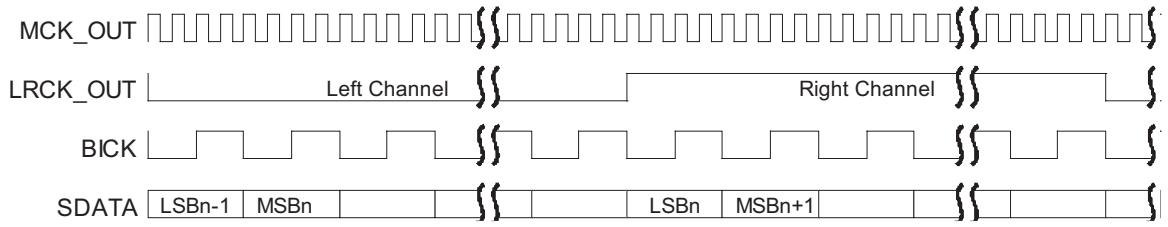


Figure 5.9: I²S compliant operation – 32 bit word length

To achieve I²S compliant operation, we must therefore make sure that the receiver operates in the correct mode by programming the input pins as follows:

- MSB transmitted one serial clock (BICK) period after transition on LRCK_OUT i.e. DDLY = 1.
- Receive MSB transmitted first i.e. MSB = 1.
- 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. BIINV = 0.
- Inversion of LRCK_OUT so that reception of left channel data corresponds to LRCK_OUT=0 i.e. LRINV = 1.

5.7 I²S TRANSMITTERS

The DICE II contains three I²S transmitter modules, one 8 channel module and 2 4-channel modules.These are not only I²S compliant, but 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 equipment. Each of the three modules has individual clock signals and can be assigned to either of the two clock domains. Each transmitter has one set of clock lines including MCK, BICK and LRCK, and 2 or 4 data lines each communicating 2 channels of audio.

All 3 modules can transmit data at 192KHz sample rates. Since the highest sample rate that can be used internally (system) is 96KHz, when running at 192KHz rate the 8-channel module becomes a 4 channel interface. Due to the design of the 4-channel modules when running at 192KHz they will remain 4-channel interfaces, giving a total of 12 channels at 192KHz. 192KHz samples from the 4 channels are demultiplexed into consecutive pairs in the 96KHz transmitter buffers per the S-Mux standard. The diagram below illustrates the router channel configuration for 48/96KHz and 192KHz operation. 192KHz operation is selected by setting a bit in the I²S transmitter.

I2S Dataline	D0		D1		D2		D3	
I2S Channel	0	1	2	3	4	5	6	7
Audio Sequence	m	n	o	p	q	r	s	t

Table 5.23: I2S Tx 1, 96k router channel configuration
96k router channel configuration for I2S Transmitter 1 (8-channel)

I2S Dataline	D0				D1			
I2S Channel	0		1		2		3	
Audio Sequence	m	m+1	n	n+1	o	o+1	p	p+1

Table 5.24: I2S Tx 1, 192k router channel configuration)
192k router channel configuration for I2S Transmitter 1 (8-channel)

I2S Dataline	D0		D1		x		x	
I2S Channel	0	1	2	3	x	x	x	x
Audio Sequence	m	n	o	p	x	x	x	x

Table 5.25: I2S Tx 2 or 3, 96k router channel configuration
96k router channel configuration for I2S Transmitter 2 or 3 (4-channel interface)

I2S Dataline	D0				D1			
I2S Channel	0		1		2		3	
Audio Sequence	m	m+1	n	n+1	o	o+1	p	p+1

Table 5.26: I2S Tx 2 or 3, 192k router channel configuration
192k router channel configuration for I2S Transmitter 2 or 3 (4-channel interface)

Each of the three transmitters output one bit clock, one master clock and one left/right clock. All data lines from a given transmitter will be synchronized with each other and with the clock outputs regardless of which I2S channel the data is routed to. The following diagram illustrates the flow of data and clocking information through all three I2S transmitters. Note that data line D2 and D3 of I2S Tx0 will not be available when operating at high rates (192kHz).

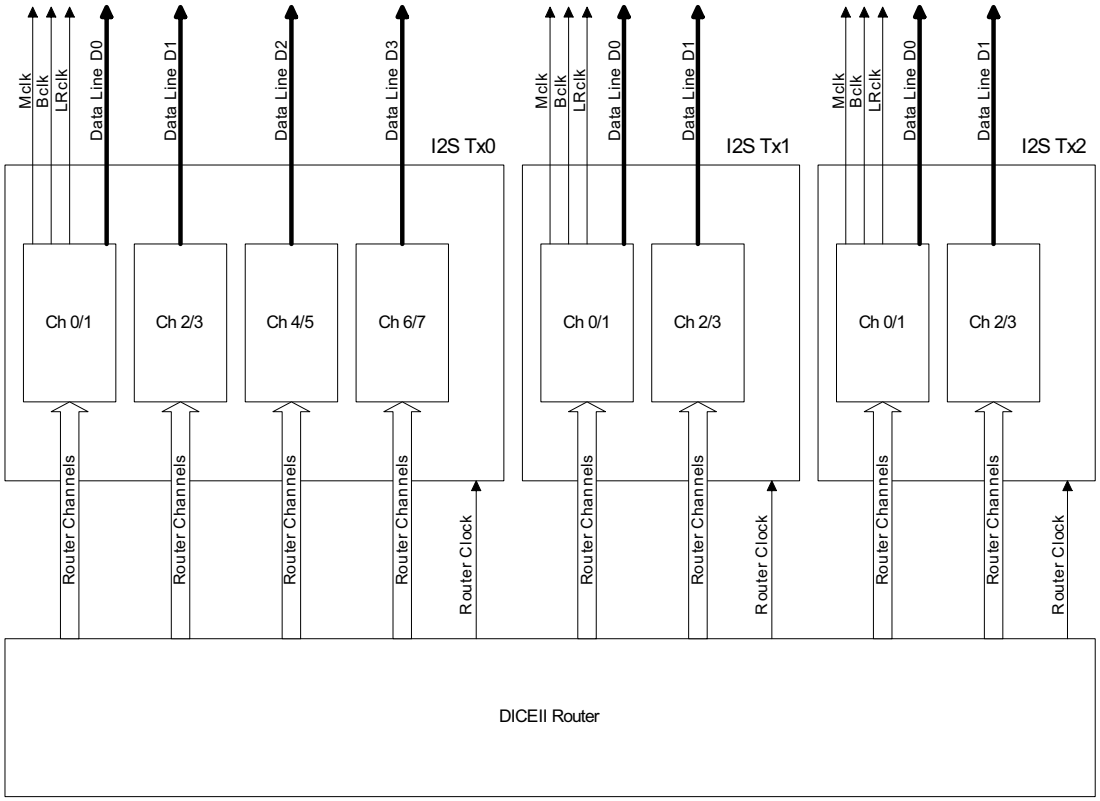


Figure 5.10: Flow of data and clock information for all three I2S transmitters
(note that D2 and D3 of Tx0 will not be available at 192kHz)

5.7.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
I2S_TX2_MCK	K19 (shared)	O	8	I2S Transmitter 2 Master Clock
I2S_TX2_BICK	K18 (shared)	O	8	I2S Transmitter 2 Bit Clock
I 2 S _ T X 2 _ LRCLK	K17 (shared)	O	8	I2S Transmitter 2 Left/Right Clock
I2S_TX2_D0	J20	O	2	I2S Transmitter 2 Data Ch.0/1
I2S_TX2_D1	J19	O	2	I2S Transmitter 2 Data Ch.2/3
I2S_TX1_MCK	G20	O	8	I2S Transmitter 1 Master Clock
I2S_TX1_BICK	G19	O	8	I2S Transmitter 1 Bit Clock
I2S_TX1_LRCK	F20	O	8	I2S Transmitter 1 Left/Right Clock
I2S_TX1_D0	G18	O	2	I2S Transmitter 1 Data Ch.0/1
I2S_TX1_D1	F19	O	2	I2S Transmitter 1 Data Ch.2/3
I2S_TX0_MCK	C20 (shared)	O	8	I2S Transmitter 0 Master Clock
I2S_TX0_BICK	E17 (shared)	O	8	I2S Transmitter 0 Bit Clock
I2S_TX0_LRCK	D18 (shared)	O	8	I2S Transmitter 0 Left/Right Clock
I2S_TX0_D0	C19	O	2	I2S Transmitter 0 Data Ch.0/1
I2S_TX0_D1	B20	O	2	I2S Transmitter 0 Data Ch.2/3
I2S_TX0_D2	C18	O	2	I2S Transmitter 0 Data Ch.4/5
I2S_TX0_D3	B19	O	2	I2S Transmitter 0 Data Ch.6/7

Table 5.27: I²S Transmitter Signal Description

5.7.2 MODULE CONFIGURATION

Address	Register
0xce11 0000	I2S_TX0_MODE
0xce11 0004	I2S_TX0_D0_CTRL
0xce11 0008	I2S_TX0_D1_CTRL
0xce11 000c	I2S_TX0_D2_CTRL
0xce11 0010	I2S_TX0_D3_CTRL
0xce11 0014	I2S_TX0_MUTE
0xce13 0000	I2S_TX1_MODE
0xce13 0004	I2S_TX1_D0_CTRL
0xce13 0008	I2S_TX1_D1_CTRL
0xce13 0014	I2S_TX1_MUTE
0xce15 0000	I2S_TX2_MODE
0xce15 0004	I2S_TX2_D0_CTRL
0xce15 0008	I2S_TX2_D1_CTRL
0xce15 0014	I2S_TX2_MUTE

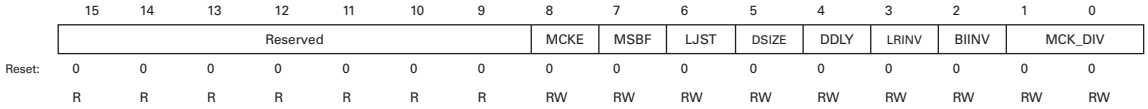
Table 5.28: I²S Transmitter Memory Map

5.7.3 I2S_TXN_MODE

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														M192	CKINFRQ
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
CKINFRQ	0	0	RW	This bit selects the frequency of the clock input to the I²S Receiver block from the clock controller. 0: 256fs (256 x the sampling frequency) 1: 512fs (512 x the sampling frequency) – for special use only
M192	1	0	RW	This bit selects the sample rate mode for the interface. 0: Normal mode, 8 (4) channels of base rate or 2 x base rate 1: Double mode, 4 (2) channels of 4 x base rate. Note: In Double mode the channels are packed in pairs at half the sample rate from the router point of view.

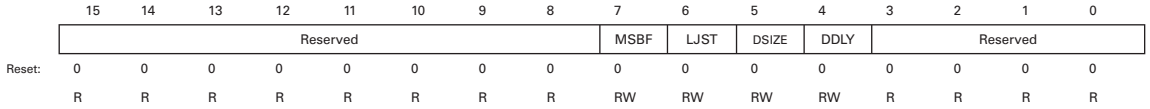
5.7.4 I2S_TXN_D0_CTRL



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved							MCKE	MSBF	LJST	DSIZE	DDLY	LRINV	BIINV	MCK_DIV	
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

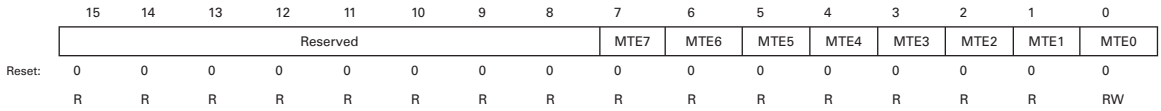
Name	Bit	Reset	Dir	Description
MCKE	8	0	RW	This bit enables the MCK signal. On some of the transmitters the MCK pin has alternate functions. (see GPCSR for details) 0: MCK disabled 1: MCK Enabled Note: This bit is only valid in the CH0 register. All channels in one transmitter share the clock pins.
MSBF	7	0	RW	This bit selects the ordering of the bits in the data frame. 0: LSB is sent first 1: MSB is sent first
LJST	6	0	RW	This bit selects the justification of the bits in the data frame when 24 bit data size is selected. 0: Right justified (padded MSB's 1: Left justified (padded LSB's)
DSIZE	5	0	RW	This bit selects the data size. 0: 24 bit data 1: 32 bit data
DDLY	4	0	RW	This bit selects delayed data to comply with I²S. 0: No Delay, first bit aligned with LRCK edge. 1: One bit delay after LRCK edge.
LRINV	3	0	RW	This bit selects the phase of the LRCK pin. On some of the transmitters the LRCK pin has alternate functions. (see GPCSR for details) 0: LRCK is high for left channel data, low for right channel data. 1: LRCK is low for left channel data, high for right channel data. Note: This bit is only valid in the D0 register. All channels in one transmitter share the clock pins.
BIINV	2	0	RW	This bit selects the phase of the BICK pin. On some of the transmitters the LRCK pin has alternate functions. (see GPCSR for details) 0: The frame starts with a negative edge. 1: The frame starts with a positive edge. Note: This bit is only valid in the D0 register. All channels in one transmitter share the clock pins.
MCK_DIV	1:0	0	RW	This bit selects the frequency of the MCK pin. . On some of the transmitters the LRCK pin has alternate functions. (see GPCSR for details) 00: 256fs (128fs in Double mode) 01: 128fs (64fs in Double mode) 10: 64fs (32 fs in Double mode) 11: Reserved Note: This bit is only valid in the D0 register. All channels in one transmitter share the clock pins.

5.7.5 I2S_TXN_D1/D2/D3_CTRL



Name	Bit	Reset	Dir	Description
MSBF	7	0	RW	This bit selects the ordering of the bits in the data frame. 0: LSB is sent first 1: MSB is sent first
LJST	6	0	RW	This bit selects the justification of the bits in the data frame when 24 bit data size is selected. 0: Left justified (padded LSB's) 1: Right justified (padded MSB's)
DSIZE	5	0	RW	This bit selects the data size. 0: 24 bit data 1: 32 bit data
DDLY	4	0	RW	This bit selects delayed data to comply with I²S. 0: No Delay, first bit aligned with LRCK edge. 1: One bit delay after LRCK edge.

5.7.6 I2S_TXN_MUTE



Name	Bit	Reset	Dir	Description
MTEx	7:0	0	RW	This field controls the muting of individual channels in the transmitter. For Tx2 and Tx3 only the first 4 bits are valid. 0: Not muted 1: Muted Note: In Double mode the channels should be muted in pairs.

5.8 ADAT RECEIVER

The DICE II contains one Alesis ADAT compatible receiver. The receiver can receive 8 channels of audio at the base rates and 4 channels of audio at twice the base rate using the S-MUX scheme.

When operating in S-Mux mode the router interface becomes a 4 channel interface, reading 4 channels of audio every 96KHz sample period. Data is demultiplexed into the 96KHz system buffers from a single ADAT lightpipe as follows:

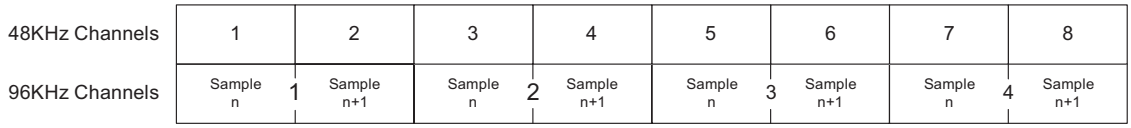


Figure 5.11: S-Mux ADAT lightpipe channel configuration

5.8.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
OPTI	W14	I	-	Optical Audio In (5V)

Table 5.29: ADAT Receiver Signal Description

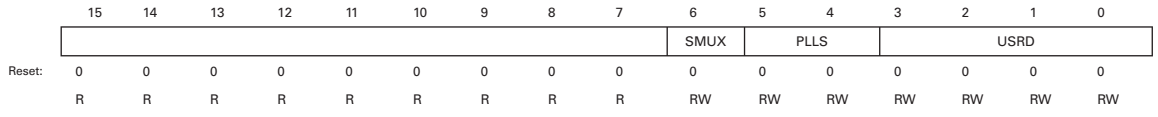
5.8.2 MODULE CONFIGURATION

Address	Register
0xce04 0000	ADATRX_CTRL
0xce04 0004	ADATRX_STAT

Table 5.30: ADAT Receiver Memory Map

5.8.3 ADATRX_CTRL

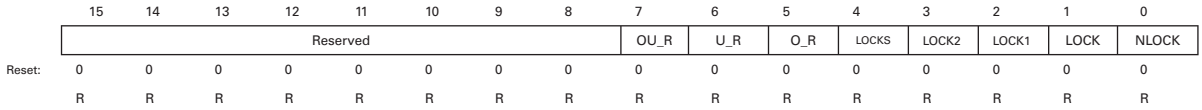
0xce04 0000



Name	Bit	Reset	Dir	Description
SMUX	6	0	RW	Writing a '1' to this location enables the S-MUX scheme. It is assumed that the router assigned to the ADAT interface is running 2 x base rate when S-MUX is enabled.
PLLS	5:4	0	RW	Define the number of consecutive sync patterns should be received before asserting LOCKS.
USRD	3:0	0	RW	The 4 bits of user data received in the last frame.

5.8.4 ADATRX_STAT

0xce04 0004



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								OU_R	U_R	O_R	LOCKS	LOCK2	LOCK1	LOCK	NLOCK
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 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_R	5	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.
LOCKS	4	0	R	Asserted when a consecutive number of sync patterns defined by PLLS has been detected.
LOCK2	3	0	R	One consecutive sync pattern received
LOCK1	2	0	R	Two consecutive sync patterns received
LOCK	1	0	R	Three consecutive sync patterns received
NLOCK	0	0	R	An unlock condition was detected since last read. This bit is sticky and will be cleared immediately after a read.

5.9 ADAT TRANSMITTER

The DICE II chip contains one Alesis ADAT compatible transmitter. The transmitter can transmit 8 channels of audio at the base rates. The transmitter can also handle 4 channels of audio at twice the base rate using the S-MUX scheme.

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:

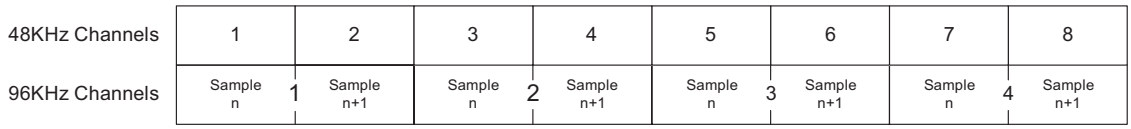


Figure 5.12: S-Mux ADAT lightpipe channel configuration

5.9.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
OPTO	Y14	O	8	Optical Audio Out

Table 5.31: ADAT Transmitter Signal Description

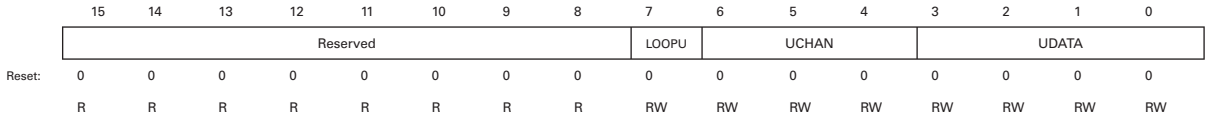
5.9.2 MODULE CONFIGURATION

Address	Register
0xce05 0000	ADATTX_CTRL1
0xce05 0004	ADATTX_CTRL2
0xce05 0008	ADATTX_CTRL3

Table 5.32: ADAT Transmitter Memory Map

5.9.3 ADATTX_CTRL1

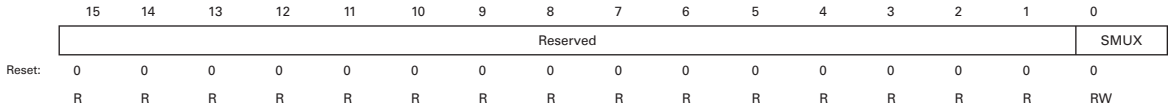
0xce05 0000



Name	Bit	Reset	Dir	Description
LOOPU	7	0	RW	Enables looping of user data from the upper 4 bits of the 32 bit audio word from the channel defined by UCHAN. 0: No Loop (use UDATA) 1: Loop (user data with audio)
UCHAN	6:4	0	RW	In loop mode this field defines which audio channel carries the user bits.
UDATA	3:0	0	RW	Used in non-loop mode to specify static user data.

5.9.4 ADATTX_CTRL2

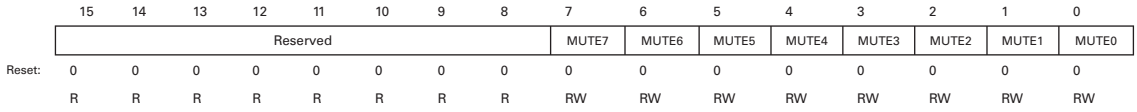
0xce05 0004



Name	Bit	Reset	Dir	Description
SMUX	0	0	RW	Writing a '1' to this location enables the S-MUX scheme. It is assumed that the router assigned to the ADAT interface is running 2 x base rate when S-MUX is enabled.

5.9.5 ADATTX_CTRL3

0xce05 0008



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 TDIF TRANSMITTER

The DICE II contains one TDIF-1 compliant transmitter. The TDIF transmitter includes 4 transmit data lines, as per the TDIF-1 specification. It also includes a Left/Right Clock output, fs0/fs1 outputs, and an emphasis output. Note that each TDIF signal includes two audio channels (left and right), for a total of 8 output channels.

5.10.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
TDF_O0	Y9 (shared)	O	6	TDIF audio data output 1
TDF_O1	W10 (shared)	O	6	TDIF audio data output 2
TDF_O2	V10 (shared)	O	6	TDIF audio data output 3
TDF_O3	Y10 (shared)	O	6	TDIF audio data output 4
TDF_OLR	Y11 (shared)	O	6	TDIF left right clock output
TDF_OFS0	W11 (shared)	O	6	TDIF sample rate 0 output
TDF_OFS1	V11 (shared)	O	6	TDIF sample rate 1 output
TDF_OEM	U11 (shared)	O	6	TDIF emphasis output

Table 5.33: TDIF Transmitter Signal Description

Note that all pins used by the TDIF module are multi-purpose or shared. The function of these pins is software configurable via the GPCSR module, specifically register GPCSR_VIDEO_SELECT – 0xc700 0010. Refer to the GPCSR module documentation for more information.

To set the shared pins to function as TDIF pins, the register mentioned above should be set as shown below.

GPCSR_VIDEO_SELECT 0xc700 0010 = 0x0009 5555

Note that this register only configures the TDIF output pins. For inputs, the pin will contain whatever signal has been connected to it.

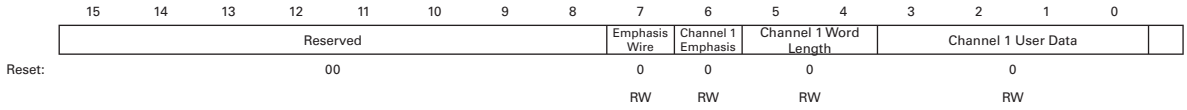
5.10.2 MODULE CONFIGURATION

Address	Register
0xce07 0000	TDIF_TX_EMPH_CH0_CFG
0xce07 0004	TDIF_TX_ FS0_CH1_CFG
0xce07 0008	TDIF_TX_FS1_CH2_CFG
0xce07 000c	TDIF_TX_CH3_CFG
0xce07 0010	TDIF_TX_CH4_CFG
0xce07 0014	TDIF_TX_CH5_CFG
0xce07 0018	TDIF_TX_CH6_CFG
0xce07 001c	TDIF_TX_CH7_CFG
0xce07 0020	TDIF_TX_MUTE
0xce07 0024	TDIF_TX_INV_CTRL

Table 5.34: TDIF Transmitter Memory Map

5.10.3 TDIF_TX_CH0_CFG_EMPH

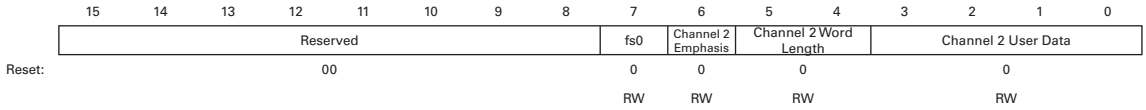
0xce07 0000



Name	Bit	Reset	Dir	Description
Emphasis Wire	7	0	RW	Memory mapping of TDIF interface emphasis wire. Applies common emphasis sub data information. 0 – on, 1 – off.
Channel 1 Emphasis	6	0	RW	TDIF Channel 1 emphasis information bit. 0 – off, 1 – on.
Channel 1 Word Length	5:4	0	RW	TDIF Channel 1 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 1 User Data	3:0	0	RW	TDIF Channel 1 user data information.

5.10.4 TDIF_TX_FS0_CH1_CFG

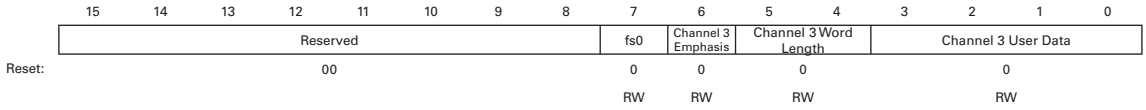
0xce07 0004



Name	Bit	Reset	Dir	Description
fs0	7	0	RW	Memory mapping of TDIF interface fs0 wire. Indicates sample frequency applied by TDIF transmitter. fs1 fs0 indicates the following sample rates. 00 – 48kHz, 01 – 44kHz, 10 – 32kHz, 11 – other.
Channel 2 Emphasis	6	0	RW	TDIF Channel 2 emphasis information bit. 0 – off, 1 – on.
Channel 2 Word Length	5:4	0	RW	TDIF Channel 2 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 2 User Data	3:0	0	RW	TDIF Channel 2 user data information.

5.10.5 TDIF_TX_ FS1_CH2_CFG

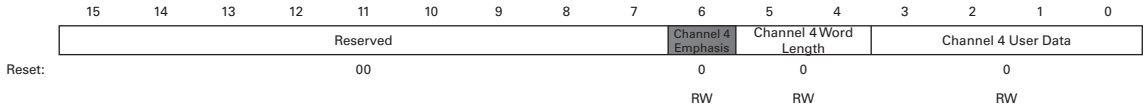
0xce07 0008



Name	Bit	Reset	Dir	Description
fs1	7	0	RW	Memory mapping of TDIF interface fs1 wire. Indicates sample frequency applied by TDIF transmitter. fs1 fs0 indicates the following sample rates. 00 – 48kHz, 01 – 44kHz, 10 – 32kHz, 11 – other.
Channel 3 Emphasis	6	0	RW	TDIF Channel 3 emphasis information bit. 0 – off, 1 – on.
Channel 3 Word Length	5:4	0	RW	TDIF Channel 3 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 3 User Data	3:0	0	RW	TDIF Channel 3 user data information.

5.10.6 TDIF_TX_CH3_CFG

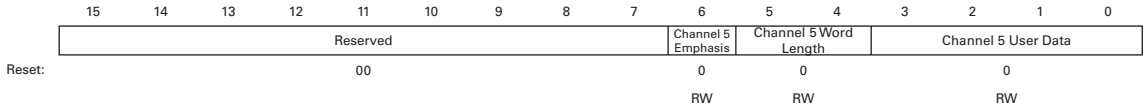
0xce07 000c



Name	Bit	Reset	Dir	Description
Channel 4 Emphasis	6	0	RW	TDIF Channel 4 emphasis information bit. 0 – off, 1 – on.
Channel 4 Word Length	5:4	0	RW	TDIF Channel 4 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 4 User Data	3:0	0	RW	TDIF Channel 4 user data information.

5.10.7 TDIF_TX_CH4_CFG

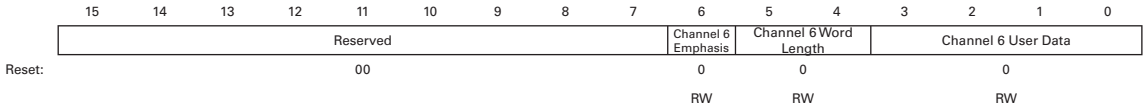
0xce07 00010



Name	Bit	Reset	Dir	Description
Channel 5 Emphasis	6	0	RW	TDIF Channel 5 emphasis information bit. 0 – off, 1 – on.
Channel 5 Word Length	5:4	0	RW	TDIF Channel 5 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 5 User Data	3:0	0	RW	TDIF Channel 5 user data information.

5.10.8 TDIF_TX_CH5_CFG

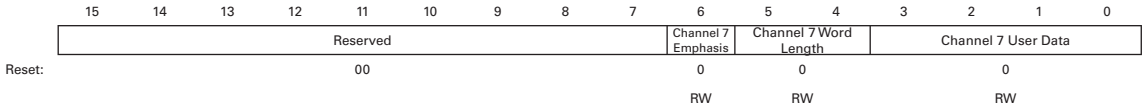
0xce07 0014



Name	Bit	Reset	Dir	Description
Channel 6 Emphasis	6	0	RW	TDIF Channel 6 emphasis information bit. 0 – off, 1 – on.
Channel 6 Word Length	5:4	0	RW	TDIF Channel 6 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 6 User Data	3:0	0	RW	TDIF Channel 6 user data information.

5.10.9 TDIF_TX_CH6_CFG

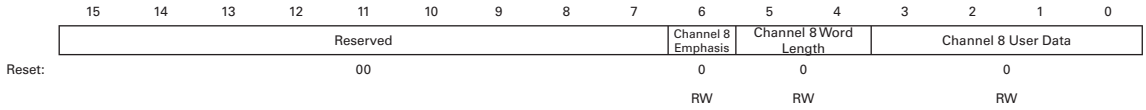
0xce07 0018



Name	Bit	Reset	Dir	Description
Channel 7 Emphasis	6	0	RW	TDIF Channel 7 emphasis information bit. 0 – off, 1 – on.
Channel 7 Word Length	5:4	0	RW	TDIF Channel 7 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 7 User Data	3:0	0	RW	TDIF Channel 7 user data information.

5.10.10 TDIF_TX_CH7_CFG

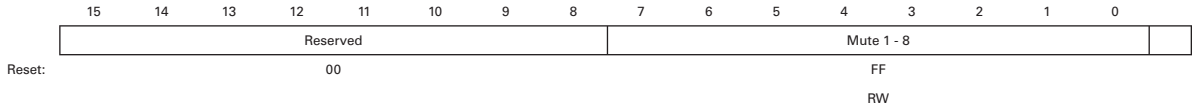
0xce07 001c



Name	Bit	Reset	Dir	Description
Channel 8 Emphasis	6	0	RW	TDIF Channel 8 emphasis information bit. 0 – off, 1 – on.
Channel 8 Word Length	5:4	0	RW	TDIF Channel 8 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Channel 8 User Data	3:0	0	RW	TDIF Channel 8 user data information.

5.10.11 TDIF_TX_MUTE

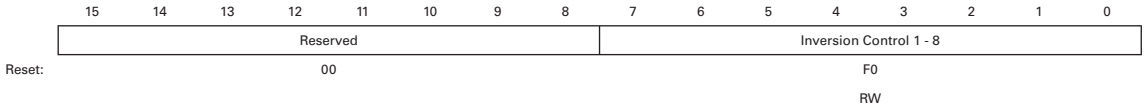
0xce07 0020



Name	Bit	Reset	Dir	Description
Mute 1 - 8	7:0	FF	RW	Each bit associated with one of the 8 TDIF transmitter channels. If a given bit is set, the associated channel will be muted. 0 – off, 1 – on

5.10.12 TDIF_TX_INV_CTRL

0xce07 0024



Name	Bit	Reset	Dir	Description
Inversion Control 1 - 8	7:0	F0	RW	Makes the TDIF transmitter independent of variants of the TDIF-1 description, by allowing users to invert the 8 interface signals individually. Register entries are considered active high (1 - signal inversion is conducted). Each bit represents one of the 8 interface signals as follows: 0: TDIF Tx 1 1: TDIF Tx 2 2: TDIF Tx 3 3: TDIF Tx 4 4: LR Clock Tx 5: fs0 Tx 6: fs1 Tx 7: Emphasis Tx

5.11 TDIF RECEIVER

The DICE II contains one TDIF-1 compliant receiver. The TDIF receiver includes 4 receive data lines, as per the TDIF-1 specification. It also includes a Left/Right Clock input, fs0/fs1 inputs, and an emphasis input. Note that each TDIF signal includes two audio channels (left and right), for a total of 8 input channels.

5.11.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
TDF_I0	W7 (shared)	I (S)	6	TDIF audio data input 1 (5V)
TDF_I1	Y7 (shared)	I (S)	6	TDIF audio data input 2 (5V)
TDF_I2	V8 (shared)	I (S)	6	TDIF audio data input 3 (5V)
TDF_I3	W8 (shared)	I (S)	6	TDIF audio data input 4 (5V)
TDF_ILR	Y8 (shared)	I (S)	6	TDIF left right clock input (5V)
TDF_IFS0	U9 (shared)	I (S)	6	TDIF sample rate 0 input (5V)
TDF_IFS1	V9 (shared)	I (S)	6	TDIF sample rate 1 input (5V)
TDF_IEM	W9 (shared)	I (S)	6	TDIF emphasis input (5V)

Table 5.35: TDIF Receiver Signal Description

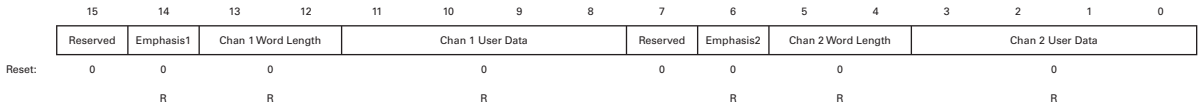
5.11.2 MODULE CONFIGURATION

Address	Register
0xce06 0000	TDIF_RX_CH0/1_CFG
0xce06 0004	TDIF_RX_CH2/3_CFG
0xce06 0008	TDIF_RX_CH4/5_CFG
0xce06 000c	TDIF_RX_CH6/7_CFG
0xce06 0010	TDIF_RX_STAT
0xce06 0014	TDIF_RX_CFG
0xce06 0018	TDIF_RX_PHASE_DIFF
0xce06 001c	TDIF_RX_INV_CTRL

Table 5.36: TDIF Receiver Memory Map

5.11.3 TDIF_RX_CH0/1_CFG

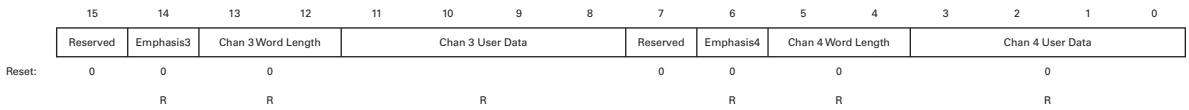
0xce06 0000



Name	Bit	Reset	Dir	Description
Chan1 Emphasis	14	0	R	TDIF Channel 1 emphasis information bit. 0 – off, 1 – on.
Chan1 Word Length	13:12	0	R	TDIF Channel 1 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan1 User Data	11:8	0	R	TDIF Channel 1 user data information.
Chan2 Emphasis	6	0	R	TDIF Channel 2 emphasis information bit. 0 – off, 1 – on.
Chan2 Word Length	5:4	0	R	TDIF Channel 2 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan2 User Data	3:0	0	R	TDIF Channel 2 user data information.

5.11.4 TDIF_RX_CH2/3_CFG

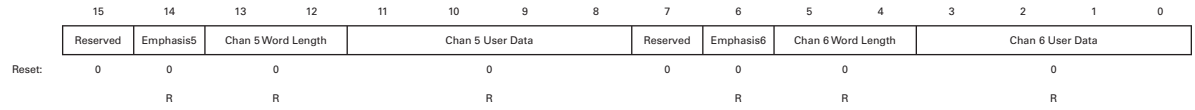
0xce06 0004



Name	Bit	Reset	Dir	Description
Chan3 Emphasis	14	0	R	TDIF Channel 3 emphasis information bit. 0 – off, 1 – on.
Chan3 Word Length	13:12	0	R	TDIF Channel 3 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan3 User Data	11:8	0	R	TDIF Channel 3 user data information.
Chan4 Emphasis	6	0	R	TDIF Channel 4 emphasis information bit. 0 – off, 1 – on.
Chan4 Word Length	5:4	0	R	TDIF Channel 4 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan4 User Data	3:0	0	R	TDIF Channel 4 user data information.

5.11.5 TDIF_RX_CH4/5_CFG

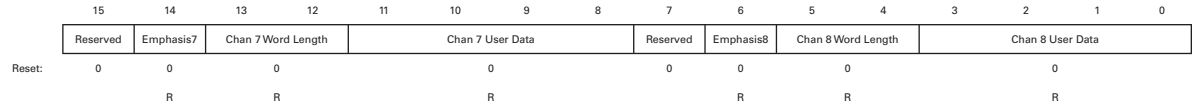
0xce06 0008



Name	Bit	Reset	Dir	Description
Chan5 Emphasis	14	0	R	TDIF Channel 5 emphasis information bit. 0 – off, 1 – on.
Chan5 Word Length	13:12	0	R	TDIF Channel 5 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan5 User Data	11:8	0	R	TDIF Channel 5 user data information.
Chan6 Emphasis	6	0	R	TDIF Channel 6 emphasis information bit. 0 – off, 1 – on.
Chan6 Word Length	5:4	0	R	TDIF Channel 6 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan6 User Data	3:0	0	R	TDIF Channel 6 user data information.

5.11.6 TDIF_RX_CH6/7_CFG

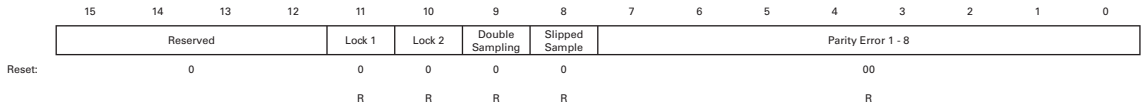
0xce06 000c



Name	Bit	Reset	Dir	Description
Chan7 Emphasis	14	0	R	TDIF Channel 7 emphasis information bit. 0 – off, 1 – on.
Chan7 Word Length	13:12	0	R	TDIF Channel 7 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan7 User Data	11:8	0	R	TDIF Channel 7 user data information.
Chan8 Emphasis	6	0	R	TDIF Channel 8 emphasis information bit. 0 – off, 1 – on.
Chan8 Word Length	5:4	0	R	TDIF Channel 8 word length information field. 00 – 16bits, 01 – 20bits, 10 – 24bits, 11 – reserved.
Chan8 User Data	3:0	0	R	TDIF Channel 8 user data information.

5.11.7 TDIF_RX_STAT

0xce06 0010

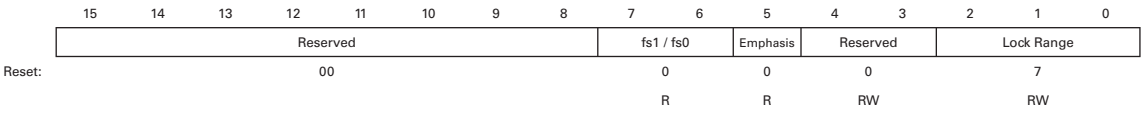


Name	Bit	Reset	Dir	Description
Lock 1	11	0	R	Status bit from Lock Detector signaling whether regenerated left/right clock (pll_lrck) is in lock with TDIF left right clock (lrck_in). 0 – no lock, 1 – lock.
Lock 2 *	10	0	R	Sticky representation of Lock 1. Lock 2 keeps an out of lock state until the bit has been accessed by the APB bus.
Double Sampling *	9	0	R	Audio sample data double sampling error flag. 0 – no error, 1 – error.
Slipped Sample *	8	0	R	Audio sample data slip error flag. 0 – no error, 1 – error.
Parity Error 1-8 *	7:0	0	R	TDIF channel 1 – 8 parity error array. 0 – no error, 1 – error.

* Sticky

5.11.8 TDIF_RX_CFG

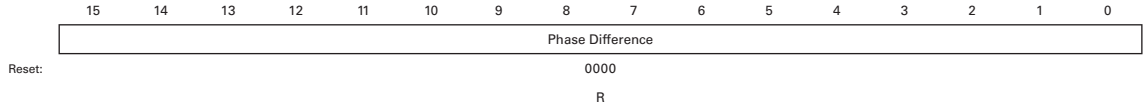
0xce06 0014



Name	Bit	Reset	Dir	Description
fs1 / fs0	7:6	0	R	Memory mapping of TDIF interface fs1 wire and fs0 wire. Indicates sample frequency applied by TDIF transmitter. 00 – 48kHz, 01 – 44kHz, 10 – 32kHz, 11 – other.
Emphasis	5	0	R	Memory mapping of TDIF interface emphasis wire. Applies common emphasis sub data information. 0 – on, 1 – off.
Lock Range	2:0	7	RW	Setup for Lock Detector, providing a maximum allowable signal skew to signal lock within. Interpreted as the number of Router clock cycles (~20ns).

5.11.9 TDIF_RX_PHASE_DIFF

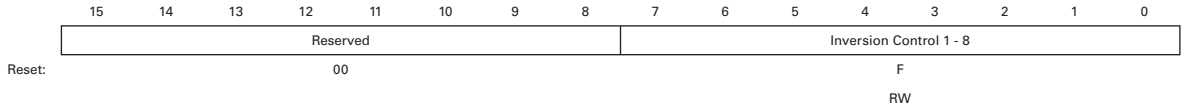
0xce06 0018



Name	Bit	Reset	Dir	Description
Phase Difference	15:0	0	R	Phase difference between sys_1fs and tdif_1fs. The contents read is a sign extended 16bit number (2's complement), indicating how the time tdif_1fs lags sys_1fs in terms of Router clock cycles (~20ns).

5.11.10 TDIF_RX_INV_CTRL

0xce06 001c



Name	Bit	Reset	Dir	Description
Inversion Control 1-8	7:0	F	RW	Makes the TDIF transmitter independent of variants of the TDIF-1 description, by allowing users to invert the 8 interface signals individually. Register entries are considered active high (1 - signal inversion is conducted). Each bit represents one of the 8 interface signals as follows: 0: TDIF Rx 1 1: TDIF Rx 2 2: TDIF Rx 3 3: TDIF Rx 4 4: LR Clock Rx 5: fs0 Rx 6: fs1 Rx 7: Emphasis Rx

5.12 DSAI RECEIVERS

The DICE II chip contains 4 Digital Serial Audio Interface (TDM/I^{8}S) receivers. Each receiver has a clock and sync generator which is independent of the receiver itself. The DICE II has 4 full sets of clock and sync pins which can be programmed as either outputs or inputs, as shown in the diagram below. When programmed as outputs the clock and sync pins can source from any of the 4 receiver clock generators, and any of the 4 transmitter clock generators.

Each receiver can use the clocks from any of the for clock sync pairs named A to D. Please refer to the GPCSR_CTRL section (4.1) for details on routing the clock signals. The diagram illustrates options for DSAI Receiver 0 and clock sync pair D.

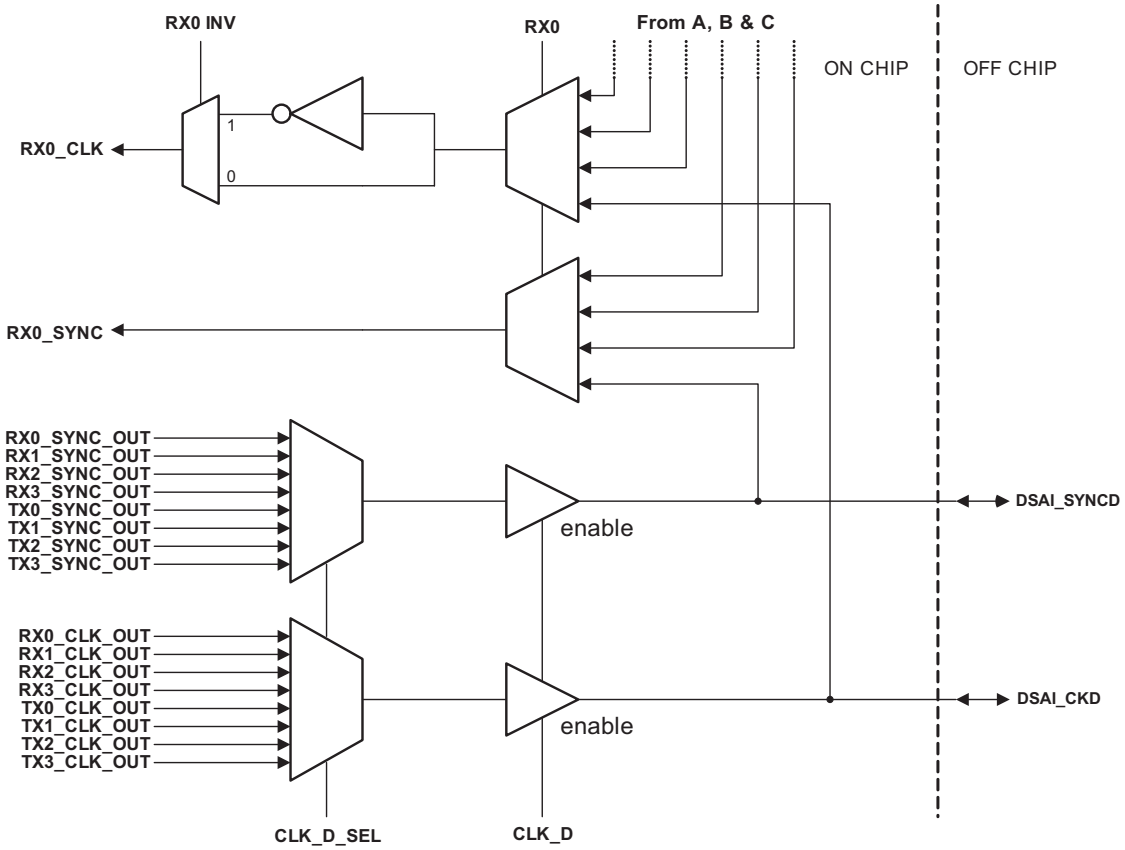


Figure 5.13: DSAI Receiver clock and sync selection.

5.12.1 PIN DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
DSAI_RX0	Y18	I	-	DSAI Receiver 0 data line (5V)
DSAI_RX1	U16	I	-	DSAI Receiver 1 data line (5V)
DSAI_RX2	V17	I	-	DSAI Receiver 2 data line (5V)
DSAI_RX3	W18	I	-	DSAI Receiver 3 data line (5V)
DSAI_CKA	Y19	I/O (S)	8	DSAI Clock A
DSAI_SYNCA	V18	I/O (S)	8	DSAI Sync A
DSAI_CKB	W19	I/O (S)	8	DSAI clock B
DSAI_SYNCB	Y20	I/O (S)	8	DSAI Sync B
DSAI_CKC	W20	I/O (S)	8	DSAI Clock C
DSAI_SYNCC	V19	I/O (S)	8	DSAI Sync C
DSAI_CKD	U19	I/O (S)	8	DSAI Clock D
DSAI_SYNCD	U18	I/O (S)	8	DSAI Sync D

Table 5.37: DSAI Recevier Signal Description

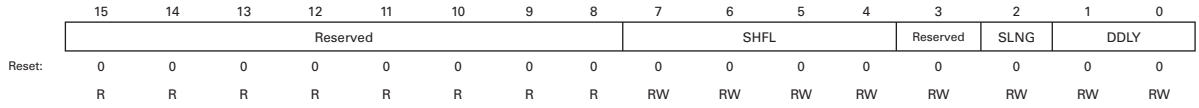
5.12.2 MODULE CONFIGURATION

Address	Register
0xce08 0000	DSAI1_RX_CTRL
0xce08 0008	DSAI1_RX_STAT
0xce0a 0000	DSAI2_RX_CTRL
0xce0a 0008	DSAI2_RX_STAT
0xce0c 0000	DSAI3_ RX_CTRL
0xce0c 0008	DSAI3_ RX_STAT
0xce0e 0000	DSAI4_ RX_CTRL
0xce0e 0008	DSAI4_ RX_STAT

Table 5.38: DSAI Receiver Memory Map

5.12.3 DSAIN_RX_CTRL

0xce08 0000



Name	Bit	Reset	Dir	Description
SHFL	7:4	0	RW	Defines the order and alignment of the data bits for this receiver. Refer to the table below for the definition of the different shuffle modes.
SLNG	2	0	RW	Selects the length of the sync pulse. 0: 32 bit clocks 1: 1 bit clock
DDLY	1:0	0	RW	Selects the delay of the data in relation to the sync pulse. 00: Data starts 0 clocks after Sync. 01: Data starts 1 clock after Sync. 10: Data starts 2 clocks after Sync. 11: Data starts 3 clocks after Sync.

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 5.39: DSAI Rx Data Shuffle

5.12.4 DSAIN_RX_STAT

0xce08 0008

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														SLIP	
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
SLIP	1:0	0	RW	These two read only sticky bits indicates underrun and overrun conditions when running the receiver in slave mode. Bit 1: Slipped sample Bit 0: Repeat sample

5.12.5 DSAI RX TIMING

5.12.5.1 DSAI RX TIMING IN SLAVE MODE

In this mode DSAI RX block inside the chip gets clocks and sync signal from the outside world. Chosen DSAI clock for a particular DSAI engine should be configured as an input in GPCSR_DSAI_SELECT register. In addition, clock used by DSAI receivers could be inverted. The following figures illustrate the timing relationship between DSAI clocks, data and sync signals in case of both rising and falling edge of the clocks being used.

Conditions:
Commercial range (0-70 °C ambient, $VDD_{1.8V}$=1.8V±0.15V, $VDD_{3.3V}$=3.3±0.3V)
Output load C_{LOAD} = 30 pF

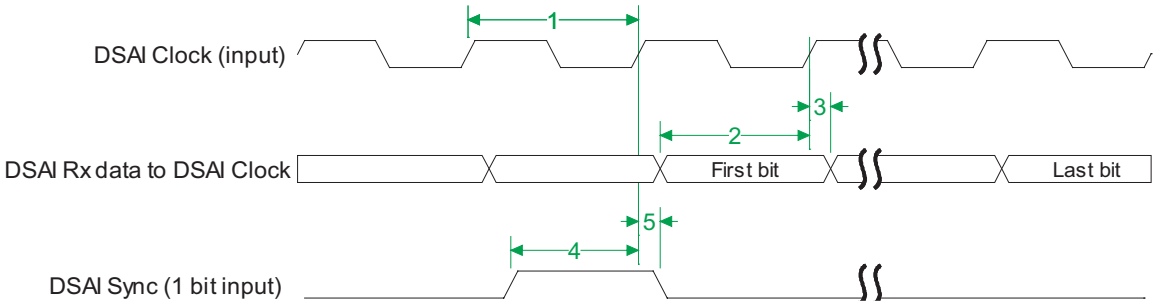


Figure 5.14: DSAI Rx Slave Timing when using rising edge of clock (dcka/b/c/d)

No.	Name	Min	Max	Comment
1	DSAI Clock cycle time	39		DSAI clock cycle time (256FS)
2	DSAI Rx data to DSAI Clock setup	4		DSAI data required setup time before DSAI rising clock
3	DSAI Rx data to DSAI clock hold	0		DSAI data required hold time after DSAI rising clock
4	DSAI Sync to DSAI clock setup	5		DSAI sync required setup time before DSAI rising clock
5	DSAI Sync to DSAI Clock hold	0		DSAI sync required hold time after DSAI rising clock

Table 5.40: DSAI Rx Timing to the rising edge of DSAI clock (dcka/b/c/d)

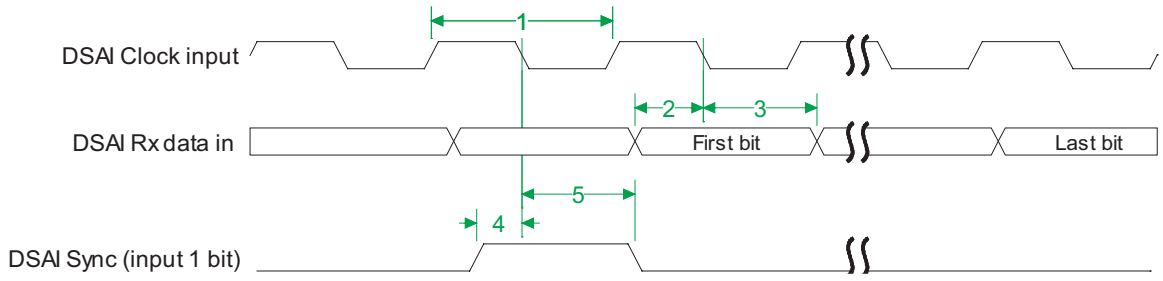


Figure 5.15: DSAI Rx Slave Timing when using falling edge of clock (dck*)

No.	Name	Min	Max	Comment
1	DSAI Clock cycle time	39		DSAI clock cycle time (256FS)
2	DSAI Rx Data setup to DSAI Clock	4		DSAI data required setup time before DSAI falling clock
3	DSAI Rx data hold time	0		DSAI data required hold time after DSAI falling clock
4	DSAI Sync setup time	5		DSAI sync required setup time before DSAI falling clock
5	DSAI Sync hold time	0		DSAI sync required hold time after DSAI falling clock

Table 5.41: DSAI Rx Slave Timing (falling edge of dck*)

5.12.5.2 DSAI RX TIMING IN MASTER MODE

In this mode DSAI block inside the chip gets clocks and sync signal from the DSAI clock generators and in this case DSAI clocks would be synchronized to one of the Routers. Chosen DSAI clock and sync for a particular DSAI engine should be configured as an output in GPCSR_DSAI_SELECT register. In addition, clock used by DSAI receivers could be inverted. Note, that DSAI Sync is generated of the positive edge of the clock anyway. The following figures illustrate the timing relationship between DSAI clocks, data and sync signals in case of both rising and falling edge of the clocks being used.

Conditions:
Commercial range (0-70 °C ambient, $VDD_{1.8V}$=1.8V±0.15V, $VDD_{3.3V}$=3.3±0.3V)
Output load C_{LOAD} = 30 pF

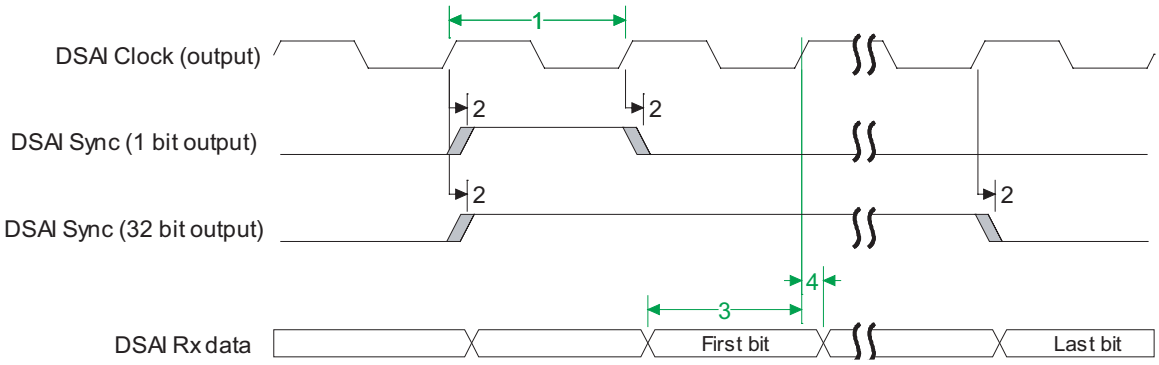


Figure 5.16: DSAI Rx Master Timing when using rising edge of clock (dcka/b/c/d)

No.	Name	Min	Max	Comment
1	DSAI Clock cycle time	39		DSAI clock cycle time (256FS)
2	DSAI Clock to DSAI Sync out delay	1	4	DSAI rising clock to DSAI sync out delay
3	DSAI Rx data setup to DSAI	4		DSAI data required setup time before DSAI rising clock
4	DSAI Rx data hold to DSAI Clock	0		DSAI data required hold time after DSAI rising clock

Table 5.42: DSAI Rx Master Timing to the rising edge of DSAI clock (dcka/b/c/d)

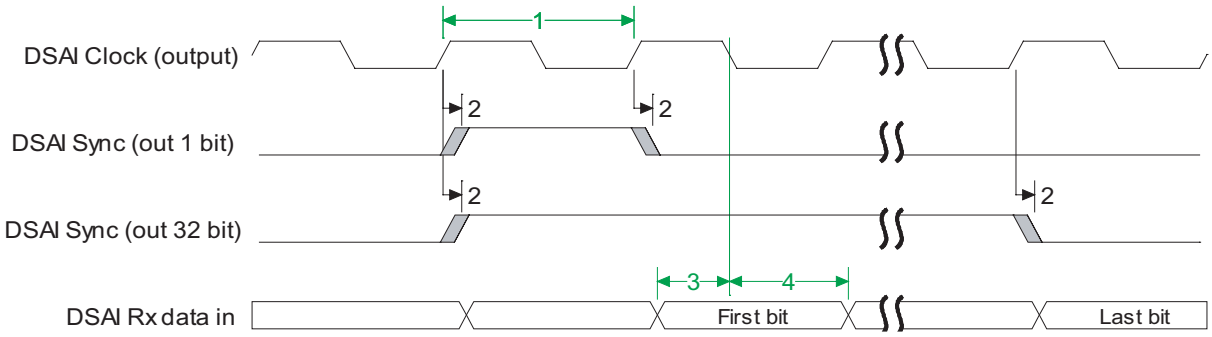


Figure 5.17: DSAI Rx Master Timing when using falling edge of clock (dcka/b/c/d)

No.	Name	Min	Max	Comment
1	DSAI Clock cycle time	39		DSAI clock cycle time (256FS)
2	DSAI Sync Out to DSAI Clock	1	4	DSAI rising clock to DSAI sync out delay (from rising edge)
3	DSAI Rx data setup	4		DSAI data required setup time before DSAI falling clock
4	DSAI Rx data hold	0		DSAI data required hold time after DSAI falling clock

Table 5.43: DSAI Rx Master Timing to the falling edge of DSAI clock (dcka/b/c/d)

5.13 DSAI TRANSMITTERS

The DICE II chip contains 4 Digital Serial Audio Interface (TDM/I^{8}S) transmitters. Each transmitter has a clock and sync generator which is independent of the transmitter itself. The DICE II has 4 full sets of clock and sync pins which can be programmed as either outputs or inputs, as shown in the diagram below. When programmed as outputs these clock and sync pins can source from any of the 4 receiver clock generators, and any of the 4 Transmitter clock generators.

Each Transmitter can use the clocks from any of the for clock sync pairs named A to D. Please refer to the GPCSR_CTRL section for details on routing the clock signals. The diagram illustrates options for DSAI Transmitter 0 and clock sync pair D.

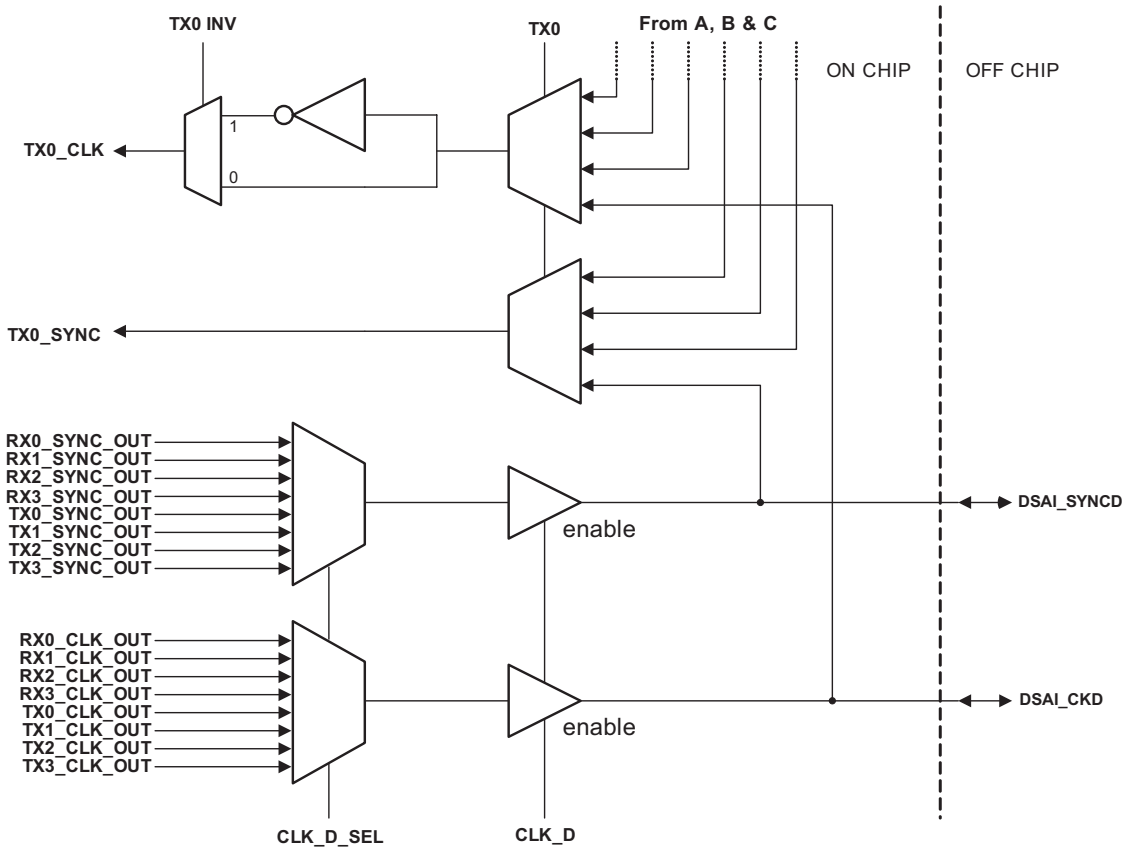


Figure 5.18: DSAI Transmitter clock and sync selection.

5.13.1 SIGNAL DESCRIPTION

Signal	PBGA Pin	I/O	Drive (mA)	Description
DSAI_CKA	Y19	I/O (S)	8	DSAI Clock A
DSAI_SYNCA	V18	I/O (S)	8	DSAI Sync A
DSAI_CKB	W19	I/O (S)	8	DSAI clock B
DSAI_SYNCB	Y20	I/O (S)	8	DSAI Sync B
DSAI_CKC	W20	I/O (S)	8	DSAI Clock C
DSAI_SYNCC	V19	I/O (S)	8	DSAI Sync C
DSAI_CKD	U19	I/O (S)	8	DSAI Clock D
DSAI_SYNCD	U18	I/O (S)	8	DSAI Sync D
DSAI_TX0	T17	O	4	DSAI Transmitter 0 data line
DSAI_TX1	V20	O	4	DSAI Transmitter 1 data line
DSAI_TX2	U20	O	4	DSAI Transmitter 2 data line
DSAI_TX3	T18	O	4	DSAI Transmitter 3 data line

Table 5.44: DSAI Transmitter Signal Description

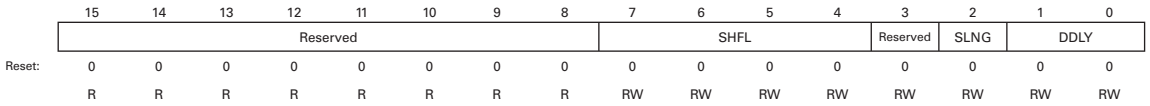
5.13.2 MODULE CONFIGURATION

Address	Register
0xce09 0000	DSAI1_TX_CTRL
0xce09 0008	DSAI1_TX_STAT
0xce09 000c	DSAI1_TX_MUTE
0xce0b 0000	DSAI2_TX_CTRL
0xce0b 0008	DSAI2_TX_STAT
0xce0b 000c	DSAI2_TX_MUTE
0xce0d 0000	DSAI3_TX_CTRL
0xce0d 0008	DSAI3_TX_STAT
0xce0d 000c	DSAI3_TX_MUTE
0xce0f 0000	DSAI4_TX_CTRL
0xce0f 0008	DSAI4_TX_STAT
0xce0f 000c	DSAI4_TX_MUTE

Table 5.45: DSAI Transmitter Memory Map

5.13.3 DSAIN_TX_CTRL

0xce09 0000



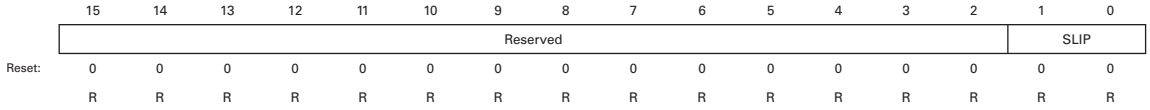
Name	Bit	Reset	Dir	Description
SHFL	7:4	0	RW	Defines the order and alignment of the data bits for this transmitter. Refer to the table below for the definition of the different shuffle modes.
SLNG	2	0	RW	Selects the length of the sync pulse. 0: 32 bit clocks 1: 1 bit clock
DDLY	1:0	0	RW	Selects the delay of the data in relation to the sync pulse. 00: Data starts 0 clocks after Sync. 01: Data starts 1 clock after Sync. 10: Data starts 2 clocks after Sync. 11: Data starts 3 clocks after Sync.

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 5.46: DSAI TX Data Shuffle

5.13.4 DSAIN_TX_STAT

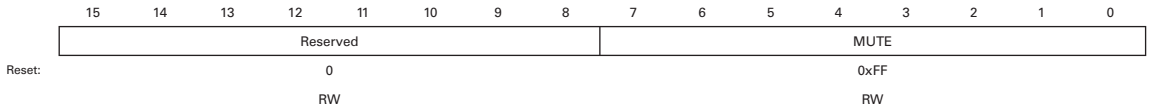
0xce09 0008



Name	Bit	Reset	Dir	Description
SLIP	1:0	0	RW	These two read only sticky bits indicate underrun and overrun conditions when running the receiver in slave mode. Bit 1: Slipped sample Bit 0: Repeat sample

5.13.5 DSAIN_TX_MUTE

0xce09 000c



Name	Bit	Reset	Dir	Description
MUTE	7:0	0	RW	This field controls the muting of individual channels in the transmitter. 0: Not muted 1: Muted

5.13.6 DSAI TX TIMING

5.13.6.1 DSAI TX TIMING IN SLAVE MODE

Conditions:
Commercial range (0-70 °C ambient, $VDD_{1.8V}$=1.8V±0.15V, $VDD_{3.3V}$=3.3±0.3V)
Output load C_{LOAD} = 30 pF

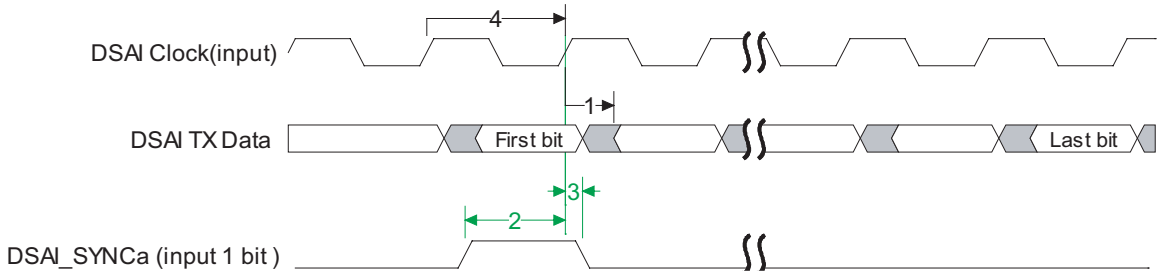


Figure 5.19: DSAI Tx Slave Timing when using rising edge of clock (dcka/b/c/d)

No	Name	Min	Max	Comment
1	DSAI_DATA delay to DSAI output clock	5	14	DSAI rising clock to DSAI data out delay
2	DSAI_SYNC to DSAI_CK setup time	5		DSAI sync required setup time before DSAI rising clock
3	DSAI_SYNC to DSAI_CK hold	0		DSAI sync required hold time before DSAI rising clock
4	DSAI clock period	39		DSAI clock cycle time (256FS)

Table 5.47: DSAI Tx Timing to the rising edge of DSAI clock (dcka/b/c/d)

5.13.6.2 DSAI TX TIMING IN MASTER MODE

Conditions:
Commercial range (0-70 °C ambient, $VDD_{1.8V}$=1.8V±0.15V, $VDD_{3.3V}$=3.3±0.3V)
Output load C_{LOAD} = 30 pF

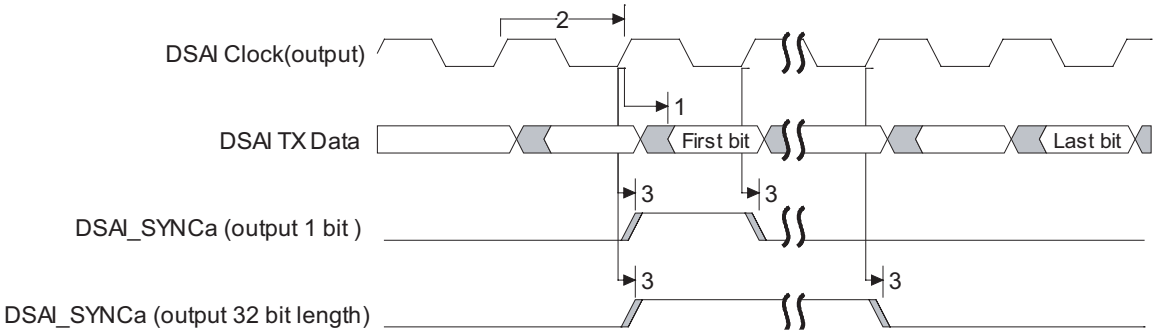


Figure 5.20: DSAI Tx Master Timing when using rising edge of clock (dcka/b/c/d)

No	Name	Min	Max	Comment
1	DSAI_DATA from DSAI clock output delay	5	14	DSAI rising clock to DSAI data out delay
2	DSAI clock period	39		DSAI clock cycle time (256FS)
3	DSAI Sync from DSAI Clock Output delay	1	4	

Table 5.48: DSAI Tx Master Timing to the rising edge of DSAI clock (dcka/b/c/d)

5.14 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.

5.14.1 MODULE CONFIGURATION

Address	Register
0xce16 0000 – 0xce16 0080	ARMAUDIO_BUF
0xce16 0100	ARMAUDIO_CTRL

Table 5.45: ARM Transceiver Memory Map

5.14.2 ARMAUDIO_BUF

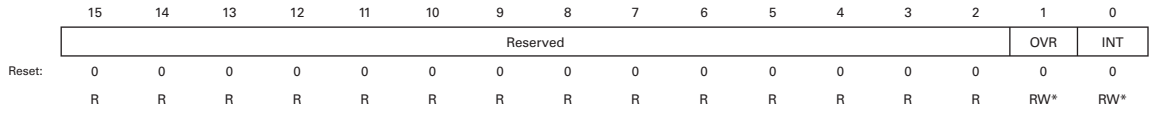
0xce16 0000 – 0xce16 0080

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.

5.14.3 ARMAUDIO_CTRL

0xce16 0100



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.

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. To solve this problem the AVS uses a module called the Internal Time Processor (ITP), to smooth the local clock over any of these small correction jumps. Due to the isolation of the ITP controlled AVS local timer from the 1394 cycle timer, the AVS is also immune to clock jumping caused by a change in master node or an arbitrary bus reset.

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 8 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 4 independent audio receivers each capable of extracting 16 audio channels and 8 MIDI plugs.

6.1.1 MODULE CONFIGURATION

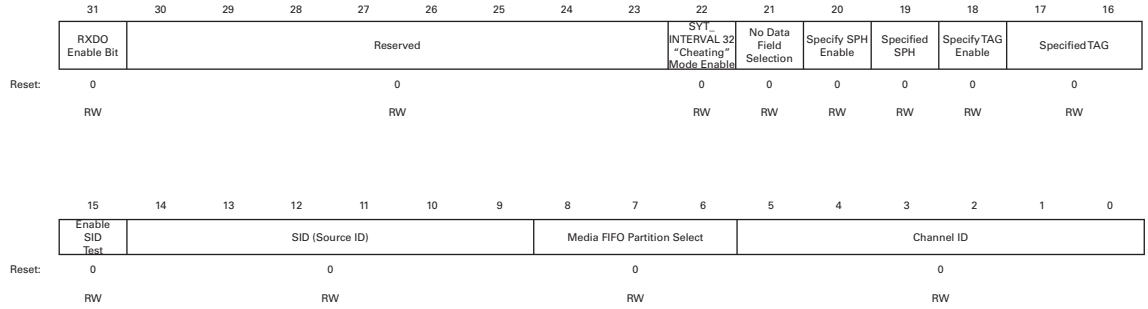
Address	Register
0xcf00 0000	ARX1_CFG0
0xcf00 0004	ARX1_CFG1
0xcf00 0008	ARX1_QSEL0
0xcf00 000c	ARX1_QSEL1
0xcf00 0010	ARX1_QSEL2
0xcf00 0014	ARX1_QSEL3
0xcf00 0018	ARX1_QSEL4
0xcf00 001c	ARX1_PHDR
0xcf00 0020	ARX1_CIP0
0xcf00 0024	ARX1_CIP1
0xcf00 0028	ARX1_ADO_CFG
0xcf00 002c	ARX1_ADO_MIDI
0xcf00 0030	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
0xcf00 0060	ARX3_CFG0
0xcf00 0064	ARX3_CFG1
0xcf00 0068	ARX3_QSEL0
0xcf00 006c	ARX3_QSEL1
0xcf00 0070	ARX3_QSEL2
0xcf00 0074	ARX3_QSEL3

0xcf00 0078	ARX3_QSEL4
0xcf00 007c	ARX1_PHDR
0xcf00 0080	ARX1_CIP0
0xcf00 0084	ARX1_CIP1
0xcf00 0088	ARX3_ADO_CFG
0xcf00 008c	ARX3_ADO_MIDI
0xcf00 0090	ARX4_CFG0
0xcf00 0094	ARX4_CFG1
0xcf00 0098	ARX4_QSEL0
0xcf00 009c	ARX4_QSEL1
0xcf00 00a0	ARX4_QSEL2
0xcf00 00a4	ARX4_QSEL3
0xcf00 00a8	ARX4_QSEL4
0xcf00 00ac	ARX1_PHDR
0xcf00 00b0	ARX1_CIP0
0xcf00 00b4	ARX1_CIP1
0xcf00 00b8	ARX4_ADO_CFG
0xcf00 00bc	ARX4_ADO_MIDI

Table 6.1: AVS Audio Receiver Memory Map

6.1.2 ARXN_CFG0

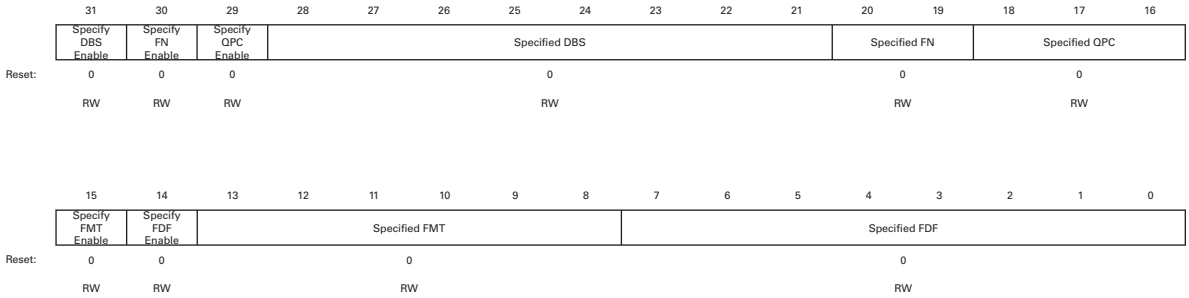
0xcf00 0000



Name	Bits	Reset	Dir	Description
RXDO Enable Bit	31	0	RW	RXDO Enable Bit. Setting this bit enables operation of the RXDO block.
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.
Media FIFO Partition Select	8:6	0	RW	Media FIFO Partition Select. Select which Media FIFO partition this RXDO block shall use. Media FIFO partitions 1, 2, and 3 can be encrypt/decrypt paths; the other 5 partitions cannot support M6 functionality.
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

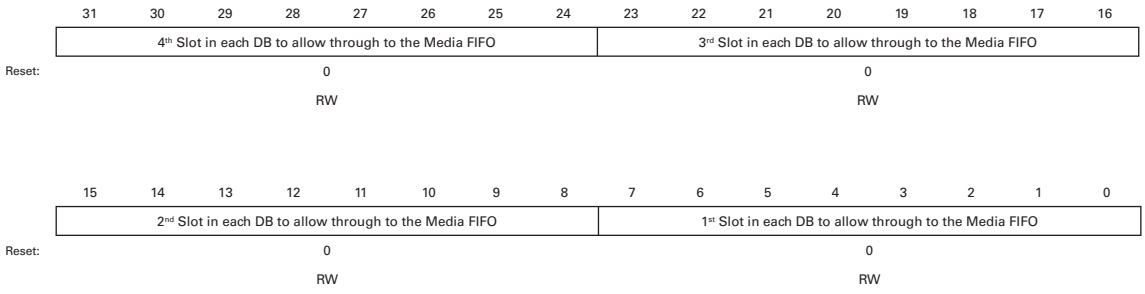
0xcf00 0004



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

0xcf00 0008



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 DICEII 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 DICEII 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 DICEII router.

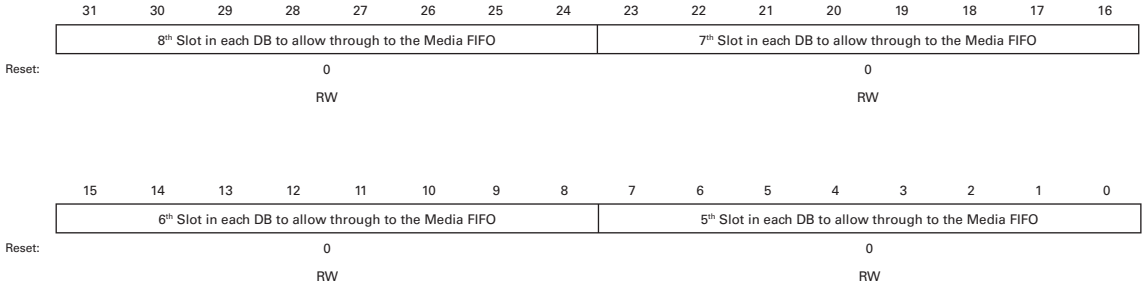
The reason for having 17 QSEL entries is to be able to handle 16 audio and 1 MIDI. Note that this does not mean that QSEL 17 is reserved for MIDI. Rather, MIDI must be referenced by the last valid QSEL entry. Note that this could even be the first QSEL, in the case of having 1 MIDI sequence and NO audio 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

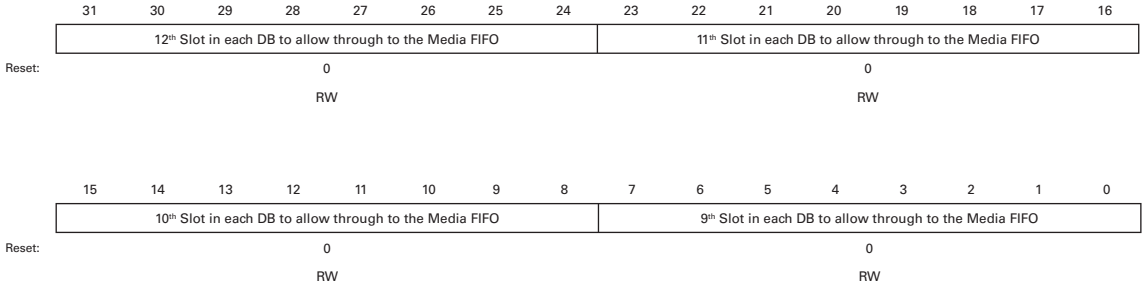
0xcf00 000c



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

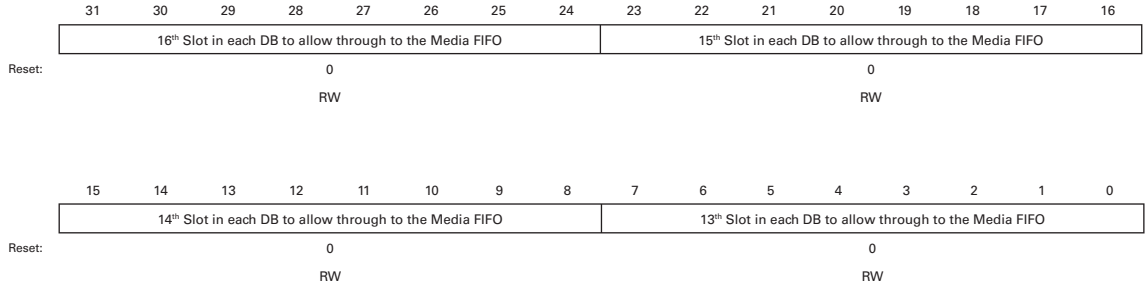
0xcf00 0010



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

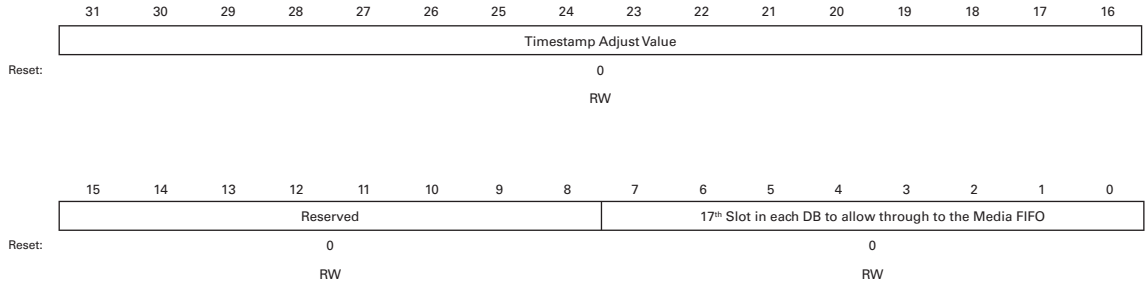
0xcf00 0014



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

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:

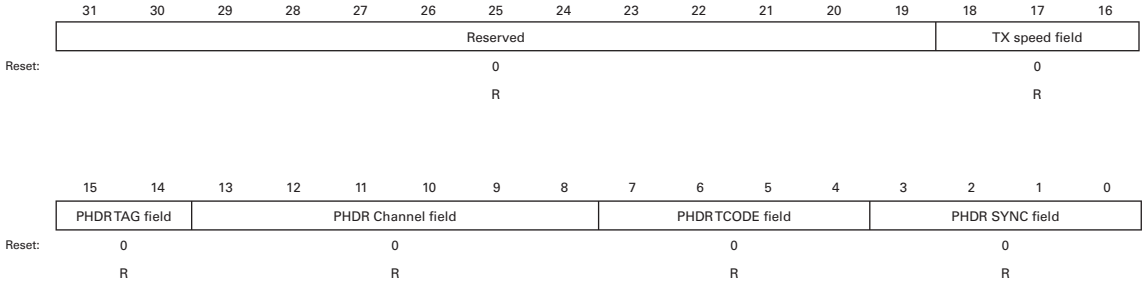
QSEL Slots 1-8: 8'h02, 8'h04, 8'h06, 8'h08, 8'h0A, 8'h0C, 8'h0E, 8'h10 (respectively).

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

0xcf00 001c

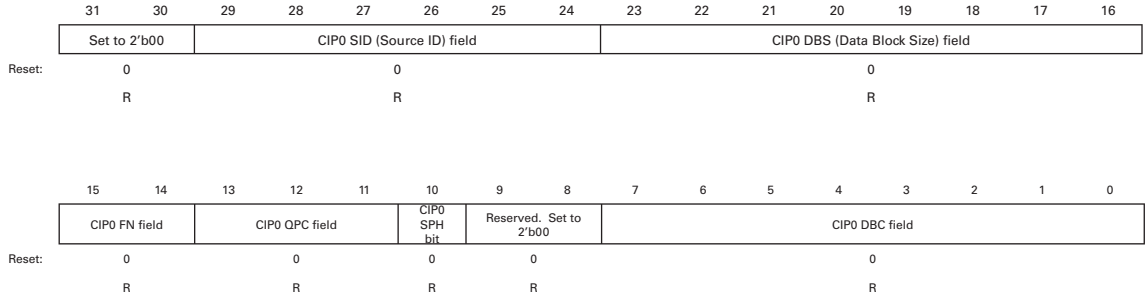


Name	Bits	Reset	Dir	Description
TX speed field	18:16	0	R	Set to 3'h0 for S100, 3'h1 for S200, and any other value indicates S400 isochronous transmit speed
PHDR TAG field	15:14	0	R	
PHDR Channel field	13:8	0	R	
PHDR TCODE field	7:4	0	R	
PHDR SYNC field	3:0	0	R	

This register is read only. It contains the last received PHDR quadlet.

6.1.10 ARXN_CIP0

0xcf00 0020

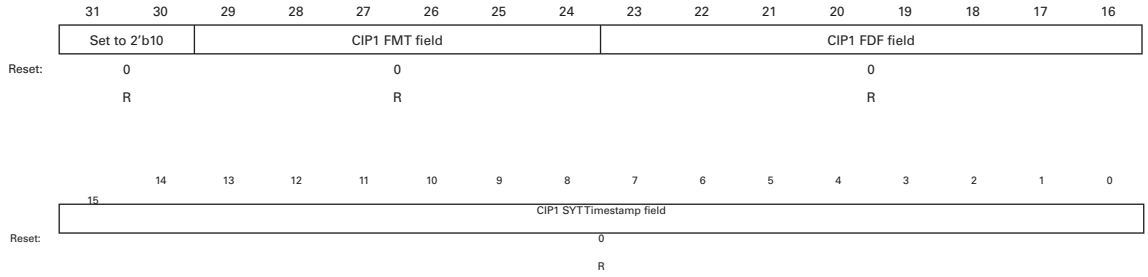


Name	Bits	Reset	Dir	Description
Reserved	31:30	0	R	
CIP0 SID (Source ID) field	29:24	0	R	CIP0 SID (Source ID) field
CIP0 DBS (Data Block Size) field	23:16	0	R	CIP0 DBS (Data Block Size) field
CIP0 FN field	15:14	0	R	CIP0 FN field
CIP0 QPC field	13:11	0	R	CIP0 QPC field
CIP0 SPH bit	10	0	R	CIP0 SPH bit
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

0xcf00 0024

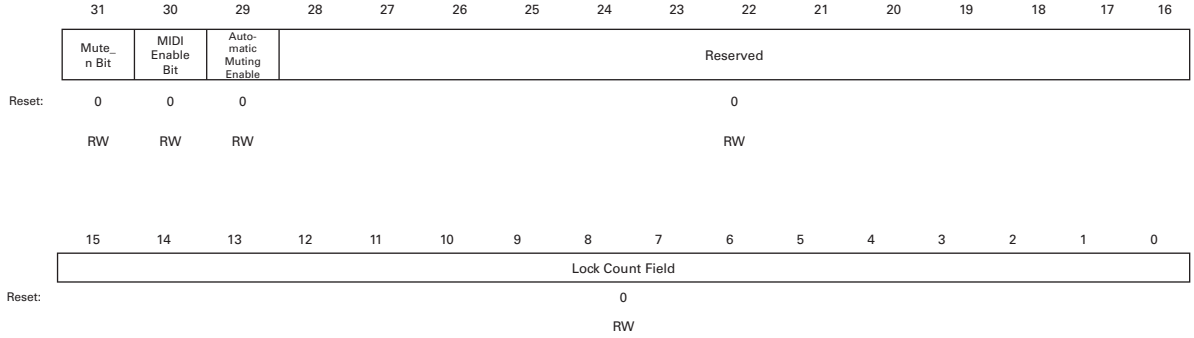


Name	Bits	Reset	Dir	Description
Set to 2'b10	31:30	0	R	
CIP1 FMT field	29:24	0	R	
CIP1 FDF field	23:16	0	R	
CIP1 SYTTimestamp field	15:0	0	R	

This register is read only. It contains the last received CIP1 quadlet.

6.1.12 ARXN_ADO_CFG

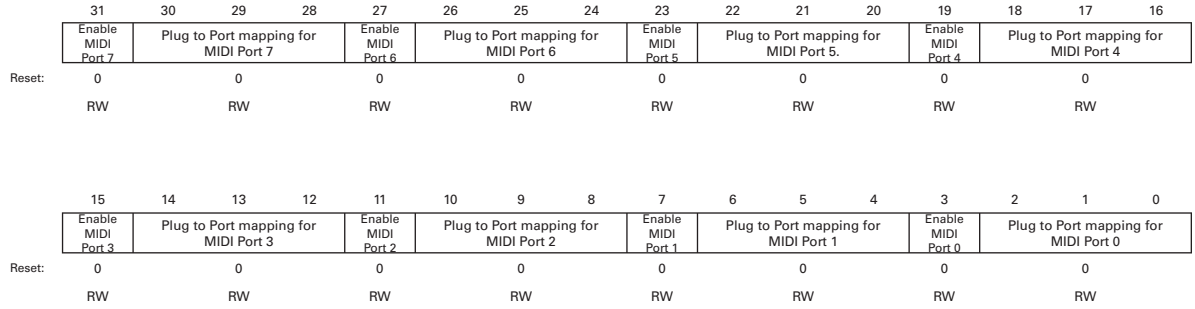
0xcf00 0028



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.
Lock Count Field	15:0	0	RW	Lock Count Field. This sets the number of samples that must pass through the ADO without slipping or repeating 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

0xcf00 002c



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.

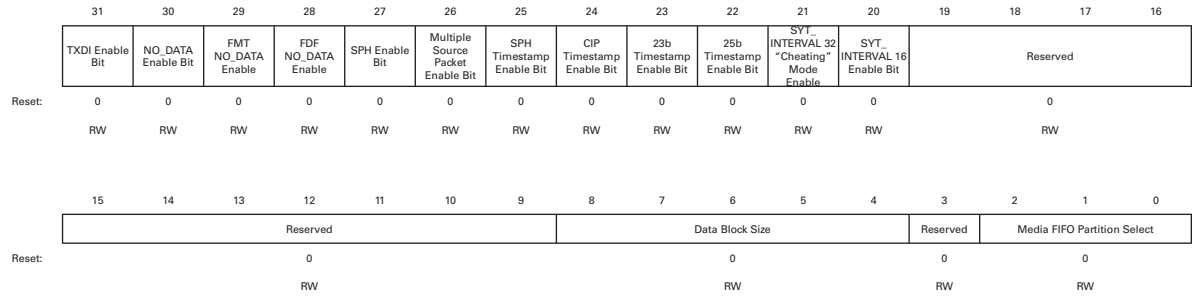
6.2.1 MODULE CONFIGURATION

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 6.2: AVS Audio Transmitter Memory Map

6.2.2 ATXN_CFG

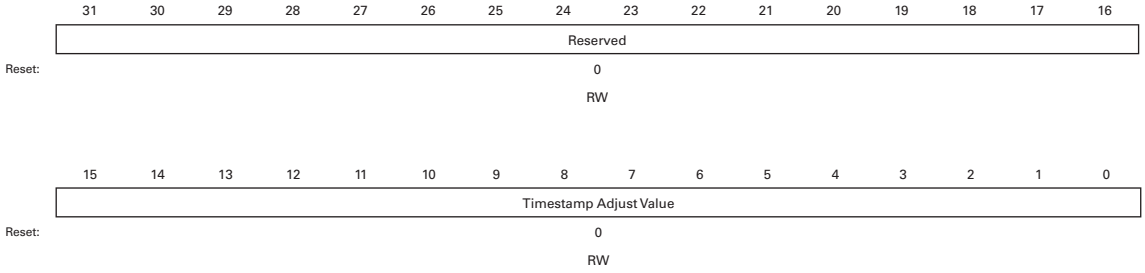
0xcf00 00c0



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.
SYT_INTERVAL 32 "Cheating" Mode Enable	21	0	RW	Setting this bit puts the ATX DB counters into a cheat mode, allowing SYT_INTERVAL 16 streams from the Router to be output as if they were SYT_INTERVAL 32 streams with Data Blocks half the size of what was input from the Router. Proper set up of the rest of the ATX (Data Block Size, SYT_ INTERVAL 16 Enable) for SYT_INTERVAL 16 is required for this mode to work.
SYT_INTERVAL 16 Enable Bit	20	0	RW	Setting this bit tells the ATX to expect 16 Data Blocks of quadlets per timestamp instead of just 8. This is for 88.2kHz and 96kHz data streams.
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.
Media FIFO Partition Select	2:0	0	RW	Select which Media FIFO partition this TXDI block shall use. Media FIFO partitions 1, 2, and 3 can be encrypt/ decrypt paths, the other 5 partitions cannot support M6 functionality.

6.2.3 ATXN_TSTAMP

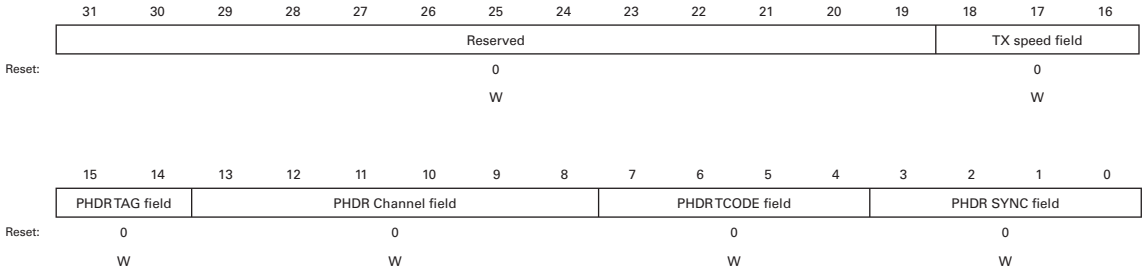
0xcf00 00c4



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

0xcf00 00c8

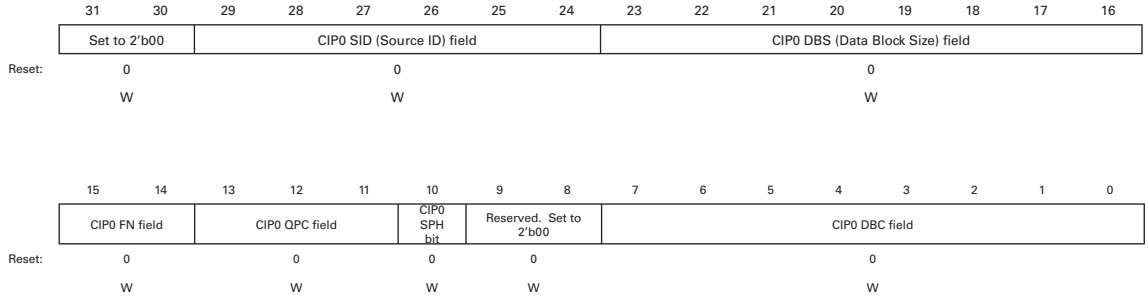


Name	Bits	Reset	Dir	Description
TX speed field	18:16	0	W	Set to 3'h0 for S100, 3'h1 for S200, and any other value will set S400 isochronous transmit speed
PHDR TAG field	15:14	0	W	Used when constructing PHDR quadlet to send
PHDR Channel field	13:8	0	W	Used when constructing PHDR quadlet to send
PHDR 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

0xcf00 00cc

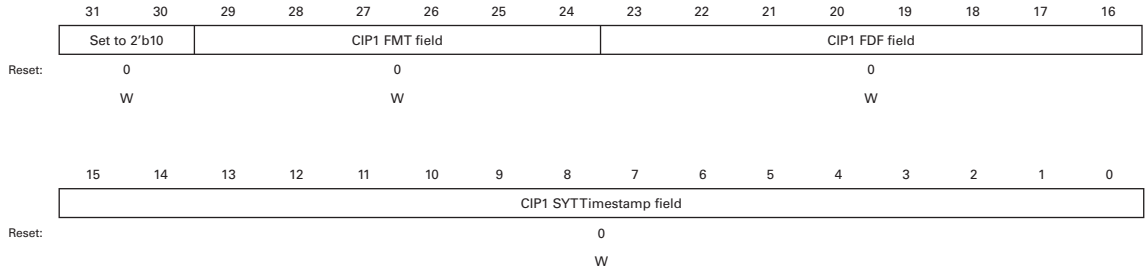


Name	Bits	Reset	Dir	Description
Set to 2'b00	31:30	0	W	Set to 2'b00
CIP0 SID (Source ID) field	29:24	0	W	CIP0 SID (Source ID) field. (used when constructing CIP0 quadlet to send)
CIP0 DBS (Data Block Size) field	23:16	0	W	CIP0 DBS (Data Block Size) field. (only used when constructing CIP0 quadlet to 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

0xcf00 00d0

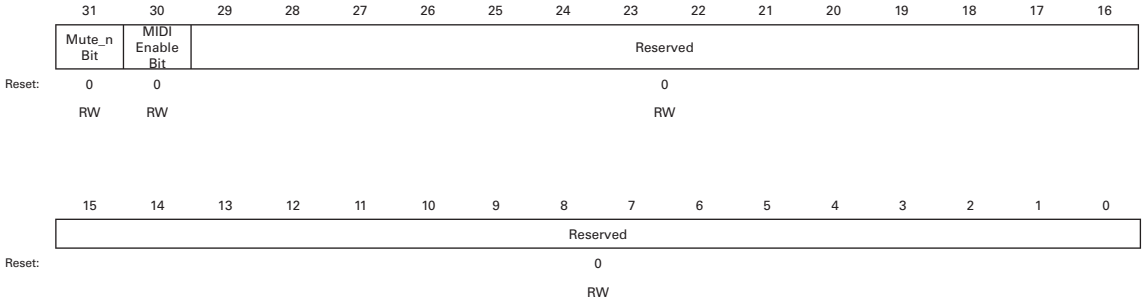


Name	Bits	Reset	Dir	Description
Set to 2'b10	31:30	0	W	Set to 2'b10
CIP1 FMT field	29:24	0	W	(used when constructing CIP1 quadlet to send)
CIP1 FDF field	23:16	0	W	(used when constructing CIP1 quadlet to send)
CIP1 SYTTimestamp field	15:0	0	W	This will be filled in by the ATX when sending the CIP1 quadlet.

This register is Write Only.

6.2.7 ATXN_ADI_CFG

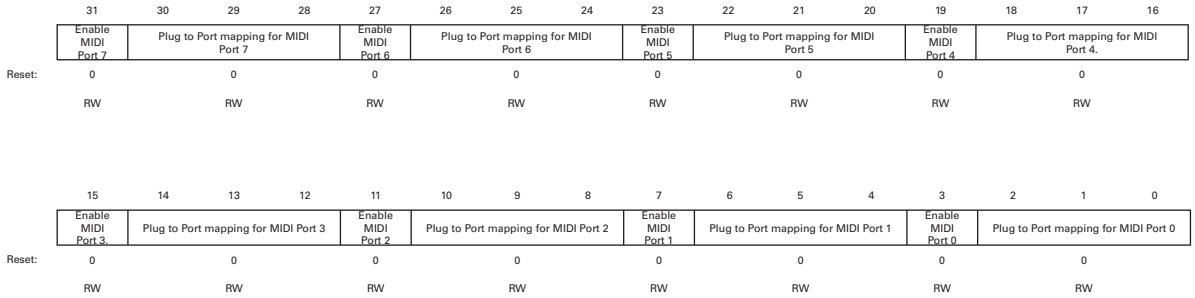
0xcf00 00d4



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

0xcf00 00d8



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.

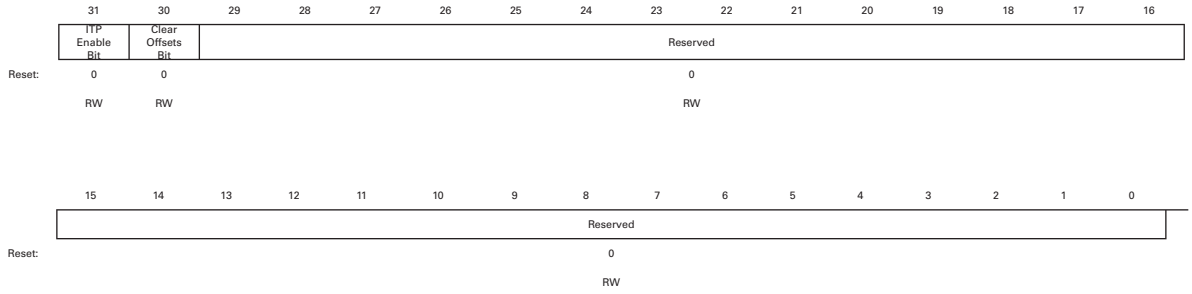
6.3.1 MODULE CONFIGURATION

Address	Register
0xcf00 01f8	ITP_CFG

Table 6.3: AVS ITP Memory Map

6.3.2 ITP_CFG

0xcf00 01f8



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 II should handle Channel Status, User Bits, Validity and Block Sync in a similar way as is done by the DICE AES transceivers.

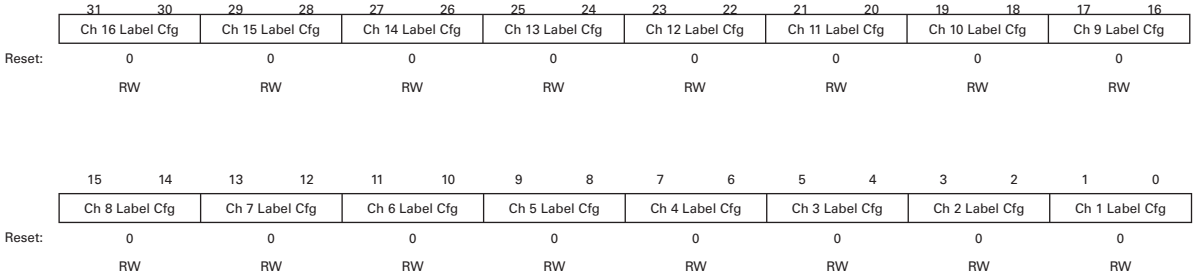
6.4.1 MODULE CONFIGURATION

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 6.4: AVS Audio Transmitter Format Handler Memory Map

6.4.2 FMT_TXDIN_CFG0

0xcf00 02c0

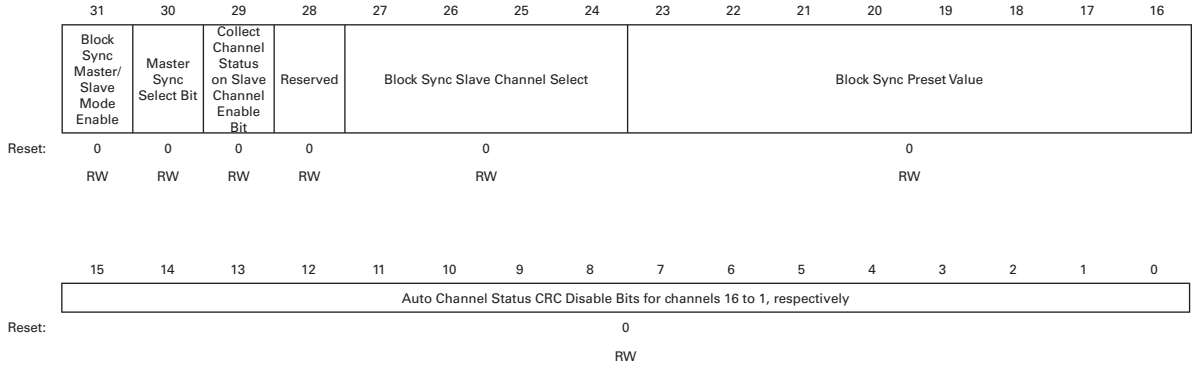


Name	Bits	Reset	Dir	Description
Channel 16 Label Configuration	31:30	0	RW	Channel 16 Label Configuration. (see below for detail of bits)
Channel 15 Label Configuration	29:28	0	RW	Channel 15 Label Configuration. (see below for detail of bits)
Channel 14 Label Configuration	27:26	0	RW	Channel 14 Label 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

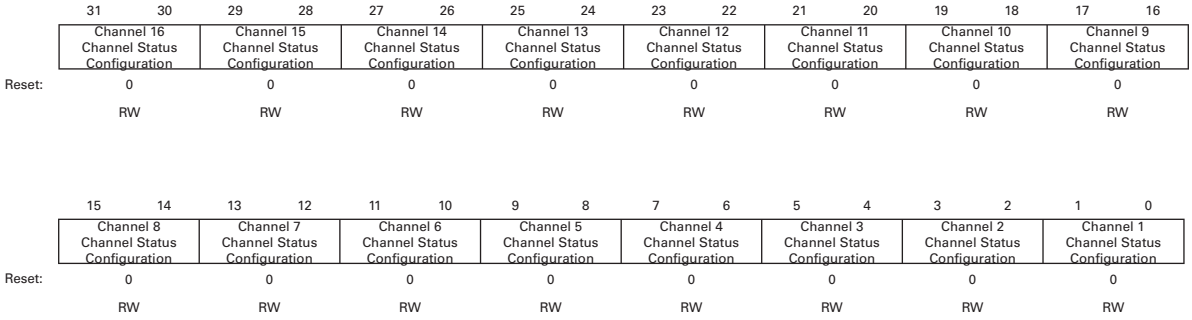
0xcf00 02c4



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

0xcf00 02c8

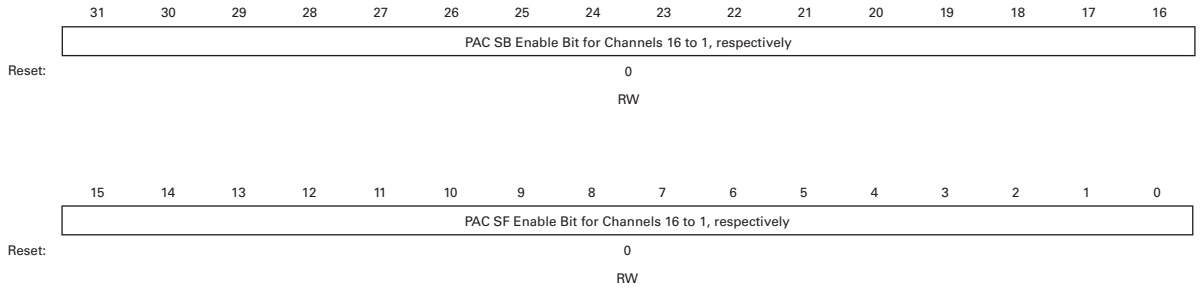


Name	Bits	Reset	Dir	Description
Channel 16 Channel Status Configuration	31:30	0	RW	Channel 16 Channel Status Configuration. (see below for detail of bits)
Channel 15 Channel Status Configuration	29:28	0	RW	Channel 15 Channel Status Configuration. (see below for detail of bits)
Channel 14 Channel Status 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

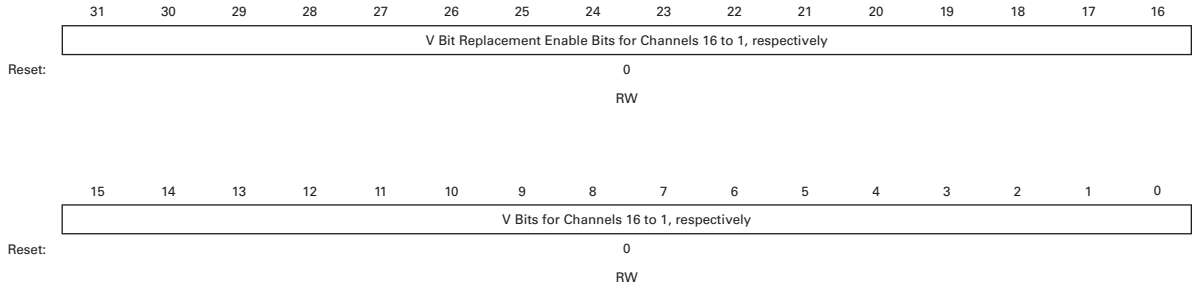
0xcf00 02cc



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

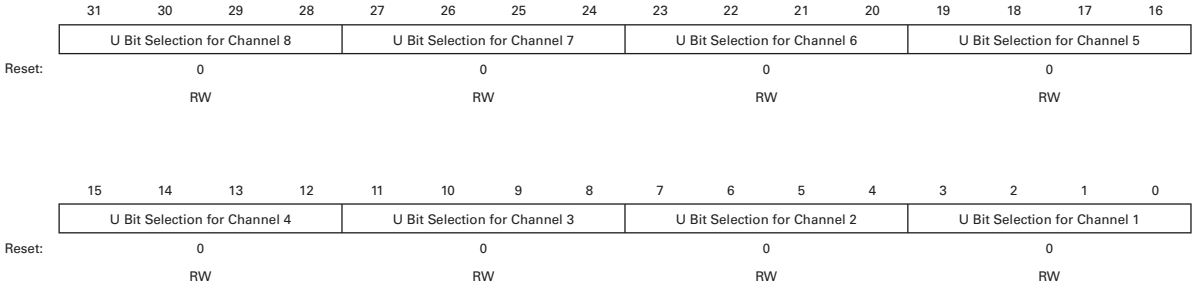
0xcf00 02d0



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

0xcf00 02d4



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.

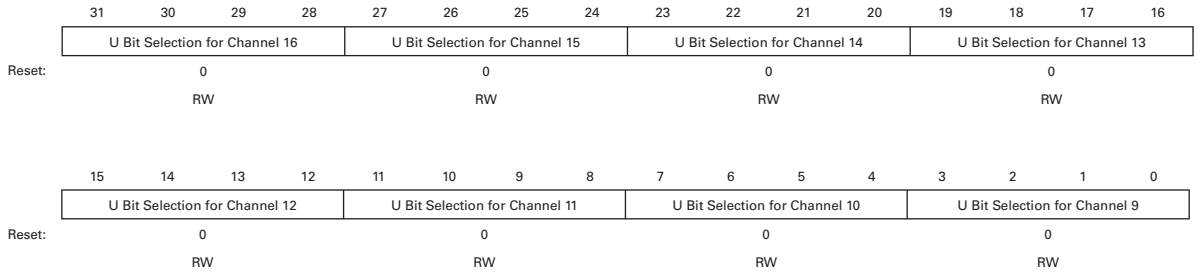
4'b0100: Take U bit input from AVS RX 4.

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

0xcf00 02d8



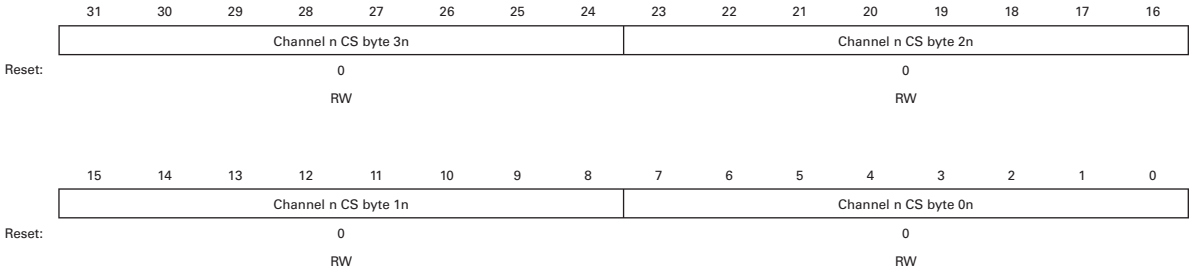
Name	Bits	Reset	Dir	Description
User Bit for Channel 16	31:28	0	RW	U Bit Selection for Channel 16. (see detail of bits below)
User Bit for Channel 15	27:24	0	RW	U Bit Selection for Channel 15. (see detail of bits below)
User Bit for Channel 14	23:20	0	RW	U Bit Selection for Channel 14. (see detail of 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

0xcf00 02dc

These registers are used to write the entire channel status block (192 bits) for one selected channel.



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

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.

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 II should handle Channel Status, User Bits, Validity and Block Sync in a similar way as is done by the DICE AES transceivers.

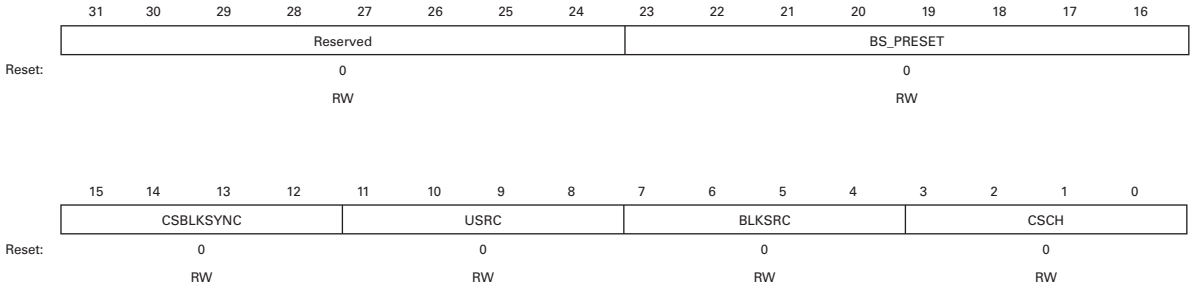
6.5.1 MODULE CONFIGURATION

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
0xcf00 0260	FORMAT_RXDI3_CFG
0xcf00 0264	FORMAT_RXDI3_LABELn
0xcf00 0274	FORMAT_RXDI3_CSBLOCKn
0xcf00 0290	FORMAT_RXDI4_CFG
0xcf00 0294	FORMAT_RXDI4_LABELn
0xcf00 02a4	FORMAT_RXDI4_CSBLOCKn

Table 6.5: AVS Audio Receiver Format Handler Memory Map

6.5.2 FORMAT_RXDIN_CFG

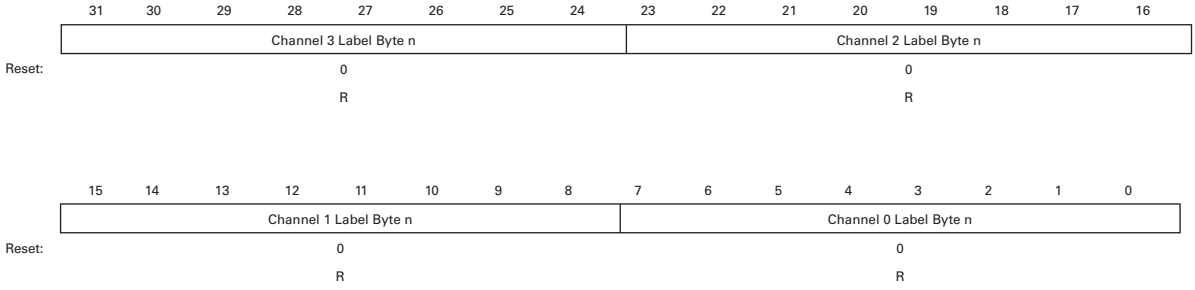
0xcf00 0200



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

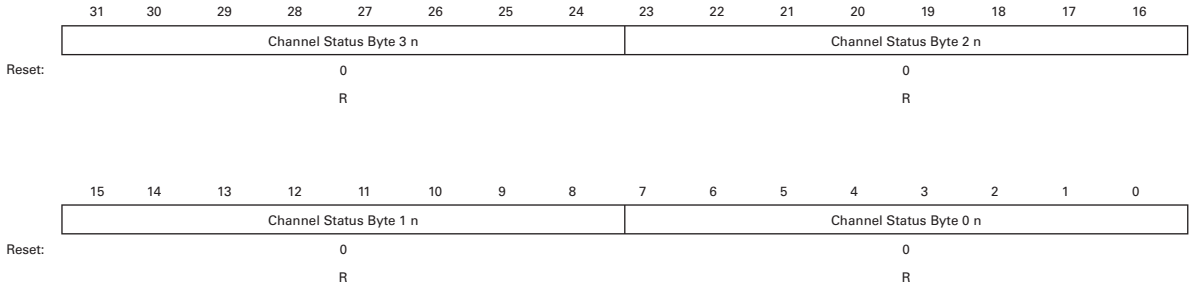
0xcf00 0204



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

0xcf00 0214



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 one interrupt 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
0xcf00 014c	AVSI_INT2_STATUS
0xcf00 0150	AVSI_INT2_MASK

Table 6.6: AVS INT CTRL Memory Map

6.6.2 APBA_INT0_STATUS

0xcf00 013c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VRX_CFG_FAIL	VRX_CIP_FAIL	VRX_DBC_FAIL	VRX_LONG_PKT	VRX_PKT_ABORT	VRX_STATUS_ERR	VTX_BOUNDRY_ERR	VTX_FRAME_AGEOUT	VTX_PKT_ABORT	CIPHER0_KEY_REQ	CIPHER1_KEY_REQ	CIPHER2_KEY_REQ	ADO1_LOCKED	ADO1_STREAM_END	ADO1_STREAM_START	ADO2_LOCKED
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
	ADO2_STREAM_END	ADO2_STREAM_START	ADO3_LOCKED	ADO3_STREAM_END	ADO3_STREAM_START	ADO4_LOCKED	ADO4_STREAM_END	ADO4_STREAM_START	ATX1_STREAM_END	ATX1_STREAM_START	ATX2_STREAM_END	ATX2_STREAM_START	VDO_STREAM_END	VDO_STREAM_START	VTX_STREAM_END	VTX_STREAM_START
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

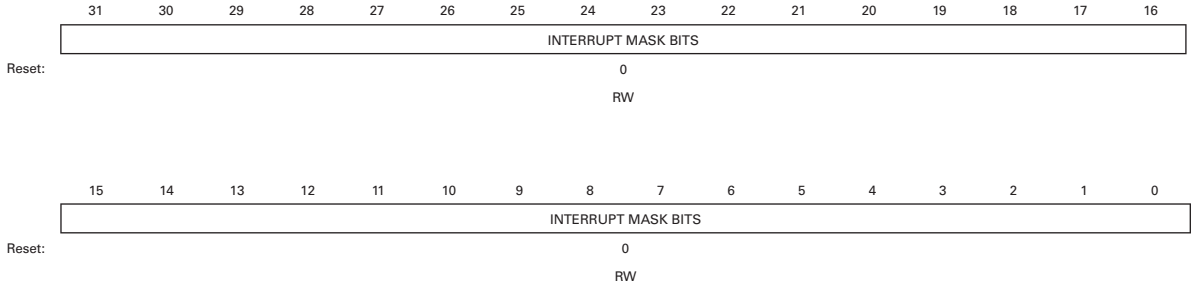
Name	Bits	Reset	Dir	Description
VRX_CFG_FAIL	31	0	RW	The VRX module detected that a "forced" value in the CFG registers did not match what the stream is actually sending.
VRX_CIP_FAIL	30	0	RW	The VRX module detected an error in the CIP format of the received stream.
VRX_DBC_FAIL	29	0	RW	The VRX module detected a discontinuity in the DBC of the received stream. This will always fire once when the stream starts up.
VRX_LONG_PKT	28	0	RW	The VRX module received an isoch packet that was too long to store in its local memory.

VRX_PKT_ABORT	27	0	RW	The VRX was forced to abort an isoch packet due to an error.
VRX_STATUS_ERR	26	0	RW	The VRX received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
VTX_BOUNDRY_ERR	25	0	RW	The VTX encountered a problem with the isoch packet boundary as it was sending a packet.
VTX_FRAME_AGEOUT	24	0	RW	The VTX had to age-out a frame of data that was waiting to be sent because it became stale and was not transmitted in time.
VTX_PKT_ABORT	23	0	RW	The VTX encountered a problem and had to abort transmission of an isoch packet.
CIPHER0_KEY_REQ	22	0	RW	The cipher 0 block requests a new key.
CIPHER1_KEY_REQ	21	0	RW	The cipher 1 block requests a new key.
CIPHER2_KEY_REQ	20	0	RW	The cipher 2 block requests a new key.
ADO1_LOCKED	19	0	RW	The output of ADO 1 has not slipped/repeated samples in a long enough time that it can be considered in a "lock" state.
ADO1_STREAM_END	18	0	RW	The data stream output by ADO 1 has ended.
ADO1_STREAM_START	17	0	RW	The data stream output by ADO 1 has started.
ADO2_LOCKED	16	0	RW	The output of ADO 2 has not slipped/repeated samples in a long enough time that it can be considered in a "lock" state.
ADO2_STREAM_END	15	0	RW	The data stream output by ADO 2 has ended.
ADO2_STREAM_START	14	0	RW	The data stream output by ADO 2 has started.
ADO3_LOCKED	13	0	RW	The output of ADO 3 has not slipped/repeated samples in a long enough time that it can be considered in a "lock" state.
ADO3_STREAM_END	12	0	RW	The data stream output by ADO 3 has ended.
ADO3_STREAM_START	11	0	RW	The data stream output by ADO 3 has started.
ADO4_LOCKED	10	0	RW	The output of ADO 4 has not slipped/repeated samples in a long enough time that it can be considered in a "lock" state.
ADO4_STREAM_END	9	0	RW	The data stream output by ADO 4 has ended.
ADO4_STREAM_START	8	0	RW	The data stream output by ADO 4 has started.
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.
ATX2_STREAM_END	5	0	RW	The data stream transmitted by ATX 2 has ended.
ATX2_STREAM_START	4	0	RW	The data stream transmitted by ATX 2 has started.
VDO_STREAM_END	3	0	RW	The data stream output by VDO has ended.
VDO_STREAM_START	2	0	RW	The data stream output by VDO has started.
VTX_STREAM_END	1	0	RW	The data stream transmitted by VTX has ended.
VTX_STREAM_START	0	0	RW	The data stream transmitted by VTX has started.

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

0xcf00 0140

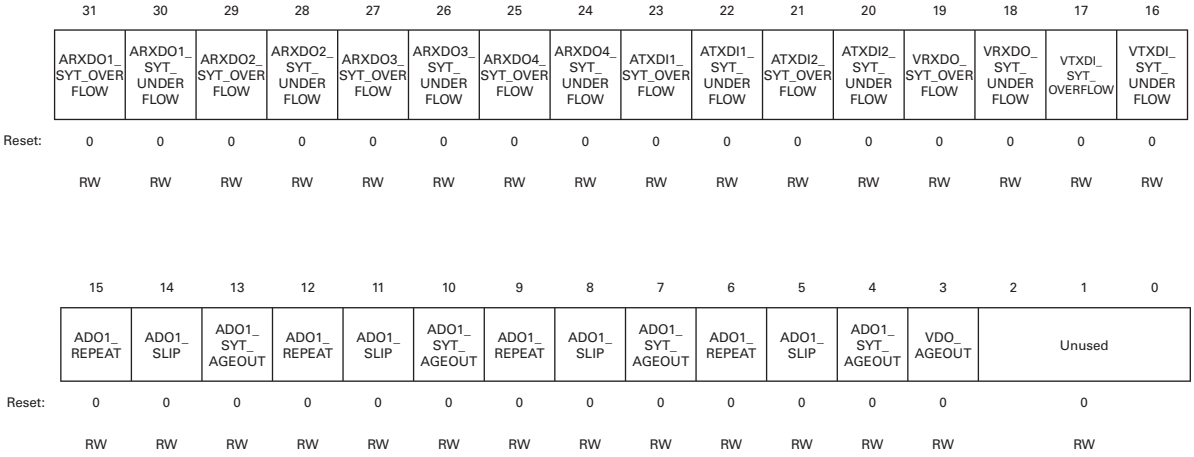


Name	Bits	Reset	Dir	Description
Interrupt Mask	31:0	0	RW	Interrupt mask bits 0: Ignore interrupt 1: Allow interrupt

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

0xcf00 0144



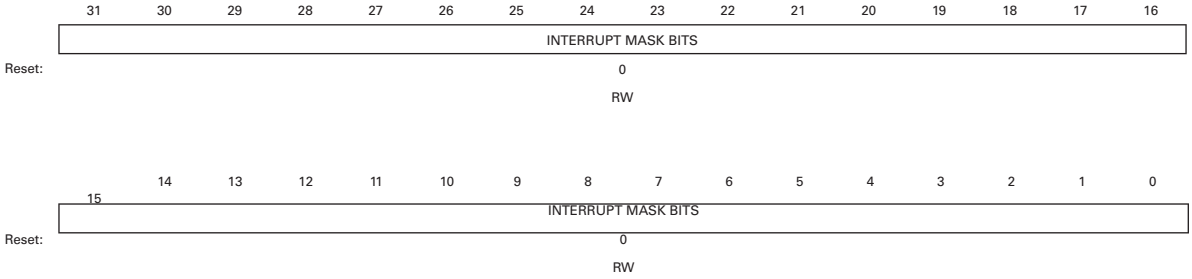
Name	Bits	Reset	Dir	Description
ARXDO1_SYT_ OVERFLOW	31	0	RW	The timestamp FIFO of ARXDO 1 has overflowed.
ARXDO1_SYT_ UNDERFLOW	30	0	RW	The timestamp FIFO of ARXDO 1 has underflowed.
ARXDO2_SYT_ OVERFLOW	29	0	RW	The timestamp FIFO of ARXDO 2 has overflowed.
ARXDO2_SYT_ UNDERFLOW	28	0	RW	The timestamp FIFO of ARXDO 2 has underflowed.

ARXDO3_SYT_ OVERFLOW	27	0	RW	The timestamp FIFO of ARXDO 3 has overflowed.
ARXDO3_SYT_ UNDERFLOW	26	0	RW	The timestamp FIFO of ARXDO 3 has underflowed.
ARXDO4_SYT_ OVERFLOW	25	0	RW	The timestamp FIFO of ARXDO 4 has overflowed.
ARXDO4_SYT_ UNDERFLOW	24	0	RW	The timestamp FIFO of ARXDO 4 has underflowed.
ATXDI1_SYT_OVERFLOW	23	0	RW	The timestamp FIFO of ATXDI 1 has overflowed.
ATXDI1_SYT_ UNDERFLOW	22	0	RW	The timestamp FIFO of ATXDI 1 has underflowed.
ATXDI2_SYT_OVERFLOW	21	0	RW	The timestamp FIFO of ATXDI 2 has overflowed.
ATXDI2_SYT_ UNDERFLOW	20	0	RW	The timestamp FIFO of ATXDI 2 has underflowed.
VRXDO_SYT_OVERFLOW	19	0	RW	The timestamp FIFO of VRXDO has overflowed.
VRXDO_SYT_ UNDERFLOW	18	0	RW	The timestamp FIFO of VRXDO has underflowed.
VTXDI_SYT_OVERFLOW	17	0	RW	The timestamp FIFO of VTXDI has overflowed.
VTXDI_SYT_UNDERFLOW	16	0	RW	The timestamp FIFO of VTXDI has underflowed.
ADO1_REPEAT	15	0	RW	The ADO 1 had to repeat a sample of data.
ADO1_SLIP	14	0	RW	The ADO 1 had to slip a sample of data.
ADO1_SYT_AGEOUT	13	0	RW	The ADO 1 had to age-out a stale frame of data.
ADO2_REPEAT	12	0	RW	The ADO 2 had to repeat a sample of data.
ADO2_SLIP	11	0	RW	The ADO 2 had to slip a sample of data.
ADO2_SYT_AGEOUT	10	0	RW	The ADO 2 had to age-out a stale frame of data.
ADO3_REPEAT	9	0	RW	The ADO 3 had to repeat a sample of data.
ADO3_SLIP	8	0	RW	The ADO 3 had to slip a sample of data.
ADO3_SYT_AGEOUT	7	0	RW	The ADO 3 had to age-out a stale frame of data.
ADO4_REPEAT	6	0	RW	The ADO 4 had to repeat a sample of data.
ADO4_SLIP	5	0	RW	The ADO 4 had to slip a sample of data.
ADO4_SYT_AGEOUT	4	0	RW	The ADO 4 had to age-out a stale frame of data.
VDO_AGEOUT	3	0	RW	The VDO had to age-out a stale frame of data.

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

0xcf00 0148



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.

6.6.6 APBA_INT2_STATUS

0xcf00 014c

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITP_EP_TOO_BIG	ARX1_CFG_FAIL	ARX1_CIP_FAIL	ARX1_DBC_FAIL	ARX1_LONG_PKT	ARX1_PKT_ABORT	ARX1_STATUS_ERR	ARX2_CFG_FAIL	ARX2_CIP_FAIL	ARX2_DBC_FAIL	ARX2_LONG_PKT	ARX2_PKT_ABORT	ARX2_STATUS_ERR	ARX3_CFG_FAIL	ARX3_CIP_FAIL	ARX3_DBC_FAIL
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
	ARX3_LONG_PKT	ARX3_PKT_ABORT	ARX3_STATUS_ERR	ARX4_CFG_FAIL	ARX4_CIP_FAIL	ARX4_DBC_FAIL	ARX4_LONG_PKT	ARX4_PKT_ABORT	ARX4_STATUS_ERR	ATX1_BOUNDRY_ERR	ATX1_FRAME_AGEOUT	ATX1_PKT_ABORT	ATX2_BOUNDRY_ERR	ATX2_FRAME_AGEOUT	ATX2_PKT_ABORT	Unused
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
ITP_EP_TOO_BIG	31	0	RW	The ITP's internally-calculated error value has become too large to handle. This indicates a massive clocking problem in the 1394 network.
ARX1_CFG_FAIL	30	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	29	0	RW	The ARX 1 module detected an error in the CIP format of the received stream.
ARX1_DBC_FAIL	28	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.

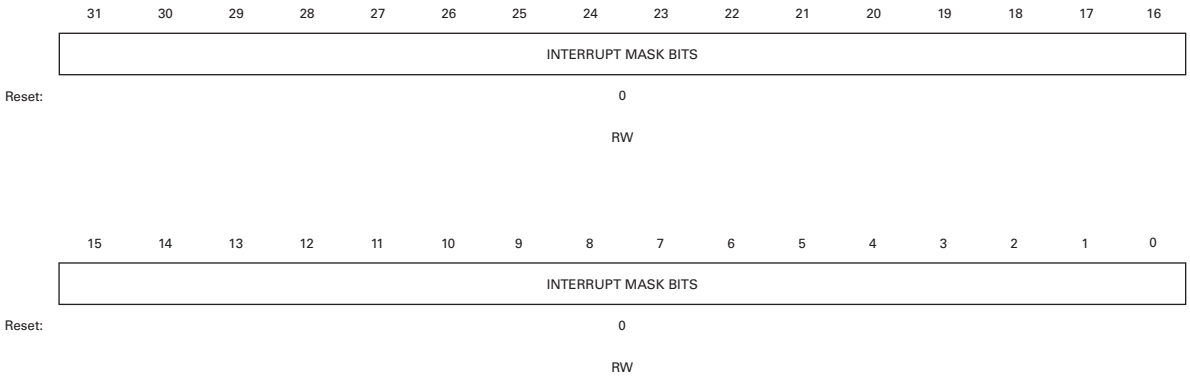
Name	Bits	Reset	Dir	Description
ARX1_LONG_PKT	27	0	RW	The ARX 1 module received an isoch packet that was too long to store in its local memory.
ARX1_PKT_ABORT	26	0	RW	The ARX 1 was forced to abort an isoch packet due to an error.
ARX1_STATUS_ERR	25	0	RW	The ARX 1 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
ARX2_CFG_FAIL	24	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	23	0	RW	The ARX 2 module detected an error in the CIP format of the received stream.
ARX2_DBC_FAIL	22	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.
ARX2_LONG_PKT	21	0	RW	The ARX 2 module received an isoch packet that was too long to store in its local memory.
ARX2_PKT_ABORT	20	0	RW	The ARX 2 was forced to abort an isoch packet due to an error.
ARX2_STATUS_ERR	19	0	RW	The ARX 2 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
ARX3_CFG_FAIL	18	0	RW	The ARX 3 module detected that a "forced" value in the CFG registers did not match what the stream is actually sending.
ARX3_CIP_FAIL	17	0	RW	The ARX 3 module detected an error in the CIP format of the received stream.
ARX3_DBC_FAIL	16	0	RW	The ARX 3 module detected a discontinuity in the DBC of the received stream. This will always fire once when the stream starts up.
ARX3_LONG_PKT	15	0	RW	The ARX 3 module received an isoch packet that was too long to store in its local memory.
ARX3_PKT_ABORT	14	0	RW	The ARX 3 was forced to abort an isoch packet due to an error.
ARX3_STATUS_ERR	13	0	RW	The ARX 3 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.
ARX4_CFG_FAIL	12	0	RW	The ARX 4 module detected that a "forced" value in the CFG registers did not match what the stream is actually sending.
ARX4_CIP_FAIL	11	0	RW	The ARX 4 module detected an error in the CIP format of the received stream.
ARX4_DBC_FAIL	10	0	RW	The ARX 4 module detected a discontinuity in the DBC of the received stream. This will always fire once when the stream starts up.
ARX4_LONG_PKT	9	0	RW	The ARX 4 module received an isoch packet that was too long to store in its local memory.
ARX4_PKT_ABORT	8	0	RW	The ARX 4 was forced to abort an isoch packet due to an error.
ARX4_STATUS_ERR	7	0	RW	The ARX 4 received a status quadlet (signaling the end of an isoch packet) before or after it was expected.

Name	Bits	Reset	Dir	Description
ATX1_BOUNDRY_ERR	6	0	RW	The ATX 1 encountered a problem with the isoch packet boundary as it was sending a packet.
ATX1_FRAME_AGEOUT	5	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	4	0	RW	The ATX 1 encountered a problem and had to abort transmission of an isoch packet.
ATX2_BOUNDRY_ERR	3	0	RW	The ATX 2 encountered a problem with the isoch packet boundary as it was sending a packet.
ATX2_FRAME_AGEOUT	2	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	1	0	RW	The ATX 2 encountered a problem and had to abort transmission of an isoch packet.

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.

6.6.7 APBA_INT2_MASK

0xcf00 0150



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 8 partitions which can be allocated freely from the memory pool.

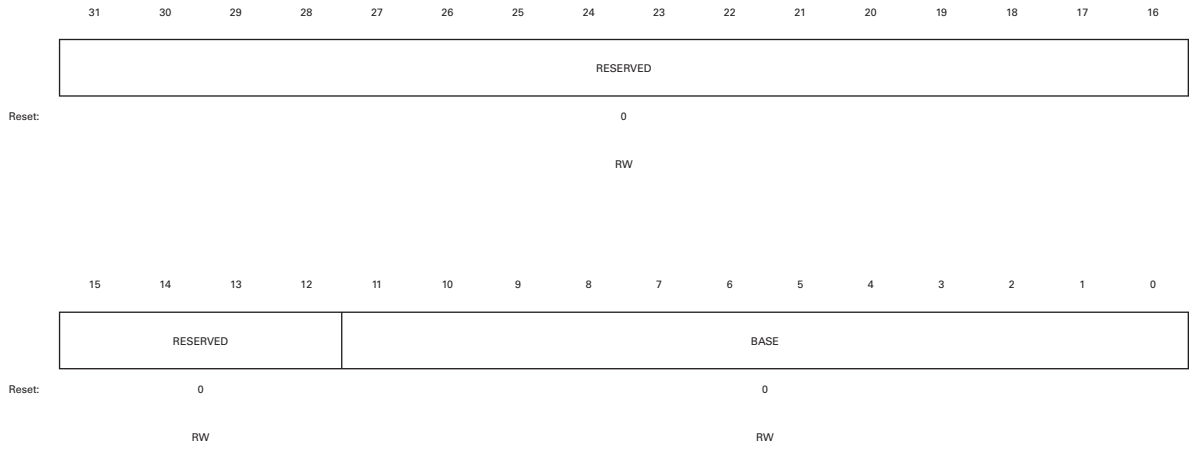
6.7.1 MODULE CONFIGURATION

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 01b4	AVSFIFO_PART4_BASE
0xcf00 01b8	AVSFIFO_PART4_LIMIT
0xcf00 01bc	AVSFIFO_PART4_FLUSH
0xcf00 01c0	AVSFIFO_PART5_BASE
0xcf00 01c4	AVSFIFO_PART5_LIMIT
0xcf00 01c8	AVSFIFO_PART5_FLUSH
0xcf00 01cc	AVSFIFO_PART6_BASE
0xcf00 01d0	AVSFIFO_PART6_LIMIT
0xcf00 01d4	AVSFIFO_PART6_FLUSH
0xcf00 01d8	AVSFIFO_PART7_BASE
0xcf00 01dc	AVSFIFO_PART7_LIMIT
0xcf00 01e0	AVSFIFO_PART7_FLUSH
0xcf00 01fc	AVSFIFO_STAT

Table 6.6: AVS Media FIFO Memory Map

6.7.2 AVSFIFO_PARTN_BASE

0xcf00 0184

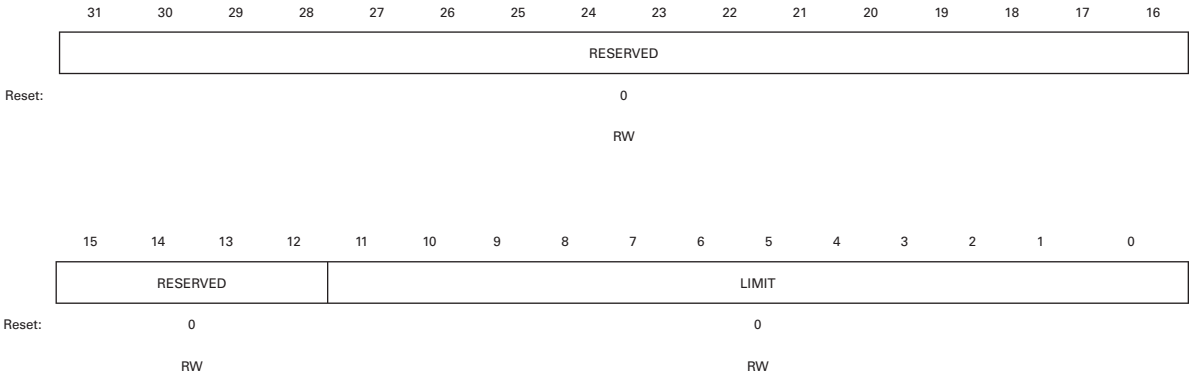


Name	Bits	Reset	Dir	Description
Base	11:0	0	RW	The lowest RAM address at which this partition can store data.

The BASE register is both readable and writeable.

6.7.3 AVSFIFO_PART0_LIMIT

0xcf00 0188

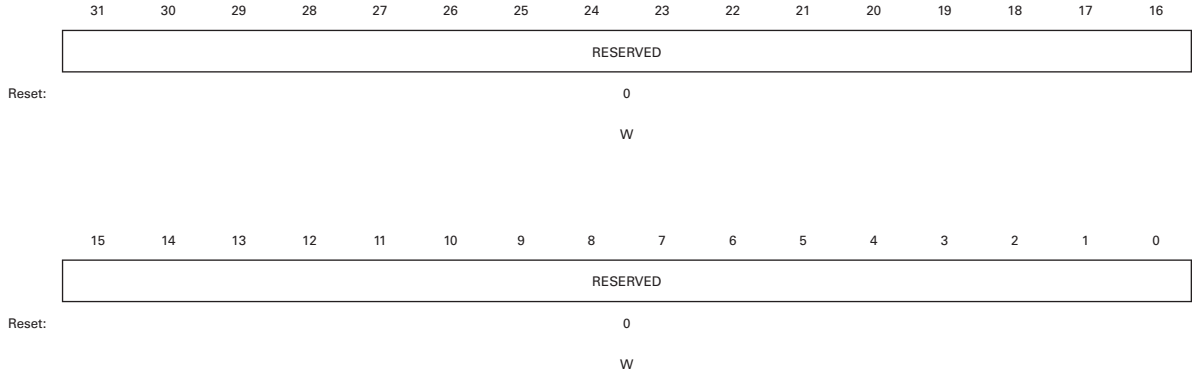


Name	Bits	Reset	Dir	Description
Limit	11:0	0	RW	The highest RAM address at which this partition can store data

The LIMIT register is both readable and writeable.

6.7.4 AVSFIFO_PART0_FLUSH

0xcf00 018c

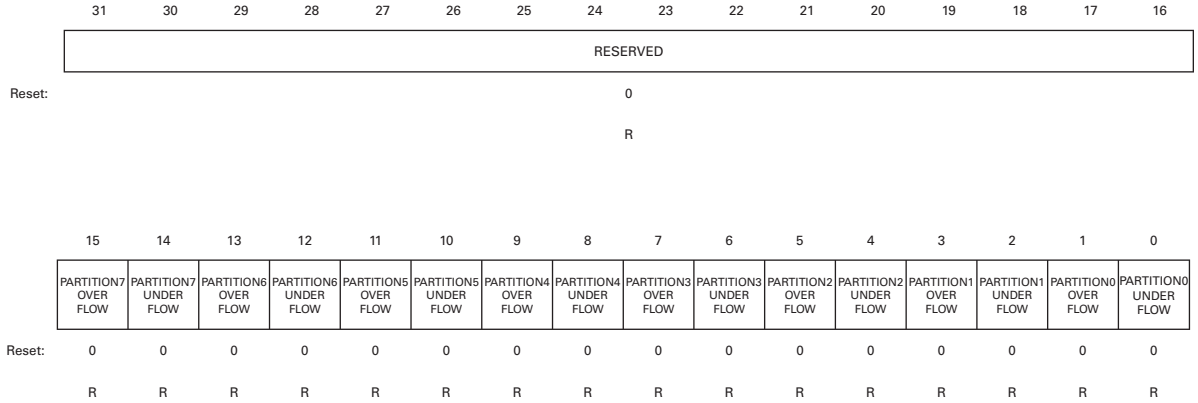


Name	Bits	Reset	Dir	Description
Flush	31:0	0	W	This register does not store data.

The FLUSH register is write-only.

6.7.5 AVSFIFO_STAT

0xcf00 01fc



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.

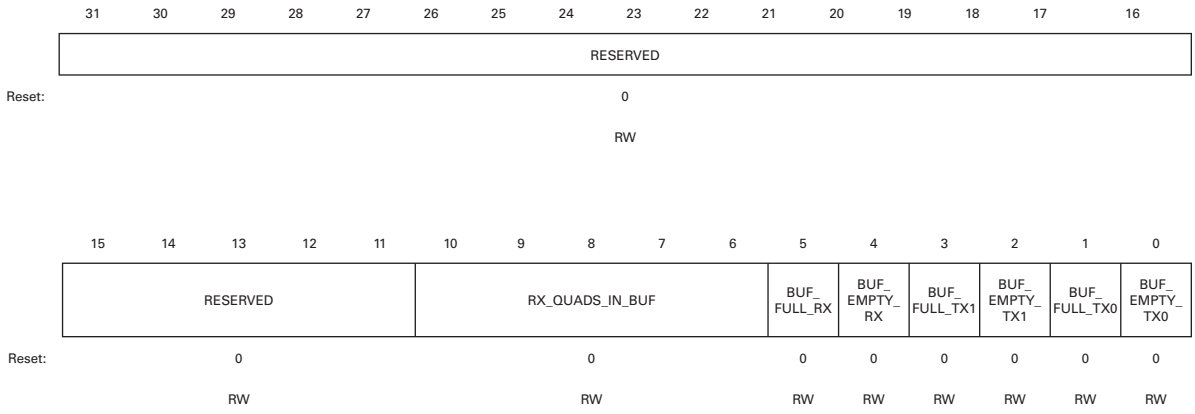
6.8.1 MODULE CONFIGURATION

Address	Register
0xcf00 01e4	AVSMIDI_STAT
0xcf00 01e8	AVSMIDI_CTRL
0xcf00 01ec	AVSMIDI_RX
0xcf00 01f0	AVSMIDI_TX0
0xcf00 01f4	AVSMIDI_TX1

Table 6.7: AVS MIDI Memory Map

6.8.2 AVSMIDI_STAT

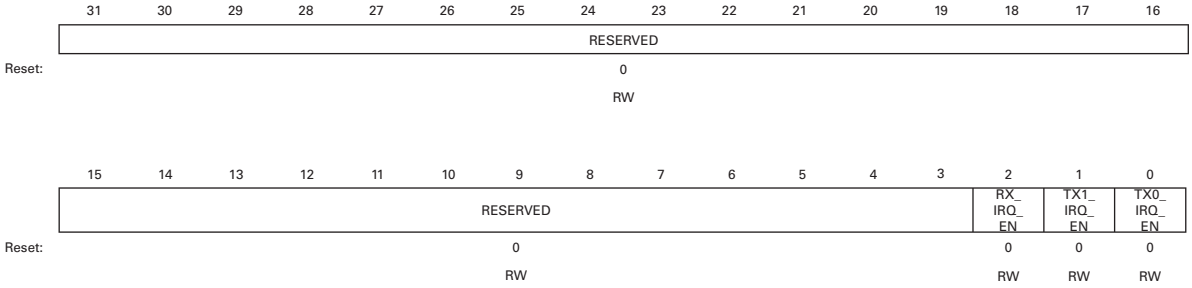
0xcf00 01e4



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

0xcf00 01e8

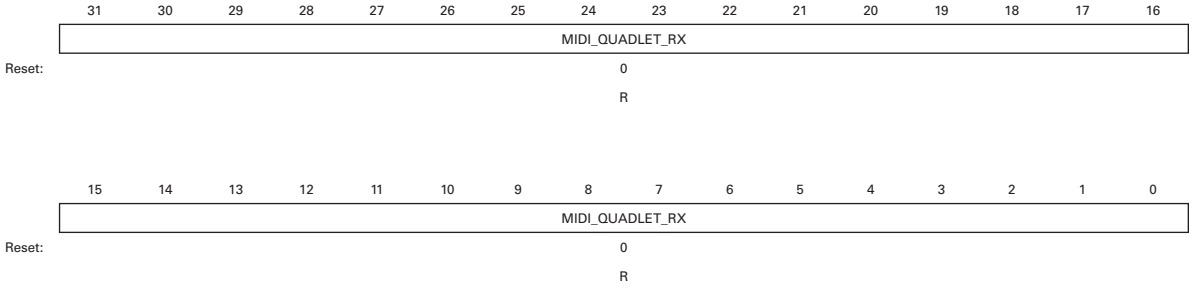


Name	Bits	Reset	Dir	Description
rx_irq_en	2	0	RW	Rx interrupt enable
tx1_irq_en	1	0	RW	Tx1 interrupt enable
tx0_irq_en	0	0	RW	Tx0 interrupt enable

The CTRL register is both readable and writable.

6.8.4 AVSMIDI_RX

0xcf00 01ec

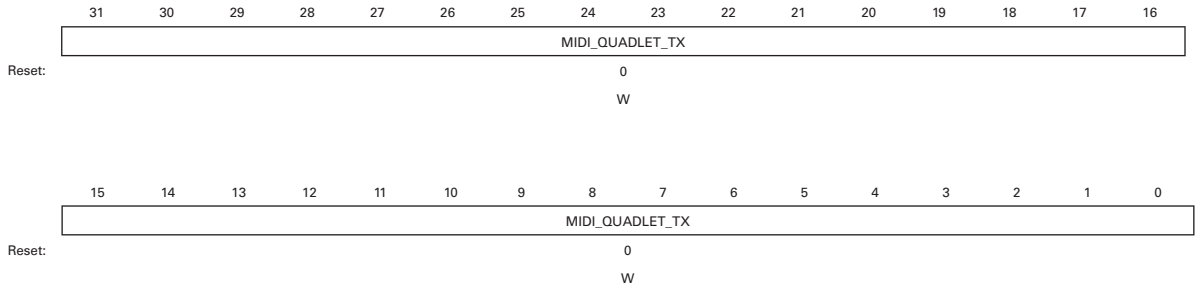


Name	Bits	Reset	Dir	Description
MIDI quadlet data	31:0	0	R	MIDI quadlet data with the following format: [31:29] MIDI port mapping [28:27] source MIDI machine number (AVS Rx0-3) [26] '0' [25:24] counter (number of valid bytes) [23:16] MIDI byte 1 [15:8] MIDI byte 2 [7:0] MIDI byte 3 This format is similar to the one defined in IEC 61883-6.

The RX register is read-only.

6.8.5 AVSMIDI_TXN

0xcf00 01f0



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

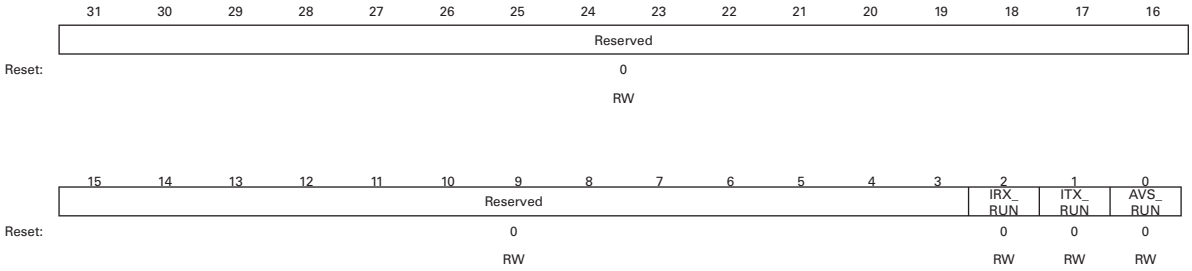
6.9.1 MODULE CONFIGURATION

Address	Register
0xc800 0000	PDB_INT (AVC_CTRL)

Table 6.8: AVS General Memory Map

6.9.2 PDB_INT (AVC_CTRL)

0xc800 0000



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

DICE II, 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 below figure shows the typical connections to DICE II.

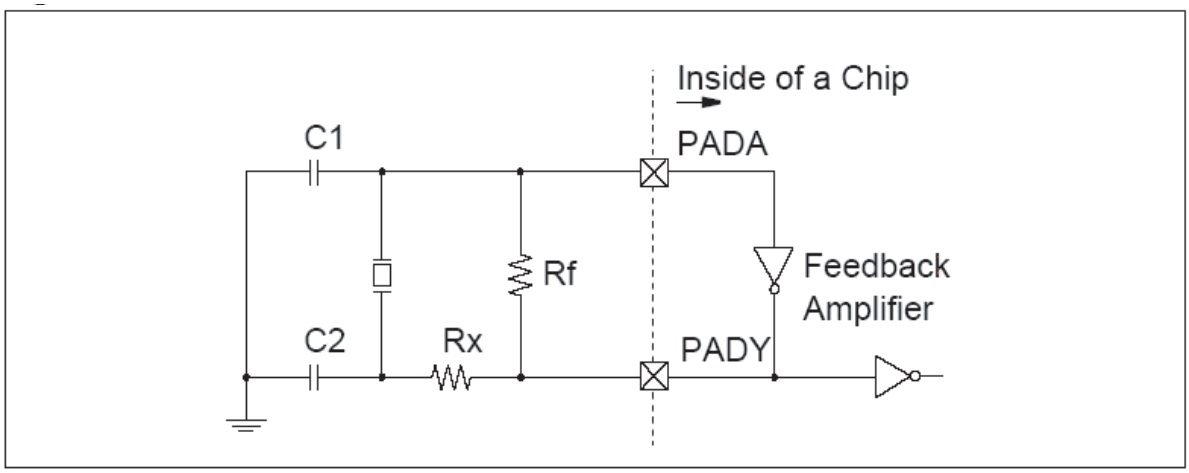


Figure 7.1: On-Chip oscillator typical connections

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.

7.0.1 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.

7.0.1.1 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-to-pin capacitances are about 7pF each.

7.0.1.2 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.

7.0.1.3 RF AND RX SELECTION

A large Rf (1MΩ) 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 kΩ 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.

7.0.1.4 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		285 mA		Audio subsystem not started
Normal Operation		311 mA	340mA	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		< 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.

PLL supply at 3.3v and ambient temperature of 20 deg.

Condition	Min	typ	Max	Comment
All VCO's running		18mA		

IO supply at 3.3v, 30pF load, and ambient temperature of 20 deg.

Condition	Min	typ	Max	Comment
1394 PHY Interface		6mA		@24.576MHz
AES3 Outputs		1.4mA		At 96kHz
DSAI Outputs		10mA		At 96kHz
I2S Outputs		17mA		At 96kHz
External Bus Interface		70mA		0-D15; A0-A23; bus control pins
Remaining IO		20mA		

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
V_T	Switching threshold				1.4		V
V_{T+}	Schmitt trigger, positive-going threshold		CMOS			2.0	V
V_{T-}	Schmitt trigger, negative-going threshold		CMOS	0.8			V
I_{IH}	High level input current	Input buffer	$V_{IN} = V_{DD}$	-10		10	µA
		Input buffer with pull-down		10	33	60	µA
I_{IL}	Low level input current	Input buffer	$V_{IN} = V_{SS}$	-10		10	µA
		Input buffer with pull-down		-60	-33	-10	µA
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			V
		Type B2	I_{OH} = -2mA				V
		Type B4	I_{OH} = -4mA				V
		Type B8	I_{OH} = -8mA				V
		Type B12	I_{OH} = -12mA				V
		Type B16	I_{OH} = -16mA				V
		Type B20	I_{OH} = -20mA				V
		Type B24	I_{OH} = -24mA				V
V_{OL}	Low level output voltage	Type B1 to B24	I_{OH} = 1µA			0.05	V
		Type B1	I_{OH} = 1mA			0.4	V
		Type B2	I_{OH} = 2mA				V
		Type B4	I_{OH} = 4mA				V
		Type B8	I_{OH} = 8mA				V
		Type B12	I_{OH} = 12mA				V
		Type B16	I_{OH} = 16mA				V
		Type B20	I_{OH} = 20mA				V
		Type B24	I_{OH} = 24mA				V
I_{OZ}	Tri-state output leakage current		$V_{OUT} = V_{DD}$ or V_{SS}	-10		10	µA
I_{DD}	Quiescent supply current					100	µA
C_{IN}	Input capacitance		Any input and bi-directional buffers			4	pF
C_{OUT}	Output capacitance		Any output buffer				pF

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
V_{IN}	DC input voltage	3.3V V_{DD}	-0.5	4.8	
		3.3V input buffer	-0.5	4.8	
V_{OUT}	DC output voltage	3.3V interface/ 5V tolerant input buffer	-0.5	6.5	
		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	
V_{IN}	DC supply voltage for analog core (=V_{DDA}) DC input voltage	1.8V V_{DD}	1.8 – 5%	1.8 + 5%	
		3.3V input buffer	-0.3	V_{DCIO} + 0.3	
V_{OUT}	DC output voltage	3.3V interface/ 5V tolerant input buffer	-0.3	5.5	
		3.3V output buffer	-0.3	V_{DCIO} + 0.3	
T_A	3.3V interface/ 5V tolerant output buffer -0.3			5.5	°C
	Commercial temperature range			0 to 70	
	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	–	40	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

9.1 Thermal Ratings

9.1.1 OPERATING TEMPERATURE

	MIN	TYP	MAX	UNIT
Operating ambient temperature				°C

9.1.2 THERMAL CHARACTERISTICS

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
272 PBGA R JA, high–K board	Board mounted, no air flow				°C /W
272 PBGA R JA, low–K board	Board mounted, no air flow				°C /W

9.1.3 DISSIPATION RATING TABLE

PACKAGE	TA = 25°C POWER RATING	DERATING FACTOR ABOVE TA = 25°C	TA = 70°C POWER RATING
high–K board			
low–K board			

9.1.4 ABSOLUTE MAXIMUM RATINGS OVER OPERATING TEMPERATURE RANGES

Supply voltage range AV_{dd} -	0. X V to X V
V_{dd}	0. X V to X V
PLL_V_{dd}	0. X V to X V
Input clamp current I_{IK} ($V_I < 0$ or $V_I > V_{DD}$)	± X mA
Output clamp current I_{OK} ($V_O < 0$ or $V_O > V_{DD}$)	± X mA
Electrostatic discharge	HBM: X kV
Continuous total power dissipation	See Dissipation Rating Table
Operating free–air temperature T_A -	X °C to X °C
Storage temperature range T_{stg} -	X °C to X °C
Lead temperature 1.6 mm (1/16 inch) from cage for 10 seconds	X °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.

Chapter 10 Soldering

10.1 Soldering

10.1.1 MOISTURE SENSITIVITY LEVEL

As with all BGA components, DICE II is sensitive to moisture. The Moisture Sensitivity Level (MSL) rating per JEDEC standard J-STD-020A is an indicator of the maximum allowable time period (floor life time) that a device, once removed from the dry bag, can be exposed to an environment with a maximum temperature of 30°C and a maximum relative humidity of 60% RH. before solder reflow.

The MSL rating of DICE II is MSL-3 (168 hours) .

10.1.2 BALL COMPOSITION

DICE II is a lead-free component. The composition of its balls is: Sn 96.5%, Ag 3.0%, Cu 0.5%.

10.1.2 REFLOW PROFILE FOR THE LEAD-FREE SOLDER BALL

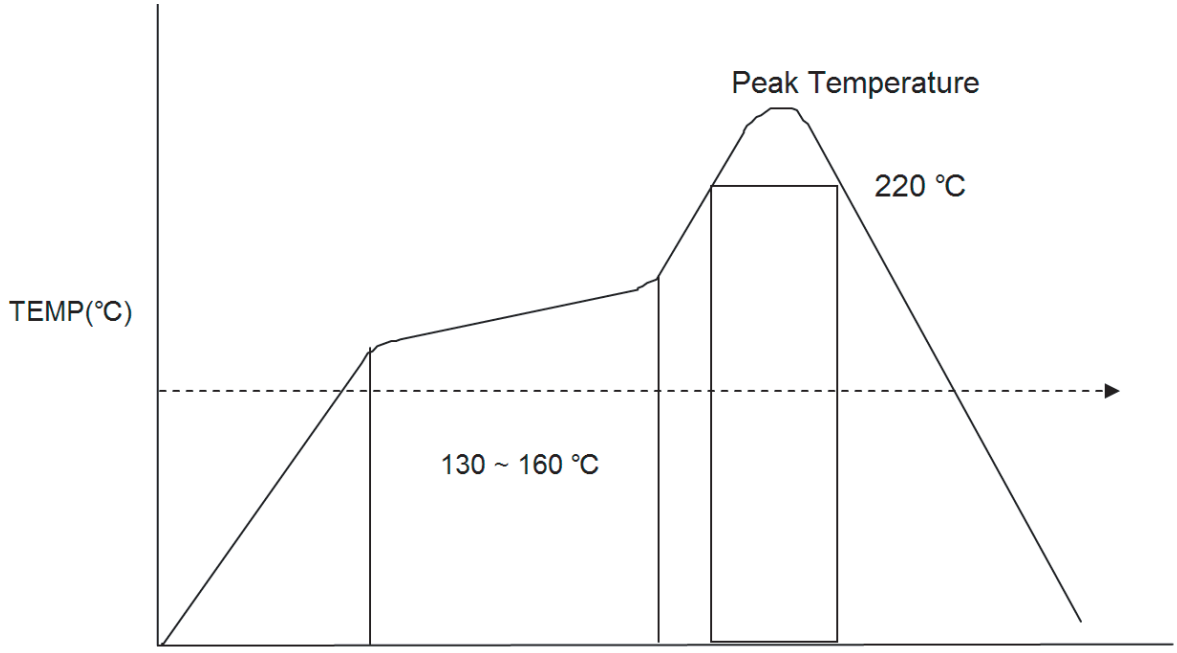


Figure 10.1: Soldering Temperature Profile

Dwell time (above 220°C)	35 - 90 second
Dwell time between (130 - 160°C)	50 - 180 second
Peak Temperature	230 - 255°C

Table 10.1: Soldering Temperature Profile

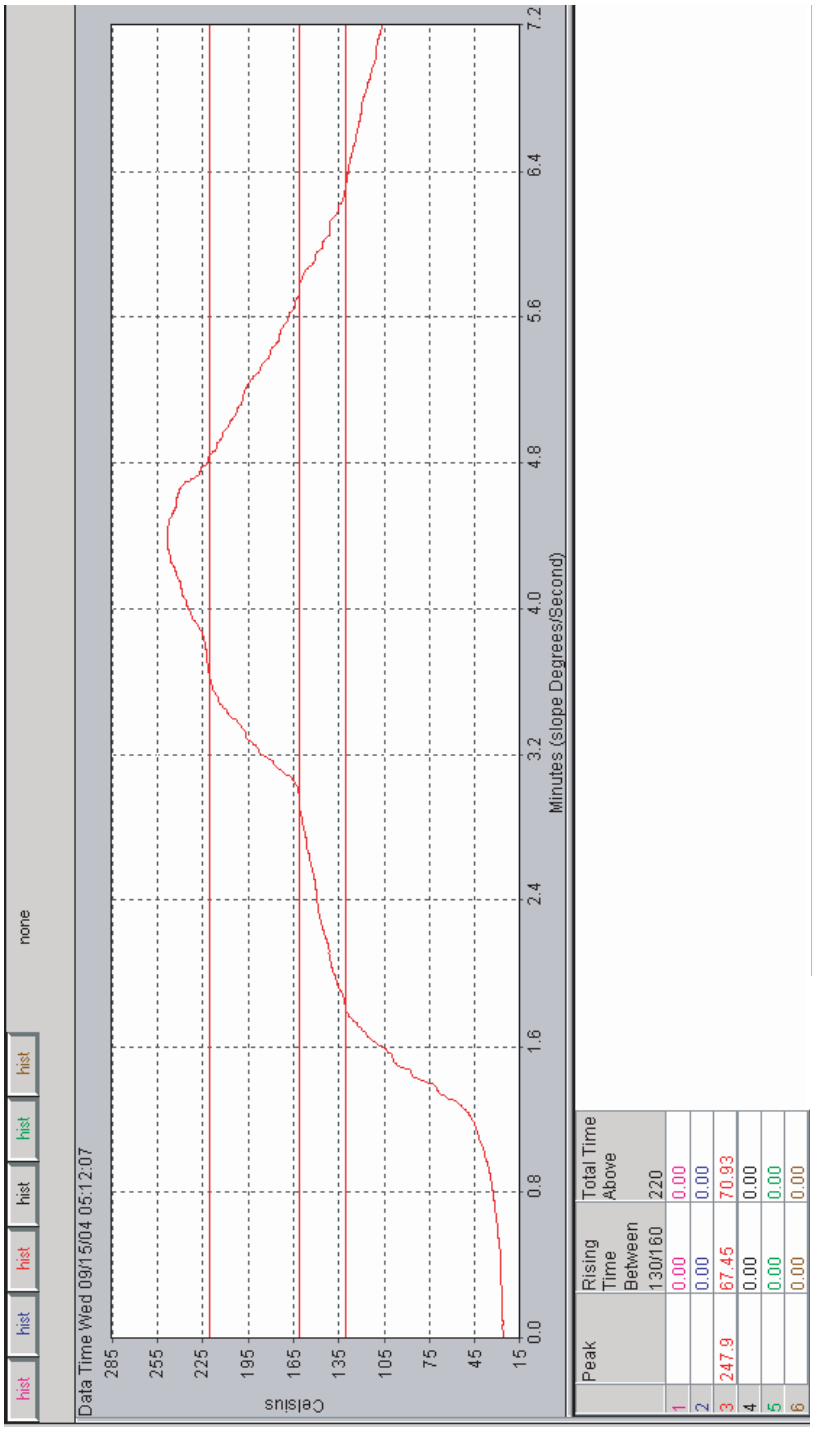


Figure 10.2: Reflow Profile Sample

Appendix 1 Memory Map and Register Summary

A.1 Memory Map

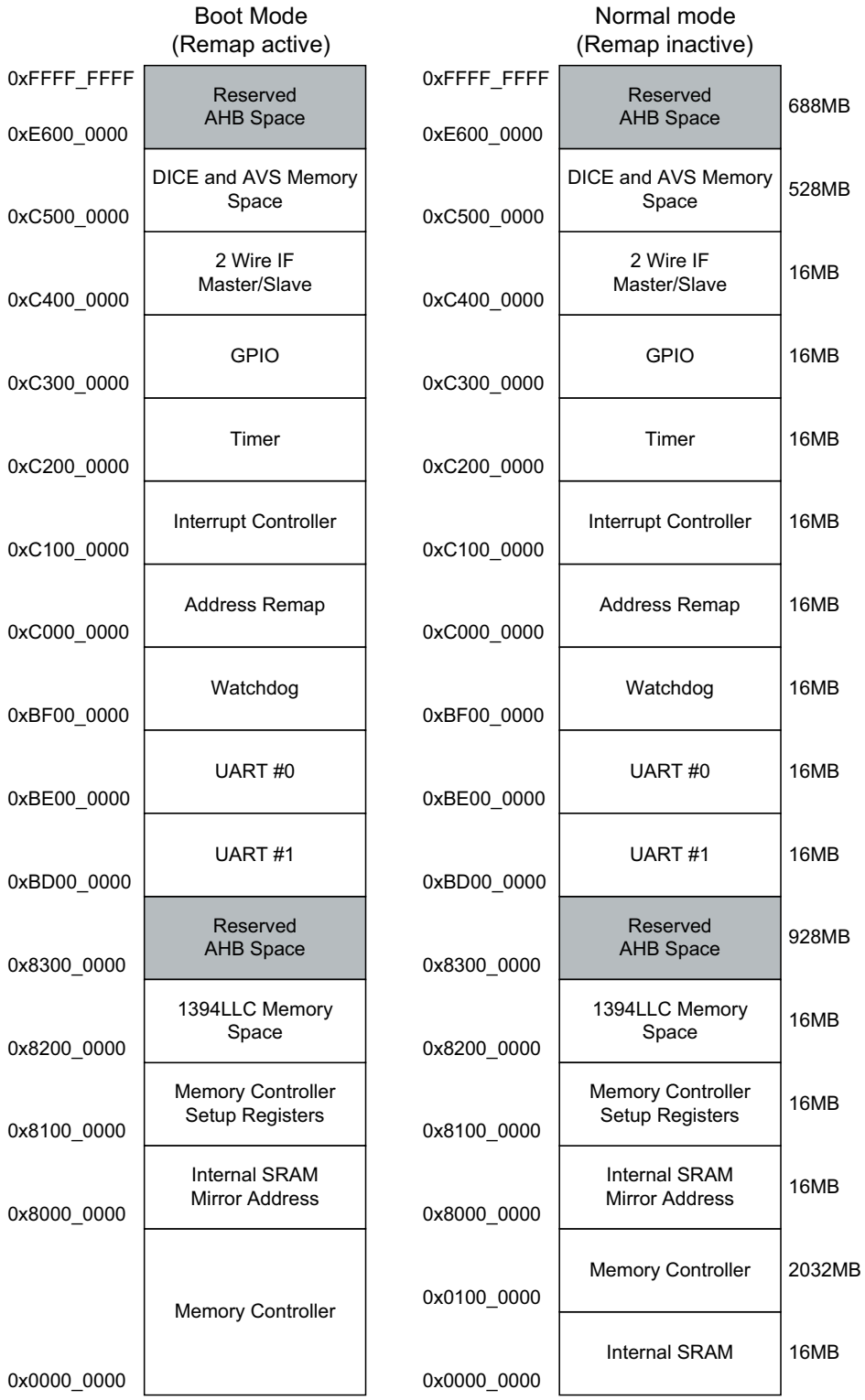


Figure A.1: Global Memory Map (allocated address space)

A.2 DICE II Register Summary

GPSR (General Purpose CSR) Memory Map	See Chapter 4 – ARM Peripherals
Address	**Register**
0xC700 0000	GPCSR_SYSTEM
0xC700 0004	GPCSR_IO_SELECT0
0xC700 0008	GPCSR_DSAI_SELECT
0xC700 000c	GPCSR_DSAI_CLOCK_INV
0xC700 0010	GPCSR_VIDEO_SELECT
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

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

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
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

UART 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
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

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

1394 Link Layer Controller - 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

1394 Link Layer Controller - Memory Map	See Chapter 4 – ARM Peripherals
Address	**Register**
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

Gray Code Interface Memory Map	See Chapter 4 – ARM Peripherals
Address	**Register**
0xc600 0000	GRAY_STAT
0xc600 0004	GRAY_CTRL
0xc600 0008	GRAY_CNT

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
0xc100 00cc	INTCTRL_FIQ_RAW
0xc100 00d0	INTCTRL_FIQ_STAT
0xc100 00d4	INTCTRL_FIQ_FINALSTAT
0xc100 00d8	INTCTRL_SYSTEM_PRIORITY_LEVEL

Watch Dog Memory Map	See Chapter 4 – ARM Peripherals
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

Dual Timer Memory Map	See Chapter 4 – ARM Peripherals
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

A2.2.2 DICE

Router Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce00 0000	ROUTER0_CTRL
0xce00 0400	ROUTER0_ENTRY0
0xce00 0404	ROUTER0_ENTRY1
:	:
0xce00 07fc	ROUTER0_ENTRY255
0xce00 0800	ROUTER1_CTRL
0xce00 0c00	ROUTER1_ENTRY0
0xce00 0c04	ROUTER1_ENTRY1
:	:
0xce00 0ffc	ROUTER1_ENTRY255

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	RX_DOMAIN
0xce01 0020	TX_DOMAIN
0xce01 0024	AES_VCO_SETUP
0xce01 0028	ADAT_VCO_SETUP
0xce01 002c	TDIF_VCO_SETUP
0xce01 0030	SMUX
0xce01 0034	PRESCALER1
0xce01 0038	PRESCALER2
0xce01 003c	HPLL_REF
0xce01 0040	SRCNT1
0xce01 0044	SRCNT2
0xce01 0048	SR_MAX_CNT1
0xce01 004c	SR_MAX_CNT2

JET 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
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

JET PLL Memory Map	See Chapter 5 – DICE
Address	Register
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
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

JET PLL Memory Map	See Chapter 5 – DICE
Address	Register
0xcc00039c	PLL1_SMALLEST_NP_$
0xcc0003d8	PLL1_GPI
0xcc0003e0	PLL1_CONFIG_AC
0xcc0003f0	PLL1_SHUTDOWN_M
0xcc0003f4	PLL1_SHUTDOWN_I
0xcc0001f8	PLL1_SHUTDOWN_E
0xcd00 0000	PLL2_CAF_ENABLE
0xcd00 0004	PLL2_CAF_SELECT
0xcd00 0008	PLL2_COAST
0xcd00 0018	PLL2_REF_SEL
0xcd00 001c	PLL2_REF_EDG
0xcd00 0028	PLL2_RDIV
0xcd00 002c	PLL2_THROTTLE
0xcd00 0058	PLL2_U_THRESHOLD
0xcd00 0060	PLL2_BW_FLOOR
0xcd00 0064	PLL2_BW_CEILING
0xcd00 0068	PLL2_SHP_FIX
0xcd00 006c	PLL2_SHP_VAR
0xcd00 0070	PLL2_MAX_SLW_FIX
0xcd00 0074	PLL2_MAX_SLW_VAR
0xcd00 0078	PLL2_DCNT_LIN
0xcd00 007c	PLL2_DCNT_EXP
0xcd00 0088	PLL2_LOOSE_THR
0xcd00 0098	PLL2_MIN_PER
0xcd00 009c	PLL2_MAX_PER
0xcd00 00b0	PLL2_NDIV_F
0xcd00 00b4	PLL2_NDIV_E
0xcd00 00b8	PLL2_NDIV_B
0xcd00 00bc	PLL2_BYP_F
0xcd00 00c0	PLL2_PHASE_LAG
0xcd00 00c8	PLL2_FRACT_RES
0xcd00 00d0	PLL2_BURST_LEN
0xcd00 00d8	PLL2_GPO_EN
0xcd00 00dc	PLL2_GPO_1
0xcd00 00e0	PLL2_GPO_2
0xcd00 00e4	PLL2_GPO_3
0xcd00 00f0	PLL2_X1X2_MODE
0xcd000100	PLL2_CHAIN_I
0xcd000104	PLL2_SINK_I
0xcd000108	PLL2_ANCHOR_I

JET PLL Memory Map	See Chapter 5 – DICE
Address	Register
0xcd00010c	PLL2_IANCHOR_VAL
0xcd000110	PLL2_UNBND_I
0xcd000118	PLL2_IDET
0xcd000120	PLL2_IDIV_C
0xcd000124	PLL2_IDIV_F
0xcd000128	PLL2_IDIV_S
0xcd000130	PLL2_INV_CDI
0xcd000134	PLL2_HBL_CDI
0xcd000144	PLL2_SINK_E
0xcd000148	PLL2_ANCHOR_E
0xcd00014c	PLL2_E_ANC_VAL
0xcd000150	PLL2_UNBIND_E
0xcd000158	PLL2_EDET_X1
0xcd00015c	PLL2_EDET_X2
0xcd000160	PLL2_EDIV_C
0xcd000164	PLL2_EDIV_F
0xcd000168	PLL2_EDIV_S
0xcd000170	PLL2_INV_CDE
0xcd000174	PLL2_HBL_CDE
0xcd000180	PLL2_DIVIDE_CJ
0xcd000184	PLL2_INVERT_CJ
0xcd000280	PLL2_FAMILY_ID
0xcd000284	PLL2_FORM_ID
0xcd000288	PLL2_REVISION_ID
0xcd00028c	PLL2_INSTANCE_ID
0xcd0002b8	PLL2_MTR_SELECT
0xcd0002bc	PLL2_MTR_EDGES
0xcd0002c0	PLL2_RES_EX
0xcd0002c4	PLL2_PUNC_MP
0xcd0002cc	PLL2_MTR_PERIOD
0xcd0002d0	PLL2_GREATEST_MP
0xcd0002d4	PLL2_GREATEST_MP_$
0xcd0002d8	PLL2_SMALLEST_MP
0xcd0002dc	PLL2_SMALLEST_MP_$
0xcd000300	PLL2_TICK_RATE
0xcd000304	PLL2_TURN_RATE
0xcd000308	PLL2_MAIN_STATUS
0xcd00030c	PLL2_MAIN_STATUS_$
0xcd000320	PLL2_DETECT_R
0xcd000324	PLL2_DETECT_F
0xcd000328	PLL2_STICKY_BITS

JET PLL Memory Map	See Chapter 5 – DICE
Address	**Register**
0xcd00032c	PLL2_STICKY_BITS_$
0xcd000350	PLL2_IRQ_ENABLES
0xcd00038c	PLL2_NCO_PERIOD
0xcd000390	PLL2_GREATEST_NP
0xcd000394	PLL2_GREATEST_NP_$
0xcd000398	PLL2_SMALLEST_NP
0xcd00039c	PLL2_SMALLEST_NP_$
0xcd0003d8	PLL2_GPI
0xcd0003e0	PLL2_CONFIG_AC
0xcd0003f0	PLL2_SHUTDOWN_M
0xcd0003f4	PLL2_SHUTDOWN_I
0xcd0001f8	PLL2_SHUTDOWN_E

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
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_SEL1
0xce03 000c	CS_SEL2
0xce03 0010	CS_SEL3
0xce03 0014	MUTE
0xce03 0018	V_BIT
0xce03 0040	USR_SEL1
0xce03 0044	USR_SEL2
0xce03 0048	USR_SEL3
0xce03 004c	USR_SEL4
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

I²S Receiver Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce10 0000	I2S_RX1_MODE
0xce10 0004	I2S_RX1_CH1_CTRL
0xce10 0008	I2S_RX1_CH2_CTRL
0xce10 000c	I2S_RX1_CH3_CTRL
0xce10 0010	I2S_RX1_CH4_CTRL
0xce12 0000	I2S_RX2_MODE
0xce12 0004	I2S_RX2_CH1_CTRL
0xce12 0008	I2S_RX2_CH2_CTRL
0xce14 0000	I2S_RX3_MODE
0xce14 0004	I2S_RX3_CH1_CTRL
0xce14 0008	I2S_RX3_CH2_CTRL

I²STransmitter Memory Map	See Chapter 5 – DICE
Address	Register
0xce11 0000	I2S_TX0_MODE
0xce11 0004	I2S_TX0_D0_CTRL
0xce11 0008	I2S_TX0_D1_CTRL
0xce11 000c	I2S_TX0_D2_CTRL
0xce11 0010	I2S_TX0_D3_CTRL
0xce11 0014	I2S_TX0_MUTE
0xce13 0000	I2S_TX1_MODE
0xce13 0004	I2S_TX1_D0_CTRL
0xce13 0008	I2S_TX1_D1_CTRL
0xce13 0014	I2S_TX1_MUTE
0xce15 0000	I2S_TX2_MODE
0xce15 0004	I2S_TX2_D0_CTRL
0xce15 0008	I2S_TX2_D1_CTRL
0xce15 0014	I2S_TX2_MUTE

ADAT Receiver Memory Map	See Chapter 5 – DICE
Address	Register
0xce04 0000	ADATRX_CTRL
0xce04 0004	ADATRX_STAT

ADAT ransmitter Memory Map	See Chapter 5 – DICE
Address	Register
0xce05 0000	ADATTX_CTRL1
0xce05 0004	ADATTX_CTRL2
0xce05 0008	ADATTX_CTRL3

TDIF Transmitter Memory Map	See Chapter 5 – DICE
Address	Register
0xce07 0000	TDIF_TX_EMPH_CH0_CFG
0xce07 0004	TDIF_TX_ FS0_CH1_CFG
0xce07 0008	TDIF_TX_FS1_CH2_CFG
0xce07 000c	TDIF_TX_CH3_CFG
0xce07 0010	TDIF_TX_CH4_CFG
0xce07 0014	TDIF_TX_CH5_CFG
0xce07 0018	TDIF_TX_CH6_CFG
0xce07 001c	TDIF_TX_CH7_CFG
0xce07 0020	TDIF_TX_MUTE
0xce07 0024	TDIF_TX_INV_CTRL

TDIF Receiver Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce06 0000	TDIF_RX_CH0/1_CFG
0xce06 0004	TDIF_RX_CH2/3_CFG
0xce06 0008	TDIF_RX_CH4/5_CFG
0xce06 000c	TDIF_RX_CH6/7_CFG
0xce06 0010	TDIF_RX_STAT
0xce06 0014	TDIF_RX_CFG
0xce06 0018	TDIF_RX_PHASE_DIFF
0xce06 001c	TDIF_RX_INV_CTRL

DSAI Receiver Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce08 0000	DSAI1_RX_CTRL
0xce08 0008	DSAI1_RX_STAT
0xce0a 0000	DSAI2_RX_CTRL
0xce0a 0008	DSAI2_RX_STAT
0xce0c 0000	DSAI3_ RX_CTRL
0xce0c 0008	DSAI3_ RX_STAT
0xce0e 0000	DSAI4_ RX_CTRL
0xce0e 0008	DSAI4_ RX_STAT

DSAI Transmitter Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce09 0000	DSAI1_TX_CTRL
0xce09 0008	DSAI1_TX_STAT
0xce09 000c	DSAI1_TX_MUTE
0xce0b 0000	DSAI2_TX_CTRL
0xce0b 0008	DSAI2_TX_STAT
0xce0b 000c	DSAI2_TX_MUTE
0xce0d 0000	DSAI3_TX_CTRL
0xce0d 0008	DSAI3_TX_STAT
0xce0d 000c	DSAI3_TX_MUTE
0xce0f 0000	DSAI4_TX_CTRL
0xce0f 0008	DSAI4_TX_STAT
0xce0f 000c	DSAI4_TX_MUTE

ARM Audio Transceiver Memory Map	See Chapter 5 – DICE
Address	**Register**
0xce16 0000 – 0xce16 0080	ARMAUDIO_BUF
0xce16 0100	ARMAUDIO_CTRL

A1.2.3 AVS

AVS Audio Receiver 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

AVS Audio Receiver Memory Map	See Chapter 6 – AVS
Address	Register
0xcf00 005c	ARX2_ADO_MIDI
0xcf00 0060	ARX3_CFG0
0xcf00 0064	ARX3_CFG1
0xcf00 0068	ARX3_QSEL0
0xcf00 006c	ARX3_QSEL1
0xcf00 0070	ARX3_QSEL2
0xcf00 0074	ARX3_QSEL3
0xcf00 0078	ARX3_QSEL4
0xcf00 007c	ARX1_PHDR
0xcf00 0080	ARX1_CIP0
0xcf00 0084	ARX1_CIP1
0xcf00 0088	ARX3_ADO_CFG
0xcf00 008c	ARX3_ADO_MIDI
0xcf00 0090	ARX4_CFG0
0xcf00 0094	ARX4_CFG1
0xcf00 0098	ARX4_QSEL0
0xcf00 009c	ARX4_QSEL1
0xcf00 00a0	ARX4_QSEL2
0xcf00 00a4	ARX4_QSEL3
0xcf00 00a8	ARX4_QSEL4
0xcf00 00ac	ARX1_PHDR
0xcf00 00b0	ARX1_CIP0
0xcf00 00b4	ARX1_CIP1
0xcf00 00b8	ARX4_ADO_CFG
0xcf00 00bc	ARX4_ADO_MIDI

AVS Audio Transmitter 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 Format Handler 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
0xcf00 0260	FORMAT_RXDI3_CFG
0xcf00 0264	FORMAT_RXDI3_LABELn
0xcf00 0274	FORMAT_RXDI3_CSBLOCKn
0xcf00 0290	FORMAT_RXDI4_CFG
0xcf00 0294	FORMAT_RXDI4_LABELn
0xcf00 02a4	FORMAT_RXDI4_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
0xcf00 014c	AVSI_INT2_STATUS
0xcf00 0150	AVSI_INT2_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 01b4	AVSFIFO_PART4_BASE
0xcf00 01b8	AVSFIFO_PART4_LIMIT
0xcf00 01bc	AVSFIFO_PART4_FLUSH
0xcf00 01c0	AVSFIFO_PART5_BASE
0xcf00 01c4	AVSFIFO_PART5_LIMIT
0xcf00 01c8	AVSFIFO_PART5_FLUSH
0xcf00 01cc	AVSFIFO_PART6_BASE
0xcf00 01d0	AVSFIFO_PART6_LIMIT
0xcf00 01d4	AVSFIFO_PART6_FLUSH
0xcf00 01d8	AVSFIFO_PART7_BASE
0xcf00 01dc	AVSFIFO_PART7_LIMIT
0xcf00 01e0	AVSFIFO_PART7_FLUSH
0xcf00 01fc	AVSFIFO_STAT

AVSMIDI 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)