TECHNICAL MANUAL

DMN-8600
DVD Recorder
System Processor

July 2002

Preliminary-3

LSI Logic Confidential

LSI LOGIC®

This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified either the functional descriptions, or the electrical and mechanical specifications using production parts.

Document DB14-000198-02, Third Edition (July 2002)
This document describes the LSI Logic DMN-8600 and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

TRADEMARK ACKNOWLEDGMENT
LSI Logic, the LSI Logic logo design, DoMiNo, and ZiVA are trademarks or registered trademarks of LSI Logic Corporation. All other brand and product names may be trademarks of their respective companies.

GS

To receive product literature, visit us at http://www.lsilogic.com.

For a current list of our distributors, sales offices, and design resource centers, view our web page located at

http://www.lsilogic.com/contacts/na_salesoffices.html

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Preface

This book is the primary reference and technical manual for the DMN-8600 advanced, scalable DVD recorder processor. It contains a complete functional description for the DMN-8600 and includes complete physical and electrical specifications for the DMN-8600.

Audience

This document assumes that you have some familiarity with microprocessors and related support devices. The people who benefit from this book are

- Engineers and managers who are evaluating the processor for possible use in a system
- Engineers who are designing the processor into a system

Organization

This document has the following chapters and appendixes:

- Chapter 1, **Overview**, defines DMN-8600 features and introduces its main applications.
- Chapter 2, **Application Example**, describes the common user interface shared by all DMN-8600 applications and provides a listing of components needed to complete the design.
- Chapter 3, **Internal Architecture**, gives an overview of the main modules and their external interfaces.
- Chapter 4, **Functional Description**, provides an extended description to the main modules and their external interfaces.
- Chapter 5, **Programmability (C-Ware)**, provides an overview of how the DMN-8600 is programmed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- Chapter 6, **Signal Descriptions**, describes signals within their respective functional groups.
- Chapter 7, **Memory Mapping**, describes the addresses for different configurations, including access to SDRAM and internal registers via the Control Bus (CBus).
- Chapter 8, **Host Slave Interface**, gives an extended description to the different configurations and features of the Host slave interface; includes registers.
- Chapter 9, **Secondary Bitstream Interface**, gives an extended description to the Secondary Bitstream interface; includes registers.
- Chapter 10, **Host Async Master Interface**, gives an extended description to the different configurations and features of the Host Async master interface; includes registers.
- Chapter 11, **Video Interface**, gives an extended description to the video interface; includes registers.
- Chapter 12, **Audio Interface**, gives an extended description to the audio interface; includes registers.
- Chapter 13, **SDRAM Interface**, gives an extended description to the SDRAM interface; includes registers.
- Chapter 14, **Bitstream I/O (Storage) Port**, gives an extended description to the bitstream I/O interface; includes registers.
- Chapter 15, **Serial I/O Port**, gives an extended description to the serial I/O interface; includes registers.
- Chapter 16, **Clock Control and Power Management**, describes clock PLLs and power modes; includes registers.
- Chapter 17, **JTAG Boundary Scan**, describes JTAG boundary scan interface in accordance with IEEE 1149.1; includes cell listing.
- Chapter 18, **Specifications**, describes electrical characteristics as well as AC timing; includes pin list alphabetized according to both pin name and pin bondout location.
- Chapter A, **Register Listing**, is an index of the DMN-8600 registers.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Related Publications

DoMiNo Full Feature Network Media Processor Family Preliminary Technical Manual, Document number DB14-000197-00

DoMiNo Dual Stream Network Media Processor Family Advance Product Datasheet, Document number TD10-1016, Rev 0.10

DMN-8600 Applications User Guide, Document number TD10-1032 (Pre-Advance)

DMN-8600 DVD Recorder System Processor Programming Guide, Volume 1, Document number DB15-000233-01
Volume 2, Document number DB15-000246-00

DMN-8100 Advanced Multiformat A/V Processor for Video Peripheral Applications Technical Manual, Document number DB14-000217-01

Galileo CE Development Board Preliminary User's Guide, Document number DB15-000229-01

Galileo Light (Galileo LT) Development Board Advance User's Guide, Document number DB15-000240-00

Conventions Used in This Manual

The first time a word or phrase is defined in this manual, it is *italicized.*

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive. Signals that are active LOW are marked with an overbar symbol.

Hexadecimal numbers are indicated by the prefix "0x" —for example, 0x32CF. Binary numbers are indicated by the prefix "0b" —for example, 0b0011.0010.1100.1111.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 1 Revision History

Date	Part No.	Description
February 20, 2001	91-E5SD-SS-ADV-1	Initial 308 BGA datasheet with signal descriptions.
May 3, 2001	TD10-1017, Rev. 0.10	Pin numbers assigned to signals and part productized as DMN-8600.
August 15, 2001	TD10-1017, Rev. 0.20	Added register, extended interface descriptions, and AC/DC information.
August 30, 2001	TD10-1017, Rev. 0.21	Special Edition
September 18, 2001	TD10-1017, Rev. 0.22	Special Edition - 2nd Edition
October 2, 2001	DB14-000198-00	Preliminary Edition
January 31, 2002	DB14-000198-01	Preliminary-2 Edition. Chapter 15 expanded, etc.
July 31, 2002	DB14-000198-02	Preliminary-3 Edition. Assorted changes made.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Contents

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figures

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Tables

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 1
Overview

The DMN-8600 is the LSI Logic system-on-a-chip Audio/Video/System (AVS) recorder processor product for DVD recorder applications based on the LSI Logic DoMiNo™ architecture. The DMN-8600 is intended for demanding, yet low-cost, single-stream AV applications in consumer, prosumer, and professional markets.

The DMN-8600 handles MPEG-1, MPEG-2, and DV compression/ decompression. In addition, the DMN-8600 is capable of transcoding between and transrating within these formats. The DMN-8600 represents a quantum leap in performance, allowing multiple A/V pipelines to execute simultaneously.

Relying on its internal host processor, the DMN-8600 delivers leading performance for advanced image processing techniques such as deinterlacing and noise reduction by using motion-compensation algorithms in combination with other features, allowing delivery of the highest quality MPEG-1, MPEG-2 and DV compressed images.

The DMN-8600 can also be used as a powerful audio processor that encodes audio into M1L2 and MP3 formats, and decodes AC3, M2L2, DTS, DVD-Audio, and MP3 formats. It can also be programmed to perform post-processing tasks such as 3D audio and Bass management.

The DMN-8600's true multiformat A/V recorder capabilities and high level of on-chip system integration bring unprecedented video quality and functionality to the market for audio/video single-stream product applications, including:

- Consumer Applications
  - Low cost DVD recorders (portable and nonportable)
  - Personal video recorders (PVRs)
  - Home media servers

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

  - Internet enabled networked home media appliances
  - AV capture/playback PC peripherals
  - Digital VHS Recording
- Prosumer/Professional Applications
  - Nonlinear Editing
  - Networked video (production studios)
  - Prosumer and professional MPEG camcorders
  - Professional MPEG VTRs
  - Transraters and transcoders

The DMN-8600 delivers four times the processing performance compared to previous generation DVx processors.

The DMN-8600 integrates dual SPARC processors with pixel-level coprocessors and a complete set of host and I/O interfaces. This architecture balances hardware that has been optimized for performance-critical functions at the bit- and pixel-operations level with firmware that has been optimized for performance and scalability. This balance delivers the flexibility needed to cut time to market and lower development cost. See Table 1.1.

Table 1.1 DMN-8600 Features Summary

Video	A/V Encode/Decode	Multiformat audio and video encode/decode
	Transcoding/rating	Multiformat transcoding, multirate transrating
	Resolutions	Horizontal: 720, 704, 640, 544, 480, 352 Vertical: NTSC (480i, 480p) and PAL (576i)
	Formats	NTSC, PAL, ITU-R BT.656/601
	Input	One 8-bit video stream at 27 MHz
	Output	One 8/16-bit video stream at 27 MHz, with progressive scan output support at 54 (480P) MHz.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 1.1 DMN-8600 Features Summary (Cont.)

Audio	Inputs	Single IDS (inter-device serial) Port, 2 channels
	Outputs	Quad IDS ports, 8 channels
	Resolutions	16 to 24 bits/sample
I/O	Serial	Dual UARTs, IEEE1394 Link, SPI, IR, IR Blaster, IDC
	Parallel	ATAPI/DVD (SD)/SBP
System	Host	16/32-bit generic host interface or internal host
	Graphics	2D, 24-bit RGB, 8-bit alpha channel, OSD, flicker filter and video scaler
	Encryption/decryption	CPRM, CPPM, CSS, 5C (via 1394), Watermark detection
Memory	Memory	8–64 Mbytes of SDRAM, SDR or DDR
	Controller	On-chip, 32-bit-wide SDRAM interface, 148.5 MHz
	Peak Bandwidth	1.2 Gbytes/s
Physical	Input Voltages	3.3 V I/O; 1.8 V Core; 2.5 / 3.3 V DRAM
	System Clock	13.5 or 27 MHz
	Operating Power	< 2.7 W @ V_{DDQ} = 1.8 V typical
	Package	308 Pin Ball Grid Array (BGA)
JTAG	Compliance	IEEE 1149.1 compliance for boundary scan testing and board assembly testing

Finally, the DMN-8600 achieves a high level of performance through system-on-chip (SOC) levels of integration. In addition to the audio and video codec, the DMN-8600 also integrates host, graphics and I/O processor subsystems. I/O and graphics sub-systems include functionality such as:

- High-speed interfaces to external peripherals (1394, ATAPI/DVD(SD)/SBP
- 32-bit 2D graphics engine with alpha blending, flicker filter, and video scaler for OSD, GUI, PiP, side-by-side display, EPG, background textures, Web browsing, advertisements, etc.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

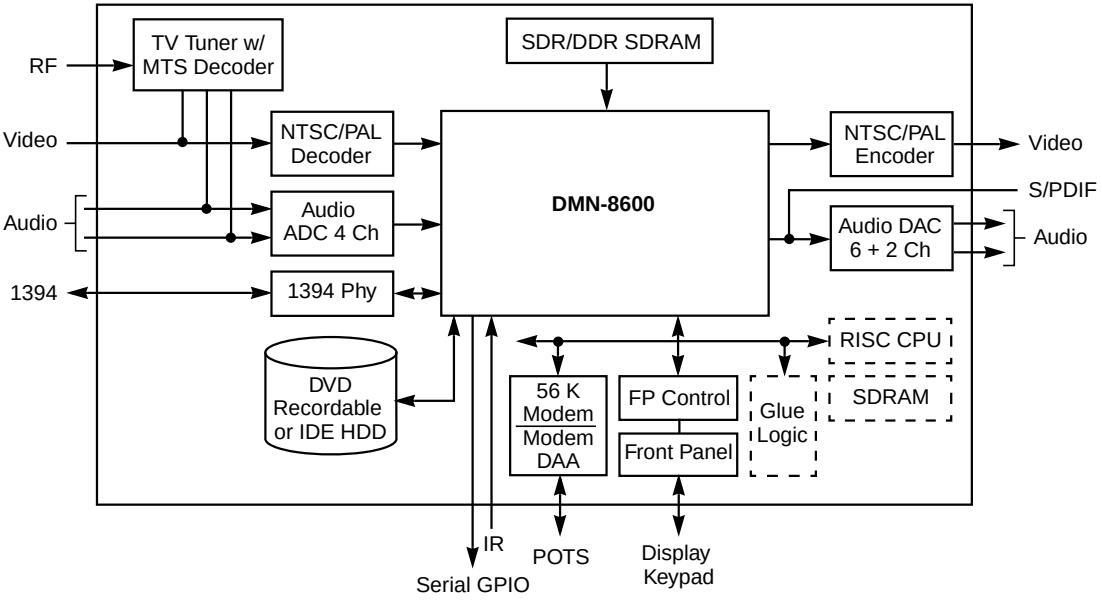
Through such high integration, the DMN-8600 is capable of a high degree of concurrency. Examples of this are simultaneous encoding and decoding in MPEG-2 MP@ML format, simultaneous decoding to MPEG-2 MP@ML and transcoding to DV25, multi-angle view decoding, DV25 to MPEG-2 with zero delay preview, CD-DA to MP3 transcoding, as well as IEEE1394 transport stream muxing and demuxing. The DMN-8600 also supports video recorders with hard disk drives and optical recorder drives in the same system with independent scrambling control.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 2
Application Example

The DMN-8600's high level of on-chip system integration drastically reduces system component count (BOM), and consequently system cost. Figure 2.1 shows a block diagram of an advanced DVD recorder appliance with analog time shift based on a DMN-8600.

Figure 2.1 System Block Diagram for a DMN-8600-Based Advanced DVD Recorder Appliance



The only major components needed to complete the design are:

- 8 to 64 Mbytes SDRAM
- TV tuner
- Analog video encoder and decoder

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- Audio ADCs and DACs
- 56 Kbytes Modem plus DAA
- Firewire 1394 PHY
- Optional host CPU

The DMN-8600 has an on-chip ATAPI controller that supports one master and one slave device from a single, bidirectional port. This makes it possible to seamlessly attach DVD-RAM or other read/writeable DVD drives in dual loader configuration mode, or in combination with a hard disk drive. The DMN-8600 is able to record a single input, either from a TV tuner, the IEEE1394 link, or the ATAPI interface, in MPEG-1 or MPEG-2 format. Similarly, it is capable of decoding any of these formats (including DVD-Audio) and sending the result to the video and audio DACs. The built-in support for progressive video (54 MHz) enables 480p video output. A DV25 camcorder may be hooked up to the IEEE1394 port for transcoding to MPEG-2 format.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 3
Internal Architecture

The DMN-8600 can accomplish very compute-intensive signal processing tasks because it integrates multiple high performance processing modules on a single chip (see Figure 3.1):

- Two SPARC processors, with integrated audio DSP, I-cache and memory, host functions, and a 2D graphics engine with DVD subpicture decode and a 32-bit RGBA OSD/GUI accelerator with scaler and flicker filter
- Video DSP
- Motion estimation unit
- High-performance memory controller (for audio, video and graphics data), with a 32-bit data bus for support of 8 to 64 Mbytes of external SDR or DDR SDRAM, SGRAM or FCRAM

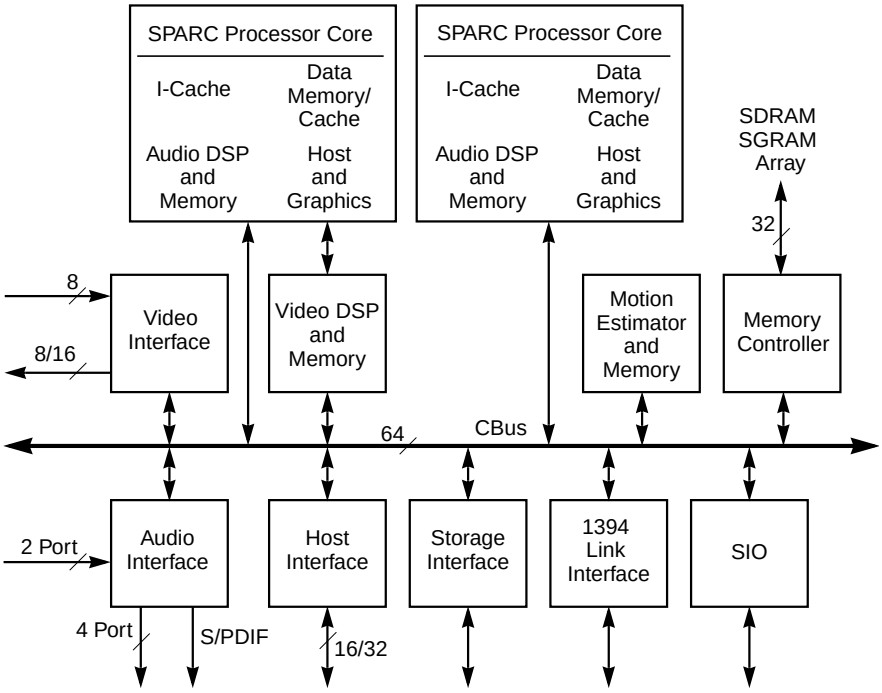
The DMN-8600 supports multiple external interfaces:

- One video input and one video output interface to external ITU-R BT.656/601 video decoder and encoder chips
- Dual-port IDS (inter-device serial) TDM (time-division multiplexed) audio input, and quad-port IDS TDM audio output interfaces to external audio decoder and encoder chips
- IEC-958 S/PDIF audio-out interface
- External generic host bus interface
- Serial I/O interface controller for infrared port, general purpose serial I/O, modem, etc.
- Storage device interface controller for IDE/ATAPI devices or non-ATAPI devices (low-cost optical loaders)
- IEEE1394 link layer interface (with 5C encryption) to a required external Physical Layer chip

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The features and functions of these modules are discussed in “Chapter 4, “Functional Description.”

Figure 3.1 DMN-8600 Internal Architecture Diagram



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 4
Functional Description

This section describes the various modules and capabilities of the DMN-8600 and contains the following sections.

4.1 RISC Engine

Two 32-bit SPARC processors perform three classes of functions: system processing, audio processing, and high-level control flow and decision-making tasks for video processing. Optionally they can also perform 2D graphics and host functions.

The SPARC processors have a programmable, scalable architecture that includes an internal 16 Kbyte instruction cache and an internal configurable 16 Kbyte data memory/cache. Data memory is used instead of a data cache when the software needs predictable real-time performance and overlapped DMA transfers. Each SPARC processor delivers 150 MIPS of processing power.

4.2 Video Processing

As a highly integrated network media processor engine, the DMN-8600 is designed to implement motion-compensated, block-DCT-based video compression algorithms. To this end, the DMN-8600 includes a Video

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

DSP and Motion Estimation coprocessor that off-load compute-intensive tasks from the SPARC processors.

4.2.1 Motion Estimation Coprocessor

The programmable motion estimation (ME) coprocessor has a throughput of 29 billion arithmetic operations per second (BOPS). It takes the motion estimation commands from the SPARC processor and generates results for each target. A single interrupt is generated at the end of each ME stage.

4.2.2 Video DSP Coprocessor

The video DSP coprocessor performs vector memory-to-memory instructions. This improves code density and off-loads the SPARC processors. Its 64 Kbyte data memory is double-buffered (two banks) to allow concurrent DMA and DSP operations. Some of the functions that the DSP coprocessor performs include:

- Detelecine
- Activity measures
- Motion compensation
- Adaptive temporal and de-interlace filtering
- Linear filtering/decimation
- DCT/IDCT (discrete cosine/inverse discrete cosine transforms, up to 12 bits)
- Quantization / Dequantization
- Variable length encoding / decoding

The video DSP coprocessor operates at approximately 16 BOPS.

4.3 Audio Processing

As shown in Figure 2.1, the audio processing hardware is physically integrated into each SPARC processor. The hardware consists of two parallel 63-bit MAC units with 32-bit precision, and a DSP instruction set to support efficient encoding and decoding of a wide variety of audio algorithms. These include MPEG-1 Layer 2, DTS, MP3, AC-3, AAC, etc.,

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

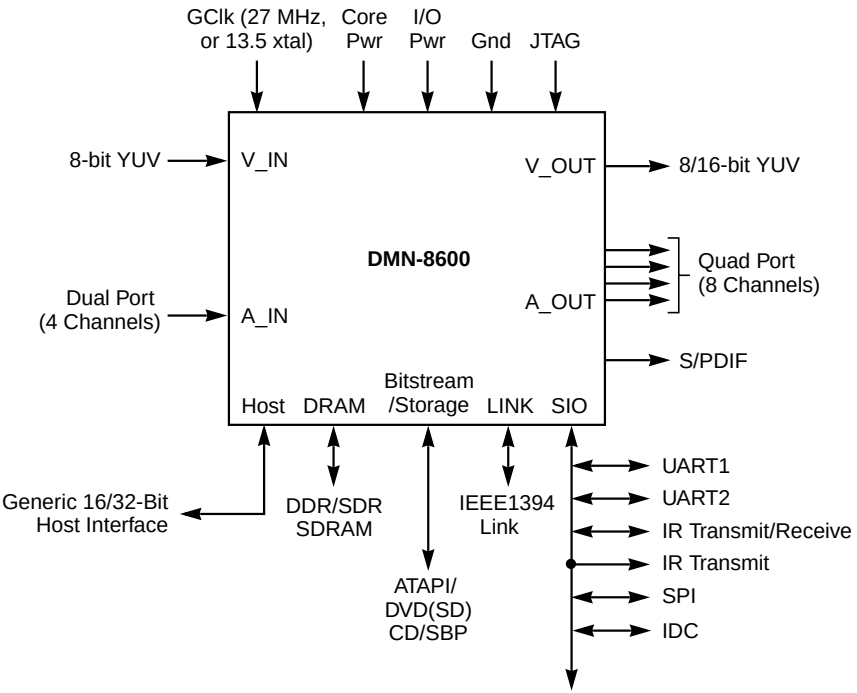
as well as audio special effects such as 3D audio, Dolby ProLogic™, echo, harmonizing, Bass management, FIR and IIR filters.

Each SPARC processor can execute two instructions in parallel, a MAC operation (DSP instruction) and a regular SPARC processor instruction; e.g., there can be an overlap between the load-and-store operations and the multiply-and- accumulate operations.

4.4 Interfaces and I/O

The DMN-8600's connections to the outside world are summarized in Figure 4.1. The following subsections briefly describe each corresponding interface.

Figure 4.1 DMN-8600 System I/O Diagram



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

4.4.1 Host Interface

The generic host interface provides a simple, efficient, low-cost interface to a 16-bit or 32-bit host microcontroller. Three embedded host/async master configurations are supported: async master only, embedded host only, and embedded host plus limited master. The configurable storage and default serial I/O configurations for each of these interfaces are shown in Table 4.1.

Table 4.1 DMN-8600 Storage and Serial I/O Selection

	Embedded Host	Embedded Host plus Limited Master	Embedded Master
Storage	ATAPI or SD and SBP or CD and SBP	ATAPI or SD or SBP	ATAPI or SD and SBP or CD and SBP
Serial I/O	UART2, IDC, IRRX	UART2, IDC, IRRX	UART2, IDC, IRRX
	UART1, SPI, IRTX		UART1, SPI, IRTX

When embedded host plus limited master is selected, the serial I/O pins for UART1, SPI and IR transmit are replaced by the master control pins M_A[26:22, 5:1], $\overline{\text{M\_CS}}$[0], $\overline{\text{M\_OE}}$, $\overline{\text{M\_WE}}$. The master interface M_A[21:6]/M_D[15:0] pins are shared with ATAPI_DATA[15:0] pins when the storage interface is in ATAPI mode, and SD_DATA[7:0]/ SBP_DATA[7:0] when the storage interface is in SD mode.

The limited master is primarily intended to support an external PROM accessed independently of the host processor. Consequently, it only supports one chip select with self-paced cycles.

The host interface provides three distinct functions:

- Local SDRAM access
- Internal register access
- Compressed data input /output (if the dedicated bitstream interfaces are not used for this purpose)

The host communication functions include initializing the DMN-8600, downloading microcode to the local SDRAM, sending commands, monitoring status, and downloading graphics data such as OSD bitmaps.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The host interface supports two types of read/write protocols:

- Individual (I) mode
- Multiplexed (M) mode

In I mode, the decoder requires the host to drive separate signals (RD and WR) to indicate whether a transfer is a read or a write. In M mode, the decoder requires one direction signal (RD or WR) and a data strobe. This selection allows the DMN-8600 to operate with most popular microcontrollers.

4.4.2 Bitstream/Storage Interface

ATAPI, DVD (SD), CD, and SBP interfaces are multiplexed on a single interface port.

ATAPI is an asynchronous, 120 ns, 16-bit word interface commonly used on IDE hard disk drives, CD-R, CD-RW, and DVD recorder drives.

DVD (SD) is an 8-bit word low-level interface to the DSP on consumer electronics DVD optical loaders. It can process DVD sector data to locate the header of the 2048-byte pack and perform DVD descrambling on scrambled sectors. The resulting data is sent to the SDRAM using the DMA interface.

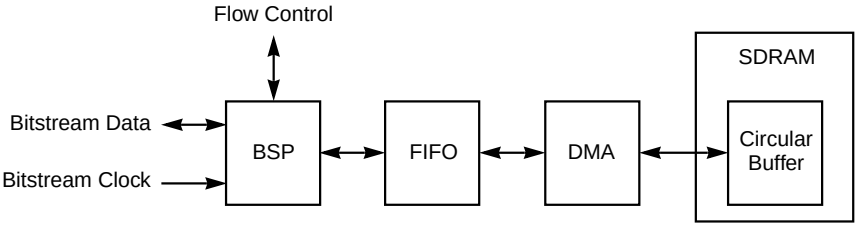
CD is a serial low-level interface to the DSP on consumer electronics CD-DA optical loaders. Low bit-rate (1.5 MHz) CD-DA/CD-ROM sector input coming from the ATAPI or a CD drive's bit-serial interface is directly streamed to SDRAM where the DMN-8600 performs sector framing by locating the sector sync header.

Software is responsible for further input processing, including PCR recovery, PTS handling, DVD navigation, CD descrambling and CRC check, demultiplexing the DVD/CD data in SDRAM and copying the AV bitstreams to proper VBV buffers during a vertical blanking interval (VBI).

For output, DMA channels transfer data from the SDRAM to the bitstream/storage interfaces. The SBP is a synchronous flow-controlled 8-bit data port. It supports peak transfer rates up to 27 Mbytes/s without flow control (internal clock at least 4x higher) or 50 Mbyte/s with flow control. The data transfer flow for the bitstream port is shown in Figure 4.2.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 4.2 DMN-8600 Bitstream Interface



The SBP transfers bitstreams between an 8-bit bitstream bus and a circular buffer in SDRAM under DMA control from a FIFO buffer. The SBP operates only in half duplex mode, receiving (incoming) or transmitting (outgoing) 8-bits of data per clock cycle.

Note: ATAPI transfers can be interleaved with async master accesses, but async master accesses can not be performed while an SD, CD or secondary bitstream port transfer is in progress.

4.4.3 IEEE 1394 Interface

The IEEE 1394 interface can receive MPEG-2 transport stream or DV stream data contained in isochronous packets (IPs). The DMN-8600 filters the packets by matching channel ID's and performs 1394 descrambling on the scrambled data. In case of MPEG-2 transport streams the DMN-8600 performs descrambling on the data scrambled with 5C encryption. Software is responsible for transport section processing and demultiplexing.

For output, DMA channels transfer data from SDRAM to the IEEE 1394 interface. In the case of IEEE 1394 isochronous output, software prepares the proper IEEE 1394 time stamp in the packet header and multiplexes packets from different channels together.

4.4.4 Video Interface

The video interface is a programmable high-speed I/O port that supports the transfer of uncompressed digital video into and out of the DMN-8600 processor. It captures one stream of 8-bit digital video data from an

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ITU-R 656[1] source such as low-cost video decoder chips. The video interface outputs 8- or 16-bit YUV digital video data to low-cost ITU-R 656 encoders.

A compelling feature is the support for 480p (54 MHz, progressive scan) output, resulting in dramatic video quality improvements on Progressive TV screens. Integration of this feature on the DMN-8600 lowers system cost and makes it affordable for the mass consumer.

4.4.5 Audio Interface

The DMN-8600 has a dual serial audio input port capable of two channels of audio input for stereo recording, and four serial audio output ports capable of eight channels of audio output for up to eight channel surround sound reproduction. In addition, it supports an S/PDIF output interface port, which is fully IEC-958 compliant.

The serial audio input port receives uncompressed 16- to 24-bit serial digital audio data from external audio A/D or AES/SPDIF devices. An externally or internally generated clock provides bit serial clocking of the data coming from an external ADC pair. An externally or internally generated frame sync provides synchronization of frames. Frames are organized as two samples per frame.

The four serial audio output ports send uncompressed 16- to 24-bit digital audio to external audio DACs or S/PDIF devices. An externally or internally generated clock provides synchronous bit serial clocking for all four ports. An internally generated frame sync provides synchronization of frames for all four ports. Each port is capable of handling two channels of serial audio data, with two samples per frame.

4.4.6 SDRAM Interface

The 32-bit SDRAM interface generates the address and control signals needed to support one to four 64 Mbit or 128 Mbit SDRAM chips. The SDRAM clock operates at up to 148.5 MHz. This means that DDR (double data rate) SDRAM can transfer data at a maximum effective rate of 297 MHz with two data transfers per clock cycle on the 148.5MHz bus clock. The DRAM configurations supported are one to two 2 M x 32 or

1. “ITU-R 656” refers to the ITU-R BT.656 standard (formerly known as CCIR 656).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

4 M x 32 SDR or DDR SDRAMs or two to four 8 M x 16 SDR or DDR SDRAMs.

The DMN-8600 uses its external SDRAM to store both code and data, allowing memory allocation to be under complete software control. For example, when encoding MPEG video, the SDRAM contains:

- Application program code
- Video capture buffers for incoming frames
- Subsampled video buffers used for motion estimation
- Reconstructed frame buffers for prediction
- Rate buffers
- Local tables and other data used by the encoding algorithm

4.4.7 Serial I/O Interface

The DMN-8600 serial I/O (SIO) interface consists of five peripheral modules: SPI, two UARTs, and two IRs (one IR transmit/receive, one IR transmit only). All modules interface to the internal bus as well as to a DMA channel, so they can be used either as programmed I/O or with DMA transfers.

The UARTs operate in full duplex mode. UART1 supports synchronous communication as well as hardware flow control using the RTS/CTS pins.

Note: UART1 is not available when the external host port is configured in embedded Host-plus-Limited-Master mode.

The SPI (Serial Peripheral Interface) port provides a bus for serial devices such as SmartMedia flash cards and EPROM devices. Up to 4 chip selects are supported. The serial bus clock is generated internally. The user may specify that the last 32-bit word transferred can have anywhere from 1 to 31 bits (transfers less than1 byte not allowed).

Note: The SPI port is not available when the external host port is configured in Host-plus-Limited-Master mode.

The IR (infrared) interface supports both IR receive and IR Blaster transmit functionality for most consumer electronic IR protocols. This two-module interface (one IR transmit/receive, one IR transmit only) is

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

highly programmable and compatible with a large variety of signal timing specifications. The ability of the interface to directly access SDRAM allows for the storage of received sequences and the transmission of preassembled sequences with minimal processor support.

Note: The IR Blaster transmit ports are not available when the external host port is configured in Host-plus-Limited-Master or mode.

The IDC (inter-device communication) interface allows background communication to occur between the various components in the system. The IDC bus is a simple, two-wire, bidirectional communication bus. The two signals, clock and data, are common to every device connected to the bus.

4.4.8 JTAG Interface (Test Access Port)

The Joint Test Action Group (JTAG) interface includes a 5-pin port as outlined in IEEE standard 1149.1. The JTAG interface provides for boundary scan testing utilizing a multiplexor and latches on every pin of the DMN-8600 device that can be forced to a known state. The actual data that is latched depends on multiplexor functions controlled by the TAP (test access port) controller. The TAP can capture pin data based on test signals from outside the DMN-8600, or can capture normal activity on the pin.

4.5 Graphics Accelerator

The DMN-8600 integrates a 2D graphics engine on the dual SPARC processors. This graphics engine is capable of supporting OSD, 2D graphical user interfaces, and 32-bit RGBA with an 8-bit alpha blending channel, flicker filters and a high-quality video scaler. It is capable of supporting multiple video inputs, windowed video and graphics with arbitrarily relocatable and resizable windows, PiP (Picture-in-Picture), letterbox, and side-by-side display of SD sources.

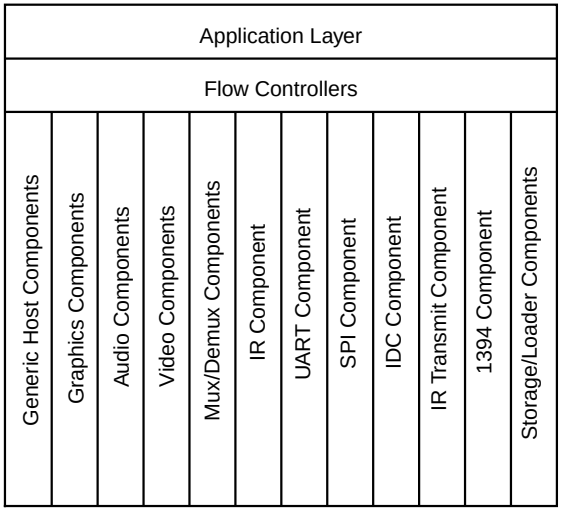
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 5 Programmability (C-Ware)

The DMN-8600 is a programmable device. It is compatible with the LSI Logic unique C-Ware environment. C-Ware is an object oriented software architecture that abstracts the underlying hardware. This promotes portability and application reusability within and across LSI Logic product families. Figure 5.1 shows the basic C-Ware architecture.

Figure 5.1 C-Ware Architecture



This architecture is layered, starting with the “Component” layer. Components abstract the hardware and provide access via a set of standardized APIs. The next layer up is the “Flow Control” layer. Flow controllers build complete functions, such as a DVD encode function. In this case the flow controller controls the video and audio capture, the encoding, and the storage on hard disk or DVD recordable disk. Flow

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

controllers also have a set of standardized APIs. These APIs allow developers to build applications on top of flow controllers.

This drastically reduces software efforts, engineering investments and time-to-market, while increasing the ability to leverage a common platform across different products and the ability to be able to upgrade products.

Support for an external, generic host (16- or 32-bit) is provided through a set of generic host APIs. All components and flow controllers provided by LSI Logic are fully optimized for performance and code compactness.

Note: The DMN-8600 shares its API with LSI Logic ZiVA®-5 family of products, enabling the development and porting of Playback navigation from either platform to the other.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 6
Signal Descriptions

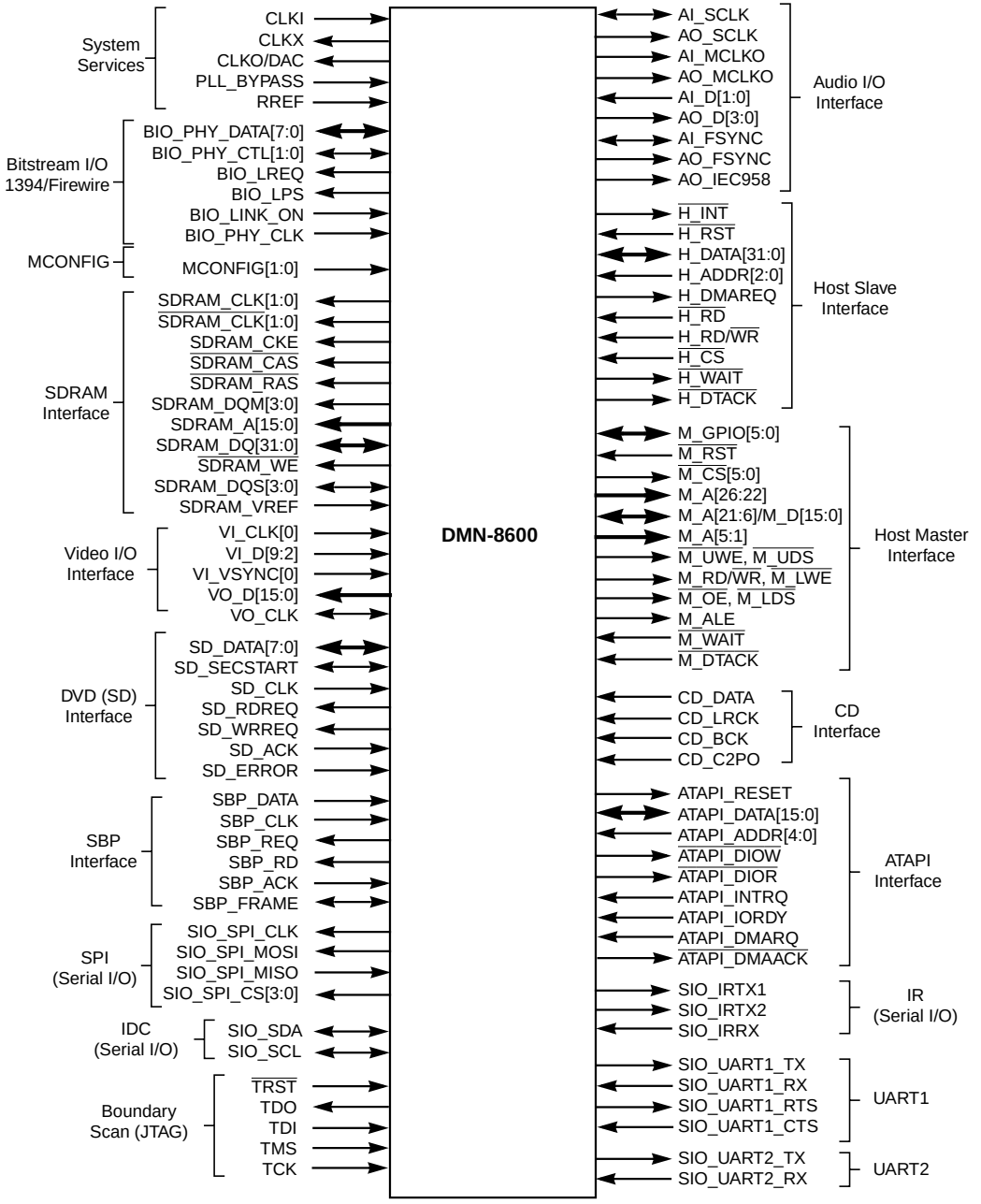
This chapter describes the signals that comprise the external physical interface to the DMN-8600.

Figure 6.1 shows a diagram of the DMN-8600 with all external interface signals grouped together.

Table 6.1 provides the pin name, pin number, type, and description of each signal. Multiplexed pins are selected by some combination of DMN-8600 input pins using MCONFIG[2:0] and (for Bitstream I/O only) a control register. All other interface pins have dedicated signal assignments.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 6.1 DMN-8600 System Interfaces



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions

Name	Pin No.	Type[1]	Description
System Services			
The following pins provide the internal clock source, reset and power-down indications from the chip. They are 3.3 V LVTTL compatible.			
CLKI	A10	I	Clock. This input provides the timing reference for internally generated DMN-8600 clocks. Nominally 13.5 or 27 MHz LVTTL signal or 13.5 MHz crystal, this is internally multiplied to yield the internal processing and audio/video clocks.
CLKX	B10	O	Connected to other pin of 13.5 MHz crystal. This pin should be unconnected if an LVTTL clock signal is connected to CLKI.
CLKO/DAC	A12	O	The output of the internal 13.5 MHz crystal oscillator or output of a train of digital pulses controlled by register **TCdacCtl**. The function is selected by the clock control register.
PLL_BYPASS	A7	I	Bypasses the PLL used to generate the internal processing clock (also for I/O interfaces that do not have their own clock). (For testing purposes only.)
RREF	C11	I	Analog reference resistor. Connecting to pin VSS_RREF through a 1.18 K ± 1% resistor is recommended.
NC	C6, B5, B4, C5, D6, A4, B1, A3, A6, D2	I/O	No connect.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
Power and Ground			
VDD_5	U5	5	Nominal supply bias to support 5V tolerant I/O circuits.
VDD_2.5	D15, D17, C16, J16, K16, L16	2.5/3.3	Nominal DDR-DRAM/SDRAM I/O power supply.
VDD_3.3	D5, E9, E11, T11, T12, T4, L5, K5, E4	3.3	Nominal digital I/O power supply
VDD_1.8	E10, E12, T10, T9, U4, M5, J5, D4	1.8	Digital power pins which supply power to the core.
VDD_DLL	U15, U16	1.8	Digital power for internal clock DLL.
VDD_A	B8, D9, B9, D10	3.3	3.3 V nominal isolated analog supplies. These are isolated nets which supply power to each analog block in the design.
VDD_RREF	D11	3.3	3.3 V analog power. This is an isolated net which supplies power to the internal Bandgap block.
VDD_X	A11	3.3	Isolated 3.3 V nominal crystal oscillator supply.
VSS	H8, H9, H10, H11, H12, H13, J8, J9, J10, J11, J12, J13, K8, K9, K10, K11, K12, K13, L8, L9, L10, L11, L12, L13, M8, M9, M10, M11, M12, M13, N8, N9, N10, N11, N12, N13	GND	Ground for core logic and I/O signals.
VSS_DLL	T17, U17	GND	Analog ground for internal clock DLL.
VSS_RREF	B11	GND	Analog ground for the internal Bandgap block.
VSS_A	C8, A8, C9, A9	GND	Analog ground for isolated analog supplies.
VSS_X	C10	GND	Ground for crystal oscillator supply.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
Bitstream I/O 1394/Firewire			
BIO_PHY_DATA[7:0]	K3, K2, K1, J1, M2, M3, L3, J2	I/O	Data 7 through data 0 of the phy-link data bus. Data is expected on D0–D1 for 100 Mbits/s packets, D0-D3 for 200 Mbits/s and D0–D7 for 400 Mbits/s.
BIO_PHY_CTL[1:0]	L2, J3	I/O	Control 0 and 1 of the phy-link control bus. These two bits together indicate the four operations that can occur in this interface.
BIO_LREQ	M1	O	Link Request. It makes the bus request to access the PHY layer.
BIO_LPS	L4	O	Link Power Status. Indicates that the link is powered and functional. It requests the PHY to disable or enable the PHY/link interface.
BIO_LINK_ON	K4	I	Occurrence of a link-on event.
BIO_PHY_CLK	L1	I	PHY_CLK is the 49.152 MHz clock supplied by the PHY device.
MCONFIG			
MCONFIG[1:0]	Y16, M16	I	These mode pins indicate the configuration of DMN-8600 as defined below. They must be stable from power on. MCONFIG[1:0] = 00 = Async slave only. MCONFIG[1:0] = 01 = Async slave + Async master (no UART1, SPI, IRTX). MCONFIG[1:0] = 10 = Async master only.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
SDRAM Interface			
Between 8 MBytes and 64 MBytes SDRAM or DDR SDRAM can be used, with clock speeds between 80 MHz and 150 MHz. With x32 parts one or two chips may be used, while with x16 parts two or four chips may be used. SDRAM signals can be configured for either 3.3 V LVTTL for SDR SDRAM or 2.5 V SSTL_2 for DDR SDRAM. SDRAM_CLK, SDRAM_CKE, $\overline{\text{SDRAM\_RAS}}$, $\overline{\text{SDRAM\_CAS}}$, $\overline{\text{SDRAM\_WE}}$ and SDRAM_A timing are specified at 30 pF load. SDRAM_DQM and SDRAM_DQ timing are specified at 12 pF load.			
SDRAM_CLK[1:0]	G20, P20	O	SDRAM clock. These outputs are buffered versions of the internal clock.
$\overline{\text{SDRAM\_CLK}}$[1:0]	H20, N20	O	Active LOW SDRAM differential clock. Used by DDR parts.
SDRAM_CKE	C15	O	SDRAM clock enable. This output is used to put the SDRAMs in low-power mode when the chip is put in a power-down state.
$\overline{\text{SDRAM\_CAS}}$	B17	O	Active LOW SDRAM Column Address Strobe. Connects directly to $\overline{\text{CAS}}$ inputs.
$\overline{\text{SDRAM\_RAS}}$	B16	O	Active LOW SDRAM Row Address Strobe. Connects directly to $\overline{\text{RAS}}$ inputs.
SDRAM_DQM[3:0]	G19, K20, L17, N17	O	These pins are the byte masks corresponding to SDRAM_DQ[7:0], [15:8], [23:16] and [31:24]. They allow for byte reads/writes to SDRAM, and connect gluelessly to the UDQM/LDQM inputs.
SDRAM_A[15:0]	A18, C20, E18, E17, F17, C19, D19, D18, B20, C18, A20, B18, C17, A17, B19, A19	O	SDRAM address, bank select, chip select. These outputs connect directly to the SDRAM address inputs, bank address pins, and chip select inputs as shown in the table below. Bank Address can be two or three pins, depending on whether there are four or eight banks: **Memory Configuration** / **Chip Select Pin** / **Bank Address Pins, Address Pins** 32 MBytes x 2 [15:14] [13:0] 32 MBytes x 1 [15] [13:0] 16 MBytes x 2 [15:14] [13:0] 16 MBytes x 1 [15] [13:0] 8 MBytes x 2 [15:14] [12:0] 8 MBytes x 1 [15] [12:0]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
SDRAM_DQ[31:0]	F20, F19, H17, G18, G17, F18, E19, D20, H19, H18, J17, J18, J20, K17, K18, K19, N18, M18, L20, M17, L19, M20, N19, L18, P19, P18, R20, P17, T20, R18, T19, R17	I/O	SDRAM read/write data.
$\overline{\text{SDRAM\_WE}}$	A16	O	Active LOW SDRAM Write Enable. Specifies transaction: read (= 1) or write (= 0). These pins connect gluelessly to the DRAM $\overline{\text{WE}}$ inputs.
SDRAM_DQS[3:0]	E20, J19, M19, R19	I/O	SDRAM data strobe for DDR parts.
SDRAM_VREF	D16	I	SDRAM reference voltage for DDR parts. Use a regulated VDD_2.5 2 voltage source.
Video Input/Output			
Video port signals VI_CLK[0], VI_D[9:2], and VI_SYNC[0] are 3.3 V LVTTL and 5 V TTL compatible. The remainder are all 3.3 V LVTTL compatible only. All outputs are specified with a 25 pF load and 10 mA Drive.			
VI_CLK[0]	A5	I	Video input clock. VI_CLK[0] is the sample clock for video input stream 1 (single stream). Input data is sampled on the rising edge. For 8-bit video input, VI_CLK is twice the pixel clock frequency given the 4:2:2 I/O format and should be 27 MHz for CCIR 601 format video. Input capture is synchronized to EAV or SAV codes in the input video, with vertical sync optionally provided by the VI_VSYNC pins for nonstandard video inputs. VI_CLK is asynchronous to all other DMN-8600 clocks, 0 to 74.25 MHz.
VI_D[9:2]	A1, A2, B3, C4, D3, C2, C3, B2	I	Video Input Data - For 8-bit video data, interleaved luma and chroma samples are captured on the VI_D[9:2] pins. The data order is either (Cb, Y, Cr, Y) or (Y, Cb, Y, Cr) as selected by the Yfirst bit in the video control register.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
VI_VSYNC[0]	C1	I	Video Input Vertical Sync. If VSP is set, then VI_VSYNC[0] will define the start of the frame by setting the vertical line counter to zero when asserted. VI_VSYNC[0] is active HIGH if ISpol is set, otherwise it is active LOW.
VO_D[15:0]	F4, E3, D1, E2, F3, E1, G4, F2, F1, G3, G2, G1, H4, H3, H2, J4	O	Video Output Data. For 8-bit YCrCb video data interleaved luma and chroma samples are output on the VO_D[7:0] pins. The data order is either (Cb, Y, Cr, Y) or (Y, Cb, Y, Cr), as selected by the Yfirst bit in the video control register. For 16-bit YCrCb video data luma and chroma samples are output on VO_D[7:0] and VO_D[15:8] pins, respectively.
VO_CLK	H1	I/O	Video Output Clock. VO_CLK is the sample clock for video output. The direction of VO_CLK is programmable. Output data and control signals are driven on the rising edge if olckr is set, otherwise they are driven on the falling edge. For 8-bit video output, VO_CLK is twice the pixel clock frequency given the 4:2:2 I/O format and should be 27 MHz for CCIR 601 format video. VO_CLK is the same as the pixel clock frequency and should be 54 MHz for 8-bit data and 27 MHz for 16-bit data. VO_CLK is asynchronous to all other chip clocks, 0 to 74.25 MHz.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
Audio I/O Interface			
Audio port signals are 3.3 V LVTTL compatible and 5 V tolerant. All outputs are specified with a 25 pF load and 10 mA drive.			
AI_SCLK	B12	I/O	Serial audio bit clock used for audio input. Input data and AI_FSYNC as an input are sampled on the rising edge if Iclkr is set; otherwise they are sampled on the falling edge. AI_FSYNC, as an output, is driven on the falling edge if Iclkr is set, otherwise it is driven on the rising edge. AI_SCLK is asynchronous to all other chip clocks. This clock can be driven from outside or it can be internally generated and driven out.
AO_SCLK	A14	O	Serial audio bit clock for audio output. Output data and AO_FSYNC are driven on the falling edge if Oclkr is set; otherwise they are driven on the rising edge. AO_SCLK is asynchronous to all other chip clocks. This clock is internally generated and driven out.
AI_MCLKO	A13	O	Audio master input clock output for the internally generated master input clock.
AO_MCLKO	B13	O	Audio Master Output Clock output for the internally generated master output clock.
AI_D[1:0]	D12, C13	I	Audio Stream Input Data. Up to four channels of serial audio data are clocked in from the AI_D pin. With two samples/frame, channel 2n and 2n + 1 use pin AI_D[n]. With one sample/frame, channel n uses pin AI_D[n].
AO_D[3:0]	A15, B15, C14, D14	O	Audio Stream Output Data. Up to eight channels of serial audio data are clocked out from the four AO_D pins by the clock on the AO_SCLK pin. With two samples/frame, channel 2n and 2n + 1 use pin AO_D[n]. With one sample/frame, channel n uses pin AO_D[n].
AI_FSYNC	C12	I/O	AI_FSYNC determines the start or end of the next input sample or frame as specified by frForm in the audio input control register. AI_FSYNC can be internally or externally generated.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
AO_FSYNC	D13	O	AO_FSYNC determines the start or end of the next output stream 1 sample or frame as specified by frForm in the audio output control register. AO_FSYNC is always internally generated.
AO_IEC958	B14	O	IEC-958 Interface Output. This output provides single-wire, self-clocking IEC-958 digital audio synchronized to the audio output framing and master clock. It must be connected to an external line driver.
Host Slave Interface			
The Host Interface port is the communication port with an external host. The host interface shares its pins with the (Async) Master interface, allowing up to two signals per pin: Slave or Master, with two possible configurations as selected by MCONFIG[1:0] as follows: MCONFIG[1:0] = 00 = Async host (slave) only MCONFIG[1:0] = 01 = Async host (slave) + Async master (no UART1, SPI, IRTX) signals			
$\overline{\text{H\_INT}}$	U7	O	Host interrupt request (open drain output). This pin is shared with M_GPIO[0].
$\overline{\text{H\_RST}}$	W16	I	Host Reset signal. This signal is shared with $\overline{\text{M\_RST}}$.
H_DATA[31:16]	W4, Y3, V5, W5, Y4, V6, Y5, Y6, U6, W6, V7, W7, Y8, Y7, V8, W8	I/O	Host interface data bus [31:16]. H_DATA[31:30] (slave) pins are shared with $\overline{\text{M\_CS}}$[5:4], H_DATA[29:26] (slave) pins are shared with $\overline{\text{M\_CS}}$[3:0], H_DATA[25:21] (slave) pins are shared with M_A[26:22], and H_DATA[20:16] (slave) pins are shared with M_A[5:1].
H_DATA[15:0]	Y11, U10, V11, V10, Y12, V12, W11, U11, V13, W12, U12, Y14, V14, W13, W14, W15	I/O	Host interface data bus [15:0]. These pins (slave) pins are shared with M_A[21:6]/M_D[15:0].
H_ADDR[2:0]	W9, W10, Y13	I	Host access address bus (16-bit word address). This pin (slave) is shared with M_GPIO[3:1].
H_DMAREQ	V15	O	Host DMA transfer request. This pin (slave) is shared with $\overline{\text{M\_UWE}}$/$\overline{\text{UDS}}$.
$\overline{\text{H\_RD}}$	Y10	I	Host I-mode read strobe. This pin (slave) is shared with M_GPIO[4].
H_RD/$\overline{\text{WR}}$	U8	I	Host I-mode write strobe/M-mode direction signal. H_RD/$\overline{\text{WR}}$ (slave) is shared with M_RD/$\overline{\text{WR}}$.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
H_CS	Y9	I	Host chip select. This pin is shared with M_GPIO[5].
H_WAIT	V9	O	Wait signal to the host. This pin is shared with M_WAIT.
H_DTACK	U9	O	Host data transfer acknowledge. This pin is shared with M_DTACK.
Host Master Interface			
The master interface shares all of its pins with the host (slave) interface and some of its pins with the Serial I/O interface pins. The host interface shares its pins with the (Async) Master interface, allowing up to two signals per pin, Slave or Master, with three possible configurations selected by MCONFIG[1:0] as follows: MCONFIG[1:0] = 00 = Async slave only MCONFIG[1:0] = 01 = Async slave + Async master (no UART1, SPI, IRTX) MCONFIG[1:0] = 10 = Async master only When Async Host plus Async Master is selected, the low order address and control pins other than M_ALE (M_A[10:1], M_CS[0], M_OE, M_RD/WR) replace the Bitstream I/O pins for UART1, SPI and IR transmit (UART2, IR receive and IDC are still available). The limited master is primarily intended to support an external PROM accessed independently of the host processor; consequently, it only supports one chip select with self-paced cycles.			
M_GPIO[5:0]	Y9, Y10, W9, W10, Y13, U7	I/O	General purpose I/O [5:0]. These pins are shared with H_CS, H_RD, H_ADDR[2:0], and H_INT, respectively.
M_RST	W16	I	Master Reset signal. This signal is shared with H_RST (slave).
M_CS[5:0] M_CS[0]	W4, Y3, V5, W5, Y4, V6 (master) W19 (slave/master)	O	Master chip select [5:0] outputs. These pins are shared with H_DATA[31:26] (slave) pins when in master mode. M_CS[0] is shared with UART1_RX when in slave/limited master mode.
M_A[26:22]	V6, Y5, Y6, U6, W6 (master) Y17, U18, Y19, V18, V16, (slave/master)	O	Master address [26:22]. These pins are shared with H_DATA[25:21] (slave) pins when in master mode, and are shared with SPI pins when in slave/limited master mode.
M_A[21:6]/ M_D[15:0]	Y11, U10, V11, V10, Y12, V12, W11, U11, V13, W12, U12, Y14, V14, W13, W14, W15	I/O	Master muxed address [21:6] and data [15:0]. These pins are shared with H_DATA[15:0] (slave) pins.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
M_A[5:1]	W7, Y8, Y7, V8, W8 (master mode) W18, U20, V20, U14, T18 (slave/master)	O	Master address [5:1]. These pins are shared with H_DATA[20:16] (slave) pins when in master mode, and with SPI, UART and IRTX pins when in slave/limited master mode.
$\overline{\text{M\_UWE}}$ $\overline{\text{M\_UDS}}$	V15	O	Master upper write enable (SRAM-mode). Upper data strobe (68K mode). This pin is shared with H_DMAREQ.
M_RD/$\overline{\text{WR}}$ $\overline{\text{M\_LWE}}$	U8 (master mode) V17 (slave/limited master mode)	O	Master Lower Write Enable (SRAM-mode), direction (68K mode). M_RD/$\overline{\text{WR}}$ is shared with H_RD/$\overline{\text{WR}}$ when in master mode, and shared with SIO_UART1_CTS when in slave/limited master mode.
$\overline{\text{M\_OE}}$ $\overline{\text{M\_LDS}}$	Y15 (master mode) W17 (slave/limited master mode)	O	Master $\overline{\text{OE}}$ (SRAM-mode). Lower data strobe (68K mode). Shared with IRTX2 when in slave/limited master mode.
M_ALE	U13	O	Master address latch enable.
$\overline{\text{M\_WAIT}}$	V9	I/O	Master ready. This pin is shared with $\overline{\text{H\_WAIT}}$.
$\overline{\text{M\_DTACK}}$	U9	I/O	Master data transfer acknowledge. This pin is shared with $\overline{\text{H\_DTACK}}$.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
DVD (SD) Interface			
As a Bitstream I/O pin, DVD (SD) signals are shared with up to two other signals per pin. One of these three possible configurations is selected by MCONFIG[2:0] and the DVD control register (see description of DVD control register on page 14-5). When the Bitstream I/O interface is in SD/SBP mode, the master interface M_A[21:6]/M_D[15:0] pins are shared with SD_DATA[7:0]/SBP_DATA[7:0] pins.			
SD_DATA[7:0]	N1, N2, N4, P1, P3, P4, R3, R4	I/O	SD_DATA is the bidirectional parallel data bus. When SD DSP transfers 32-bit data, it must write the most significant byte first. Multiplexed with ATAPI_DATA[7:0]
SD_SECSTART	U1	I/O	The SD sector start indicator. Multiplexed with ATAPI_DMAACK.
SD_CLK	T3	I	SD data clock in both sync or async mode. Multiplexed with ATAPI_IORDY. If SD_CLK = 1, then input data and control signals are sampled on the rising edge; otherwise they are sampled on the falling edge.
SD_RDREQ	V3	O	The chip asserts SD_RDREQ to indicate that the internal buffer has available space and it is ready to perform a read (SD_WRREQ = 1) or that it will be reading data (SD_WRREQ = 0). Multiplexed with ATAPI_DIOR.
SD_WRREQ	T1	O	The chip asserts SD_WRREQ to indicate that it has writable data (SD_WRREQ = 1) or that it is ready to take or give data (SD_WRREQ = 0). In the latter case, the direction of the transfer is indicated by SD_RDREQ. Multiplexed with ATAPI_DIOW.
SD_ACK	W2	I	Transfer of data acknowledged by system. Active low if SD_ACK = 0, otherwise active high. Multiplexed with ATAPI_INTRQ.
SD_ERROR	W1	I	The DSP asserts this signal to indicate that an error has occurred. If DSP does not provide an error signal, ground this pin. Multiplexed with ATAPI_DMARQ.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
CD Interface			
As a Bitstream I/O pin, CD signals are shared with up to two other signals per pin. One of these three possible configurations is selected by MCONFIG[2:0] and the DVD control register (see description of DVD control register on page 14-5).			
CD_DATA	R4	I	Serial data from CD DSP. Multiplexed with SD_DATA[0]/ATAPI_DATA[0]
CD_LRCK	R3	I	CD Left-Right Clock. Multiplexed with SD_DATA[1]/ATAPI_DATA[1].
CD_BCK	P4	I	CD DSP Bit Clock. Multiplexed with SD_DATA[2]/ATAPI_DATA[2].
CD_C2PO	P3	I	CD-ROM Error. Multiplexed with SD_DATA[3]/ATAPI_DATA[3]
ATAPI Interface			
As a Bitstream I/O pin, ATAPI signals are shared with up to two other signals per pin. One of these three possible configurations is selected by MCONFIG[2:0] and the DVD control register (see description of DVD control register on page 14-5).			
ATAPI_RESET	Y1	O	The chip reset for connected ATAPI devices.
ATAPI_DATA[15:0]	Y2, U2, T2, R2, R1, P2, N3, M4, N1, N2, N4, P1, P3, P4, R3, R4	I/O	ATAPI_DATA is the bidirectional data bus. ATAPI_DATA[15:8] are shared with SBP_DATA[7:0], and ATAPI_DATA[7:4] are shared with SD_DATA[7:4]. ATAPI_DATA[3:0] are shared with SD_DATA[3:0], and CD_CDPO, CD_BCK, CD_LRCK, and CD_DATA, respectively.
ATAPI_ADDR[4:0]	V4, U3, V1, V2, W3	O	Device register address. Pins 4 to 0 correspond to ATAPI pins CS0,CS1,DA2,DA1 and DA0, respectively. ATAPI_ADDR[4:0] pins are shared with SBP_CLK, SBP_REQ, SBP_RD, SBP_ACK, and SBP_FRAME, respectively.
$\overline{\text{ATAPI\_DIOW}}$	T1	O	ATAPI I/O write request. This pin is shared with SD_WRREQ.
$\overline{\text{ATAPI\_DIOR}}$	V3	O	ATAPI I/O read request. This pin is shared with SD_RDREQ.
ATAPI_INTRQ	W2	I	Device interrupt request. This pin is shared with SD_ACK.
ATAPI_IORDY	T3	I	Device I/O ready. This pin is shared with SD_CLK.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
ATAPI_DMARQ	W1	I	Device DMA request. This pin is shared with SD_ERROR.
$\overline{\text{ATAPI\_DMAACK}}$	U1	O	ATAPI DMA acknowledgement. This pin is shared with SD_SECSTART.
SBP Interface			
As Bitstream I/O pins, SBP signals are shared with up to two other signals per pin. One of these three possible configurations is selected by MCONFIG[2:0] and the DVD control register (see description of DVD control register on page 14-5).			
SBP_DATA[7:0]	Y2, U2, T2, R2, R1, P2, N3, M4	I	SBP data bus. Multiplexed with ATAPI_DATA[15:8] only during non-ATAPI mode.
SBP_CLK	V4	I	SBP clock. All signals are sampled and driven on the rising edge.
SBP_REQ	U3	O	SBP request for data transfer (WrReq = 0) or write transfer request (WrReq = 1). Active low if POL = 1, otherwise active high.
SBP_RD	V1	O	SBP read/$\overline{\text{write}}$ transfer (WrReq = 0) or read transfer request (WrReq = 1). Active low if POL = 1 and WrReq = 1, otherwise active high.
SBP_ACK	V2	I	Transfer of data acknowledged by system. Active low if POL = 1, otherwise active high.
SBP_FRAME	W3	I/O	Indicates first byte of each frame.
SPI (Serial I/O)			
As Serial I/O pins, SIO_SPI pins are shared with up to one other signal per pin. MCONFIG[1:0] = 00 selects the SIO_SPI signals listed below.			
SIO_SPI_CLK	Y17	O	SPI clock. Multiplexed with M_A[26].
SIO_SPI_MOSI	U18	O	SPI master out slave in. Multiplexed with M_A[25].
SIO_SPI_MISO	T18	I	SPI master in slave out. Multiplexed with M_A[1].
SIO_SPI_CS[3:0]	W18, V16, V18, Y19	O	SPI chip selects. Multiplexed with M_A[24:22], M_A[5].

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
IR (Serial I/O)			
As Serial I/O pins, SIO_IRTX pins are shared with up to one other signal per pin. MCONFIG[1:0] = 00 selects the SIO_IRTX1 and SIO_IRTX2 signals listed below.			
SIO_IRTX1	U15	O	IR transmit 1. Multiplexed with M_A[2].
SIO_IRTX2	W17	O	IR transmit 2. Multiplexed with $\overline{\text{M\_OE}}$.
SIO_IRRX	Y20	I	IR receive input.
IDC			
SIO_SDA	Y18	I/O	IDC data (open drain).
SIO_SCL	V19	I/O	IDC clock (open drain).
UART1			
As Serial I/O pins, SIO_UART1 pins are shared with up to one other signal per pin. MCONFIG[1:0] = 00 selects the SIO_UART1 signals listed below.			
SIO_UART1_TX	U20	O	UART1 transmit. Multiplexed with M_A[4].
SIO_UART1_RX	W19	I	UART1 receive. Multiplexed with $\overline{\text{M\_CS}}$[0].
SIO_UART1_RTS	V20	O	UART1 request to send. Multiplexed with M_A[3].
SIO_UART1_CTS	V17	I	UART1 clear to send. Multiplexed with $\overline{\text{M\_WR}}$.
UART2			
As Serial I/O pins, SIO_UART2 pins are shared with up to one other signal per pin. MCONFIG[1:0] = 00 selects the SIO_UART2 signals listed below.			
SIO_UART2_TX	U19	O	UART2 transmit.
SIO_UART2_RX	W20	I	UART2 receive.
Boundary Scan (JTAG)			
$\overline{\text{TRST}}$	D7	I	Test reset. BST reset - resets the TAP controller. This signal must be pulled high during normal mode.
TDO	B7	O	Test data Out. BST serial data output.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 6.1 DMN-8600 Pin Descriptions (Cont.)

Name	Pin No.	Type[1]	Description
TDI	C7	I	Test data In. BST serial data chain input.
TMS	B6	I	Test mode select. Controls state of test access port (TAP) controller.
TCK	D8	I	Test clock. Boundary scan test (BST) serial data clock.

1. I - input, O - output, OD - open drain, PU - requires external pull-up resistor.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 7
Memory Mapping

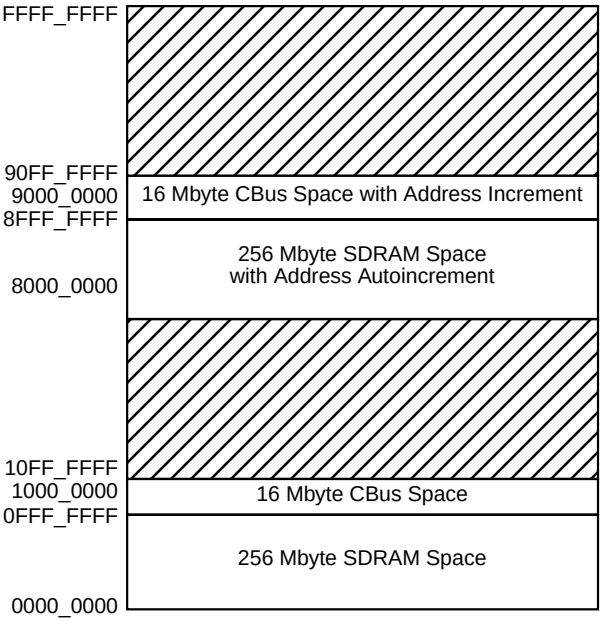
The following sections summarize the DMN-8600 memory map for different configurations. These configurations include SDRAM access and access to internal registers via the internal Control Bus (CBus).

7.1 Host Interface Address Mapping

A 32-bit address is used to access either the SDRAM or the CBus registers, as shown in Figure 7.1.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 7.1 Host Interface Address Mapping



7.2 SPARC Processor Address Mapping

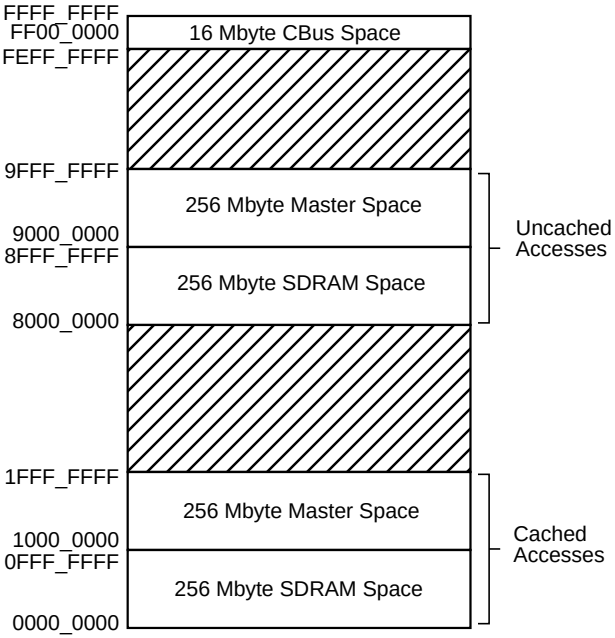
The SPARC processor always boots from address 0x0. There are two address mappings for the SPARC processor, depending on the system implementation.

7.2.1 In System with an External Master Processor

DMN-8600 SPARC processors boot from SDRAM — the external master must load SPARC boot code into SDRAM via the DMN-8600 device's Host Interface. Figure 7.2 shows the resulting memory map.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 7.2 SPARC Processor Memory Map with External Host Processor

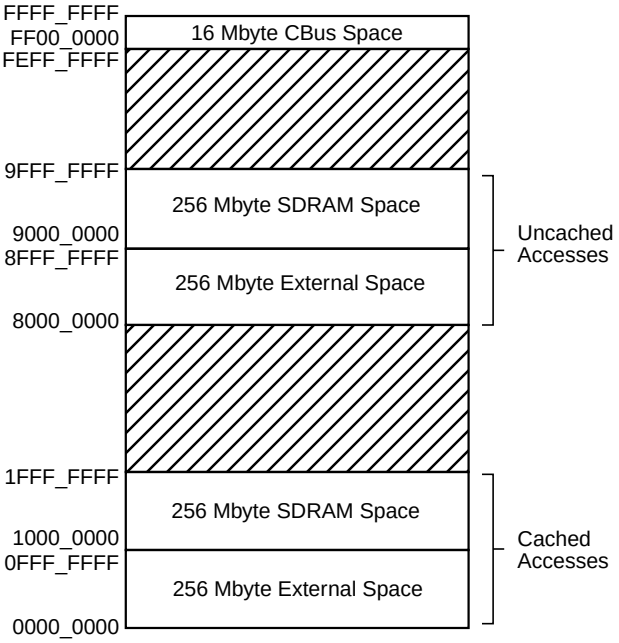


7.2.2 In System with No External Master Processor

DMN-8600 SPARC processors boot from an external ROM. Figure 7.3 shows the resulting memory map.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 7.3 SPARC Processor Memory Map without External Host Processor



7.3 Control Bus Address Mapping

The 16 Mbyte Control Bus (CBus) address space can be accessed through different masters, as shown in Figures 7.1 through 7.3. The address range where CBus elements are located varies with the master that is trying to access them.

CBus address space is divided into 256 regions of 16 K words each. The resources in the Control Bus address space are the control registers, status registers, Audio Memory (AMem), and RISC memory (Rmem). A 24-bit address offset is used to index into the CBus space shown in Figures 7.1 through 7.3. Table 7.1 shows the assignments of these blocks to the various modules within the DMN-8600 device.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Note: Control Bus addresses refer to 32-bit words and thus are one-fourth the offsets shown in Table 7.1.

Table 7.1 CBus Module Address Assignments

Address Range	Resources
0x00_0000–0x01_FFFF	Reserved
0x02_0000–0x02_FFFF	Motion Estimator
0x03_0000–0x03_FFFF	SDRAM Controller
0x05_0000–0x05_FFFF	Audio I/O
0x06_0000–0x06_FFFF	Host
0x07_0000–0x07_FFFF	Reserved
0x08_0000–0x08_FFFF	SBP (Secondary Bitstream)
0x09_0000–0x70_FFFF	Reserved
0x80_0000–0x80_FFFF	AMem
0x81_0000–0xBD_FFFF	Reserved
0xBE_0000–0xBF_FFFF	Serial I/O (UART, IR, SPI)
0xC0_0000–0xC0_FFFF	RMem
0xC1_0000–0xC1_FFFF	Video DSP
0xC2_0000–0xC2_FFFF	CLK Controller
0xC3_0000–0xC3_FFFF	DSP DMem
0xC4_0000–0xC4_FFFF	Video I/O
0xC5_0000–0xF8_FFFF	Reserved
0xF9_0000 –0xF9_FFFF	Reserved
0xFA_0000–0xFA_FFFF	Reserved
0xFB_0000–0xFB_FFFF	Bitstream I/O
0xFC_0000–0xFC_FFFF	Audio DSP
0xFD_0000–0xFD_FFFF	Video SPARC Processor
0xFE_0000–0xFF_FFFF	reserved

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 8
Host Slave Interface

This chapter describes the host slave interface and contains the following sections:

The Host processor port is configured differently for different processors. It supports the following general features:

- 16- or 32-bit Host data bus
- Host DMA target
- WAIT and DTACK signalling for slow response time
- Separate or combined Read and Write strobes

The DMN-8600 embedded host interface supports two modes:

- *Async Slave Interface with host DMA*: An asynchronous, demultiplexed address/data bus (See Figure 8.1) that provides local DRAM and control register access to an external microcontroller (Host) which may perform various control functions at the system level.

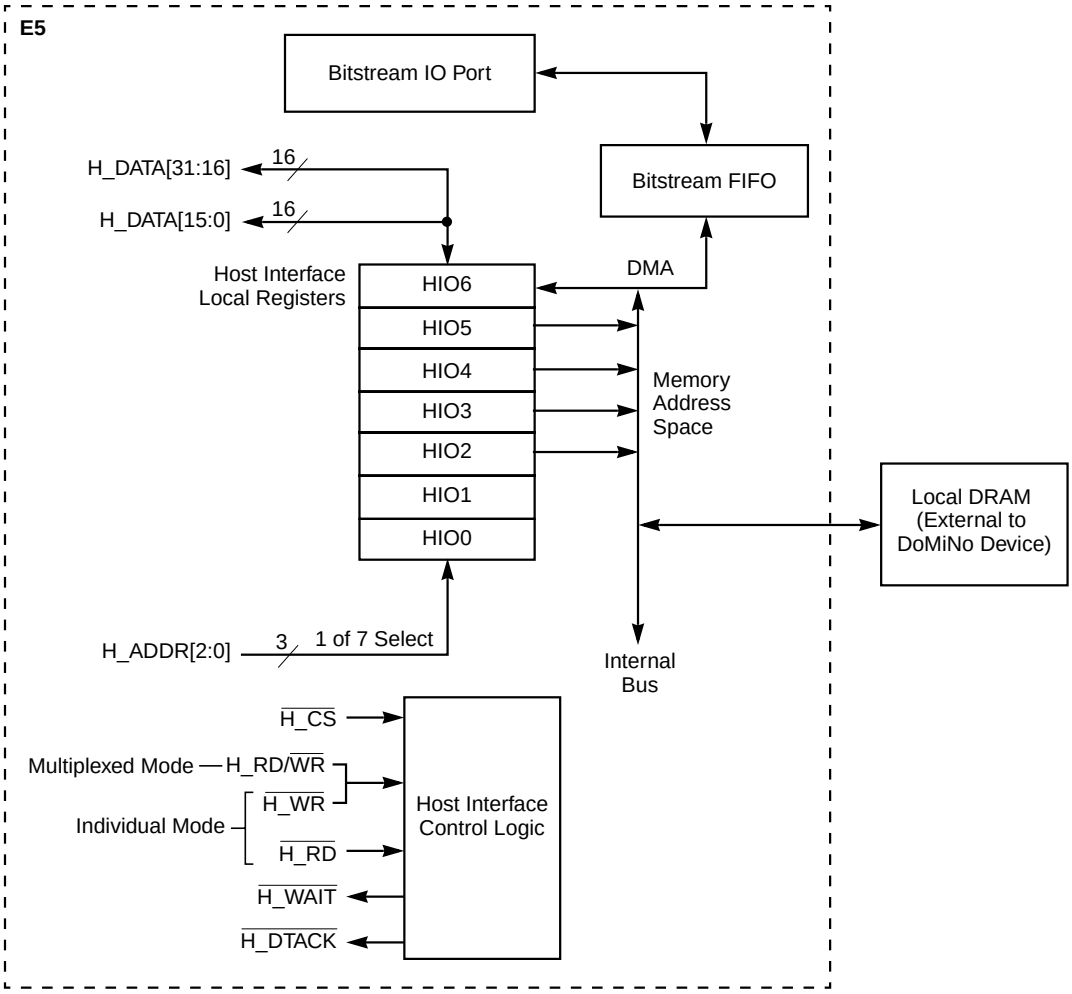
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Note: The master interface uses a multiplexed synchronous 26-bit address/16-bit data bus with multiple chip selects and interrupt inputs to access external slave devices.

- *Async Master Interface with master DMA*: using its own DMA engine to perform master mode DMA transfers, this interface replaces either the:
  - *Slave interface* – operating without an external host, the master interface replaces the slave interface to allow use of the serial I/O interfaces.
  - *Serial I/O interface* – shares pins with the storage interface, with separate control signals used to allow dynamic arbitration between storage and master mode transfers. In this configuration, master accesses can be performed in parallel with host accesses to allow code download from an external PROM without interfering with host processing.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 8.1 Host Interface (Showing Slave Pins Only)



The Host DMA and Secondary Bitstream interfaces support peak transfer rates up to 50 Mbytes/s. The microcode must ensure that the internal DMA channel is programmed to match the transfers being performed by the host. Each interface is described in the following sections

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.1 Async Slave Interface with Host DMA

The flexible DMN-8600 Host bus interface offers many options for configuring the following host interface controls:

- Configurable 16/32-bit demultiplexed address/data bus
- Separate $\overline{WR}$ or $\overline{RD}$ strobes (I-mode) or single $\overline{CS}$ strobe with a single $\overline{WR}$ signal (M-mode)
- Asynchronous control signals
- $\overline{WAIT}$ and $\overline{DTACK}$ signals for slow response time
- Host DMA target

The Host interface provides internal register and SDRAM accesses to an external microcontroller (Host) to configure and control DMN-8600 data transfers. The Host Processor interface receives slave transfer requests from an external host and internally arbitrates the transfers. The Host interface does not support burst transfers.

The slave interface provides a FIFO buffer between the internal DMA channel and the bitstream pins. All internal DMA reads transfer data from the bitstream FIFO to SDRAM, and internal DMA writes transfer data from SDRAM to the bitstream FIFO.

8.2 Async Slave Interface Transfer Modes

The Host interface offers two distinct ways to manage bitstream transfers:

- 32-bit and DMA 32-bit I/O mode (Mode A)
- 16-bit and DMA 16-bit I/O mode (Mode B)

Each mode requires specific configuration through the Host interface.

8.2.1 Transfer Mode A

The Host interface can be configured such that both the programmed I/O transfers and DMA transfers are 32 bits wide. This transfer mode uses the Host interface read/write protocols called I-mode and M-mode.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.2.2 Transfer Mode B

For Mode B, the Host interface is configured for 16-bit I/O transfers and 16-bit DMA transfers. This transfer mode uses the Host interface read/write protocols called I-mode and M-mode.

Table 8.1 summarizes the transfer modes and port availability.

Table 8.1 Transfer Modes and Port Configurations

Mode	Port Transfer Description
A	32-bit and DMA 32-bit I/O
B	16-bit and DMA 16-bit I/O

8.3 Async Slave WRITE and READ Protocols

The Host interface supports either separate $\overline{RD}$ and $\overline{WR}$ strobes (I-mode) or a single $\overline{CS}$ strobe with a single $\overline{WR}$ signal (M-mode).

8.3.1 I-Mode WRITE and READ Operations

To select I-mode, the Host should perform a dummy read cycle to the Host Address Register (address 0x4) using the $\overline{RD}$ strobe in the first Host access after reset.

In I-mode, the address is sampled when $\overline{WAIT}$ and $\overline{DTACK}$ are driven once $\overline{WR}$ or $\overline{RD}$ go LOW.

8.3.2 I-Mode Incoming Transfers

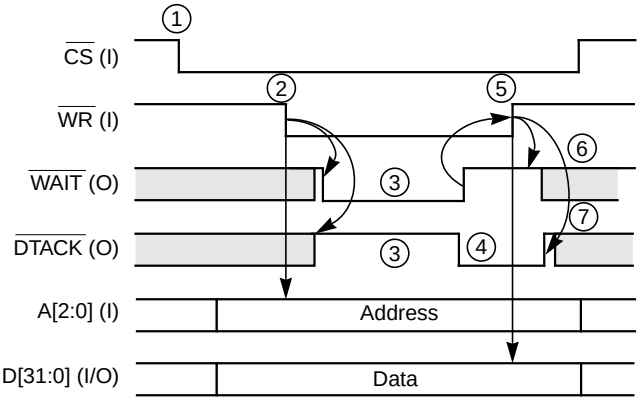
Figure 8.2 shows an I-mode write. The sequence is as follows:

- The Host drives the address and $\overline{CS}$ (1), and asserts $\overline{WR}$ (2) after the address and chip select are stable.
- The DMN-8600 processor asserts both $\overline{WAIT}$ and $\overline{DTACK}$ from a 3-state level to a high level (shaded area is 3-state) (3) in response to the assertion of $\overline{WR}$.
- The DMN-8600 processor deasserts $\overline{WAIT}$ (4) and then asserts $\overline{DTACK}$ (4) when it is ready to take data in.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- The Host deasserts $\overline{WR}$ (5) in response to either the $\overline{WAIT}$ deassertion or $\overline{DTACK}$ assertion.
- The DMN-8600 processor uses the LOW-to-HIGH edge of $\overline{WR}$ (6) to latch in data and to stop driving $\overline{WAIT}$.
- The DMN-8600 processor deasserts $\overline{DTACK}$, then stops driving $\overline{DTACK}$ on the LOW-to-HIGH ⑦ edge of $\overline{WR}$.

Figure 8.2 I-Mode WRITE



8.3.3 I-Mode Outgoing Transfers

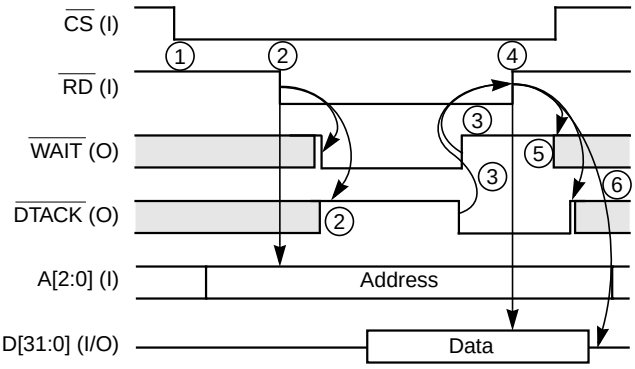
Figure 8.3 shows an I-mode read. The sequence is as follows:

- The Host drives the address on the A[2:0] bus and asserts $\overline{RD}$ (1) after the address is stable.
- On the HIGH-to-LOW edge of $\overline{RD}$ (2), the DMN-8600 processor samples the address, asserts $\overline{WAIT}$, and asserts $\overline{DTACK}$ from a 3-state level (shaded area is 3-state) to a HIGH level.
- The DMN-8600 processor deasserts $\overline{WAIT}$ (3) and asserts $\overline{DTACK}$ after it drives the read data out on the D[31:0] bus.
- The Host deasserts $\overline{RD}$ (4) in response to $\overline{WAIT}$ deassertion or $\overline{DTACK}$ assertion.
- The DMN-8600 processor stops driving the D[31:0] bus and $\overline{WAIT}$ (5) in response to the $\overline{RD}$ deassertion.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- The processor deasserts $\overline{\text{DTACK}}$ (6) also in response to $\overline{\text{RD}}$ deassertion.

Figure 8.3 I-Mode READ



8.3.4 M-Mode WRITE and READ Operations

To select M-mode, the Host performs a dummy write cycle to the Host Address Register (address 0x4) with the $\overline{\text{CS}}$ strobe asserted after $\overline{\text{WR}}$ in the first Host access after reset. The $\overline{\text{RD}}$ pin is held HIGH throughout M-mode.

In M-mode, the address is sampled when $\overline{\text{WAIT}}$ and $\overline{\text{DTACK}}$ are driven when $\overline{\text{CS}}$ goes LOW.

8.3.5 M-Mode Incoming Transfers

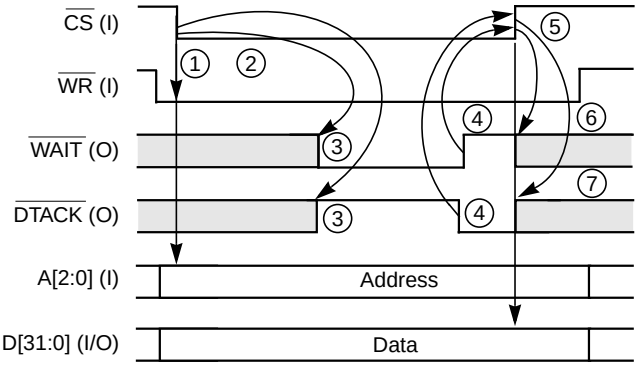
Figure 8.4 shows an M-mode write. The sequence is as follows:

- The Host drives the address and $\overline{\text{WR}}$ and asserts $\overline{\text{CS}}$ (1) after the address and $\overline{\text{WR}}$ (2) are stable.
- The DMN-8600 processor asserts $\overline{\text{WAIT}}$ and deasserts $\overline{\text{DTACK}}$ (3) in response to a $\overline{\text{CS}}$ HIGH-to-LOW edge.
- The processor then deasserts $\overline{\text{WAIT}}$ (4) and asserts $\overline{\text{DTACK}}$ when it is ready to take data in.
- The Host deasserts $\overline{\text{CS}}$ (5) in response to either the $\overline{\text{WAIT}}$ deassertion or $\overline{\text{DTACK}}$ assertion.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- The DMN-8600 processor uses the LOW-to-HIGH edge of $\overline{CS}$ to latch in data and to stop driving $\overline{WAIT}$ (6).
- The DMN-8600 processor deasserts $\overline{DTACK}$, then stops driving $\overline{DTACK}$ on the LOW-to-HIGH (7) edge of $\overline{CS}$.

Figure 8.4 M-Mode WRITE



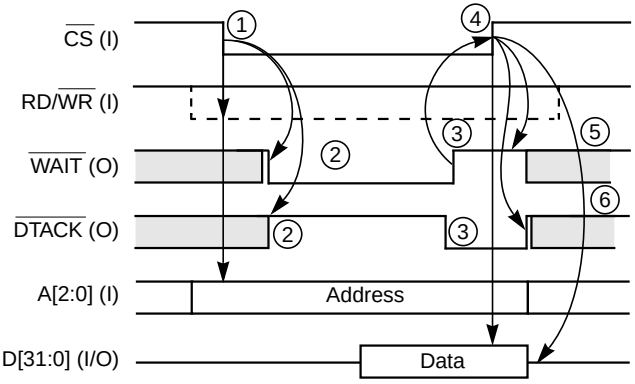
8.3.6 M-Mode Outgoing Transfers

Figure 8.5 shows an M-mode read. The sequence is as follows:

- The DMN-8600 processor samples the Address and RD/$\overline{WR}$ (1) on the HIGH-to-LOW edge of $\overline{CS}$.
- The processor asserts $\overline{WAIT}$ (2) and deasserts $\overline{DTACK}$ from a 3-state to HIGH level (shaded area is 3-state) in response to the HIGH-to-LOW transition of $\overline{CS}$.
- The DMN-8600 processor deasserts $\overline{WAIT}$ (3) and asserts $\overline{DTACK}$ after driving data out.
- Deassertion of $\overline{WAIT}$ or assertion of $\overline{DTACK}$ causes $\overline{CS}$ (4) to be deasserted.
- The DMN-8600 processor stops driving data and $\overline{WAIT}$ (5) in response to the deassertion of $\overline{CS}$.
- The DMN-8600 processor stops driving $\overline{DTACK}$ and then stops driving $\overline{DTACK}$ (6) also in response to the $\overline{CS}$ deassertion.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

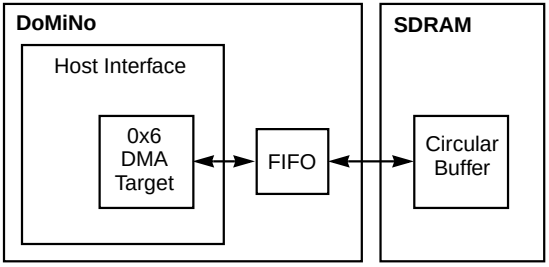
Figure 8.5 M-Mode READ



8.4 Host DMA Read/Write Protocol

The host DMA is connected to a host DMA channel (see Figure 8.6). As a host DMA target, the primary bitstream interface supports dual address DMA transfers where the DMA target is the Host DMA Data Register (address 0x6). The Host DMA target register is accessed using a standard host read or write (no separate DMA acknowledge is used).

Figure 8.6 Host DMA Target Transfers



To configure the DMA target register for little-endian Host transfers, set the LE bit in the Host Control register. In this case, data bytes are swapped when transferring data between SDRAM and the Host register.

The BSRD bit in the Host Configuration register controls the direction of the bitstream transfers for the primary bitstream interface. The

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

H_DMAREQ pin uses an edge-sensitive protocol as described in the following two subsections.

8.4.1 Incoming Transfers to DMN-8600 Device

When BSRD = 0 in the Host Configuration register, the Host transfers bitstreams to the DMN-8600 device as follows:

- H_DMAREQ is asserted when there is space in the bitstream FIFO and a DMA transfer is active.
- H_DMAREQ is deasserted on the falling edge of a write strobe ($\overline{WR}$ in I-mode, $\overline{CS}$ in M-mode) and data is sent to the Host DMA Data Register.
- H_DMAREQ is reasserted as early as the next rising edge of the write strobe if there is remaining space in the FIFO and additional data is still required to complete the DMA transfer.
- Any data sent when H_DMAREQ is deasserted is ignored and lost.

8.4.2 Outgoing Transfers from DMN-8600 Device

When BSRD = 1 in the Host Configuration register, the Host receives bitstreams from the DMN-8600 device as follows:

- H_DMAREQ is asserted when there is data in the bitstream FIFO.
- H_DMAREQ is deasserted on the falling edge of a read strobe ($\overline{RD}$ in I-mode, $\overline{CS}$ in M-mode) to the Host DMA Data register.
- H_DMAREQ is reasserted as early as the rising edge of the read strobe if there is additional data in the bitstream FIFO.
- Any DMA read that occurs when H_DMAREQ is deasserted will result in undetermined data (that is, spurious data).

8.5 Power Management

When DMN-8600 is in standby mode, asserting the $\overline{CS}$ pin causes a chip reset, which clears standby mode and reactivates the internal PLL.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.6 Host Interface Registers

The Host interface contains a set of seven 32-bit registers that are accessed directly using address [2:0]. These registers configure the Host interface for operation.

Table 8.2 and Table 8.3 show the mapping of the Host Interface registers. The mapping depends on the width of the data bus and the endian setting. The H32 and BSSEP bits in the Host Control Register determine whether the Host interface transfers bitstreams in Mode A (Time-multiplexed 32-bit and DMA 32-bit I/O mode), Mode B (Time-multiplexed16-bit and DMA 16-bit I/O mode), or Mode C (16-bit and a separate 8-bit Primary Bitstream Port I/O mode)

The LE bit in the Host Control Register selects between big-endian and little-endian byte ordering.

Table 8.2 Host Register Mapping (32 bits wide)

Address	Big Endian		Little Endian	
	DATA[31:16]	**DATA[15:0]**	**DATA[31:16]**	**DATA[15:0]**
0x0	Read: Version Register Write: Ignored	Read: Host Control Write: Host Control	Read: Version Register Write: Ignored	Read: Host Control Write: Host Control
0x1	Read: Version Register Write: Ignored	Read: Host Control Write: Ignored	Read: Version Register Write: Ignored	Read: Host Control Write: Ignored
0x2/0x3	Host Data Register [31:0]		Host Data Register [31:0]	
0x4/0x5	Host Address Register [31:0]		Host Address Register [31:0]	
0x6	Host DMA Data Register [31:0]		Host DMA Data Register [7:0, 15:8, 23:16, 31:24]	
0x7	Reserved		Reserved	

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 8.3 Host Register Mapping (16 bits wide)

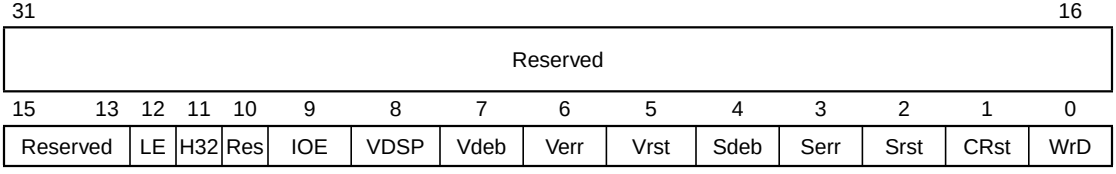
Address	Big Endian: DATA[15:0]	Little Endian: DATA[15:0]
0x0	Host Control Register	Host Control Register
0x1	Version Register (Ignore Writes)	Version Register (Ignore Writes)
0x2	Host Data Register [31:16]	Host Data Register [15:0]
0x3	Host Data Register [15:0]	Host Data Register [31:16]
0x4	Host Address Register [31:16]	Host Address Register [15:0]
0x5	Host Address Register [15:0]	Host Address Register [31:16]
0x6	DMA Data Register [15:0]	DMA Data Register [7:0, 15:8]
0x7	Reserved	Reserved

8.6.1 Host Interface Control Register

This 16-bit wide register at Host space address 0x0 and CBus address 0x60000 specifies configuration and Host control information. It is accessible by both H_ADDR[2:0] and the CBus address listed above. After reset, all bits in the Host Configuration Register are reset to zero. In 32-bit mode, this register occupies the lower 16 data bits when the external host accesses either register ADR = 0x0 or ADR = 0x1.

Host Control Register

Address: 0x0 (Host space) 0x60000 (CBus)



LE **LE (Little Endian)** **12**

If LE is set, then host DMA transfers between SDRAM and the host DMA data register are byte swapped, otherwise they are not swapped.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

If H32 is set, then bits [31:24] of the host DMA data register are swapped with [7:0] and bits [23:16] are swapped with [15:8].

If H32 is not set, then bits [15:8] are swapped with [7:0]. If H32 and LE are not set, then the least significant half of the host address register is at host address 0x5 and the most significant half is at address 0x4.

If H32 is not set and LE is set, the least significant half is at host address 0x4 and the most significant half at address 0x5.

If H32 and LE are not set, then the least significant half of the host data register is at host address 0x3 and the most significant half is at address 0x2.

If H32 is not set and LE is set, the least significant half is at host address 0x2 and most significant half at address 0x3.

1 = Host DMA transfers between SDRAM and the Host DMA Data Register are byte swapped.

0 = They are not swapped.

H32 **H32** **11**

A read/writable bit that is set by the host processor if host slave transfers are 32 bits wide, and cleared if host transfers are 16-bits wide. If H32 is set, host DMA transfers are 32-bit wide host DMA targets. If H32 is clear, then host DMA transfers are 16-bit wide host DMA targets.

1 = Host transfers are 32-bits wide.

0 = Host transfers are 16-bits wide or a separate 8-bit data port.

IOE **Interrupt Output Enable** **9**

A read/writable bit that is the active-high output enable for the Host Interrupt pin. The Host Interrupt pin is an open-drain signal that is active LOW. Hence, setting this bit results in a 0 being driven onto the INT pin. This enable bit must be reset to 0.

VDSP **VDSP Reset** **8**

A self clearing writable bit that resets the DMN-8600 Video DSP including the instruction and result queues.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Other units are not reset. This bit can be used to recover from a hang in the video DSP. This bit should not be set if there are pending memory transactions in the video DSP.

Vdeb **Video Debug Interrupt** **7**

A read/writable bit that is set by the host processor to force the Video RISC core to enter the DEBUG state. The debug handler can poll this bit to determine the cause of entering debug state. This bit is masked by the processor already being in the DEBUG state. This bit should be cleared by the interrupt handler; this serves as the acknowledge.

1 = RISC core enters debug state.

0 = Default state.

Verr **VCPU Error State** **6**

A read/writable flag that is set when the Video RISC core has entered the ERROR state. This bit will remain set until cleared by software, external reset or setting the Chip Reset bit.

1 = Send interrupt to RISC core.

0 = Cleared by software.

Vrst **Video CPU Reset** **5**

A read/writable bit that resets the DMN-8600 Video SPARC processor core and Video DSP. Other units, including SDRAM control, ME processing, video channel, system SPARC and host interface are not reset by this bit to allow debug access. This bit is automatically set by the same mechanisms that set Chip Reset. It remains set after the reset sequence and must be cleared by a control register write, either from host interface or the system SPARC. After the VCPU reset bit is set, a full chip reset must be performed with the Chip Reset bit before resuming execution (when the host resets this bit) since pending RISC core transactions are left in an undefined state.

1 = Drive $\overline{\text{H\_INT}}$ pin LOW.

0 = $\overline{\text{H\_INT}}$ pin is floating.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Sdeb **Debug Interrupt** **4**

The host sets this read/writable bit to force the processor's RISC core to enter the DEBUG state. The debug handler can poll this bit to determine the cause of entering DEBUG state. If the processor is already in the debug state, this bit is masked. The interrupt handler should be used to clear this bit; this serves as the acknowledge.

1 = Enter debug state.

0 = Normal operation.

Serr **CPU Error State or RISC Core Error Flag** **3**

The processor sets this bit when the RISC core enters the ERROR state. CPUErr remains set until cleared by software, an external reset, or when the ChipRst bit is set.

1 = RISC core in error state.

0 = Normal operation.

Srst **CPU Reset** **2**

Setting this read/writable bit resets the SPARC core and Video DSP. It does not reset other units, such as the SDRAM controller, ME, video channel, video SPARC, video DSP and Host interface--to allow debug access.

This bit is cleared by chip reset when no external host interface is present (as determined by the Mode pins) to allow the system SPARC processor to attempt booting from PROM.

It is set by reset when the host interface is present. This holds an DMN-8600 with an uninitialized SDRAM in reset until the host completes downloading, when the host clears CPU Reset to begin execution.

If the host sets CPU reset while processing is being performed for debugging purposes, a full chip reset must be performed with the Chip Reset bit before resuming execution since pending RISC core transactions are left in an undefined state.

Note: It is not possible to predict when a write to CPURst and/or ChipRST takes effect relative to other events in the DMN-8600 processor due to the unpredictable latency of the DMN-8600 internal bus arbitration.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

1 = Reset RISC core and DSP.

0 = Normal operation.

Crst **Chip Reset** **1**

This writable, self-clearing bit resets the DMN-8600 chip, except for portions of the Host interface, with all affected units are left in a quiescent state.

This bit is also set by driving the external $\overline{RST}$ signal active, and is held until $\overline{RST}$ is released.

The contents of the SDRAM are undefined after setting ChipRst. (Use CPURst instead to inspect the SDRAM.)

1 = Reset processor chip.
0 = Normal operation.

WrD **0**

The value of this bit is written in the Host Control Register. The field that needs to be written is indicated by a logic value 1 in that bit position. A logic value 0 in that bit position indicates that the field is not updated.

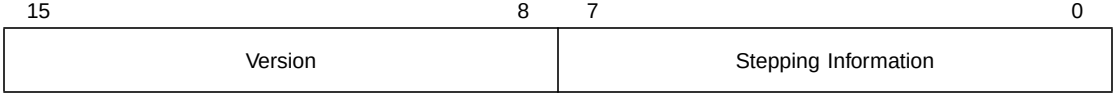
Value to write to other Host Control Register bit fields.

8.6.2 Version Register

This 16-bit register indicates the silicon chip version number. The version is in the most-significant eight bits, and stepping information is in the least significant eight bits. It is accessible as a Host space register using H_ADDR[2:0], or as a CBus register. The initial value is 0x0900.

Version Register

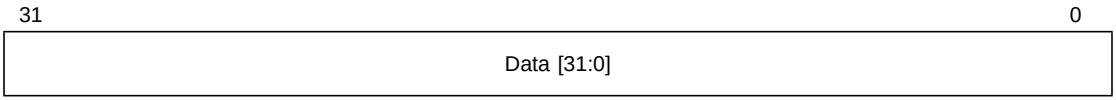
Address: 0x1 (Host space) 0x60044 (CBus)



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Host DMA Data Register

Host Space Address: 0x6



8.6.3 Host DMA Data Register

The Host DMA Data Register is used when the Host configures the DMN-8600 processor as a DMA target to transfer bitstream data to and from the DMN-8600 device. It is accessible only as a Host space register at address 0x6 via H_ADDR[2:0].

This register is the DMA target for the bitstream interface when BSSEP is cleared in the Host Configuration register.

The size of the DMA reads or writes are 16 bits if H32 is cleared and 32 bits if H32 is set.

Reads or writes to this register when the H_DMAREQ pin is not asserted are ignored.

8.6.3.1 Reading the Host DMA Data Register

Reading this register when H_DMAREQ is asserted transfers 16 or 32 bits from the internal bitstream FIFO to the host DMA channel and deasserts H_DMAREQ.

At the end of the read, if additional data is available in the bitstream FIFO, H_DMAREQ is reasserted.

8.6.3.2 Writing to the Host DMA Data Register

Writing to this register when H_DMAREQ is asserted transfers 16 or 32 bits of data from the host DMA channel to the internal bitstream FIFO and deasserts H_DMAREQ.

If additional space is available in the internal bitstream FIFO, H_DMAREQ is reasserted.

If LE is set, then Host DMA transfers between SDRAM and the Host DMA Data Register are byte swapped, otherwise they are not swapped.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.6.4 Host Data Registers

The Host uses these two 16-bit wide data registers to form a 32-bit data word for CBus register and SDRAM access. They are accessible only as Host space registers at addresses 0x3 and 0x2 via H_ADDR[2:0]. When the Host tries to read data from these registers, the $\overline{\text{H\_WAIT}}$ signal is asserted until the data from the address in the host address registers is available in the Host Data registers. When the Host writes data to the Host Data register, the data will eventually write into the address indicated in the Host Address registers.

8.6.4.1 16-bit Host Mode with LE = 1

If LE is set in 16-bit Host mode, the least significant 16 data bits are in the register at host address 0x2, and the most significant 16 data bits are in the register at host address 0x3.

When performing a read to a new address, read the 0x3 Host data register first followed by a read to 0x2.

When performing a write to a new address, write to the 0x3 data register then write to 0x2.

8.6.4.2 16-bit Host Mode with LE = 0

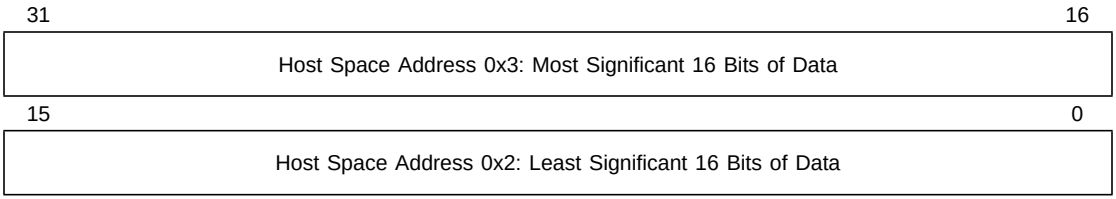
If LE is cleared in 16-bit Host mode, the most significant 16 data bits are in the register at host address 0x2, and the least significant 16 data bits are in the register at host address 0x3.

When performing a read to a new address, read the 0x2 Host data register first followed by a read to 0x3.

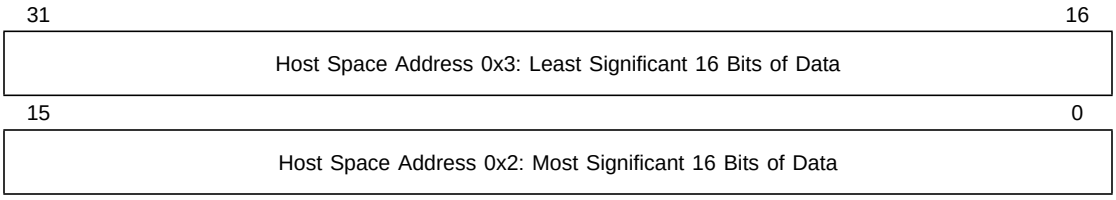
When performing a write to a new address, write to the 0x2 data register then write to 0x3.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Host Data Register (16-Bit Mode for LE = 1)

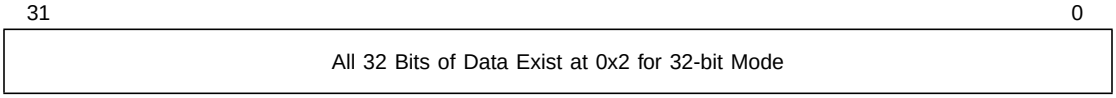


Host Data Register (16-Bit Mode for LE = 0)



Host Data Register (32-Bit Mode)

Host Address Space: 0x2



8.6.4.3 32-Bit Host Mode

In 32-bit Host mode, the two 16-bit registers are combined into a single 32-bit register at address 0x2 (LE has no effect in this case).

When performing a read to a new address, read the 0x2 Host data register first. Reading this register triggers a 32-bit read, which also loads the 0x3 Host data register.

When performing a write to a new address in 32-bit mode, the 0x3 data register should be written last. Writing the 0x3 data register triggers a 32-bit write by combining the data with the 0x2 data register.

Writing the 0x2 Host data register triggers a write in 32-bit host mode; however, in 16-bit host mode only the host register is updated.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.6.5 Host Address Register

The Host address register is composed of two 16-bit wide address registers which the Host writes via H_ADDR[2:0] to set up the address for indirect CBus and SDRAM memory address space accesses.

8.6.5.1 16-Bit Host Mode

If LE is set in 16-bit Host mode, the least significant 16 bits of the address are in the register at Host address 0x4, and the most significant 16 bits of the address are in the register at Host address 0x5.

If LE is cleared in 16-bit Host mode, the most significant 16 bits of the address are in the register at Host address 0x4, and the least significant 16 bits of the address are in the register at Host address 0x5.

8.6.5.2 32-Bit Mode

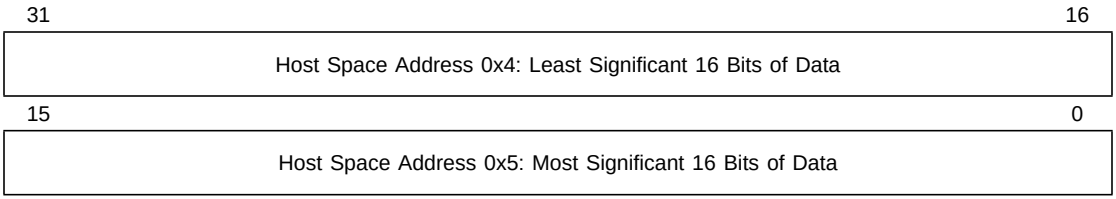
In 32-bit host mode, the two 16-bit registers are combined into a single 32-bit register at address 0x4 (LE has no effect in this case). Only bits [28:0] of the address are used for the CBus/SDRAM address. Addresses outside the SDRAM and CBus space will be ignored on writes and return undefined data on reads.

If bit 28 is set, the low order 24 bits of the address are interpreted as a CBus address. If bit 28 is clear, the low order 28 bits of the access are interpreted as an SDRAM address. If bit 31 of the address is set, then the address is autoincremented by 4 after each read or write of the data register that triggered the SDRAM access.

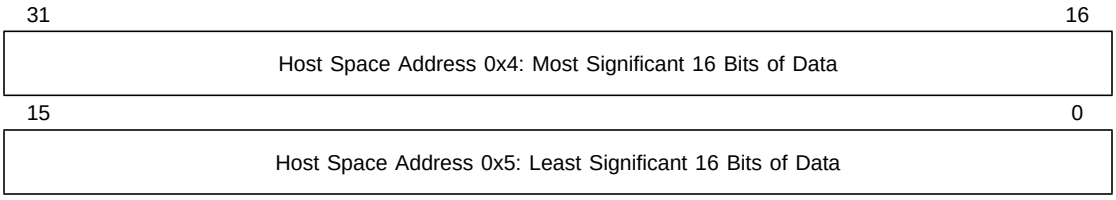
Table 8.4 shows the auto-increment support of the Host Address Register.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Host Address Register (for LE = 1 in 16-Bit Mode)



Host Address Register (for LE = 0 in 16-Bit Mode)



Host Address Register (32-Bit Mode)

Host Address Space: 0x4

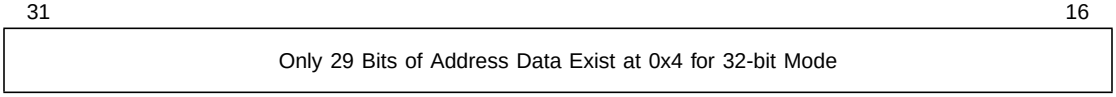


Table 8.4 Auto-increment Support of Host Address Registers

Operation	16-Bit Interface	32-Bit Interface
Read ADR = 0x2	Autoincrement	Autoincrement
Read ADR = 0x3	No autoincrement	Autoincrement
Write ADR = 0x2	No autoincrement	Autoincrement
Write ADR = 0x3	Autoincrement	Autoincrement

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.7 Host DMA Registers

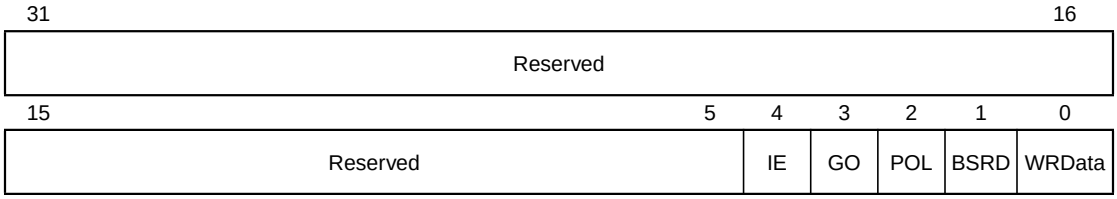
All host DMA registers are 32-bit CBus registers which are accessible to the SPARC core or Host Interface using 32-bit loads or stores.

8.7.1 Host DMA Configuration Register (CBus Address: 0x60060)

This register at control bus address 0x60060 specifies Host DMA configuration information.

Host DMA Configuration Register

CBus Address: 0x60060



IE **4**

If IE is set, then a Host DMA completion interrupt is generated when a host DMA transfer completes; otherwise no interrupt is generated.

If IE is clear, no Host completion interrupt is generated.

Incoming transfers are complete when the value of the Host DMA Next Address register equals the value of the Host DMA Stop Address register, and the DMA data has been written to SDRAM.

Outgoing transfers are complete when the value of Host DMA Next Address equals the value of the Host DMA Stop Address, and the DMA data has been transferred to the system. In other words, the DMA FIFO is empty.

1 = Host completion interrupt enabled

0 = Host completion interrupt disabled

GO **3**

GO is set by microcode or by the host processor to begin host DMA transfers.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The GO bit is cleared by hardware after the transfer is completed.

Software that clears the GO bit when BSRD is clear flushes the remaining contents of the host DMA transfer FIFO to SDRAM and terminates the transfer without generating an interrupt.

Clearing the GO bit when BSRD is set (outgoing transfer) discards the remaining contents of the host DMA FIFO and terminates the transfer. Software clearing the GO bit in either direction will generate an interrupt if enabled (IE bit is programmed to logic value 1). Since this flush will take some time to complete, software should keep polling the GO bit after writing zero until the GO bit reads as zero to ensure that the last data has been written to SDRAM.

During incoming transfers using the host DMA mode, the host interface drops the last byte if the data was received after the GO bit was cleared by software.

Note: The GO bit should be cleared, only if necessary, and read as zero before software changes the value of BSRD or any other host DMA register other than the Stop Address register.

1 = Start host DMA transfers

0 = Flush host DMA FIFO and terminate transfer without interrupt

POL **2**

If POL bit is set, then H_DMAREQ pin is active LOW; otherwise, it is active HIGH.

0 = H_DMAREQ is active HIGH

1 = H_DMAREQ is active LOW

BSRD **Bitstream READ** **1**

If BSRD is set, the Host DMA transfers are read by the system–that is, the data is being output by DoMiNo.

If BSRD is clear, the Host DMA transfers are written by the system. BSRD should not be changed while the GO bit is set.

1 = DMN-8600 processor outputs the DMA data

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

0 = System writes the Host DMA transfers

WRData **WRITE Data** **0**

The value of this bit is written to any selected bits during Host DMA configuration register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

Value is written to selected bits.

8.7.2 Host DMA NextAddress Register (CBus Address: 0x60064)

This register at control bus address 0x60064 specifies the SDRAM address for host DMA data. When GO is set, this register specifies the starting SDRAM address where host DMA data will be stored or read. As DMA data is transferred, this register is updated, automatically by hardware, to point to one byte after the last transferred byte in SDRAM. This register is read by microcode to determine how many bytes of DMA information have been transferred. The two least significant and the upper 4 bits of this register must be zero.

8.7.3 Host DMA StopAddress Register (CBus Address: 0x60068)

This register at control bus address 0x60068 specifies the transfer stop SDRAM address for host DMA data. When the Next Address reaches the value in this register and the DMA data has been transferred to SDRAM or the system, the DMA transfer is completed and the GO bit is cleared. This register can be reloaded while a DMA transfer is active to extend the length of a DMA operation. In a normal operation, this register should never be greater than or equal to the address value stored in the Limit Address register. The two least significant and the upper 4 bits of this register must be zero.

While a DMA transfer is active, the Stop Address register should not be reloaded with the address value of the Next Address register. It is possible that the Next Address hits the old Stop Address when the Stop Address Register is reloaded with a new value.

Note: Reloads of the stop address must be synchronized with DMA completion detection so that the DMA operation is not restarted after the GO bit is cleared or a completion interrupt is generated.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

8.7.4 Host DMA BaseAddress and LimitAddress Registers (CBus Address: 0x6006C and 0x60070)

The host DMA base address register at control bus address 0x6006C specifies the SDRAM address for the beginning of the host DMA circular SDRAM buffer. The host DMA limit address register at control bus address 0x60070 specifies the SDRAM address for the first byte after the host DMA buffer. When the host DMA next address reaches the value in the host DMA limit address register, it is reloaded with the host DMA Base Address register value before transferring additional data. The two least significant and the upper four bits of these registers must be zero.

8.8 Master DMA Registers

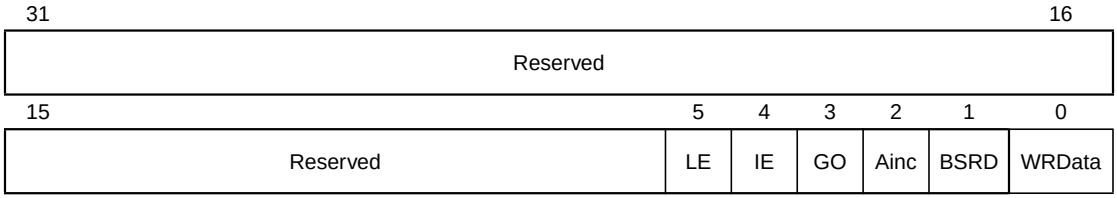
All master DMA registers are 32-bit CBus registers which are accessible to the SPARC core or Host Interface using 32-bit loads or stores.

8.8.1 Master DMA Configuration Register (Cbus Addr: 0x6F000)

This register at control bus address 0x6F000 specifies master DMA configuration information.

Master DMA Configuration Register

Memory Space Address: 0x6F000



LE **5**

If LE is set, master DMA transfers between SDRAM and the async master bus are byte swapped; otherwise, they are not swapped. Bits [15:8] of MData are swapped with [7:0].

1 = Transfers byte swapped

0 = Transfer not byte swapped

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IE **4**

If IE is set, a Master DMA completion interrupt is generated when a Master DMA transfer completes; otherwise no interrupt is generated. If IE is clear, no Master completion interrupt is generated.

Incoming transfers are complete when the value of the Master DMA Next Address register equals the value of the Master DMA Stop Address register and the DMA data has been written to SDRAM.

Outgoing transfers are complete when the value of the Master DMA Next Address equals the value of Master DMA Stop Address, and the DMA data has been transferred to the system–that is, when the DMA FIFO is empty.

1 = Master completion interrupt enabled

0 = Master completion interrupt disabled

GO **3**

GO is set by microcode to begin master DMA transfers.

The GO bit is cleared by hardware after the transfer is completed.

Software that clears the GO bit when BSRD is clear flushes the remaining contents of the master DMA transfer FIFO to SDRAM and terminates the transfer without generating an interrupt.

Clearing the GO bit when BSRD is set (outgoing transfer) discards the remaining contents of the master DMA FIFO and terminates the transfer. Software clearing the GO bit in either direction will generate an interrupt if enabled (IE bit is programmed to logic value 1). Since this flush will take some time to complete, software should keep polling the GO bit after writing zero until the GO bit reads as zero to ensure that the last data has been written to SDRAM.

During incoming transfers using the master DMA mode, the master interface drops the last byte if the data was received after the GO bit was cleared by software.

Note: The GO bit should be cleared, only if necessary, and read as zero before software changes the value of BSRD or any

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

other master DMA register other than the stop address register.

1 = Start master DMA transfers

0 = Start master DMA transfers

Flush master DMA FIFO and terminate transfer without interrupt

Ainc **2**

If Ainc is set, then master mode DMA transfers increment the host bus address by two bytes each time a master mode DMA bus cycle is performed.

If Ainc is clear, then master mode DMA transfers use the same host bus address for an entire DMA transfer.

1 = Increments host bus address by 2 bytes

0 = Uses same host bus address for entire DMA transfer

BSRD **Bitstream READ** **1**

If BSRD is set, the master DMA transfers will be read by the system-- i.e. the data is being output by DoMiNo.

If BSRD is clear, master DMA transfers will be written by the system. BSRD should not be changed while the GO bit is set.

1 = DMN-8600 processor outputs the DMA data

0 = System writes the master DMA transfers

WRData **WRITE Data** **0**

The value of this bit is written to any selected bits during host DMA configuration register writes. Bits to be written are selected by placing a “1” in each desired location as the register is written. Any bit positions that contain a “0” during register writes remain unchanged.

Value is written to selected bits.

8.8.2 Master DMA External Address Register (Cbus: 0x6F014)

This register at control bus address 0x6F014 specifies the external device address for master mode master DMA transfers. When GO is set, this register specifies the starting master bus address where master DMA data will be stored or read. As data is transferred, this register is updated if Ainc is set in the master DMA configuration register. The two

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

least significant and the upper four bits of this register must be zero. During burst reads, the three least significant bits must be zero.

8.8.3 Master DMA NextAddress Register (Cbus Addr: 0x6F004)

This register at control bus address 0x6F004 specifies the SDRAM address for master DMA data. When GO is set, this register specifies the starting SDRAM address where master DMA data will be stored or read. As DMA data is transferred, this register is updated automatically by hardware to point to one byte after the last transferred byte in SDRAM. This register is read by microcode to determine how many bytes of DMA information have been transferred. The two least significant and the upper four bits of this register must be zero.

8.8.4 Master DMA StopAddress Register (Cbus Addr: 0x6F008)

This register at control bus address 0x06F008 specifies the transfer stop SDRAM address for master DMA data. When the Next Address reaches the value in this register and the DMA data has been transferred to SDRAM or the system, the DMA transfer is completed and the GO bit is cleared. This register can be reloaded while a DMA transfer is active to extend the length of a DMA operation. In a normal operation, this register should never be greater than or equal to the address value stored in the Limit Address register. The two least significant and the upper four bits of this register must be zero.

While a DMA transfer is active, the Stop Address register should not be reloaded with the address value of the Next Address register. It is possible that the Next Address may hit the old Stop Address when the Stop Address Register is reloaded with a new value.

Note: Reloads of the stop address must be synchronized with DMA completion detection so that the DMA operation is not restarted after the GO bit is cleared or a completion interrupt is generated.

8.8.5 Master DMA BaseAddress and LimitAddress Registers (Cbus Address: 0x6F00C and 0x6F010)

The master DMA base address register at control bus address 0x6F00C specifies the SDRAM address for the beginning of the master DMA

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

circular SDRAM buffer. The master DMA limit address register at control bus address 0xdF010 specifies the SDRAM address for the first byte after the master DMA buffer.

When the master DMA next address reaches the value in the master DMA limit address register, it is reloaded with the master DMA base address register value before transferring additional data. The two least significant and the upper four bits of these registers must be zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

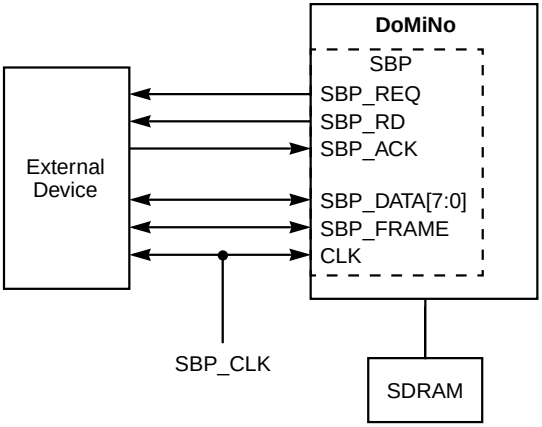
Chapter 9
Secondary Bitstream Interface

This chapter describes the secondary bitstream interface and contains the following sections:

As a separate bitstream port, the secondary bitstream interface is a synchronous, flow-controlled 8-bit data port. The secondary bitstream port pins are shared with the ATAPI interface pins. As a result, the secondary bitstream port is only available when the storage port is in SD mode. Transfers are clocked from the rising edge of SBP_CLK. The direction of the port is controlled by the BSRD bit in the Secondary Bitstream Configuration register. (See Figure 9.1).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 9.1 Secondary Bitstream Port (SBP)



Both SBP_ACK, SBP_REQ and SBP_RD signals can be either active LOW or active HIGH depending upon the setting of the POL bit and the WRREQ bit. The BSRD bit determines the direction of the Secondary Bitstream port. See Table 9.1 for transfer direction and active state details.

Table 9.1 Secondary Bitstream Pin Configuration

Bit Setting	Pin Name	BSRD=0 Incoming		BSRD=1 Outgoing	
		POL=0	POL=1	POL=0	POL=1
WRREQ = 0	SBP_ACK	active HIGH	active LOW	active HIGH	active LOW
	SBP_REQ	active HIGH	active LOW	active HIGH	active LOW
	SBP_RD	0	0	1	1
WRREQ = 1	SBP_ACK	active HIGH	active LOW	active HIGH	active LOW
	SBP_REQ	0	1	active HIGH	active LOW
	SBP_RD	active HIGH	active LOW	0	1

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

9.1 WRREQ = 0

When WRREQ = 0, the SBP may be a bitstream output (BRSD = 1) or input (BRSD = 0) as described in the following two sections.

9.1.1 Bitstream Output – Outgoing Transfers (BRSD = 1)

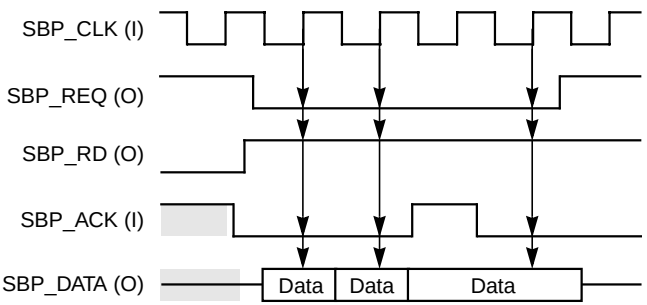
When the SBP is a bitstream output for WRREQ = 0:

- SBP_REQ is asserted along with the oldest byte of bitstream data on each clock in which the bitstream FIFO is not empty (partially filled or full).
- SBP_RD is asserted as long as the DMA transfer is enabled (Go is set in the Secondary Bitstream Configuration register and the Secondary Next Address is not equal to the Secondary Stop Address) to indicate the direction of the transfer.

When the system asserts SBP_ACK in a clock in which SBP_REQ is asserted, the oldest byte of bitstream data is removed from the FIFO at the end of the clock, and the next oldest byte (if present) in the FIFO is presented on the next clock.

Figure 9.2 shows outgoing transfers from the bitstream port with WRREQ = 0, POL = 1 and BSRD = 1. In the figure, note that SBP_ACK and SBP_REQ are shown as active LOW. The DMN-8600 processor drives SBP_DATA.

Figure 9.2 Bitstream Port Outgoing Transfers with WRREQ = 0, POL = 1 and BSRD = 1



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

9.1.2 Bitstream Input – Incoming Transfers (BSRD = 0)

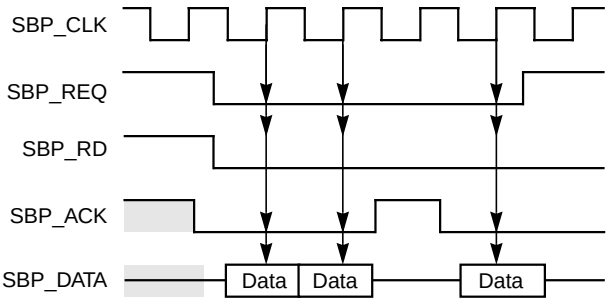
When the SBP is a bitstream input for WRREQ = 0:

- SBP_REQ is asserted on each clock in which the bitstream FIFO is empty (not full).
- SBP_RD is LOW to indicate the direction of the transfer.

When SBP_ACK is asserted by the system in a clock in which SBP_REQ is asserted, one byte of bitstream data is added to the FIFO at the end of the clock. As long as the DMA operation is enabled and SBP_CLK is running at 27 MHz or slower while the DMN-8600 internal clock is running at 148.5 MHz, SBP_REQ will be asserted continuously. This allows a non-flow-controlled transfer to be used, if necessary.

Figure 9.3 shows incoming transfers to the bitstream port with WRREQ = 0, POL = 1 and BSRD = 0. In the figure, note that SBP_ACK and SBP_REQ are shown as active LOW. The system drives SBP_DATA.

Figure 9.3 Bitstream Port Incoming Transfers with WRREQ = 0, POL = 1 and BSRD = 0



When SBP_ACK is asserted by the system in a clock in which SBP_REQ is asserted, one byte of bitstream data is added to the FIFO at the end of the clock. As long as the DMA operation is enabled and SBP_CLK is running at 27 MHz or slower while the DMN-8600 internal clock is running at 148.5 MHz, SBP_REQ will be asserted continuously. This allows a non-flow-controlled transfer to be used, if necessary.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

9.2 WRREQ = 1

If WRREQ is set in the Secondary Bitstream Control register, the SBP_REQ pin is defined as a DMA write request and the SBP_RD pin is redefined as a DMA read request.

9.3 FRAME Pin Transfers

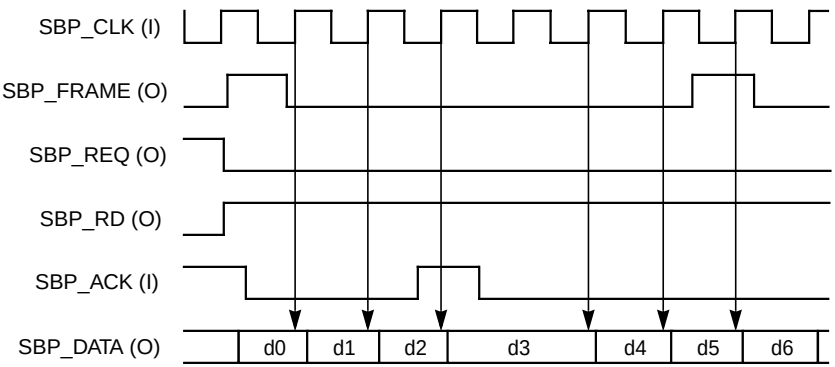
Secondary bitstream transfers optionally can be framed with the SBP_FRAME pin as follows:

- The SBP_FRAME signal is asserted with the first byte of an incoming or outgoing packet and is deasserted with the next byte of the packet.
- The size of Secondary Bitstream packets is specified by the PACKSIZE register.

9.3.1 Outgoing FRAME Transfers

Figure 9.4 shows outgoing transfers using SBP_FRAME.

Figure 9.4 Outgoing FRAME Transfers



For outgoing FRAME transfers, the following occurs:

- SBP_FRAME is asserted on the first clock of all outgoing packets (outgoing transfers are assumed to start on a packet boundary).

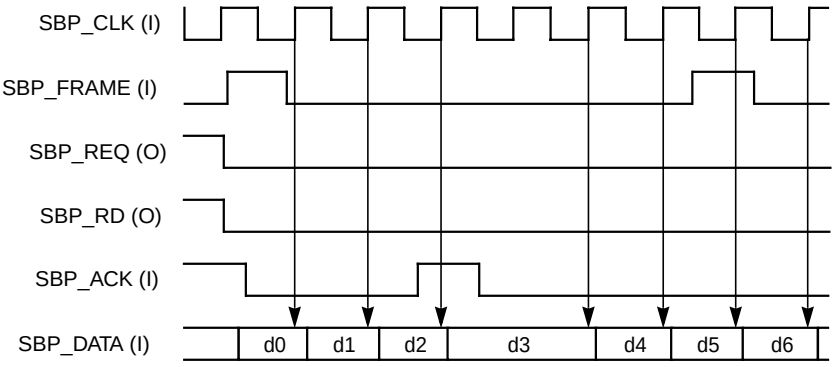
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- When packet framing is enabled, the number of clocks from the start of one outgoing packet to the start of the next outgoing packet is determined by both the PACKDELAY register and the PACKSIZE register. Only when both of their values are equal can the next outgoing SBP_FRAME be asserted.

9.3.2 Incoming FRAME Transfers

Figure 9.5 below displays incoming FRAME transfers.

Figure 9.5 Incoming FRAME Transfers



For incoming FRAME transfers (BSRD = 0), the following occurs:

- After each PACKSIZE byte is captured (long packet), incoming bytes are discarded until SBP_FRAME is asserted.
- If SBP_FRAME is asserted before PACKSIZE bytes are captured (Early Frame):
  - Zero bytes are inserted to fill out PACKSIZE bytes in the SDRAM (preserving the packet boundary alignment in the SDRAM).
  - SBP_REQ is deasserted the cycle after Early Frame until zero stuffing in completed.
  - After zero stuffing is completed, capture resumes on the next packet boundary.
- If zero stuffing occurs and the contents of the Next Address register equals the contents of the Stop Address register:
  - The GO bit in the bitstream configuration is cleared.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

  - Transfer stops.
- If hardware flow control is being used with the Stop Address to break up large transactions, then early packets must not occur.

9.4 FIFO and Buffer Operation

The FIFO is implemented as a dual-port 64-word by 64-bit memory READ/WRITE port. Note that this section applies only to the Secondary port.

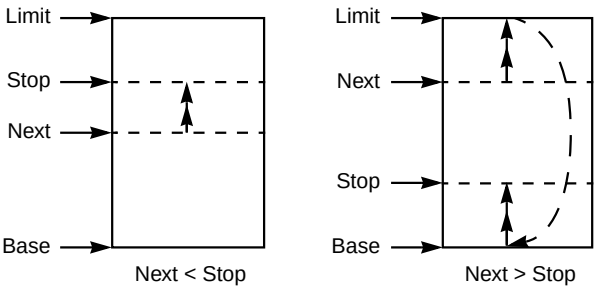
For incoming transfers, data must not remain in the bitstream FIFO for more than one millisecond. For cases where the FIFO buffer is not completely filled within the one millisecond time constraint, the partially filled buffer will be flushed to SDRAM.

The FIFO is flushed under two situations:

1. There is no SDRAM access for one millisecond (only the contents in the FIFO are flushed to SDRAM). It is possible that a few bytes may remain in the buffer.
2. Software clears the GO bit (the contents in the Packet register are first moved into the FIFO, the GO bit is cleared to 0 if the Packet register is not full, then the FIFO contents are flushed into SDRAM).

The remaining FIFO contents are discarded as outgoing bitstreams, if software clears the GO bit. The bitstream is stored to or fetched from SDRAM in the form of a circular buffer (see Figure 9.6).

Figure 9.6 Circular Buffer



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The bitstream is located in SDRAM between the Base Address and the Limit Address. When the bitstream transfer starts, the SDRAM address is in the Next Address register. The Next Address is then incremented; it points to the next transfer address after the current SDRAM access has completed.

If the Next Address reaches the Stop Address, the transfer is completed. If the Next Address reaches the Limit Address, the next SDRAM address is the Base Address. Section 9.5, “Secondary Bitstream Interface Registers,” describes the Next Address, Stop Address, Base Address, and Limit Address registers in detail with respect to the SBP.

9.5 Secondary Bitstream Interface Registers

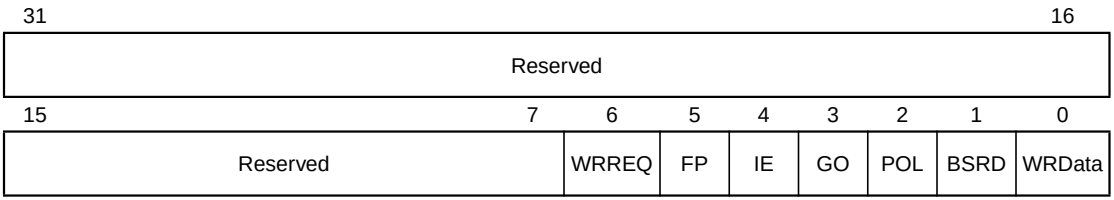
All secondary bitstream interface registers are 32-bit CBus registers which are accessible to the SPARC core or Host interface using 32-bit loads or stores.

9.5.1 Secondary Bitstream Configuration Register

This register resides at memory space address 0x080820. The Secondary Bitstream Configuration register specifies Secondary Bitstream configuration information.

Secondary Bitstream Configuration Register

CBus Address: 0x080820



WRREQ **6**

If WRREQ is set, the SBP_REQ pin is redefined as a DMA write request, which is only asserted on outgoing transfers when the bitstream FIFO is not empty.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

If WRREQ is set, the SBP_RD pin is redefined as a DMA read request, which is asserted on incoming transfers when the bitstream FIFO is not full.

1 = SBP_REQ is a DMA write request and SBP_RD is a DMA read request

0 = SBP_REQ is a DMA request and SBP_RD is the direction of transfer

FP **5**

If FP is set, then incoming packets on the Secondary Bitstream Port are framed with the SBP_FRAME pin.

If FP is clear, then SECFRAME is deasserted on incoming packets and SBP_FRAME is ignored on incoming packets.

1 = SBP incoming packets are framed

0 = SBP_FRAME is ignored on incoming, not sent on outgoing

IE **4**

If IE is set, a SBP interrupt is generated when a Secondary Bitstream DMA transfer is completed; otherwise, no interrupt is generated.

If IE is clear, no SBP interrupt is generated.

Incoming transfers are complete when the value of the Secondary Next Address register reaches the value of the Secondary Stop Address Register, and the bitstream data has been written to SDRAM.

Outgoing transfers are complete when the value of the Secondary Next Address register reaches the value of the Secondary Stop Address and the bitstream data has been transferred to the system–that is, the bitstream FIFO is empty.

Note: Hardware should not assert SBP_REQ for bytes after the stop address. The number of bytes to capture can be precomputed at the beginning of the transfer.

1 = Host completion interrupt enabled

0 = Host completion interrupt disabled

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

GO **3**

Microcode or the host processor sets GO to begin Secondary Bitstream transfers.

Hardware clears the GO bit after the transfer is completed.

Software clears the GO bit when BSRD is cleared, flushes the remaining contents of the Secondary Bitstream transfer FIFO to SDRAM, and terminates the transfer.

Clearing the GO bit when BSRD is set (outgoing transfer) discards the remaining contents of the Secondary Bitstream FIFO and terminates the transfer. Because this flush takes some time to complete, software should keep polling the GO bit after writing zero until the GO bit reads as zero to ensure that the last data has been written to SDRAM.

Note: The GO bit should be cleared, only if necessary, and read as zero before software changes the value of BSRD or any other Secondary Bitstream DMA register other than the Stop Address register.

Clearing the GO bit by hardware or software resets the packet framing. Software should set the stop address so the number of bytes transferred in a DMA operation is a multiple of SECPACKSIZE.

1 = Start SBP transfers

0 = Flush SBP FIFO and terminate transfer.

POL **2**

If POL is set and WRREQ is clear, SBP_REQ and SBP_ACK pins are active LOW; otherwise, they are active HIGH.

If POL and WRREQ are set, the SBP_RD, SBP_REQ and SBP_ACK pins are active LOW; otherwise, they are active HIGH.

1 = SBP_RD/SBP_REQ/SBP_ACK are active HIGH

0 = SBP_REQ/SBP_ACK are active LOW

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

BSRD **Bitstream READ** **1**

If BSRD is set, the Secondary Bitstream transfers are read by the system. The DMN-8600 processor outputs the bitstream.

If BSRD is clear, then the system writes Secondary Bitstream transfers. BSRD should not be changed while the GO bit is set. All bits in this register are set to zero on reset.

1 = DMN-8600 processor outputs the bitstream

0 = System writes the SBP transfers

WRData **WRITE Data** **0**

The value of this bit is written to any selected bits during Secondary Bitstream Configuration Register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

Value is written to selected bits.

9.5.2 Secondary NextAddress Register (CBus Addr: 0x080824)

This register at control bus address 0x080824 specifies the SDRAM address for secondary bitstream data. When GO is set, this register specifies the starting SDRAM address where secondary bitstream data will be transferred. As bitstream data is transferred, this register is updated to point to one byte after the last transferred byte in SDRAM. This register is read by microcode to determine how many bytes of bitstream information have been transferred. The two least significant bits and the upper four bits of this register must be zero.

9.5.3 Secondary Stop Address Register (Cbus Addr: 0x080828)

This register at control bus address 0x080828 specifies the transfer stop SDRAM address for secondary bitstream data. When the value in the Next Address reaches the value in this register and the bitstream data has been transferred to SDRAM or the system, the DMA transfer is completed and the GO bit is cleared. This register can be reloaded while a DMA transfer is active to extend the length of a DMA operation. In a normal operation, this register should never be greater than or equal to Limit Address register. The two least significant and the upper four bits of this register must be zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

While a DMA transfer is active, the Stop Address register should not be reloaded with the address value of the Next Address register. It is possible that the Next Address may hit the old Stop Address when the Stop Address register is reloaded with a new value.

Note: Reloads of the stop address must be synchronized with DMA completion detection so that the DMA operation is not restarted after the GO bit is cleared or a completion interrupt is generated.

9.5.4 Secondary Base Address and Limit Address Registers (CBus Addr: 0x08082C and 0x080830)

The Secondary Base Address register at control bus address 0x08082C specifies the SDRAM address for the beginning of the circular SDRAM buffer. The Secondary Limit Address register at control bus address 0x080830 specifies the SDRAM address for the first byte after the buffer. When the next address reaches the value in the Secondary Limit Address register, it is reloaded with the secondary base address register value before transferring additional data. The two least significant and the upper four bits of these registers must be zero.

9.5.5 Secpacksize Register (CBus Addr: 0x080834)

The Secondary Pack Size register at control bus address 0x080834 specifies the packet framing size for the secondary bitstream port in bytes. The two least significant and the upper 16 bits must be zero.

9.5.6 SecPacketDelay Register (CBus Addr: 0x080838)

The secondary packet delay register at control bus address 0x080838 specifies the number of clocks from the start of one secondary bitstream outgoing packet to the start of the next outgoing packet when packet framing is enabled.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 10
Host Async Master Interface

This chapter describes the host async master interface and contains the following sections:

The DMN-8600 async master interface has three pin configurations, described below, to optimize the total pin-count for different package and system configurations. One of three configurations is selected by the MCONFIG[1:0] pins. The configuration pins must not change after reset.

10.1 No Master

In this configuration (MCONFIG = 00), the master interface is not used. The master space is swapped with the DRAM space in the SPARC address map, changing the start of SDRAM to zero. The SCPU rst bit will remain set after reset to allow the host processor to download microcode before booting.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

10.2 Host (Slave) plus Limited Master

In this configuration (MCONFIG = 01), the master interface pins are separate from the host slave interface pins. The master space is swapped with the DRAM space in the SPARC address map, changing the start of SDRAM to zero. The SCPU rst bit will remain set after reset to allow the host processor to download microcode before booting. In this configuration, only self-paced SDRAM signals are possible and only one chip select, $\overline{\text{M\_CS}}$[0], is available.

The upper and lower order address and control pins other than M_ALE (M_ADDR[26:22], M_ADDR[5:1], $\overline{\text{M\_CS}}$[0], $\overline{\text{M\_OE}}$, M_RD/$\overline{\text{WR}}$) replace serial I/O pins for UART1, SPI and IR transmit (UART2, IR receive, and IDC are still available). The master interface pins M_ADDR[21:6]/M_D[15:0] are shared with ATAPI_DATA[15:0] pins when the storage interface is in ATAPI mode and SD_DATA[7:0]/SBP_DATA[7:0] when the storage interface is in SD mode. Since the master and ATAPI control lines are separate, both may operate concurrently.

ATAPI cycles are round robin arbitrated with master interface cycles. SD and secondary bitstream transfers can not be interrupted. Consequently SPARC master accesses will generate a host error and Master DMA accesses will abort the DMA operation when the SD DMA or Secondary Bitstream DMA GO bits are on. (See Section 10.7, “Async Master Status/Time-Out Register,” page 10-17.)

10.3 Master Replaces Slave

In this configuration (MCONFIG = 10), master interface pins replace slave interface pins. The master space is at zero of the SPARC address map, and the SCPU rst bit is cleared after reset, causing the system SPARC to begin fetching instructions from a PROM at address zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

10.4 Cycle Types

The async master interface generates master-mode cycles for a variety of asynchronous slave peripherals. Each chip select can be configured to perform one of several different types of transfers, with different transfer types selected by specific feature bits in the configuration register. In addition, individual components of the cycle timing may be varied by changing the value of various timing parameter fields in the chip select configuration. Both multiplexed address/data and demultiplexed master cycles may be performed. Demultiplexed cycles support up to a 2 Kbyte address range per chip select, and multiplexed cycles support up to 64 Mbytes of addressing.

The various cycle type options are explained in separate sections below. Unless otherwise noted, the cycle type variations are independent of each other and can be mixed and matched as the system designer sees fit.

10.4.1 Data Strobe Mode

There are two types of data strobes: 68K mode and SRAM mode. The mode selection changes the functions of three pins, as described in Table 10.1 below:

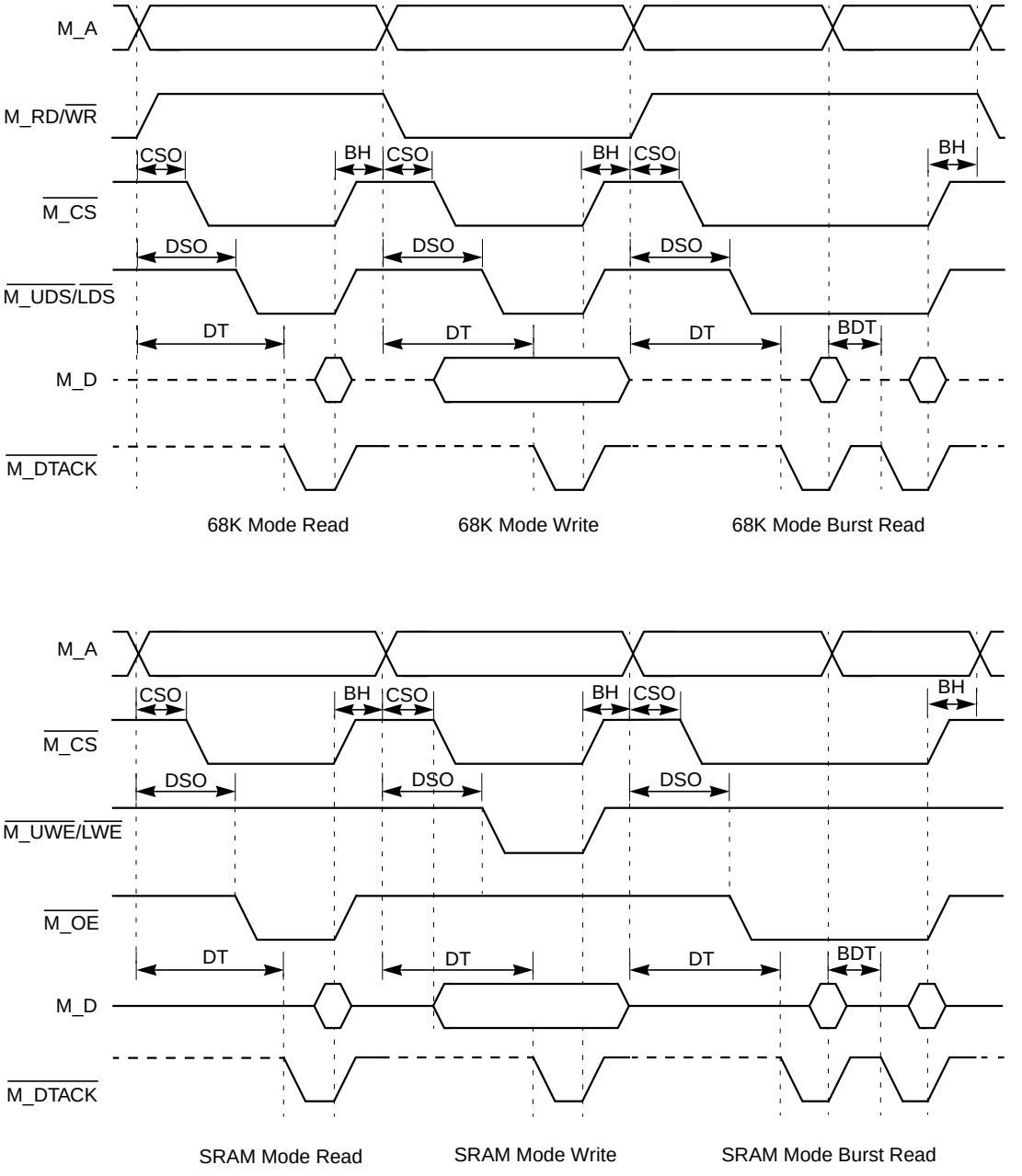
Table 10.1 Pin Functions in Different Modes

68K Mode	SRAM Mode
M_RD/$\overline{WR}$ – This signal is HIGH for a read cycle and LOW for a write cycle. It has the same timing as the address bits.	$\overline{M\_LWE}$ – This signal strobes LOW during writes if the lower byte lane, D[7:0], is to be written in this transfer. For reads it remains unasserted.
$\overline{M\_UDS}$ – This signal strobes LOW during both read and write cycles if the upper byte lane, D[15:8], is used in the transfer.	$\overline{M\_UWE}$ – This signal strobes LOW during writes if the upper byte lane, D[15:8], is to be written in this transfer. For reads, it remains unasserted.
$\overline{M\_LDS}$ – This signal strobes LOW during both read and write cycles if the lower byte lane, D[7:0], is used in the transfer.	$\overline{M\_OE}$ – This signal strobes LOW during a read cycle, but remains HIGH during a write.

Figure 10.1 shows the basic operation of these two cycle types.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 10.1 Async Master Read and Write Cycles



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

10.4.2 Transfer Acknowledge Mode

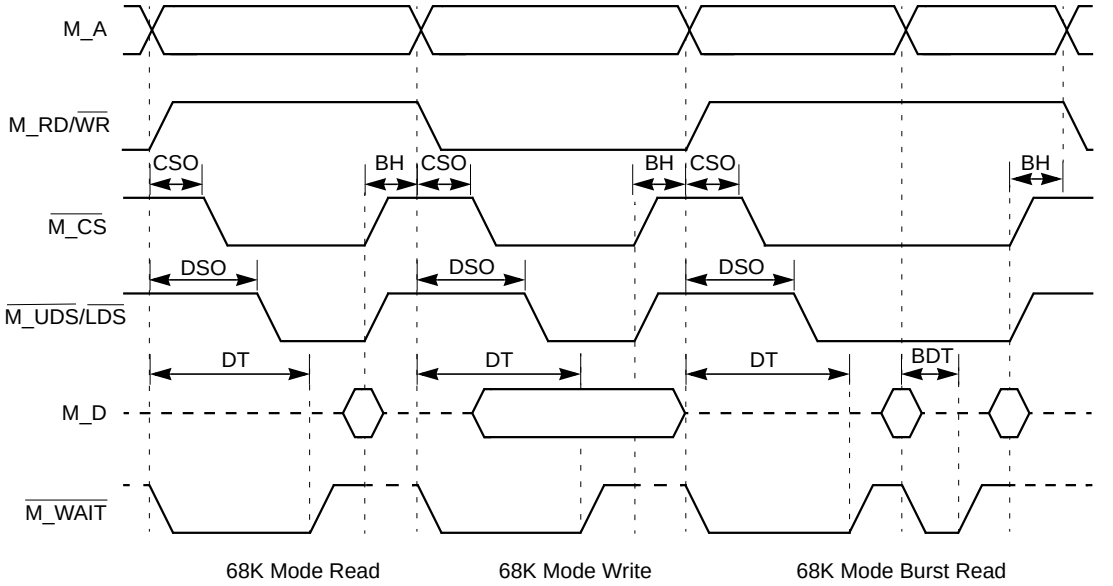
A transfer can use one of two methods of indicating cycle completion: the M_DTACK (data transfer acknowledge) signal or the M_WAIT signal. M_DTACK indicates the end of a transfer by driving low until the master deasserts M_CS. M_WAIT, on the other hand, will assert later in the cycle to indicate that the device needs more time to finish the transaction. The cycle is complete when the M_WAIT signal is deasserted.

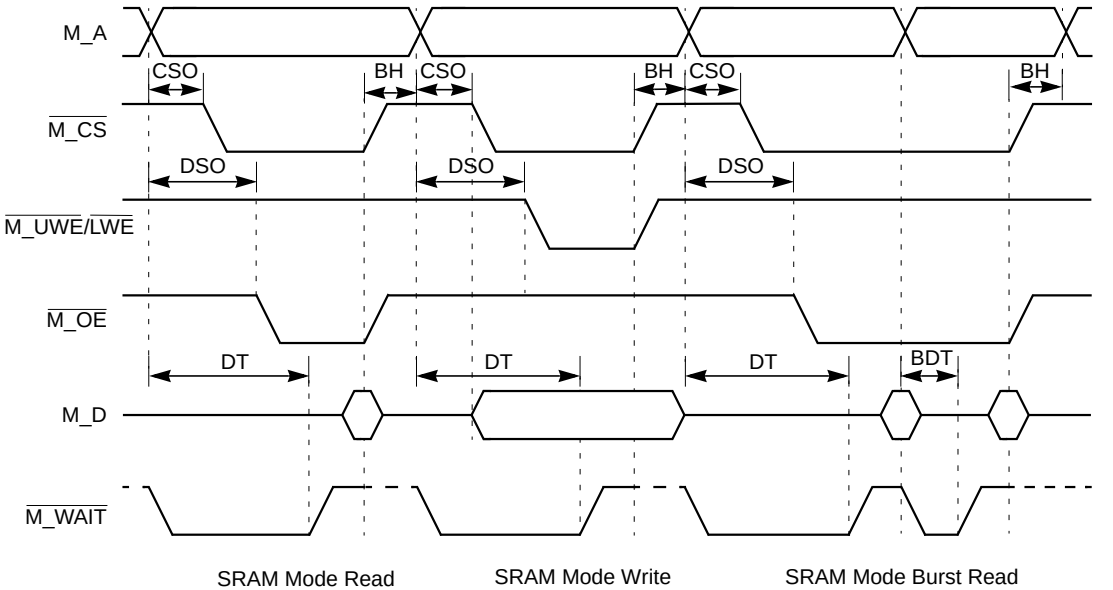
For self-paced transfers, as shown in Figure 10.2, the choice of M_WAIT or M_DTACK affects only the assertion level and the pin used for the acknowledge signal. The master waits a programmed amount of time before asserting M_DTACK or deasserting M_WAIT and terminating the cycle.

For device-paced accesses, the master waits a programmed amount of time before sampling M_DTACK or M_WAIT, as described in Section 10.4.5, “Device-Paced Transfers,” page 10-8.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 10.2 Self-Paced Async Master Cycles with $\overline{\text{M\_WAIT}}$





Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

10.4.3 Timing Parameters

Several parameters can be used to extend the cycle length by adding delays in particular places. These delays are specified in internal clock period units.

The timing parameters for the basic cycle are illustrated in Figures Figure 10.1 and Figure 10.2, with a description of their function in Table 10.2 below.

Table 10.2 Timing Parameters

Parameters	Range of Values	Description
CSO	0–15	Delay from M_A and M_RD/$\overline{WR}$ driven to fall of $\overline{M\_CS}$. For write cycles, Data is driven out along with fall of $\overline{M\_CS}$.
DSO	0–15	Delay from Addr and M_RD/$\overline{WR}$ driven to fall of data strobes ($\overline{M\_UDS}$ and $\overline{M\_LDS}$ or $\overline{M\_OE}$, $\overline{M\_UWE}$ and $\overline{M\_LWE}$).
DT	0–127	For self-timed accesses, delay from M_A and M_RD/$\overline{WR}$ driven to $\overline{M\_DTACK}$ asserted or $\overline{M\_WAIT}$ deasserted. For self-timed accesses, M_DTACK is pulsed low for one clock, then brought high for one clock before being 3-stated whereas $\overline{M\_WAIT}$ is driven LOW at the start of the cycle then brought HIGH for one clock before being 3-stated. On a read cycle, data is latched in at the clock cycle after $\overline{M\_DTACK}$ is asserted or $\overline{M\_WAIT}$ deasserted. For device-paced $\overline{M\_DTACK}$ transfers, see Section 10.4.5, “Device-Paced Transfers,” page 10-8. For device-paced $\overline{M\_WAIT}$ transfers, this value indicates the amount of time to wait before sampling the $\overline{M\_WAIT}$ pin for cycle completion. For more information, see Section 10.4.5, “Device-Paced Transfers,” page 10-8.
BDT	0–15	For self-timed burst accesses, delay from end of the previous data transfer phase to assertion of $\overline{M\_DTACK}$ or deassertion of $\overline{M\_WAIT}$ for the next phase. For device-paced transfers, see Section 10.4.5, “Device-Paced Transfers,” page 10-8.
BH	0–15	Added delay from end of cycle ($\overline{M\_CS}$ and data strobes deasserted) to start of next cycle. During the BH time, Addr and M_RD/$\overline{WR}$ are held stable and all other signals are held inactive. On a write cycle, the Data bus is also driven with the write data for the BH time.
AS	0–3	Multiplexed address set up time from address and M_ALE asserted to fall of M_ALE.
AH	0–1	Multiplexed address hold time from fall of M_ALE to Addr/Data change.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

10.4.4 Burst Transactions

Each chip select can be programmed to respond to burst as well as single-cycle transactions. Bursts are done only for read transactions, write requests are always broken up into multiple single-cycle writes. The basic burst transactions are illustrated on the right side of Figure 10.1.

A burst can consist of two or more beats, or individual transfers. Only the low-order (nonmultiplexed) address bits change between beats, no data strobes change. Typically, bursting is enabled for devices like Flash ROM with fast page mode which can respond to the address changes with new data.

The DT and BDT parameters are used to control the rate at which the burst progresses. The other parameters are only used at the beginning and end of the entire burst, as shown in Figure 10.1 or Figure 10.2. The DT parameter controls the delay associated with the first beat of the burst, and BDT the delay for subsequent beats. The fastest self-timed burst uses DT = BDT = 0 and takes one bus time unit per beat. With this timing, $\overline{\text{M\_DTACK}}$ would remain low for all of the beats, going high for a single cycle after and then going 3-state in the next cycle after that.

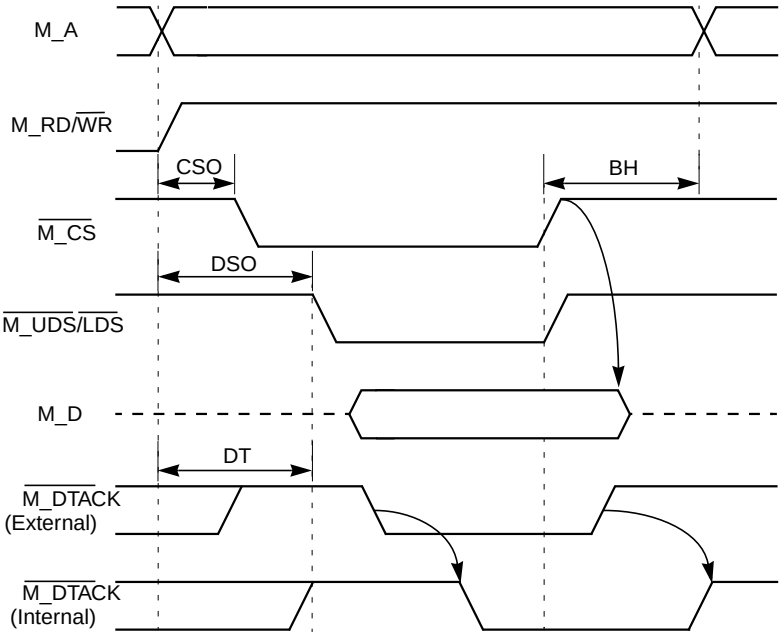
Note: the number of beats in a burst varies depending on the particular access. DMN-8600 supports single-beat accesses (for 8- or 16-bit reads), two-beat transfers (for 32-bit reads) and 4-beat transfers for aligned 64-bit reads in master DMA transfers and cache line fills.

10.4.5 Device-Paced Transfers

Some devices require the host to wait an indefinite time for the transfer to complete. These devices drive the $\overline{\text{M\_DTACK}}$ or $\overline{\text{M\_WAIT}}$ signal themselves, instead of relying on the host to self-time the transfer. Figure 10.3 shows an example of this kind of cycle.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 10.3 Async Master Device-paced Transfer



The external M_DTACK signal is used as an input to generate an internal M_DTACK signal synchronized to the internal clock. The external M_DTACK signal is asynchronous, which must be asserted for at least two rising edges of the internal clock.

As usual, after the internal M_DTACK signal has been active for a clock, read data is sampled and M_CS and the data strobe signals are deasserted. The designer must be careful to ensure that the target device has deasserted M_DTACK before the next cycle samples it (starting with the clock edge that asserts the data strobes); otherwise, it may cause premature termination of the following cycle. Typically, this requires an extension of the BH parameter.

Note that for device-paced burst transfers, the M_DTACK or M_WAIT line is sampled only for the first beat of the transfer. Subsequent beats use the BDT timing.

When using device-paced transfers, the DT timing parameter controls the delay before the master begins sampling M_DTACK or M_WAIT. This is most useful when the data acknowledge strobe is a M_WAIT signal,

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

since it gives the target device more time to decode the transfer and assert the M_WAIT signal. The delay is from the start of the cycle to the beginning of the first possible data phase (the next clock is the first one in which the data and transfer acknowledge will be sampled). The DT parameter can also be used to delay the sampling of M_DTACK.

10.4.6 Multiplexed Address Cycles

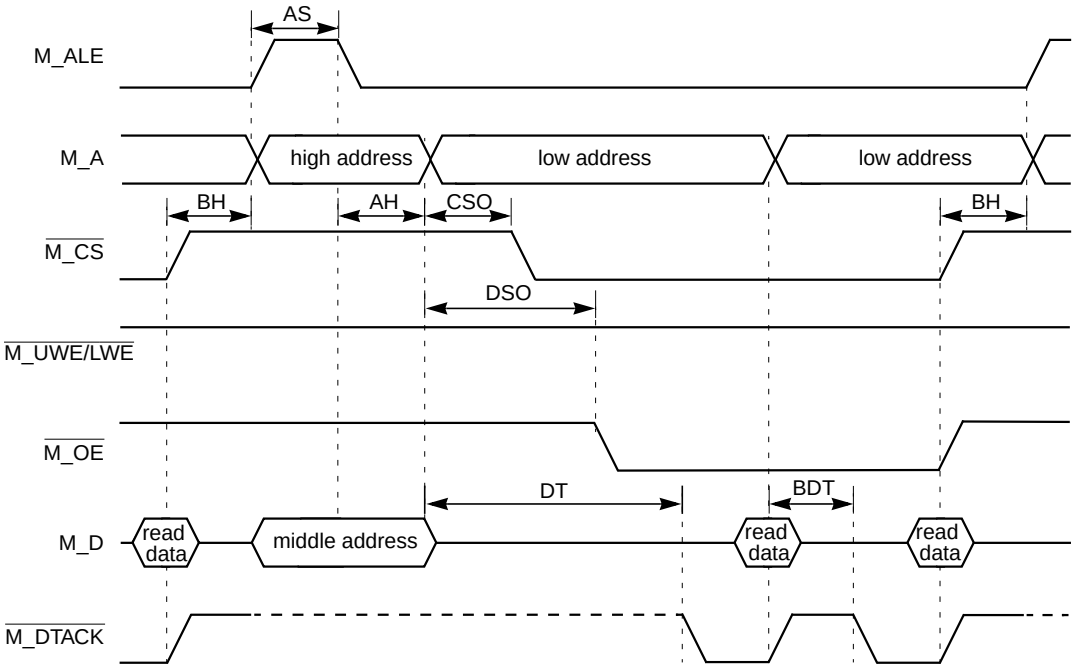
The DMN-8600 multiplexes the middle address bits 21 to 6 on the Addr/Data lines. For these cycles, the address bits are captured with an external latch. An address latch enable pin (M_ALE) is provided to control the latch.

The address multiplexing feature is selectively enabled on a per-chip-select basis. This allows the designer to have both large address space devices (like Flash ROMs) that use the multiplexing and small address space devices (UARTs, etc.) that do not. The large address space devices take a small cycle hit in performance per burst, while the smaller devices are directly addressed by the limited number of dedicated address pins.

Figure 10.4 below shows a multiplexed address cycle:

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 10.4 Multiplexed Master Cycle



Two additional timing parameters, AS and AH, are added for this mode, as defined in Table 10.2. AS controls the delay from the assertion of M_ALE and address to the falling edge of M_ALE--in other words, the address setup time. AH controls the delay from the fall of M_ALE to the Addr/Data lines changing. If the designer can guarantee sufficient hold time for the latch (due to loading on the Addr and Data busses, for instance), this hold time can be set to zero.

10.5 Chip Select Configuration Registers

The DMN-8600 has six chip select pins, $\overline{\text{M\_CS}}$[5:0]. Each of these chip selects are controlled by a pair of 32-bit chip-select configuration registers: (0x6F020, 0x6F024), (0x6F028, 0x6F02C), (0x6F030, 0x6F034), (0x6F038, 0x6F03C), (0x6F040, 0x6F044), (0x6F048, 0x6F04C). The first of the two registers contains the personality-independent part (which is used by the decode logic to select

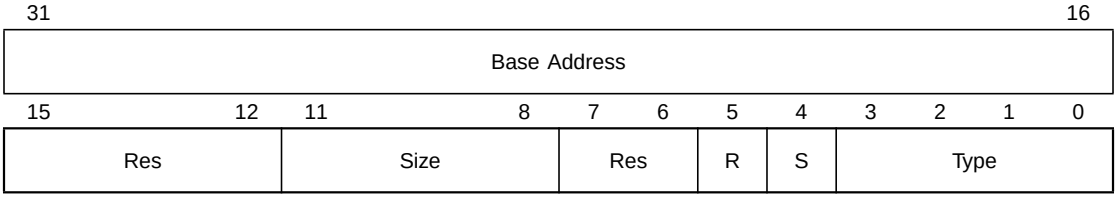
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

a chip-select and personality) and the second register contains async master specific parameters. The DMN-8600 only implements an async master personality.

The personality-independent register is shown below.

Chip Select Configuration Registers

Personality-Independent Address = 0x6F020 - 0x6F048



Base Address **[31:16]**

Base address of this chip select. In DoMiNo, the most significant 4 bits are zero. Reset value is zero.

Size **[11:8]**

Chip-select size. Controls size of chip-select region in power-of- two sizes. The Base Address parameter must be a multiple of this size as well. Valid values are:

- 0000 - Chip select disabled.
- 0001 - Region is 64K
- 0010 - Region is 128K
- 0011 - Region is 256K
- 0100 - Region is 512K
- 0101 - Region is 1M
- 0110 - Region is 2M
- 0111 - Region is 4M
- 1000 - Region is 8M
- 1001 - Region is 16M
- 1010 - Region is 32M
- 1011 - Region is 64M
- 1100 - Region is 64M

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- 1101 - Region is 64M
- 1110 - Region is 64M
- 1111 - Region is 64M

Note: Regarding Default value: Personality-Independent register CS0, the Size bit is reset to 1111; for the rest of these registers, however, the Size bit is reset to 0.

R **5**

Read-only. If set, writes to this chip-select region cause a host error interrupt, with status set in the Host Timeout register. Reset to zero.

S **4**

Supervisor-only. If set, SPARC accesses to this chip-select region from user mode causes a host error interrupt, with status set in the Host Timeout register. Reset to zero.

Type **[3:0]**

Personality module for this chip-select. In DoMiNo, the value is always zero which is the suggested ID of the async master module.

Note: When a master range address with no $\overline{\text{M\_CS}}$ pin selected occurs on a SPARC read, the cycle is aborted and a data access exception trap is induced in the respective SPARC.

When a master range address with no $\overline{\text{M\_CS}}$ pin selected occurs on a SPARC write the cycle is aborted and a host error interrupt is generated.

When a master address with no $\overline{\text{M\_CS}}$ pin selected occurs on an async master DMA cycle, the entire transfer is aborted and the GO bit is cleared.

The personality-dependent register is shown below:

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chip Select Configuration Register

Personality-dependant Address = 0x6F024 - 0x6F04C

31	30 29	28	27	26	25 24	23	22 19	18
D	TA	M	B	X	AS	A	CS	DS

15	14 11	10 7	6 0
DS	BH	BD	DT

D **31**

Data transfer acknowledge type. If clear, $\overline{\text{M\_DTACK}}$ will be used for transfer acknowledge. If set, $\overline{\text{M\_WAIT}}$ will be used. Reset to zero.

TA **[30:29]**

Data transfer acknowledge control, one of the following:

00	Transfer is self-paced. (Reset value)
01	Reserved
10	Transfer is device-paced with asynchronous acknowledge (input is clocked twice)
11	Reserved

M **28**

Model select. If clear, SRAM mode cycles will be run. If set, 68K mode cycles will be run. See Figure 10.2. Reset to zero.

B **27**

Burst-enable. If set, burst reads will be allowed to this CS as in Figure 10.2. If clear, bursts will be broken up into single-beat transfers. Reset to zero.

X **26**

Multiplexed address enable. If set, an address multiplex cycle will be run as in Figure 10.4. Reset to one.

AS **[25:24]**

Address setup time, corresponding to AS in Table 10.2. Reset to three.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

A **23**

Address hold time, corresponding to AH in Table 10.2. Reset to one.

CS **[22:19]**

Chip-select on time, corresponding to CSO in Table 10.2. Reset to 12.

DS **[18:15]**

Data strobe on time, corresponding to DSO in Table 10.2. Reset to 15.

BH **[14:11]**

Bus hold time, corresponding to BH in Table 10.2. Reset to 15.

BD **[10:7]**

Burst $\overline{\text{M\_DTACK}}$ timing, corresponding to BDT in Table 10.2. Reset to 15.

DT **[6:0]**

Data Transfer Acknowledge timing. For self-paced transfers this is the delay to assertion of $\overline{\text{M\_DTACK}}$/$\overline{\text{M\_WAIT}}$ (see Figure 10.2). For device-paced transfers, this is the delay before sampling the $\overline{\text{M\_DTACK}}$/$\overline{\text{M\_WAIT}}$ input (see Figure 10.3). Reset to 63.

Note: If the personality-dependant register is written by software in the middle of a master will be used to complete the current bus cycle before using the new timing parameters.

10.6 Interrupt/GPIO Configuration and Value Registers

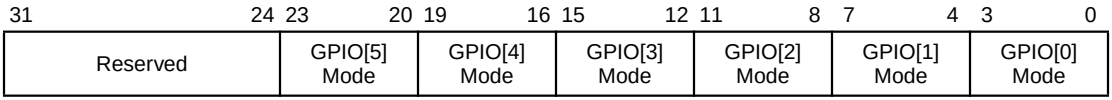
The Interrupt/GPIO Configuration register at control bus address 0x6F050 specifies async master interrupt and GPIO configuration information for six shared function/interrupt/GPIO pins. This register is set to zero on reset.

The register contains a 4-bit field for each of the six shared function/interrupt/GPIO pins. The field for GPIO[0] is stored in the least significant four bits of the register; the field for GPIO[5] is stored in the most significant four bits of the register, as shown below.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Interrupt/GPIO Configuration Register

0x6F050



The operating modes, shown in Table 10.3, are specified by the least significant 3 bits of the 4-bit field.

Table 10.3 GPIO Pin Operating Modes

Value	Mode
0	Shared function mode (not interrupt or GPIO). The shared function is defined in the pinout description (Table 6.1 and Table 18.34). Pins that do not have a shared function will act as inputs.
1	General Purpose Output
4	Negative edge triggered interrupt
5	Positive edge triggered interrupt
6	General purpose input: For pins 0 to 3, corresponding GPIO value bit is complement of input. For pins 4 to 7, corresponding GPIO value bit is same as input.
7	General purpose input or positive asserted level triggered interrupt

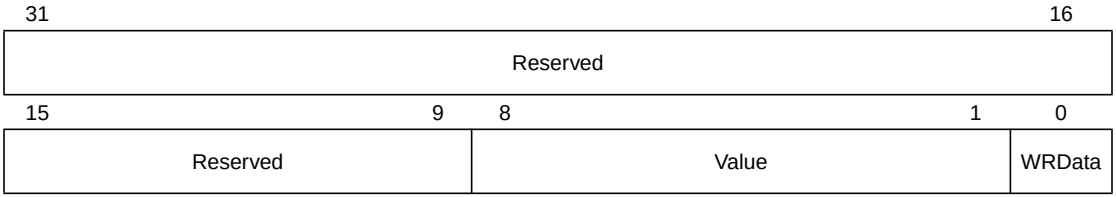
The distinction between interrupt and input is based on whether the interrupt is masked in the interrupt control register in the SPARC (some GPIOs are not connected to the interrupt controller and never generate interrupts). For edge-triggered interrupts, a rising or falling edge on the associated interrupt pin will be converted to an internal level interrupt request to the SPARCs. When interrupt acknowledge for that interrupt is received from the SPARC (or after chip reset), the internal level interrupt request will be reset. The next rising or falling edge after the level is reset will cause a new interrupt to be generated.

The most significant bit of each 4-bit field is a write enable. Writes to the Interrupt/GPIO configuration register will only update fields which have the write-enable bit set to one. Any fields that have a 0 write enable bit during register writes remain unchanged.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

GPIO Value Register

Memory Space Address: 0x6F054



Value **[8:1]**

The value field contains a 1-bit value reflecting the state of the eight GPIO/interrupt pins. The value for GPIO[0] is stored in the least significant bit of Value, the value for GPIO[5] is stored in the most significant bit of Value. For input pins, the value field is read-only value of the corresponding pin's current state. For output pins, the software can write the desired pin state to the corresponding bit.

WRData **0**

WRITE Data bit. The value of this bit is written to any selected bits during GPIO value register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

Value is written to selected bits.

10.7 Async Master Status/Time-Out Register

This register at control bus address 0x6F018 specifies the async master error status and the time-out interval for self-paced master mode cycles. The time-out interval to receive a $\overline{\text{M\_DTACK}}$ or $\overline{\text{M\_WAIT}}$ acknowledge is specified in units of 1024 internal clocks, and has a range of 0 to 255 units. A value of zero disables the time-out.

When a time-out, SD/Secondary bitstream conflict (SD or secondary bitstream DMA GO bit on and MConfig = 01 or 11) or master range address with no $\overline{\text{M\_CS}}$ pin selected occurs on a SPARC read, the cycle is aborted and a data access exception trap is induced in the respective SPARC. When a time-out, SD/Secondary bitstream conflict or master

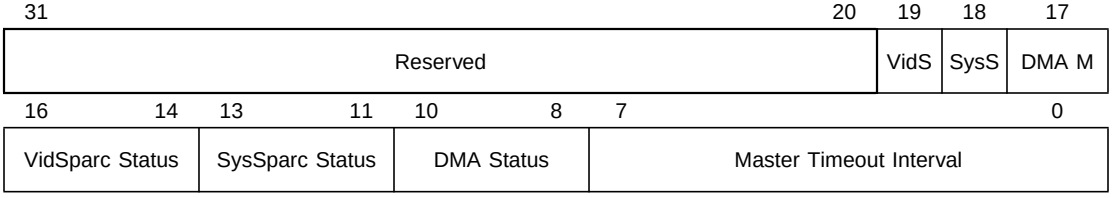
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

range address with no $\overline{M\_CS}$ pin selected occurs on a SPARC write the cycle is aborted and a host error interrupt is generated. In either case, error status is stored in this register as described below.

When a time-out, SD/Secondary bitstream conflict or master address with no $\overline{M\_CS}$ pin selected occurs on an async master DMA cycle, the entire transfer is aborted and the GO bit is cleared, which will cause a Master DMA completion interrupt. Master DMA completion interrupt handlers should check this register to see if the DMA transfer completed normally.

Async Master Status/Time-out Register

Memory Space Address: 0x6F018



VidSPARC MErr **19**

Set when multiple VidSPARC errors occur before the register is cleared.

SysSPARC MErr **18**

Set when multiple SysSPARC errors occur before the register is cleared

DMA MErr **17**

Set when multiple DMA errors occur before the register is cleared.

VidSparc Status **[16:14]**

Reflects the last master mode access error caused by video SPARC.

SysSparc Status **[13:11]**

Reflects the last master mode access error caused by the system SPARC.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

DMA Status **[10:8]**

DMA status reflects the last master mode access error caused by the async master DMA as shown below.

Value	Error Type
0	No error.
1	Write to read only chip select.
2	User mode access to supervisor chip select (only for SPARC accesses).
3	Time-out.
4	Master access with no CS selected.
5	Master access with SD or secondary bitstream DMA GO bit on and MConfig = 01 or 11.

Master Timeout Interval **[7:0]**

Time-out interval in which to receive a $\overline{\text{M\_DTACK}}$ or $\overline{\text{M\_WAIT}}$ acknowledge is specified in units of 1024 internal clocks, with a range of 0 to 255 units.

All fields of this register, except Master Timout Interval, are cleared on a read through CBus access. Writes to this register is ignored, except for the Master Timeout Interval field. As a consequence, the only way to clear the status fields and the MErr fields of this register is to read the register.

Any software routing that polls the Master Timeout Interval field also must handle the Error condition, if applicable.

This register is reset to 0.

10.8 Async Master SPARC Error Address Register

This register at control bus address 0x6F01C contains the address of the last SPARC write to generates a host error. The upper six bits are always zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

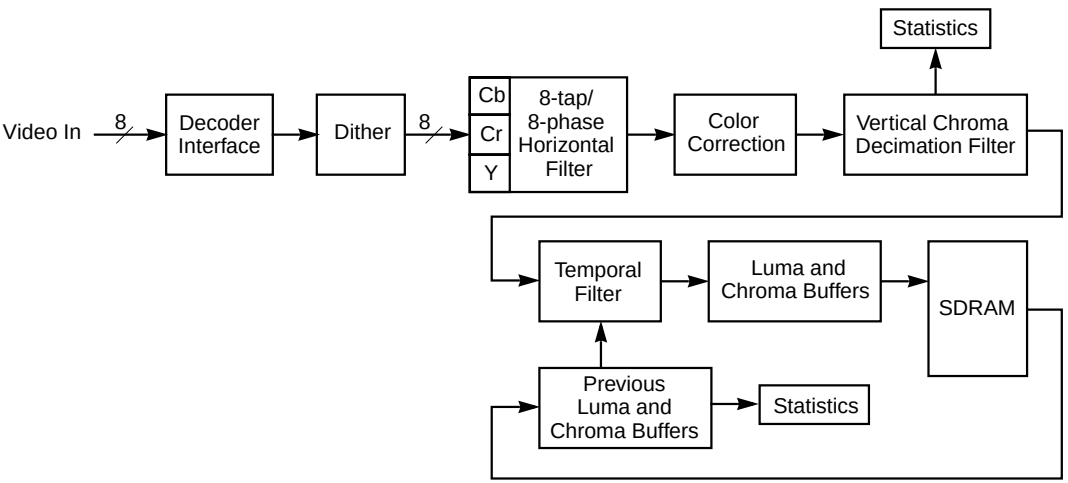
Chapter 11
Video Interface

This chapter describes the video interface and contains the following sections:

- Section 11.1, “Video Output Format”
- Section 11.2, “Video Operation”
- Section 11.3, “Video Control Register”

The DMN-8600 processor provides a video input port and a video output port. Figure 11.1 shows the data flow for the video input port. The input port captures 8-bit digital video from ITU-R 656[1] (parallel D1) or low-cost video decoder chips such as the Philips SAA7111.

Figure 11.1 Video Input Channel Data Flow



1. “ITU-R 656” refers to the ITU-R BT.656 standard (formerly known as CCIR 656).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Two programmable rectangular windows are captured from each field without software intervention, allowing for VITC and closed caption capture. Additional windows can be captured by loading a new capture window in to the input channel at the completion of each window.

In a multiple DMN-8600 system, each chip will capture the video region it requires. 8-bit values are passed through an 8-tap, 8-phase horizontal filter for resampling and noise reduction. Double buffered luma and chroma buffers are used to hold the filtered values. The DMA transfer captures up to an entire field between video interrupts. Luma and chroma information are stored separately in SDRAM using the DMN-8600 tiled image data format.

The video input channel can convert from 4:2:2 to 4:2:0 on the fly using a 3-tap filter with software specified coefficients. Frame or field filtering is supported. The video input channel can also perform on-the-fly color correction with software specified coefficients and luma and chroma temporal filtering. This combined with 4:2:2 to 4:2:0 conversion eliminates the need for chroma preprocessing and eliminates the need for loading previous data and storing current data in luma preprocessing.

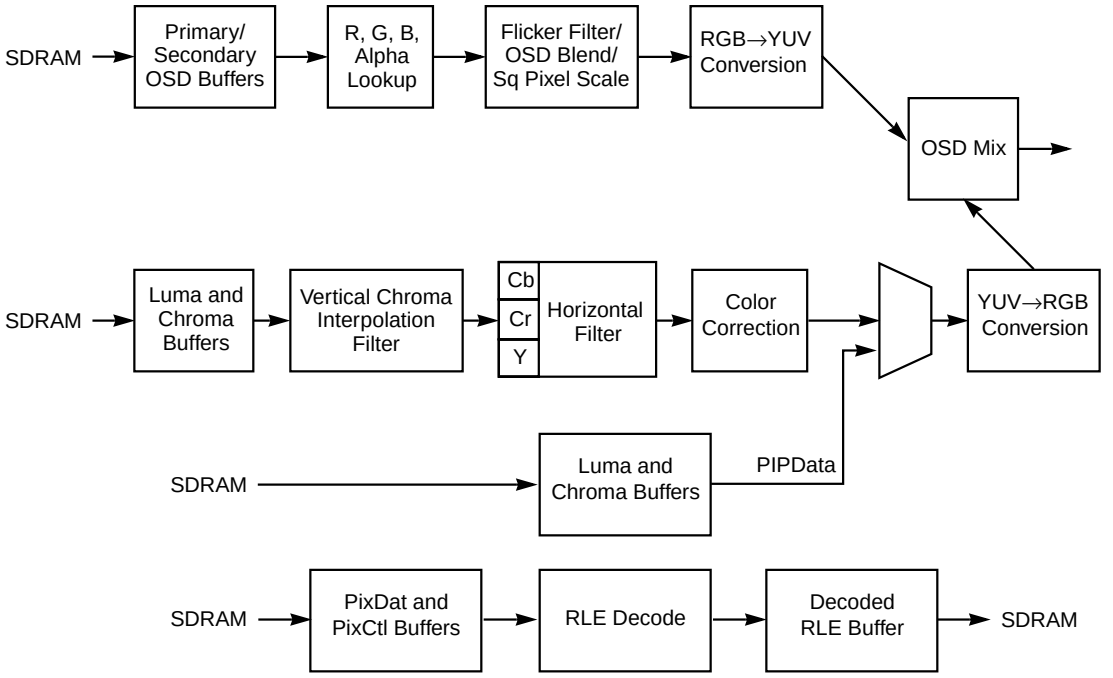
The DMN-8600 computes statistics during capture, including DC measurements, horizontal activities for strips and same-parity IT detection scores.

The input horizontal filter can be used to perform one or two pass horizontal scaling of result data from the scale(i) instructions. Extra luma and chroma input buffers are used for data coming from the DSP, and extra chroma and luma filter output buffers are used as SDRAM data buffers. When decimating data, the filter processes an average of the scaling ratio input pels (up to 2 pels) per clock. When interpolating, the 8-tap filter is split into two 4-tap filters.

Figure 11.2 shows the data flow for the video output port. The output port outputs 8-bit video data to ITU-R 656 or low-cost encoder chips such as the SAA7125 and Bt856. Two programmable rectangular windows are output for each field without software intervention. Additional data windows can be output by loading new data windows into the output channel at the completion of each window.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 11.2 Video Output Channel Data Flow



The video input channel capture window data can be inserted into the video channel output, to support PIP where the input and output timing is the same. This allows DoMiNos to be daisy chained from video out to video in to support arbitrary numbers of PIP channels. Double buffered luma and chroma buffers are used to hold the output values.

For DoMiNo, the maximum output rate is 74.25 MHz (smpte 260m). The maximum OSD pixel rate with flicker filtering and 32-bit pixels is 37.125 MHz. The DMN-8600 output values are optionally passed through an interpolating 4-tap, 8-phase horizontal filter.

The video output channel can convert from 4:2:0 to 4:2:2 on the fly using a 2-tap field or frame filter with software specified coefficients. The video input channel can also perform on-the-fly color correction with software specified coefficients. The video output channel supports a separate PIP video layer and a separate OSD/graphics overlay with 4, 8, 16 or 32-bit pixel formats and optional two tap flicker filter.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Two programmable OSD windows are output for each field without software intervention. Additional OSD windows can be output by loading new data windows into the output channel at the completion of each window.

The subpicture decoder decodes DVD format run length coded pixel data and DVD format pixel control data. To provide scaling flexibility, subpicture decoded data is output to SDRAM, with color compositing and scaling performed by the VDSP.

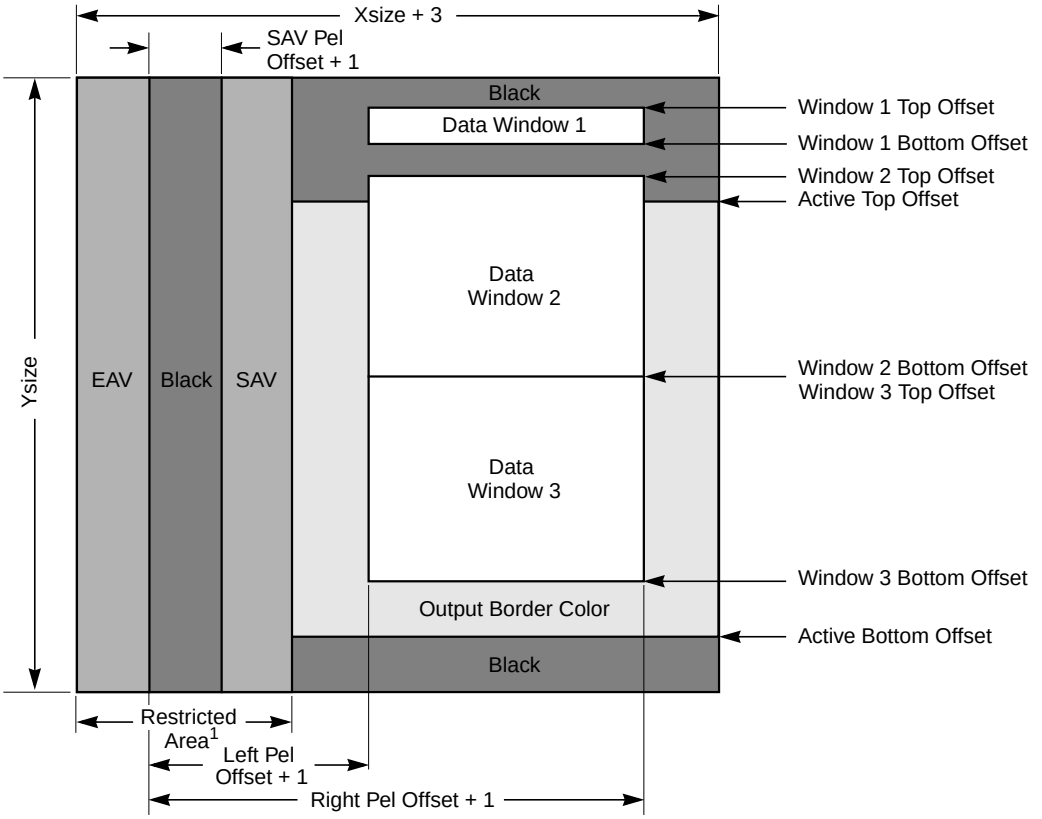
11.1 Video Output Format

Figure 11.3 shows the video output format for each interlaced field. The DMN-8600 hardware supports one blanking and one normal data window per field. However, microcode can implement multiple normal data windows. The normal data window and the blanking window may overlap the boundary of the active window.

Restriction : Pels cannot be captured within and including the EAV and SAV regions (138 pels wide).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 11.3 Output Field/Frame Format



Note: [1] Pels cannot be captured within the restricted area. The size of this area must be a minimum of 138 pels.

11.2 Video Operation

This section explains the general operation of the video interface.

11.2.1 Video Streams

The DMN-8600 processor takes as its input a single video stream synchronized with a 27 MHz Video Clock (VI_CLK[0]). The DMN-8600 processor supports both interlaced and progressive video streams. However, this document describes only interlaced streams. Figure 11.4 shows the single video stream.

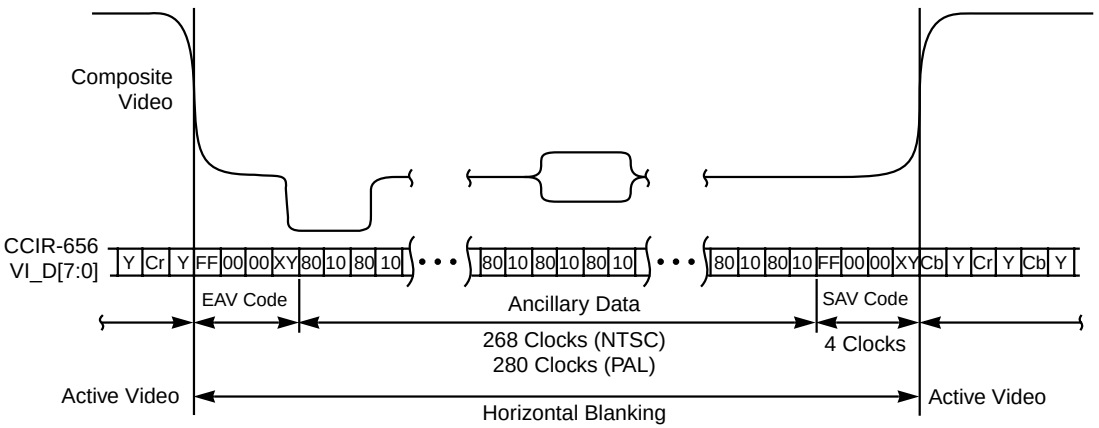
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 11.4 Single Video Stream

11.2.2 ITU-R 656 Operation

Figure 11.5 shows the input/output timing for ITU-R BT.656 operation. The VI_VSYNC signal and digital video are combined into an 8-bit input clocked at 27 MHz. Active 8-bit video data is surrounded by SAV and EAV codes. These codes are implemented using reserved 0x00 and 0xFF code sequences within the video field.

Figure 11.5 ITU-R BT.656 Timing



11.3 Video Control Register

The Video Control Register is used to control a variety of features, as described in the description below. A read-modify-write cycle must be used when accessing this register.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Video Control Register

Cbus Address: 0xC40000 0x0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res		FRC	VCP	VCL	ITS	IQF2	VCD	VCI	TF/PV	IQF	OQF	0	CC	VidInInt	CTF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FilE	0	Res	VidRGB	VideoPixFmt			PYfirst	Hfpchr	GoR	VidOutInt	Echo	GoPIP	VidFInt	GoI	GoO

FRC **29**

If set when GoI is set by software (and VCD is set), then the associated input window will perform frame based 422 to 420 vertical chroma decimation using the current and previous opposite parity input field (interlaced capture of progressive input). One chroma output line will be produced for each chroma input line, with successive chroma output lines being stored in chroma field 0 and chroma field 1. The first chroma output line is stored to chroma field 0. If clear when GoI is set (and VCD is set), then the associated input window will perform field based 422 to 420 vertical decimation of the current input field (interlaced capture of interlaced input or progressive capture of progressive input). One chroma output line will be produced for every two chroma input lines, with successive chroma output lines being stored in chroma field 0. The FRC bit setting must be the same for all windows in the same field.

VCP **28**

Vertical chroma phase. If set when GoI is set, then the input window is bottom field, otherwise it is top field for chroma decimation purposes. If set when GoO is set, then the output window is bottom field (phase 1), otherwise it is top field (phase 0) for chroma interpolation purposes. This bit is ignored for output windows with FRC clear, and input windows with FRC set.

VCL **27**

Vertical chroma last. If set and VCP and VCD are set (and FRC is not set) when GoI is set by software, then the last chroma input line will be duplicated for the last

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

two taps of the last chroma output line, otherwise the first chroma input line of the next window will be used for the last tap of the last chroma output line. If VCL is not set, then software must load the next input window before the last chroma output line is reached. VCL must be set when VCD is set and this is the last window in the current field/frame or the next window in the current field/frame has VCD clear. If VCP is clear or FRC is set when GoI is set, this bit is ignored.

ITS **26**

If set when GoI is set by software, then the associated capture window will compute inverse telecine statistics. If clear, then inverse telecine statistics are not computed. If TF is also clear, accesses are not performed to the previous field buffer in SDRAM.

IQF2 **25**

A read-only bit which is set when there are two outstanding DMA transfers in the video input stream 2 command queue. A transfer is loaded into the queue each time the GoI2 bit is set by software. A command is removed from the queue each time the transfer completes. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds) and interrupts are used for longer transfers.

VCD **24**

If set when GoI is set by software, then the associated capture window will vertically decimate the chroma output of the horizontal input filter from 422 to 420 format. If clear, then the associated capture window will not vertically decimate chroma data. If set when GoO is set by software, then the associated output window will vertically interpolate the chroma output of the horizontal output filter from 420 to 422 format. If clear, then the associated output window will not vertically interpolate chroma data. When VCD is set, the associated input or output window must have an even number of lines. VCI must be set when VCD is set and this is the first window in the current field/frame or the previous window in the current field/frame had VCD clear. VCL must be set when VCD is set and this is the last window in the current

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

field/frame or the next window in the current field/frame has VCD clear.

VCI **23**

If set and FRC and VCP are clear when GoI is set by software, then the associated capture window will use the first chroma input line for the first two taps of the first chroma output line, otherwise the last chroma input line of the previous capture window will be used for the first tap of the first chroma output line, and the first chroma input line as the second tap of the first chroma output line. VCI must be set when VCD is set and this is the first capture window in the current field/frame or the previous window in the current field/frame had VCD clear. This bit is ignored during input capture when VCP or FRC is set.

TF/PV **22**

If set when GoI is set by software, then the associated capture window will perform temporal filtering. If clear, the temporal filter is bypassed. If ITS is also clear, no accesses are performed to the previous field buffer in SDRAM. If set when GoR is set by software, then the associated rescale window will output data in interleaved YCbYCr or CbYCrY format (depending on PYfirst), otherwise the rescale window will output data in separate luma/chroma format. If set when Go0 is set by software, then the associated output window will read data in interleaved YCbYCr or CbYCr format (depending on PYfirst), otherwise the output window will read data in separate luma/chroma format. If set when GoPIP is set by software, then the associated PIP window will read data in interleaved YCbYCr or CbYCr format (depending on PYfirst), otherwise the PIP window will read data in separate luma/chroma format. Packed video data uses the luma address plane only (the chroma plane is not used), and must be in 422 format (VCD clear).

This bit should not be set with GoR, GoO or GoPIP.

IQF **21**

A read-only bit which is set when there are two outstanding echo or input capture transfers in the video input stream 1 command queue. A transfer is loaded into the queue each time the GoI or echo bit is set by software. A command is removed from the queue each

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

time the transfer completes. If both GoI and echo are set in a command, the command is removed after both echo and input capture are completed. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds) and interrupts are used for longer transfers.

OQF **20**

A read-only bit which is set when there are two outstanding DMA transfers in the video output command queue. A transfer is loaded into the queue each time the GoO bit is set by software. A command is removed from the queue each time the transfer completes. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds) and interrupts are used for longer transfers.

0 **19**

This bit must be programmed to 0.

CC **18**

If set when GoR is set by software, then the associated rescaling operation will perform color correction on the rescaled results before storing to memory. Color correction can only be performed on 4:2:2 data (bit 7 of the scale(i) instruction) or data which is vertically interpolated to 4:2:2 data (VCD bit set). If set when GoO is set by software, then the associated output window will perform color correction on the output data. Rescaling and output color correction can not be performed at the same time. If set when GoI is set by software, then the associated input window will perform color correction on the input data.

VidInInt **17**

The completion of a stream 1 echo or video capture window will generate a video completion interrupt (video input completion interrupt on DoMiNo) if VidInInt is set. If both GoI and echo are set in a command, the interrupt is signalled after both echo and input capture are completed. In DoMiNo, there are separate interrupts for input and output transfer completion, so VidInInt and VidOutInt can be set at the same time. This bit should not

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

be changed while an input or echo transfer is active or pending.

CTF **16**

If CTF and TF are set when GoI/GoI2 is set by software, then chroma and luma pels will be temporal filtered in the associated capture window. If CTF is clear and TF is set when GoI/GoI2 is set by software, then luma pels will be temporal filtered, but chroma pels will not be temporal filtered in the associated capture window. If TF is clear, then chroma pels and luma pels will not be temporal filtered.

FilE **15**

If set when GoI is set by software, then the associated capture window will be processed by the horizontal video input filter. If clear, then the associated capture window will bypass the horizontal video input filter and will not be clamped. If set when GoO is set by software, then the associated output window will be processed by the horizontal and vertical video output filter. If clear, then the associated output window will bypass the horizontal video output filter and will not be clamped at this stage. If set when GoR is set by software, then the associated rescale operation will be processed by the horizontal rescale filter. If clear, then the associated rescale operation will bypass the horizontal rescale filter and will not be clamped. Programming note: bypassing the input or output filter makes more cycles available for rescaling and vice versa. Bypassing the input filter can be used to capture unscaled data during the vertical blank interval

0 **14**

This bit must be programmed to 0.

RES **Reserved** **13**

VideoRGB **12**

If set, then video output windows in the next output field/frame are treated as a second RGB graphics layer, with a pixel format specified by videoPixFMT. The ordering between the video layer and the OSD layer is controlled by the vover bit. When VideoRGB is set, RLEalpha blending, PIP and echo windows are not allowed. The video RGB layer is affected by the

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Top/Bottom display bits in the Overlay Control register. VideoRGB should not change while an output window (video, OSD or PIP) in the current output field/frame is active.

VideoPixFmt **[11:9]**

When GoO is set in the video control register and VideoRGB is set, specifies the pixel format of the associated video window as shown below. When the pixel format is not YCrCb, then video data is interpreted as a graphics layer which underlays or overlays the OSD layer according to the Vover bit. When video is used as a graphics layer, flicker filtering is disabled. The pixel format values are shown below.

Video PixFmt Value	Pixel Format
0	4-bit Indexed Color
1	8-bit Indexed Color
2	16-bit 555, Little Endian
3	16-bit 555, Big Endian
4	16-bit 565, Little Endian
5	16-bit 565, Big Endian
6	32-bit, Little Endian
7	32-bit, Big Endian

PYfirst **8**

If set and PV is set when GoR is set by software, then the associated rescale window will output data in interleaved YCbYCr format, otherwise the rescale window will output data in CbYCrY format. If set and PV is set when Go0 is set by software, then the associated output window will read output data in interleaved YCbYCr format otherwise the output window will read output data in CbYCr format. If set and PV is set when GoPIP is set by software, then the associated PIP window will read output data in interleaved YCbYCr format otherwise the output window will read output data in CbYCr format.

Important: Packed video is not implemented in the first version of DoMiNo, and this bit is ignored.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Hfpchr **7**

If set when GoI, GoI2, GoR or GoO, the phase of chroma pel filtering of the corresponding video capture, rescale or output window is offset by half a pel relative to luma filtering as required by MPEG-1 and H.261.

GoR **6**

Set by microcode to begin horizontally rescaling a row (multiple scan lines) of luma and/or chroma data supplied by a row of scale(i) dsp instructions. The GoR bit remains set until data in the rescale filter result range has been rescaled, to allow software polling of completion. Clearing this bit is ignored.

VidOutInt **5**

The completion of a video output window will generate a video output completion interrupt if VidOutInt is set. This bit should not be changed while an output transfer is active or pending.

Echo **4**

This bit should always be 0. Set by microcode to queue up a video input stream 1capture window that will be inserted into the video output (without storing into SDRAM) using the most recently stored values in the input data vertical range register. Up to two video capture commands can be outstanding at any time. The Echo bit remains set while there is at least one video input capture outstanding to allow software polling of completion. The Echo bit must be set before the start of the capture window, or a partial window may be echoed. Typically it will be set from the video transfer interrupt handler. Clearing this bit is ignored (use the stop bit instead). Inside the capture window, data (including SAV sequences) from the stream 1 video input pins with a small delay replaces the video output or background color unless there is an active PIP window or RLEalpha is set which causes blending to occur in the RLE mixer. OSync should be set and the video output clock, output total samples per line and lines per picture (progressive) or field (interlaced) must match the incoming video. The input source should be programmed to shift its active video earlier by the input to output delay. Note that Echo and GoI can both be set time, to echo and capture input data to SDRAM at the same. Echo windows can not be

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

used in the same output field or frame as a videoRGB window, and can not be used when SSAV is set.

GoPIP **3**

Set by microcode to queue up a video PIP output window using the most recently stored values in the PIP vertical, horizontal range, luma and chroma address and row/column offset registers. Up to two PIP transfers can be outstanding at any time. The GoPIP bit must be set before the start of the PIP window, or a partial window may be output. The GoPIP bit remains set while there is at least one PIP window outstanding to allow software polling of completion. Clearing this bit is ignored. PIP windows can not be used in the same output field or frame as a videoRGB window.

VidFInt **2**

If set, a video field interrupt will be generated at the beginning of the output vsync. The video field interrupt will be aligned with the video input beginning of field if OSync is set, once the video input timing has been stable for two pairs of odd/even fields or frames.

GoI **1**

Set by microcode to queue up a video input stream 1 capture window using the most recently stored values in the input stream 1 data vertical, horizontal and result range, destination luma and chroma address and row/column offset registers. Up to two video capture commands can be outstanding at any time. The GoI bit must be set before the start of the capture window, or a partial window may be captured. Typically it will be set from the video transfer interrupt handler. The GoI bit remains set while there is at least one video input stream 1 capture outstanding to allow software polling of completion. Clearing this bit is ignored.

GoO **0**

Set by microcode to queue up a video output transfer using the most recently stored values in FI, VCD, VCI, TF and the output data vertical range, output luma address and output chroma address registers. Up to two video output transfers can be outstanding at any time. The GoO bit must be set before the start of the output window. Typically it will be set from the video field or video transfer

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

interrupt handler. The GoO bit remains set while there is at least one video output transfer outstanding to allow software polling of completion. Clearing this bit is ignored. Output data, SAV and EAV are generated even if echo is not active and the GoO bit is not set.

Video Status Register

Cbus Address: 0xC40004 Read/write 0x200000

31								23	22	21	20	19	18	17	16
Reserved									ZYO	OT	YFirst	OSync	0	Oclkr	ISpol
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stop	P-In	P-Out	0	Out16	0	Early2	RLEerr	VSP	Res	SSAV	O-Odd	Res	IOdd	Early	BOvr

ZYO **22**

If set (zero Y offset), Y and RGB are assumed to have the same offset in YUV->RGB and RGB->YUV color space conversions. If clear, a Y value of 16 corresponds to an RGB value of zero. Changes to this bit will take effect at the beginning of the next output field.

OT **21**

If set, the video channel output pins (VO_D[15:0]) are 3-stated, independent of any active output transfer. If clear, these pins are driven by DoMiNo. 3-tating allows other video sources to share the video output bus. At reset, this bit is set to avoid contention on the output pins. The reset sequence can clear this bit if DMN-8600 is the only driver. This bit should not be changed while an output transfer is active or pending.

YFirst **20**

If set, the video capture or output interleave order is (Y, CB, Y, CR); otherwise, it is (CB, Y, CR, Y). This bit should not be changed while an input or output transfer is active or pending. Changes to this bit take effect on the output at the beginning of the next output field.

OSync **19**

If set when ProgIn is clear, video output of the last byte of an EAV sequence where the F-bit toggles LOW in (the

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

fourth byte of the first line in an odd field or noninterlaced frame) is synchronized to the corresponding byte in video input stream 1 odd fields. If set when ProgIn is set, video output of the last byte of an EAV sequence where the V-bit toggles LOW (the fourth byte of the first active line in an odd field or noninterlaced frame) is synchronized to the corresponding byte in video input stream 1 frames. When OSync is set the video output clock is set to be the same as the video stream 1 input clock. If Osync is clear, then video output timing is independent of the video input.Through the use of the VIE pin, the output frame rate can be locked to a multiple of the input frame rate, e.g. two field interleaved streams at 54 MHz can be locked to a 27 MHz input stream. In all cases, synchronization occurs at the fourth byte of each input odd field or frame. The total number of clocks for all output fields between synchronization points must align the beginning of an odd output field with each odd input field. When Osync is set, the number of clocks between input odd fields be held constant for two odd/even input pairs ahead of valid output through the end of output to ensure stable output sync. Also, VOE must be asserted on all clocks that VIE is asserted. Osync should not be changed while an input or output transfer is active. Osync should not be set if VSP is set. At reset this bit is set to zero.

Note: The first byte of the EAV sequences will not be aligned if VIE and VOE are different. In particular, in the case of two output streams where VIE is toggling every other clock and VOE is not toggling, first byte of the output EAV sequence will start three 54 MHz clocks later than the input sequence. The number of clocks by which the start of output EAV is shifted relative to input is the number of clocks that VIE is asserted between the first and last byte of the input EAV sequence, minus the number of clocks that VOE is asserted.

0 **18**

This bit must be programmed to 0.

Oclkr **17**

If set, video data output pins are clocked on the rising edge of the video output clock; otherwise, they are

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

clocked on the falling edge. Typically, Oclkr is zero for slow external devices that latch on the rising edge but need long hold time, and one for fast external devices that latch on the rising edge but need zero hold time.

ISpol **16**

If set, the Input VSYNC[0] video input sync signal is active high; otherwise, it is active low. This bit should not be changed while an input transfer is active or pending.

Stop **15**

Setting this bit halts video input, OSD, RLE, and output processing for diagnostic purposes. All queue entries are flushed, and Go bits are cleared without generating completion interrupts. This bit is self-resetting, and it should be polled by software to be determine when all pending operations have been terminated. Note that rescale instructions pending in the DSP instruction queue will not complete after GoR is cleared by Stop. Software can do a syncdsp before setting Stop, or it can set GoR again later or reset the video SPARC. This bit should always be set to 0 in DoMiNo.

P-In **14**

If set (progressive input), the beginning of a video input frame of both input streams is defined to be the EAV sequence where the V bit toggles from one to zero (end of vertical blanking). If clear (field input), the beginning of a video input field is defined to be the EAV sequence where the F bit toggles. This bit should not be changed while an input transfer is active or pending.

P-Out **13**

If set (progressive output), the F bit is always zero in SAV and EAV sequences generated in the video output. Also, the odd VSync horizontal offset is used for all frames. If clear (field input), the F bit toggles at the first EAV in each field of the video output. This bit should not be changed while an output transfer is active or pending.

0 **12**

This bit must be programmed to 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Out16 **11**

If set and RGB is not set, DMN-8600 video output is 16 bits wide (SMPTE 260M). If Out16 and RGB are clear, video output is 8 bits wide (656). This bit should not be changed while an input or output transfer is active or pending.

0 **10**

This bit must be programmed to 0.

Early2 **9**

Set if an EAV code where the F bit toggles (ProgIn clear) or the V bit toggles to zero (ProgIn set) occurs between the top and bottom lines of the window. When the early bit is set, the number of bytes stored in SDRAM is undefined, but less than or equal to the normal transfer size. This is an indication that video sync has been lost on the video input stream 2 channel. This bit remains set until cleared by software.

RLEerr **8**

Set by hardware when an error is detected in the RLE pixel data of an RLE window, including the start of pixel data for a scan line not aligned to a byte boundary or the pixel data is longer than the RLE window width. This bit remains set until cleared by software.

VSP **7**

If set, input video vertical sync (first line in field or frame) of each input stream is determined by assertion of the VI_VSYNC[0] pin (and the EAV F bit is ignored). If clear, input vertical sync is determined from the EAV F bit toggle (interlaced) or V bit toggle to zero (progressive). The Input VSYNC pin (VI_VSYNC[0]) should be used when non-standard analog input is being captured that does not have consistent even/odd field sequences and timing. OSync should not be set if VSP is set.

SSAV **5**

If set or if BTC is set, the left and right edges of video capture windows are defined relative to the SAV (Start Active Video) sequence rather than the EAV sequence of the video input stream. Echo cannot be used when SSAV is set. If clear and BTC is clear, the left and right edges of the video capture window are defined relative to the

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

EAV sequence. This bit is reset to zero by a chip reset. This bit is typically set for non-line-locked video decoders that insert a variable number of clocks between EAV and SAV sequences. SSAV should not be set when BTC is set. If this bit is set, zero pels that occur between SAV and EAV sequences (active pels that are zero) are treated as invalid pels, that is, they are treated as if VIE were deasserted during that clock.

Zero pel skipping has been dropped from the DMN-8600 A step.

O-Odd **4**

Set by the hardware if the current output field (interlaced) is odd. This information comes from the internally generated output sync.

IOdd1 **2**

Set by the hardware if the current input field (interlaced) of input stream 1 is odd. This information comes from the most recent EAV/SAV F bit value on the video input 1.

Early **1**

Set if an EAV code where the F bit toggles (ProgIn clear) or the V bit toggles to zero (ProgIn set) occurs between the top and bottom lines of the window. This bit remains set until cleared by software.

BOvr **0**

Set by the hardware when excessive SDRAM latency causes an overrun to occur in the capture, previous input, OSD, RLEalpha, PIP, or previous/current field video output buffer. (In other words, if there is active video in the vertical capture range and the capture buffers are full or the previous input, OSD, RLEalpha, PIP or previous/current video output buffers are empty when data is needed.) This bit remains set until cleared by software. This bit is set when the input/output rate is out of range or when an extremely large DSP load or store instruction is executed (more than 1 Kbyte transferred).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Video Overlay Control Register

Cbus Address: 0xC4000C Read/write 0x8

31 29	28	27	26	25 24	23 20	19	18 17	16
Reserved	NoPx	SqPix	Pre-M	Alpha Mode	Reserved	Vover	VAcolor	RLEA

15	14	13 12	11 10	9 8	7	6	5	4	3	2	1	0
RLEInt	GoRLE	OSD PixFmt	ScaleH	ScaleV	Bott	Top	FlickE	0	OSDI	OQF	GOSD

NoPx **28**

If set when GoRLE is set, the associated sub-picture RLE window has no pixel control data, that is, the default colors and alpha values are used for the whole window.

SqPix **27**

When GoOSD is set, specifies whether OSD pels in the associated window are rescaled horizontally by a ratio specified by the OSD Phase Increment register. If set, OSD pels are rescaled. If clear, OSD pels are not rescaled. This bit is set to convert square aspect ratio pixels to 720H resolution output.

Pre-M **26**

If set when GoOSD is set, specifies that software has pre-multiplied the OSD RGB values in the associated OSD window by the alpha value. If not set when GoOSD is set, specifies that OSD RGB values should be multiplied by alpha before blending with the video layer.

Alpha Mode **[25:24]**

When GoOSD is set, specifies the alpha mode of the associated OSD window, as shown below. The alpha mode determines how alpha values are determined for OSD pixels in the window.

Alpha Mode Value	Alpha Mode
0	Normal Alpha
1	Inverted Alpha
2	Constant Alpha
3	Transparent Color

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Vover **19**

If set when GoOSD is set, OSD pixels in the associated OSD window is rendered under the video output/PIP data instead of over.

VAColor **[18:17]**

When RLEalpha is set, specifies whether two-color or two-video blending is used for c = 1 (VAcolor[1]) and c = 0 in (VAcolor[0]) variable alpha stream commands.

RLEA **16**

If set on the first active video line, RLE alpha stream blending is performed over the active (nonblanked) region of the current output field or frame. Data from the PIP window if active; otherwise, from the echo window if active; otherwise, the background color is blended with data from the video output window if active; otherwise, the background color. If RLEalpha is clear on the first active video line, data from the PIP window; otherwise, data from the video input stream 1 echo window overlays the video output window if active; otherwise, the background color. RLEalpha cannot set in the same output field or frame as a VideoRGB.

RLEAlpha is not implemented in the first version of DoMiNo. This bit should always be zero.

RLEInt **15**

The completion of an RLE window generates an RLE completion interrupt, if RLEInt is set. This bit should not be changed while an RLE transfer is active or pending.

GoRLE **14**

Set by microcode to queue up a sub-picture RLE window using the most recently stored values in the RLE Default Pixel Color/Alpha, RLE Vertical Range, RLE Horizontal Range, RLE Pixel Data Address, RLE Output Address, RLE Pitch, and RLE Pixel Control Address registers. Only one RLE window can be outstanding at any one time. The GoRLE bit remains set while the RLE transfer is outstanding to allow software polling for completion. Clearing this bit is ignored (use the Stop bit instead).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

OSDPixFmt **[13:11]**

When GoOSD is set, specifies the pixel format for OSD data of the associated OSD window, as shown below.

OSD PixFmt Value	Pixel Format
0	4-bit Indexed Color
1	8-bit Indexed Color
2	16-bit 555, Little Endian
3	16-bit 555, Big Endian
4	16-bit 565, Little Endian
5	16-bit 565, Big Endian
6	32-bit, Little Endian
7	32-bit, Big Endian

ScaleH **[10:9]**

Specifies the horizontal scale factor for OSD data, as shown below. This scaling is applied before the square pixel scaling selected by the SqPixel bit. Square pixel scaling is limited to pixel output rates less than or equal to one fifth of the DMN-8600 internal clock (480P at 148.5 MHz). OSD data is scaled by duplicating pels horizontally by the scale factor amount.

ScaleH, V Value	Scale Factor
0	1x
1	2x
2	4x

ScaleV **[8:7]**

Specifies the vertical scale factor for OSD data. OSD data is scaled by duplicating pels vertically by the scale factor amount.

Programming Note: 2x vertical scaling for interlaced output is best achieved by displaying both fields of OSD data in each field–that is, both the top set and the bottom set.

Bott **6**

If set, bottom field data is displayed in OSD windows and video out windows. If set at the same time as Top, data from each field is displayed on alternate lines, with top

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

field data displayed on the first line of the window. This bit should not be changed while an OSD transfer is active or pending. This bit also applies to video data, if RGB graphics data is being displayed in the video layer. This bit or the Top bit should be set if GoOSD or VideoRGB is set.

Top **5**

If set, top field data is displayed in OSD windows. If set at the same time as Bottom, data from each field is displayed on alternate lines, with top field data displayed on the first line of the window. This bit should not be changed while an OSD transfer is active or pending. This bit also applies to video data, if RGB graphics data is being displayed in the video layer. This bit or Top bit should be set if GoOSD or VideoRGB is set.

FlickE **4**

If set when the GoOSD bit is set by microcode, the associated OSD window will apply flicker filtering to primary and secondary OSD data; otherwise, flicker filtering will not be applied, and only primary OSD data is displayed for that window. This bit should not be changed while an OSD transfer is active or pending. This bit is ignored when video is used as a graphics layer, or if both Top and Bottom are set.

0 **3**

This bit must be programmed to 0.

OSDI **2**

The completion of an OSD window generates an OSD completion interrupt, if OSDInt is set. This bit should not be changed while an output transfer is active or pending.

OQF **1**

A read-only bit that is set when there are two outstanding DMA transfers in the OSD output command queue. A transfer is loaded into the queue each time the GoOSD bit is set by software. A command is removed from the queue each time the transfer completes. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds), and interrupts are used for longer transfers.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

GOSD **0**

Set by microcode to queue up an OSD window using the most recently stored values in ScaleH, ScaleV, PixFmt and the OSD Vertical Range, OSD Horizontal Range, and OSD Data Address registers. Up to two OSD windows can be outstanding at any time. The GoOSD bit must be set before the start of the OSD window. Typically, it is set from the video field or OSD interrupt handler. The GoOSD bit remains set while there is at least one video output transfer outstanding to allow software polling for completion. Clearing this bit is ignored (use the Stop bit instead).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 12
Audio Interface

This chapter describes the audio interface and contains the following sections:

For the 8600 audio interface, four channels of audio input and eight channels of audio output are supported with common input and output clocks:

- AI_D[1:0] – Incoming serial stream audio data
- AI_SCLK – Serial clock for stream audio input
- AO_SCLK – Serial clock for audio output
- AO_D[3:0] – Outgoing serial stream audio data

The 8600 uses incoming and outgoing frame sync signals and internally generated master clock signals:

- AI_FSYNC – Incoming stream frame sync
- A0_FSYNC – Audio output frame sync
- AI_MCLKO – Internal audio input master clock
- AO_MCLKO – Internal audio output master clock

The 8600 also has an IEC958 data output signal.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

12.1 Data, Serial Clocks and FSYNC

Four channels of serial audio data are clocked in by the two AI_D pins, and up to eight channels of serial audio data are clocked out of the four AO_D pins by the clock on the AO_SCLK pin. With two samples/frame, channel 2n and 2n + 1 use AudioIn/Out[n]. With one sample/frame, channel n uses AudioIn/Out[n].

AI_FSYNC determines the start or end of the next stream input sample or frame as specified by the stream frame format and the ISync field in the Audio Input Control register. If ISync is zero or two, it is an input; otherwise, it is an output.

When FSYNC is an output, frame sync is generated continuously independent of the GoI bit. FSYNC determines the start or end of the next output sample or frame as specified by the Audio Output Control register.

12.1.1 MCLK

The audio input master clock is internally generated according to Ain Master in the clock control register. The internally generated audio input master clock is output on the AI_MCLKO pin.

The audio output master clock is internally generated according to Aout Master in the clock control register. The internally generated audio output master clock is output on the AO_MCLKO pin.

12.1.2 IEC-958 Encoder

The IEC958 pin outputs IEC-958 format serial data with timing derived from the audio output master clock. The IEC958 output sample time matches the audio output sample time times the R958 field specified divisor. It is typically used to transmit compressed audio data.

The interface carries either PCM audio samples (mono or stereo) or compressed audio bitstreams (AC3 or MPEG). The sample rate of audio samples carried by this interface can be 32, 44.1, 48 or 96 kHz. The width of audio samples can be up to 24 bits.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

When carrying PCM audio samples, IEC-958 groups data into a series of blocks. Each block contains 192 frames. Each frame contains 2 subframes and each subframe contains a data payload of one audio sample. When carrying mono signals, the same audio sample can be used for both subframes. The start of a block is designated by a special subframe sync preamble.

12.2 Audio Input Control Register

The Audio Input Control Register is used to control a variety of features, as described below. A read-modify-write cycle should be used when accessing this register.

Audio Input Control Register

Cbus Address: 0x50004

31					26	25	24	23	22	21	20		18	17	16
Reserved						Icklr	IBclk			Stop	Res			FrForm	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrForm	0		ISync		ITim	Res	IS32	LE	ChCnt		AudInt	Res	IQF	0	GoI

Iclkr **25**

Specifies the clock polarity for audio input stream 1 signals. Input signals are sampled on the opposite edge that output signals are driven. If set, audio input stream 1 serial data and frame sync input (AI_D[3:0], AI_FSYNC as an input) are clocked on the rising edge of the AI_SCLK input; otherwise, they are clocked on the falling edge. If set, audio input stream 1 frame sync output (AI_FSYNC as an output) is clocked on the falling edge of the AI_SCLK input; otherwise, it is clocked on the rising edge. Programming note: This field should be

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

programmed at least one frame time before setting the GoI bit.

Iclkr Value	Driving Edge	Sampling Edge
0	Rising	Falling
1	Falling	Rising

IBclk **[24:22]**

Specifies the divisor ratio of the internally generated audio input bit clock from the audio input master clock, as shown below. When using an externally generated audio input bit clock, this field is ignored.

Value	Audio Input Master Clocks/ Internal Audio Input Bit Clock
0	2
1	4
2	6
3	8
4	12
5	16

An IBclk value of one and ITim clear or a Bclk of zero and ITim set corresponds to a 256xFs master clock. An IBclk of zero and ITim clear corresponds to a 128xFs master clock (as used with ADCs which perform clock doubling for > 48Khz sample rates). An IBclk value of two and ITim clear corresponds to a 384xFs master clock.
Programming note: This field should be programmed at least one frame time before setting the GoI bit.

Stop **21**

Setting this bit halts audio input and output processing for diagnostic purposes. All queue entries from both input and both output streams are flushed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

FrForm **[17:15]**

Specifies the external audio frame format for input stream 1 as shown below.

FrForm Value	Samples/ Frame	Sync Type	Frame Sync Transition
0	1	Pulse	First bit of each sample
1	2 + 2 Status	Pulse	First bit of first sample in frame
3	2	Left High/ Right Low	First bit of each sample
4	2	Left Low/ Right High	One clock before first bit of each sample
5	2	Left High/ Right Low	One clock after last bit of each 16-bit external sample
6	2	Left High/ Right Low	One clock after last bit of each 18-bit external sample
7	2	Left High/ Right Low	One clock after last bit of each 20-bit external sample

Pulse frame sync is a LOW-to-HIGH transition on AI_FSYNC during the first bit or the clock before the first bit of the first sample in each audio frame. The HIGH-to-LOW transition is ignored. Frform 1 requires input samples to be 16-clocks each, since there is no other timing reference to delineate sample boundaries. Frform 1 transfers four 16-bit word per frame (additional bits are discarded). The first word is the left sample, the second word is captured in the status registers, the third word is the right channel sample, and the fourth word is discarded.

Programming note: This field should be programmed at least one frame time before setting the GoI bit.

0 **14:13**

These bits must be programmed to 0.

ISync **12:11**

If Isync is set to zero, the input stream audio bit clock (AI_SCLK) and frame sync (AI_FSYNC) pins are inputs for externally generated frame timing. If Isync is set to one, AI_SCLK and AI_FSYNC are outputs and frame timing is derived from the input master clock using the divisor specified by IBclock. Internally generated input

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

timing is signaled by the AI_FSYNC pin continuously if Isync is set to one. If Isync is set to two, the stream input frame timing is the same as output frame timing. AI_SCLK and AI_FSYNC are inputs. In this case, FrForm, Iclkr and Itim are ignored, and the input stream uses the same frame format and timing as the output stream. Isync is cleared after reset to prevent contention on the AI_SCLK and AI_FSYNC pins. This field should be set to two if the input stream is using an AC-97 codec. Note: This field should be programmed at least one frame time before setting the GoI bit.

ISync Value	AI_SCLK, AI_FSYNC	SCLK/FSYNC Used	MCLK Source of SCLK Used
0	Input	AI_SCLK, AI_FSYNC	--
1	Output	AI_SCLK, AI_FSYNC	AI_MCLK
2	Input (unused)	AO_SCLK, AO_FSYNC (i.e., stream 1 output)	AO_MCLK

ITim **10**

If set and Frform is not one or two, the internally generated timing of each input sample is 64-bit clocks (as recommended for the CS4227 for optimum performance). If clear and Frform is not one, then the internally generated timing of each input sample is 32 clocks. If Frform is one then the internally generated timing of each input sample is 16 clocks. Programming note: This field should be programmed at least one frame time before setting the GoI bit.

IS32 **8**

If set, input stream 1 samples in memory occupy 32 bits, otherwise they occupy 16 bits. The number of input clocks/sample is variable between 16 and 64, depending on AI_FSYNC.

LE **7**

If set, input and output samples for audio stream is stored in little endian format in memory (LSB at lowest address); otherwise, they are stored in big endian format (MSB at lowest address).

ChCnt **[6:5]**

The number of stream audio channels to capture or playback is 2^{ChCnt}. Audio data is stored as sets of

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

interleaved samples from each channel. Within each set, channel zero is stored at the lowest address and channel 2ChCnt - 1 at the highest address. With frame formats having two samples per frame, even channels correspond to the left sample in each frame and odd channels to the right sample in each frame. With FrForm equal to one, input status is captured from all 8 audio channels, independent of the ChCnt value.

AudInt **4**

If set, an audio input stream DMA transfer completion interrupt will be generated when an audio input DMA transfer has been completed. Completion interrupts are queued by the hardware, resulting in a separate interrupt for each transfer, even if a second transfer completes before the first interrupt is taken. When the interrupt is taken, the count of outstanding interrupts is decremented by one.

Res **Reserved** **3**

IQF **2**

A read-only bit which is set when there are two outstanding DMA transfers in the audio input stream 1 command queue. A transfer is loaded into the queue each time the GoI bit is set by software. A command is removed from the queue each time the transfer completes. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds) and interrupts are used for longer transfers.

0 **1**

This bit must be programmed to 0.

GoI **0**

Set by microcode to queue up an audio input DMA transfer using the most recently stored values in the Audio Input 1 DMA address and length registers. The GOI bit remains set while there is at least one audio input 1 DMA transfer outstanding. When all audio input outstanding transfers are done, the DMA will clear the bit,

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

to allow software polling of completion. Clearing this bit is ignored (use the stop bit instead).

Set/Clear	Description
Set by software	Queue up a new transfer (max of two outstanding).
Cleared by hardware	No outstanding transfers. Last of transfer data has left AIO.

12.3 Audio Output Control Register

The Audio Output Control Register is used to control a variety of features, as described below. A read-modify-write cycle must be used when accessing this register.

Audio Output Control Register

Memory Space Address: 0x50000.

31								23	22	21	20	19	18	17	16
Reserved								V958	R958		Res		E958	OBclk	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBclk	I9S32	I958	FrForm			OTim	OS32	OClkr	ChCnt		AudInt	Res	OQF	0	GoO

V958 **23**

Value of V bit during IEC958 transfer.

R958 **[22:21]**

Specifies the rate of I958 sample output relative to audio output stream 1 samples as shown in the table below. This allows the I958 output to run at a subsampled rate (typically 44.1 kHz or 48 kHz) of the main audio output as required in DVD-Audio. Programming note: This field

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

should be programmed at least one frame time before setting the Go bit.

R958 Value	Audio Output Stream 1 Samples /I958 Samples
0	1
1	2
2	4
3	Reserved

E958 **18**

If set, the I958 output pin is a copy of the data on the SD_DATA[4] pin. If clear, then it is the output of the internally generated I958 data stream. This bit allows an externally generated I958 data stream from a CD audio front end chip to be muxed with the internally generated stream.

OBclk **[17:15]**

Specifies the divisor ratio of the audio output bit clock (AO_SCLK) from the audio output master clock, as shown in the table below:

Value	Audio Output Master Clocks/ Audio Output Bit Clock
0	2
1	4
2	6
3	8
4	12
5	16

I9S32 **[14]**

If set, IEC958 output samples in memory occupy 32 bits; otherwise, they occupy 16 bits. If 32 bit samples are selected, then only the upper 24 bits of the sample are transmitted in each output subframe. If 16 bit samples are selected, then they occupy the upper 16 bits of each

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

output subframe. I9S32 also determines the source of the U and C bits as shown below.

I9S32 cValue	Bits in Memory	How Bits Fit into 24-bit Output	U Bit	C Bit
0	16	Fits into upper 16 bits of output, the lower 8 bits being zero filled	0	Status
1	32	Only upper 24 bits of data are used	mem data LSB bit 6	mem data LSB bit 5

I958 **13**

Set by microcode to indicate that IEC958 output data should be transferred in addition to audio output stream 1 data when GoO is set. The IEC958 DMA transfer uses the most recently stored value in the IEC958 Output DMA address register. The length of the I958 transfer is the output length divided by the output sample memory size in bytes divided by the number of active channels specified by chCnt times two times the IEC958 sample size divided by the R958 sample rate divisor. When I958 is not set in an output transfer, zero data is transferred on the I958 output pin, with the usual IEC958 framing. IEC958 timing is generated from the master audio clock, and is adjusted so that one/one half or one fourth of an IEC958 sample is transmitted in each audio output sample time as determined by the OTIM bits and R958 field. The length of audio output 1 transfers with I958 set must correspond to an even number of IEC958 samples.

IEC958 length = audio output length x $2^{I9S32}/2^{OS32}$ x 2/audio_channels_per_2_samples x $1/2^{R958}$.

I958 Value	Description
0	Disabled: IEC958 will not respond to GoO; Stream audio output doesn't sync with IEC958 block 0.
1	Enabled: IEC958 responds to stream 1 GoO; Stream audio output syncs with IEC958 block 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

FrForm **12:10**

Specifies the audio stream frame output format as shown in the table below:

FrForm Value	Samples/ Frame	Sync Type	Frame Sync Transition
0	1	Pulse	First bit of each sample
1	2 + 2 Status	Pulse	First bit of first sample in frame
3	2	Left High/ Right Low	First bit of each sample
4	2	Left Low/ Right High	One clock before first bit of each sample
5	2	Left High/ Right Low	One clock after last bit of each 16-bit external sample
6	2	Left High/ Right Low	One clock after last bit of each 18-bit external sample
7	2	Left High/ Right Low	One clock after last bit of each 20-bit external sample

The number of samples/frame specifies the number of audio samples transmitted per frame sync cycle on each audio output pin. When one sample/frame is used, only four channels may be captured or transmitted. Frame sync is output on AO_FSYNC for audio output. Note: This field should be programmed at least one frame time before setting the Go bit.

Pulse frame sync is a low to high transition on out.

Fsync during the first bit or the clock before the first bit of the first sample in each audio frame. Pulse frame sync is high for one clock, and then goes low until the next frame.

Frform requires output samples to be 16-clocks each, since there is no other timing reference to delineate sample boundaries. Frform transfers four 16-bit word per frame (additional bits are discarded on input, and zero on output). For output frames, the first word is the left (lowest channel number) sample, the second word is the most significant 16 bits of the status out register, the third word is the right output sample and the fourth word is the least significant 16-bits of the status out register.

Left High/Right Low sync is high from during left (lowest number) channel sample bits and low during right

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

channel bits. Left Low/Right High is inverted. The transition of frame sync from low to high and high to low defines the beginning or end of each sample as defined in the frame sync transition column of the table below.

FrForm	S32	OTim	Bits in Memory	Bits in Output	How it Fits in Output
1	0	-	16	16	Perfect fit
1	1	-	32	16	Only upper 16 bits of data are used
2					Not implemented in DoMiNo
0, 3, 4	0	0	16	32	Fits into first 16 bits of output, the remaining 16 bits zero filled
0, 3, 4	0	1	16	64	Fits into first 16 bits of output, the remaining 48 bits zero filled
0, 3, 4	1	0	32	32	Perfect fit
0, 3, 4	1	1	32	64	Fits into first 32 bits of output, the remaining 32 bits zero filled
5-7	0	0	16	32	Left justified zero padding, right justified data
5-7	0	1	16	64	Left justified zero padding, right justified data
5-7	1	0	32	32	Left justified zero padding, right justified data (upper memdata bits used)
5-7	1	1	32	64	Left justified zero padding, right justified data (upper memdata bits used)

OTim **9**

If set and Frform is not one or two, the timing of each stream output sample is 64 bit clocks (as recommended

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

for the CS4227 for optimum performance). If clear and Frform is not one or two, then the timing of each output sample is 32 clocks. If Frform is one then the timing of each sample is 16 clocks. Output timing is signaled by the AO_FSYNC pin if there is an active output transfer (GoO is set). When OTim is changed to a new value, then FrForm should also be changed to a new value. This may necessitate writing FrForm twice, if only OTim needs to be changed. Programming note: This field should be programmed at least one frame time before setting the Go bit. See 2nd table in FrForm.

OS32 **8**

If set, output stream samples in memory occupy 32 bits; otherwise, they occupy 16 bits. See second table in FrForm.

OClkr **7**

If set, audio output stream serial data and frame sync (AO_D[3:0] and AO_FSYNC) are clocked on the falling edge of the AO_SCLK input, otherwise they are clocked on the rising edge. Programming note: This field should be programmed at least one frame time before setting the Go bit.

Oclkr Value	Driving Edge	Sampling Edge
0	Rising	Falling
1	Falling	Rising

ChCnt **[6:5]**

The number of audio output stream channels is 2^{ChCnt}. Audio data is stored as sets of interleaved samples from each channel. Within each set, channel zero is stored at the lowest address and channel $2^{ChCnt} - 1$ at the highest address. With frame formats having two samples per frame, even channels correspond to the left sample in each frame and odd channels to the right sample in each frame. When outputting less than 8 audio channels, zero data is transmitted on the unused output channels.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ChCnt Value	Total Channels	Description
0	1	1 channel
1	2	1 Left + 1 Right
2	4	2 Left + 2 Right
3	8	4 Left + 4 Right

AudInt **4**

If set, an audio output stream DMA transfer completion interrupt will be generated when an audio output DMA transfer has been completed. Completion interrupts are queued by the hardware, resulting in a separate interrupt for each transfer, even if a second transfer completes before the first interrupt is taken. When the interrupt is taken, the count of outstanding interrupts is decremented by one.

OQF **2**

A read-only bit which is set when there are two outstanding DMA transfers in the audio output stream command queue. A transfer is loaded into the queue each time the GoO bit is set by software. A command is removed from the queue each time the transfer completes. This bit can be used by software to poll for when to load the next command. Polling is typically used for short transfers (<10 microseconds) and interrupts are used for longer transfers.

0 **1**

This bit must be programmed to 0.

GoO **0**

Set by microcode to queue up an audio output stream DMA transfer using the most recently stored values in the Audio Output DMA address and length registers. The goO bit remains set while there is at least one audio output DMA transfer outstanding. When all audio output outstanding transfers are done, the DMA will clear the bit, to allow software polling of completion. If the stream output DMA queue is empty when a transfer is queued up, then the first frame of output on the audio pins will be synchronized to the start of an IEC958 block boundary to allow consistent IEC958 user status and audio data. Programming note: When changing IEC958 user status,

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

software should allow the output queue to drain, change the IEC958 status registers then queue up the next output transfer. GoO is clear when all transfers are completed. Clearing this bit is ignored (use the stop bit instead). Output timing is signaled by the AO_FSYNC pin independent of the state of the GoO bit.

Set/Clear	Description
Set by software	Queue up a new transfer (max of two outstanding).
Cleared by hardware	No outstanding transfers. Last of transfer data has left AIO.

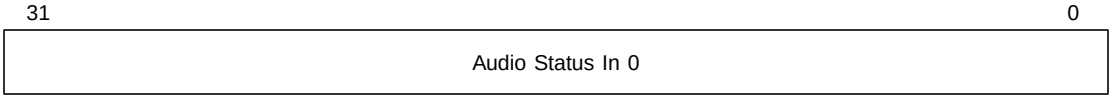
12.4 Audio Status Registers

The DMN-8600 processor can also input/output audio status information. Usually the status input/output provides status information to an audio codec connected to the processor or delivers status words from an audio codec to the processor. For the DMN-8600, three registers are used for status input and three for status output, as shown below.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

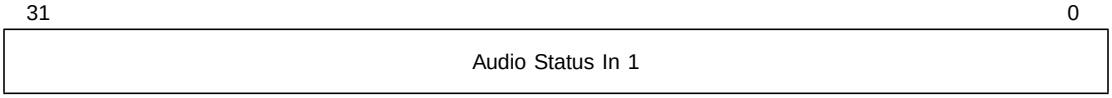
Audio Status In 0 Register

Memory Space Address: 0x050008



Audio Status In 1 Register

Memory Space Address: 0x05000C

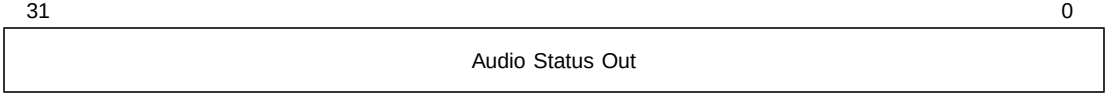


Audio Status In 0 and 1 **[31:0]**

Audio Status In 0 and 1. These registers contain the data clocked in status words in FrForm one. Frform one clocks in 16 bits of status data from each audio frame. Status data from AI_D[0] is loaded into the most significant 16-bits of the Audio Status In 0 register. Status data from AI_D[1] is loaded into the least significant 16-bits of the Audio. Data is captured only when the associated input DMA is capturing. These registers are typically used to capture codec status.

Audio Status Out Register

Memory Space Address: 0x050014



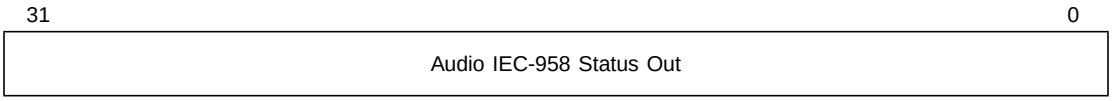
Audio Status Out **[31:0]**

Audio Status Out. Contains the data clocked in control words in Frform one. Frform one clocks out 32 bits of data from AudioStatusOut[1] to the AO_D [3:0] pins during each audio frame. Data is clocked even when the output DMA is not active. This register is typically used to control codec status.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

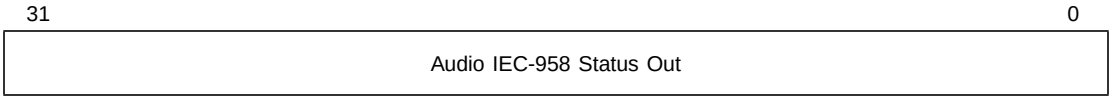
Audio IEC-958 Status Out1 Register

Memory Space Address: 0x05001C



Audio IEC-958 Status Out2 Register

Memory Space Address: 0x050020

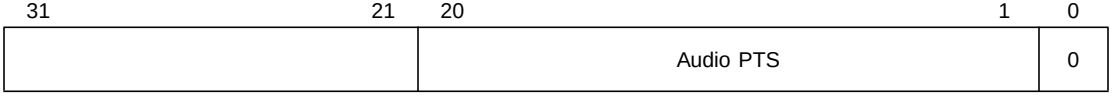


Audio IEC-958 Audio Status Out [31:0]

Audio IEC-958 Status Out 1 and 2. These two 32-bit registers are memory-mapped to Control Bus address 0x05001C and 0x050020. The 32 bits in IEC-958 Status 1 Out are clocked out on the first 32 subframe one status bits in each IEC-958 block. The 32 bits in IEC-958 Status 2 Out are clocked out on the first 32 subframe two status bits in each IEC-958 block. Bit 0 of each register corresponds to the first subframe in the block. Updates to the status registers are not transmitted on the IEC-958 until the start of the next block to avoid corrupting partially transmitted status information. Programming note: When changing IEC958 user status, software should allow the stream 1 output DMA queue to drain, change the IEC958 status registers then queue up the next output transfer.

Audio DMA In Status Register

Memory Space Address: 0x050024



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Audio DMA Out Status Register

Memory Space Address: 0x050028

31 – 21	20 – 1	0
	Audio PTS	0

Audio PTS **[20:1]**

The two audio DMA status registers listed above contain timestamps from the last completed stream input and stream output DMA transfers, respectively, as shown.

Bits[8:1] are the read only value of the 13.5 MHz divide-by-150 PTS counter, and Bits [20:9] are the read only value of the 90 kHz PTS counter as the first bit of the first sample leaves or arrives at the chip of the last completed DMA transfer of the associated audio stream. In DoMiNo, Bit 0 is always zero. Software can construct a full PTS by detecting wraparound of the 20-bit counter from one buffer to the next. Software must ensure that this register is read before the next associated outstanding audio transfer is completed to avoid losing the time stamp. Programming note: software can use Audio PTS to detect bad sclk rates.

12.5 Audio Address and Length Registers

The DMN-8600 processor provides registers containing length and SDRAM address information for audio stream transfers. For DoMiNo, four registers are used for length and two for SDRAM addresses, as shown in the following descriptions.

12.5.1 Audio Input DRAM Address Register

The 32-bit Audio Input DRAM Address register is memory-mapped to the control bus address 0x05002C, that contains the SDRAM address for audio input stream transfers. This register is initially written by the SPARC processor as an audio input stream transfer is being configured. The contents are undefined after the transfer begins. This register is part of a two-level command queue that is advanced each time the GoI bit is set by software. Reading or writing the Audio Input DRAM Address

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

register returns or updates the most recent queue entry. The lowest three bits are always zero.

12.5.2 Audio Input Length Register

The 32-bit Audio Input Length register is memory-mapped to control bus address 0x050030, which contains the length in bytes for an audio input stream transfer. This register is initially written by the SPARC processor as an audio input stream transfer is being configured. This register is part of a two-level command queue that is advanced each time the GoI bit is set by software. Reading or writing the Audio Input Length register returns or updates the most recent queue entry. The audio input time should not be longer than 45 msec to be within the wrap around interval of input PTS time stamp. The input length must be a multiple of 64 bytes.

12.5.3 Audio Output DRAM Address Register

The 32-bit Audio Output DRAM Address register, is memory-mapped to the control bus address 0x05003C, that contains the SDRAM address for an audio output stream transfer. This register is initially written by the SPARC processor as an audio output transfer is being configured. The contents are undefined after the transfer begins. This register is part of a two-level command queue that is advanced each time the GoO bit is set by software. Reading or writing the Audio Output DRAM Address register returns or updates the most recent queue entry. The lowest three bits are always zero. The audio output time should not be longer than 45 ms to be within the wrap around interval of output PTS time stamp.

12.5.4 IEC958 Output DRAM Address Register

The 32-bit IEC958 Output DRAM Address register, is memory-mapped to the control bus address 0x05004C, that contains the SDRAM address for an IEC958 output transfer. This register is initially written by the SPARC processor as an IEC958 and audio output transfer is being configured. The contents are undefined after the transfer begins. This register is part of a two-level command queue that is advanced each time the GoI bit is set by software. Reading or writing the IEC958 Output DRAM Address register returns or updates the most recent queue entry. The lowest three bits are always zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

12.5.5 Audio Output Length Register

The 32-bit Audio Output Length register, is memory-mapped to the control bus address 0x050040, that contains the length in bytes for an audio output stream transfer. This register is initially written by the SPARC processor as an audio output transfer is being configured. This register is part of a two-level command queue that is advanced each time the GoO bit is set by software. Reading or writing the Audio Output Length register returns or updates the most recent queue entry. The output length must be a multiple of 64 bytes. The length of audio output transfers with I958 set must correspond to an even number of IEC958 samples.

12.6 Audio Frame Formats

The DMN-8600 processor supports seven audio frame formats. Table 12.1 lists these formats with the products they support. To select the format needed by the application, the host must issue the Audio Configuration command defined in the DMN-8600 Programming Reference Manual.

Table 12.1 Audio Frame Formats

Format	Product Compatibility	Sample Justification	Sample Sizes (bits)
FRFORM 0	DSP chips	Left	16 to 32
FRFORM 1	Crystal 4216 and 4218 Codecs	Left	16
FRFORM 2	Reserved	–	–
FRFORM 3	Crystal ADCs and DACs	Left	16 to 32
FRFORM 4	IDS chips	Left	16 to 32
FRFORM 5	Typical multibit DACs and interpolation filters	Right	16
FRFORM 6		Right	18
FRFORM 7		Right	20

The number of samples per frame specifies the number of audio samples transmitted per frame sync cycle on each audio input and output pin.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Frame sync is input from the AI_FSYNC pin for audio capture and output on AO_FSYNC for audio output. Note that frames need not be back-to-back on input. The number of clocks in each sample must be at least 32.

Sample and status data are transmitted from the most significant to least significant bit on the pins. Formats FRFORM 0, 1, 3, and 4 left-justify the sample in the frame:

- When an output sample's memory size is 16 bits and the output timing is 32 clocks per sample, 16 trailing zeros are appended.
- When an output sample's memory size is 16 bits and the output timing is 64 clocks per sample, 48 trailing zeros are appended.
- When an output sample's memory size is 32 bits and the output timing is 64 clocks per sample, 32 trailing zeros are appended.

Formats 5 through 7 right-justify the sample in the frame:

- When an output sample's memory size is 16 bits and the output timing is 32 clocks per sample, 16 leading zeros are pre-pended.
- When an output sample's memory size is 16 bits and the output timing is 64 clocks per sample, 48 leading zeros are pre-pended.
- When an output sample's memory size is 32 bits and the output timing is 64 clocks per sample, 32 leading zeros are pre-pended.
- When outputting 32-bit memory size samples with 16 data clocks per sample in any format, the upper 16 bits are used.

12.6.1 FRFORM 0

Figure 12.2 shows the FRFORM 0 audio frame format. In FRFORM 0, only one audio sample is transmitted per frame sync (FSYNC) cycle. The FSYNC signal is of pulse type; it transitions at the first bit of each sample. The transition from HIGH to LOW is ignored. The frames need not be back to back. Samples are left-justified and, with only one sample per frame, only four channels can be transmitted.

The data has the same format as an input sample specified in the Audio Configuration command defined in the DMN-8600Programming Reference Manual.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

12.6.2 FRFORM 1

Figure 12.3 and Figure 12.4 show the FRFORM 1 audio frame format. In this format, two 16-bit audio samples and two status samples are transmitted per frame sync (FSYNC) cycle. The frame sync transition occurs at the first bit of the first sample in the frame.

FRFORM 1 requires input and output samples to be 16 clocks each, because there is no other timing reference to delineate sample boundaries. FRFORM 1 transfers four 16-bit words per frame (additional bits are discarded on input).

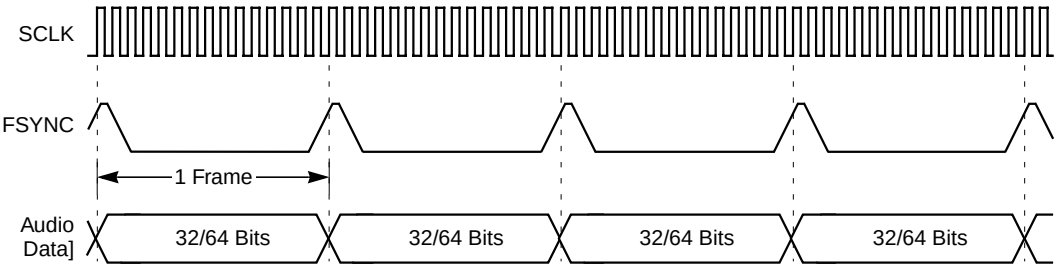
FRFORM 1 uses a pulse frame sync (FSYNC) signal. On output, FSYNC is HIGH for one clock, and then goes LOW until the next frame (Figure 12.3). On input, the HIGH-to-LOW transition is ignored (Figure 12.4).

FRFORM 1 clocks in the left status information from the AI_D[3:0] pins to the Audio Status In 0 and 1 Registers as shown in Figure 12.1. Note that the right status information is discarded.

Figure 12.1 Location of FRFORM 1 Status

Register	31 — 16	15 — 0	Address
Audio Status In 0	Left Status from AI_D0 Pin	Left Status from AI_D1 Pin	0x050008
Audio Status In 1	Left Status from AI_D2 Pin	Left Status from AI_D3 Pin	0x05000C

Figure 12.2 FRFORM 0 Audio Frame Definition



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 12.3 FRFORM 1 Audio Frame Output Definition

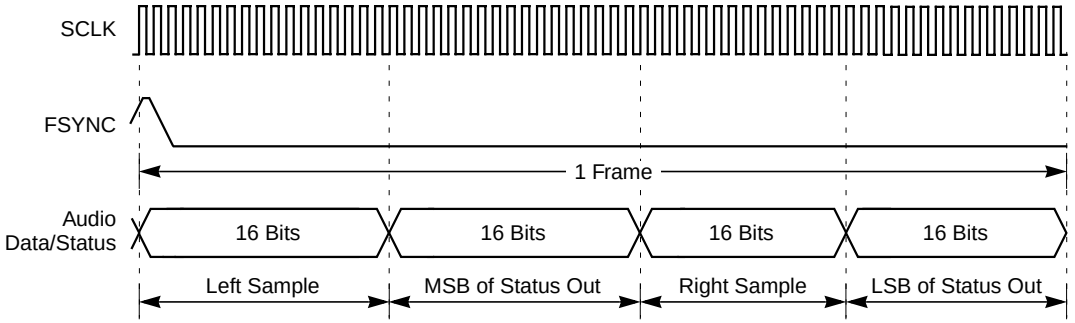
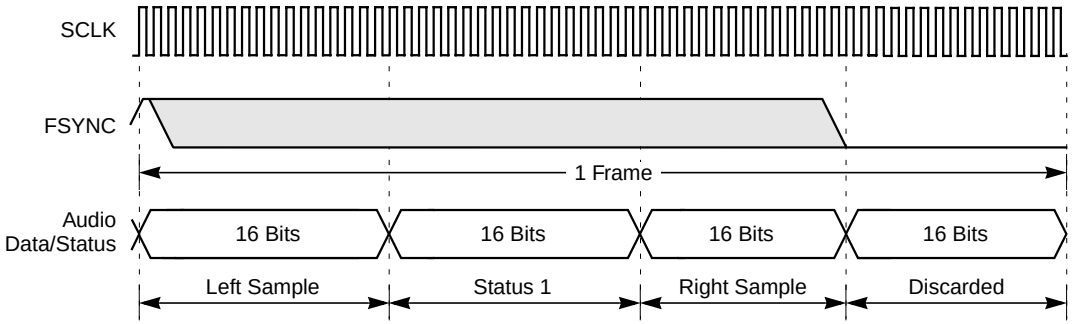


Figure 12.4 FRFORM 1 Audio Frame Input Definition



12.6.3 FRFORM 2

The FRFORM 2 audio frame format is reserved.

12.6.4 FRFORM 3

Figure 12.5 shows the FRFORM 3 audio frame format. In this format, two audio samples are transmitted per frame sync (FSYNC) cycle. FRFORM 3 uses a Left HIGH/Right LOW frame sync–that is, the Left sample is present when FSYNC is HIGH, and the Right sample is present when FSYNC is LOW. The FSYNC transition occurs at the first bit of each sample. The transition of FSYNC from LOW-to-HIGH and HIGH-to-LOW defines the beginning or end of each sample.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

12.6.5 FRFORM 4

Figure 12.6 shows the FRFORM 4 audio frame format. In this format, two audio samples are transmitted per frame sync (FSYNC) cycle. FRFORM 4 uses a Left LOW/Right HIGH frame sync–that is, the Left sample is present when FSYNC is LOW, and the Right sample is present when FSYNC is HIGH. The FSYNC transition occurs one clock cycle before the first bit of each sample. The transition of FSYNC from LOW-to-HIGH and HIGH-to-LOW defines the beginning or end of each sample.

Figure 12.5 FRFORM 3 Audio Frame Definition

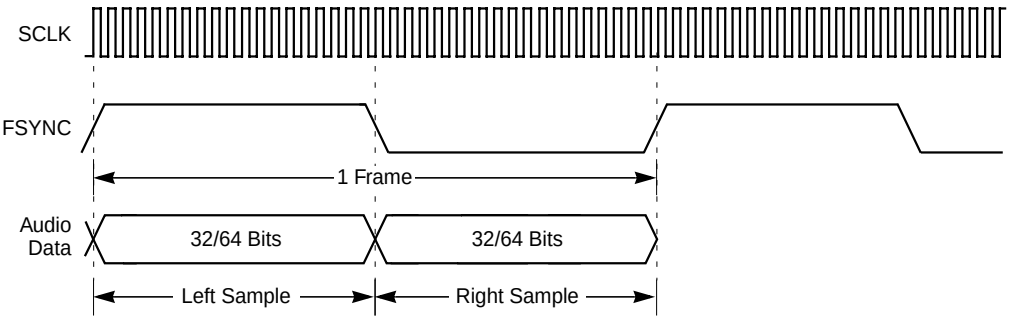
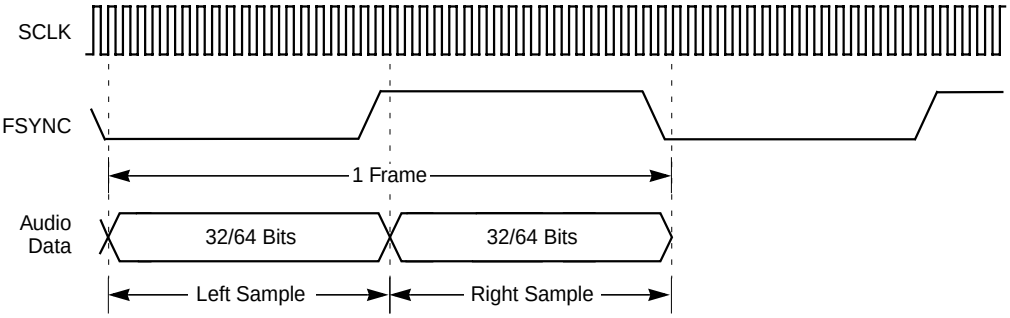


Figure 12.6 FRFORM 4 Audio Frame Definition



12.6.6 FRFORM 5

Figure 12.7 shows the FRFORM 5 audio frame format. In this format, two 16-bit audio samples are transmitted per frame sync (FSYNC) cycle. FRFORM 5 uses a Left HIGH/Right LOW frame sync–that is, the Left

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

sample is present when FSYNC is HIGH, and the Right sample is present when FSYNC is LOW. The FSYNC transition occurs one clock cycle after the last bit of each 16-bit external sample. The transition of FSYNC from LOW-to-HIGH and HIGH-to-LOW defines the beginning or end of each sample.

12.6.7 FRFORM 6

Figure 12.8 shows the FRFORM 6 audio frame format. In this format, two 18-bit audio samples are transmitted per frame sync (FSYNC) cycle. FRFORM 6 uses a Left HIGH/Right LOW frame sync–that is, the Left sample is present when FSYNC is HIGH, and the Right sample is present when FSYNC is LOW. The FSYNC transition occurs one clock after the last bit of each 18-bit external sample. The transition of FSYNC from LOW-to-HIGH and HIGH-to-LOW defines the beginning or end of each sample.

Figure 12.7 FRFORM 5 Audio Frame Definition (Right Justified)

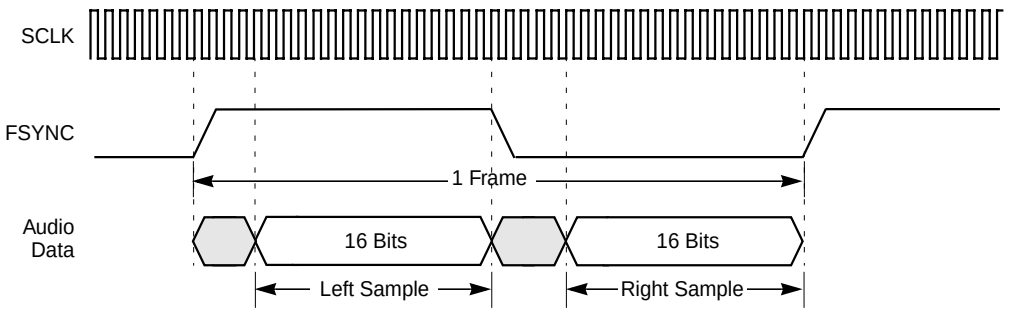
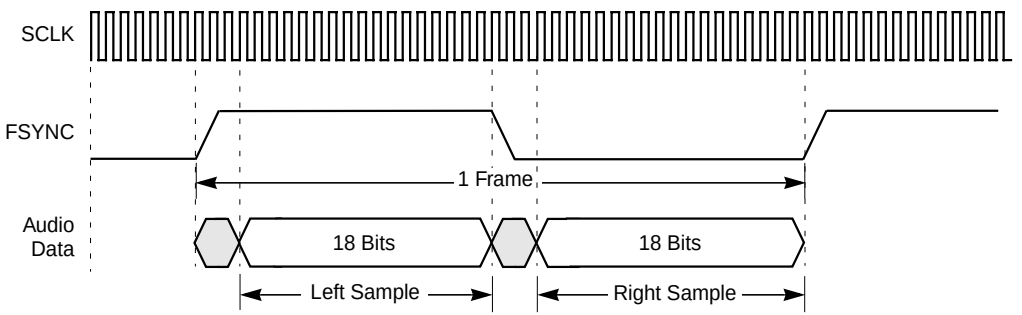


Figure 12.8 FRFORM 6 Audio Frame Definition (Right Justified)

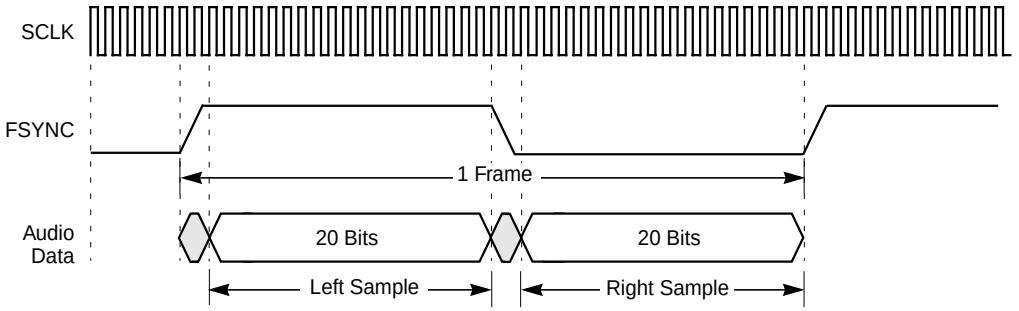


Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

12.6.8 FRFORM 7

Figure 12.9 shows the FRFORM 7 audio frame format. In this format, two 20-bit audio samples are transmitted per frame sync (FSYNC) cycle. FRFORM 7 uses a Left HIGH/Right LOW frame sync–that is, the Left sample is present when FSYNC is HIGH, and the Right sample is present when FSYNC is LOW. The FSYNC transition occurs one clock after the last bit of each 20-bit external sample. The transition of FSYNC from LOW-to-HIGH and HIGH-to-LOW defines the beginning or end of each sample.

Figure 12.9 FRFORM 7 Audio Frame Definition (Right Justified)



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 13
SDRAM Interface

This chapter describes the SDRAM interface and contains the following sections:

The DMN-8600 processor uses a 32-bit memory interface that supports up to 64 Mbytes of DRAM. The DRAM memory uses two to four 16-bit wide SDR/DDR DRAMs, or one to two 32-bit wide SDR/DDR DRAMs, capable of running at 150 MHz, 2.5 V–3.3 V. $\overline{\text{SDRAM\_RAS}}$ and $\overline{\text{SDRAM\_CAS}}$ latency and cycle times are programmable to allow for DRAMs from different vendors at different speeds to be used.

The DMN-8600 supports the following DRAM types, with parameters listed in Table 13.1:

- 8 Mbytes to 64 Mbytes
- Three types of DRAM:
  - Single Data Rate Synchronous DRAM (SDR-SDRAM)
  - Double Data Rate Synchronous DRAM (DDR-SDRAM)
  - Double Data Rate Synchronous GRAM (DDR-SGRAM)

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- Four slots, each supporting one 8 Mbytes or 16 Mbytes chip.
- 32-bit external interface.
- Support for x32 and x16 SDR/DDR DRAM chips
- 4 banks (minimum) per DRAM chip
- All external parts have the same configuration, i.e. same timing requirements, same RAS/CAS/BS sizes, and all are either SDRAMs or DDRs.

Table 13.1 Supported DRAM Types[1]

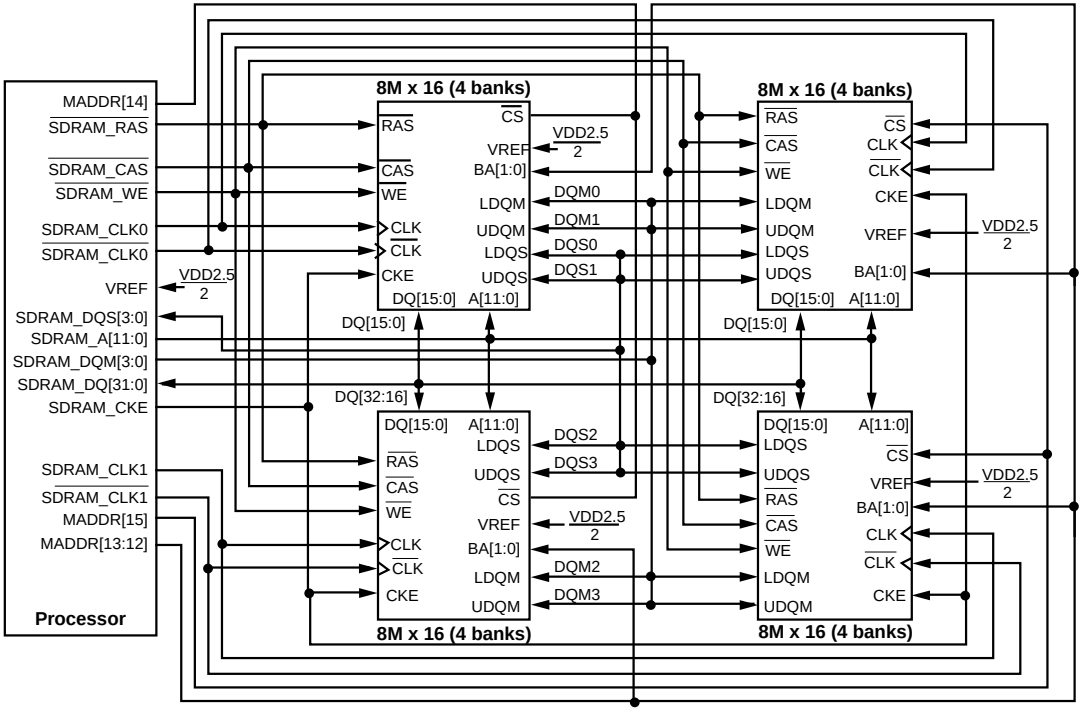
Type	Part	Quantity	Config	Total MG
SDR SDRAM	512 K x 32 b x 4 Banks	1 or 2	stacked	8–16 Mbytes
SDR SDRAM	1 M x 16 b x 4 Banks	2 or 4	stacked pairs	16–32 Mbytes
SDR SDRAM	2 M x 16 b x 4 Banks	2 or 4	stacked pairs	32–64 Mbytes
DDR SDRAM	512 K x 32 b x 4 Banks	1 or 2	stacked	8–16 Mbytes
DDR SDRAM	512 K x 32 b x 8 Banks	1or 2	stacked	16–32 Mbytes
DDR SDRAM	1 M x 32 b x 4 Banks	1or 2	stacked	16–32 Mbytes
DDR SDRAM	1 M x 32 b x 8 Banks	1or 2	stacked	32–64 Mbytes
DDR SDRAM	2 M x 32 b x 4 Banks	1or 2	stacked	32–64 Mbytes
DDR SDRAM	1 M x 16 b x 4 Banks	2 or 4	stacked pairs	16–32 Mbytes
DDR SDRAM	2 M x 16 b x 4 Banks	2 or 4	stacked pairs	32–64 Mbytes

1. HDTV requires 8 banks (two 4 bank or one 8 bank part) and 150 MHz.

Example configurations are shown in Figure 13.1–Figure 13.4.

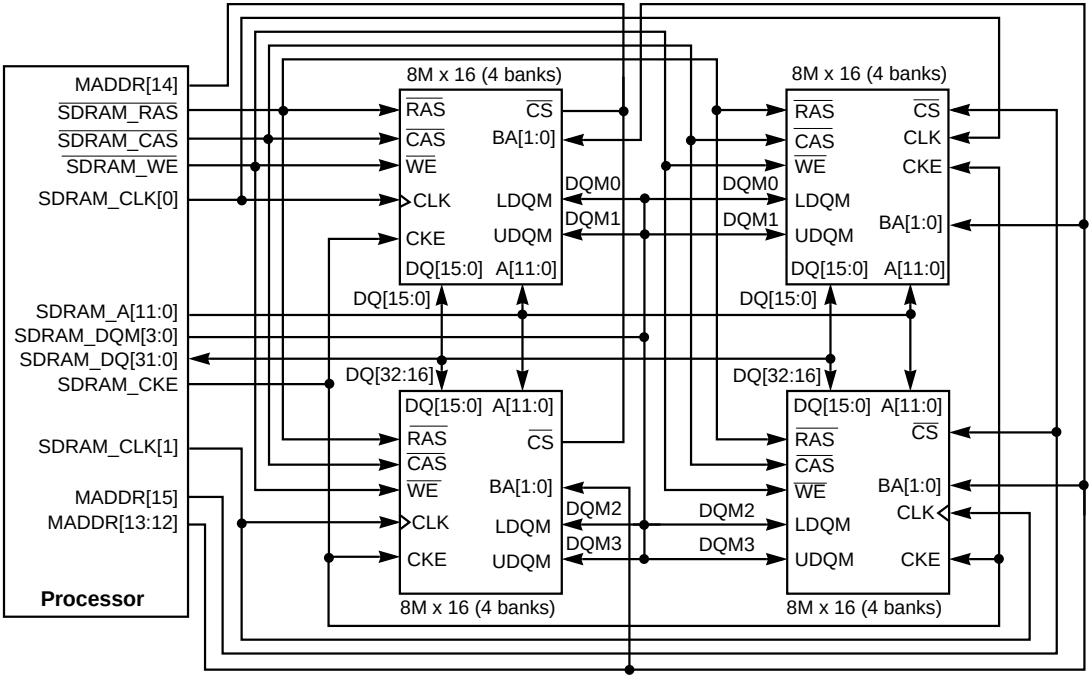
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 13.1 16-Bit DRAM Connections Using DDR SDRAM



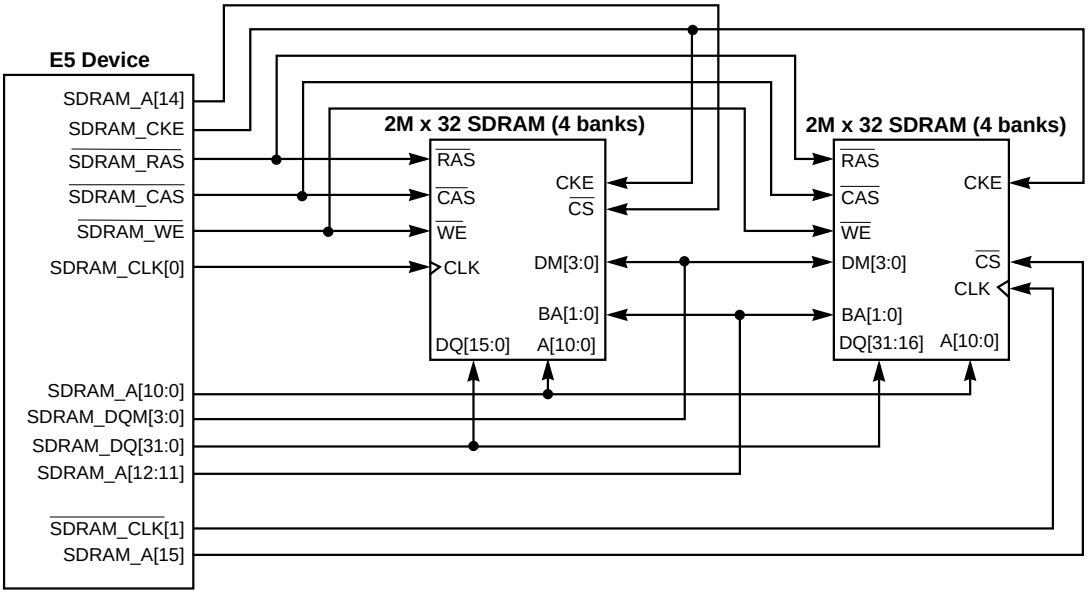
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 13.2 16-Bit DRAM Connections Using SDR SDRAM



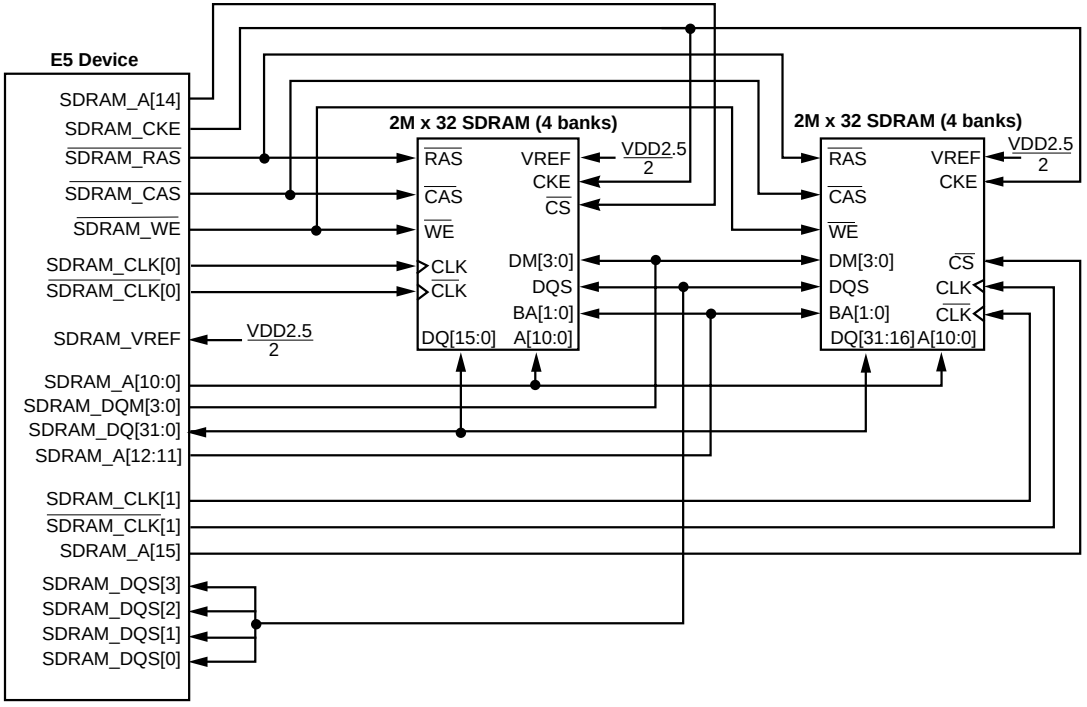
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 13.3 32-Bit DRAM Connections Using SDR SRAM



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 13.4 32-bit DRAM Connections using DDR SRAM



13.1 DRAM Address Map

Every DRAM starts at a base address that is at an 8 Mbytes boundary. The addressing can be determined from the size of the DRAM/PAGE/Bank Count. The total DRAM address space always appears contiguous without any hole in-between. To avoid any hole, external slots are populated starting at slot 0.

13.2 DRAM Address Field Description

To support 64 Mbytes (16 Mwords) with a 32-bit wide word requires 24 bits of word-addressing. The 7 to 9 LSBs are used to select word inside a page. The bits above the column address bits are used to select banks (2 bits for 4 banks, and 3 bits for 8 banks). The bits above

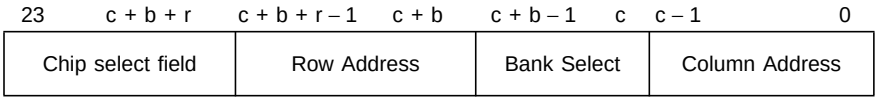
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

bank_select are the row address bits. The top-most bit is decoded into individual chip select bits in multiple DRAM configurations, except when two (x32) 4 bank parts are used. In the latter case, the chip select field is inserted between the first and second bit of the bank select field to provide tile interleaving.

For external DRAM configurations, the pinout for address SDRAM_A[15:0] is as follows:

CS Pins	BA Pins, Adr Pins
[15:14]	[13:0]

The internal linear address breakdown is shown below,



where c is the column size, b is the bank select field, r is the row size field, and the chip select field is decoded to set the CS pins: CS_field = 0 sets SDRAM_A[15], and CS_field = 1 sets SDRAM_A[14].

Note: Bank Select pin connections for DDR-based systems differ in the way the extended mode register (EMR) programming is defined in the DRAM chip on the board. If the EMR is defined as {BA1, BA0, Adr pins} for the DRAM, then the two bank select pins of DoMiNo need to be connected to the DRAM in order (more significant DoMiNo BA pin to BA1, less significant DoMiNo BA pin to BA0). However, if the EMR is defined as {BA0, BA1, Adr pins} for the DRAM (e.g., micron chips), then the two bank select pins of DoMiNo need to be connected in the reverse order (more significant DoMiNo BA pin to BA0, less significant DoMiNo BA pin to BA1).

13.3 Supported Number of Simultaneous Banks

Within a transaction, up to four active DRAM banks are used. A misaligned 9 V x 64 H transaction will touch at most 4 pages in separate banks (with the DMN-8600 tiled image format). The bank precharge and

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

RAS time for the last three banks are hidden behind the first bank access. Larger transfers are sustained without bubbles using 4 banks by closing active banks as new banks are opened.

To overlap across accesses without overhead (hide precharge of old pages and activate of future pages) in memory configurations with at least 8 banks, the first 4 banks of the next access may be activated before closing the last 4 banks of the previous access.

13.4 SDRAM Initialization

DRAMs require an initialization sequence to be performed when the DMN-8600 is reset through the reset pin or the host control register.

Note: Proper initialization assumes that the SDRAM Clock Control and External SDRAM Configuration register have been programmed, and that the controller has been taken out of Stop mode by programming the Internal Clock Control register (see Section 16.1).

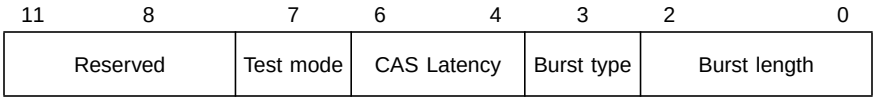
After reset, $\overline{CS}$ and SDRAM_DQM will be held high, and CKE will be low. Once the above registers are programmed, the DMN-8600 will initialize the DRAM using the following sequence:

- For SDR: Two precharge-all-banks commands are issued, followed by two refresh commands before the mode register is programmed for normal operation.
- For DDR: One precharge-all-banks command is issued, followed by an extended mode register (EMRS) write to enable the DLL. Then the mode register is programmed to reset the DLL. After a delay of at least 200 cycles, another precharge-all-banks command is issued, then at least two refresh commands are issued, followed by the mode register write to prepare the DRAM for normal operation.

The DRAM chips are configured via an internal Mode register, a 12-bit register that defines Burst length, access type, and CAS latency, as well as some vendor-specific test fields. The mode register is written during the reset sequence. A bit assignment of the register is shown in Figure .

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SDRAM Mode Register



The Mode register is written by asserting $\overline{RAS}$, $\overline{CAS}$, and $\overline{WE}$ strobes on the rising edge of clock (refer to specific DRAM timing specs for more details). The 12 bits of data are written via the Address bus. Values for the DRAM Mode register used by the DMN-8600 processor will be as follows:

Reserved **[11:8]**

Test mode **7**

These test enable bits will always be written to 00000, disabling test mode.

CAS Latency **[6:4]**

The CAS latency field will be set to 2 or 3 clocks for SDR, while for DDR it will be set to 2.5 or 3 clocks as determined by the CAS latency field ($\overline{CAS}$) in the External DRAM Configuration register (as loaded from the serial ROM).

Burst Type **3**

The Burst type field will be set to ‘0’, denoting Sequential accesses (rather than Interleaved accesses between banks).

Burst Length **[2:0]**

The Burst length field will be assigned a value of 011, denoting a burst length of eight. DoMiNo will terminate bursts by issuing another read, write or stop burst command. Bursts are needed to allow the precharge and RAS for the second bank to be issued while transferring from the first bank.

13.5 SDRAM Refresh

Since all MPEG applications will require DRAMs that remain active, the DMN-8600 processor will employ Autorefresh in it’s design. Autorefresh is enabled by driving $\overline{CS}$, $\overline{RAS}$, and $\overline{CAS}$ active and $\overline{WE}$ inactive on a

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

rising edge of clock. Refresh cycles (one to each bank) are generated at the rate programmed by the refresh interval field of the DRAM control register.

13.6 External SDRAM Configuration Register

The 32-bit Configuration Register that resides in the SDRAM is shown below. The first time the external configuration is written after reset, a DRAM initialization sequence is performed. The fields within the Configuration Register are described below.

External DRAM Configuration Register

Cbus Address: 0x30014

31	30 29	28 27	26 24	23	22	21	20 19	18	17 16
SD/DDR	Page Size	DRAM Size	Part Count	SG/SD R	Bank C	Drive Stren	Last Data to Read	LtoF L	LastDin2P Latency

15	14 13	12 11	10 9	8 7	6	5 0
WrC2D	ActiveP Latency	Prech Latency	Act L	RAS Latency	CAS L	Refresh Interval

SD/DDR **31**

SDRAM/DDR. Indicates the type of external DRAM. Not applicable to the embedded part.

0 = Single Data Rate SG/SDRAM

1 = DDR SG/SDRAM

Page Size **[30:29]**

PAGE size. All pages are 32-wide. For 16-bit wide parts, refers to the page size of a pair of parts. The tile height is determined by this field.

00 = Page size is 32x128. Tile height is 8, this requires a bank count (bit 22) of 8

01 = Page size is 32x256. Tile height is 16.

10 = Page size is 32x512. Tile height is 32.

DRAM Size **[28:27]**

DRAM size. For 16 bit wide parts, refers to the size of a pair of parts.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

0 = Reserved

1 = 8 Mbytes

2 = 16 Mbytes

3 = 32 Mbytes

Part Count **[26:24]**

Part Count. Indicates how many slots are populated by DRAM.

0 = Reserved

1–2 = Up to 2 slots can be populated.

SG/SD R **23**

SG/SDRAM. Indicates the type of external DRAM.

0 = 0SGRAM

1 = SDRAM

Bank Cnt **Bank Count** **22**

Indicates the number of banks in each part.

0 = 4 physical banks

1 = 8 physical banks (works only with page size of 512 words)

Drive Stren **21**

Drive strength at the DRAM pins.

0= Normal full drive

1= Half drive

Last Data to Read **20:19**

Last data of a read command delay. This indicates how soon after the last data for a write, a read command can be issued.

LtoF L **18**

LtoF. Last data from one data bus drivers to first data from another data bus driver. This indicates the minimum number of dead clocks between data bus drivers(bus turn around time). The actual value is (programmed_value + 1). Implementation note: Writes to different slots do not require dead clocks. Reads from different slots and read to write require dead clocks. Write to read will have at least 1.5 dead clocks because of CAS latency.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

LastDin2P Latency **[17:16]**

LastDin2P. LastData in to Row Precharge(tRDL). This indicates how soon after the last data write row precharge can take place. The actual value is (programmed_value + 1).

There is a similar parameter for read case, but it is typically tied to CAS Latency number and can be same as CAS Latency.

WrC2D **15**

WriteC2D Latency. WriteCommand to Data latency specifies the minimum number clocks between the write command and the first data for that command.

0 = 0 clock. Typically for SDRAM.

1 = 1 clock. Typically for DDR.

ActiveP Latency **[14:12]**

ActiveP latency (tRAS) specifies the minimum number of clocks between activate and precharge commands to the same bank. The actual value is (programmed_value + 4).

Prech Latency **[11:10]**

Prech latency(tRP) specifies the minimum number of clocks between a precharge command and an activate command to the same bank. The actual value is (programmed_value + 2).

Act L **9**

ActiveO latency(tRRD) specifies the minimum number clocks between successive activate commands to different banks.

0 = 2 clocks. Powers up as 0.
1= 3 clocks.

RAS Latency **[8:7]**

RAS latency(tRCD) specifies the minimum number clocks between an activate command and a read or write. Active command to Active Command duration (tRC) is not programmed since it is (tRAS + tRP). The actual value is (programmed_value + 1).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

CAS L **6**

CAS latency (tACK) specifies the minimum number clocks between a read command and data being returned. For SDR parts, the values are:

0 = 2 clocks
1 = 3 clocks

For DDR parts, the values are:

0 = 2.5 clocks
1 = 3 clocks.

Refresh Interval **[5:0]**

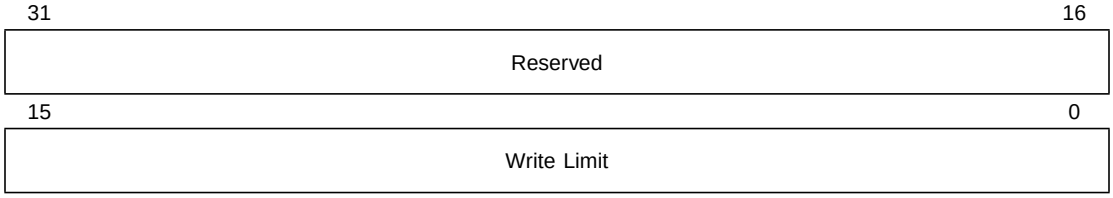
Specifies the number of clocks between refresh cycle in units of 64 clocks. Each refresh cycle will refresh all banks of the DRAM. The nominal value for 150 MHz is 2340 clocks (2048 rows every 32 ms) for SDRAMs and 1170 clocks (1024 rows every 8 ms) for DDRs.

13.7 SDRAM Control and Clock Control Registers

As shown below, the SDRAM Control and Clock Control Registers in the DMN-8600 processor set up the interface to SDRAM.

SDRAM Control Register

Cbus Address: 0x30000



Write Limit **[15:0]**

A store by an incoming IPC target or any DMA transfer to an DRAM address less than write limit * 4096 will cause a DRAM Write Protection Interrupt to the RISC core, and the write will be suppressed.

The SDRAM Clock Control Register controls the timing of the SDRAM input and output clocks. This register must be written at least 1

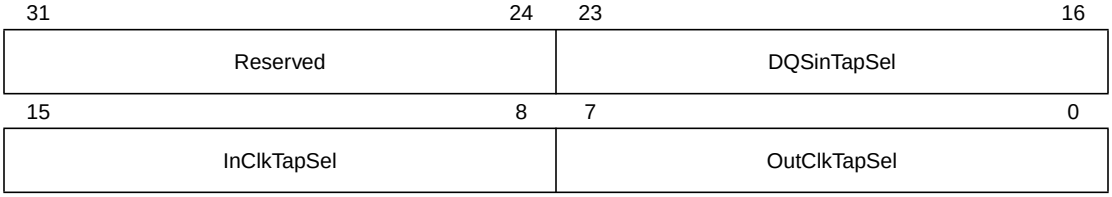
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

microseconds prior to writing the External DRAM Configuration register (Section 13.6) so the DRAM clocks are stable for the required period (per DRAM data sheet specs) prior to initialization of the DRAMs. This register should only be written once after the DMN-8600 is reset (via the Chip Reset bit in the Host Control register).

The DRAM Clock Control register is a memory mapped register at offset 0x30010 from the DMN-8600 address window, shown below

SDRAM Clock Control Register

Cbus Address: 0x30010



DQSinTapSel **[23:16]**

DQSinTapSel specifies the delay from the DQS input strobe to input capture. The delay is in units of 1/256 of the internal clock period.

InClkTapSel **[15:8]**

For SDR DRAM, InClkTapSel specifies the delay from the internal (DoMiNoClock) clock to the input capture. For DDR dram, this field specifies the delay from write data out to DQS out. The delay is in units of 1/256 of the internal clock period.

OutClkTapSel **[7:0]**

OutClkTapSel specifies the delay from the internal (DMN-8600 Clock) clock to the DRAM_CLK output clock. The delay is in units of 1/256 of the internal clock period.

13.8 SDRAM Arbitration and Throttle Registers

DRAM requests are arbitrated by priority according to the latency requirements of the requestor. The nominal request priority of each

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

requestor is specified by the register settings of the registers described in this section according to the following sequential list:

1. OSD Output Channel
2. Video PIP Output Channel
3. Video alpha Channel
4. Video Output Channel
5. Video Stream 1 Previous Field Channel
6. Video Stream 1 Opposite Field Chroma (frame chroma decimation)
7. Video Stream 1 Capture Channel
8. 1394 DMA
9. Secondary Bitstream Port DMA
10. ATAPI DMA
11. SMARTCARD_scd1MemReq
12. SMARTCARD_scd2MemReq
13. Audio Stream 1 Output Channel
14. Audio Stream 2 Output Channel
15. Audio Stream 1 Input Channel
16. Audio Stream 2 Input Channel
17. Audio IEC958 Output Channel
18. SIO DMA request
19. Async Master DMA
20. Video SPARC cache miss
21. System SPARC cache miss
22. Video rescaling write (both buffers full)
23. DSP Alpha channel
24. DSP VLC channel
25. ME Wmem/Tmem load
26. ME result write
27. ME command read
28. Video Subpicture RLE Data input

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

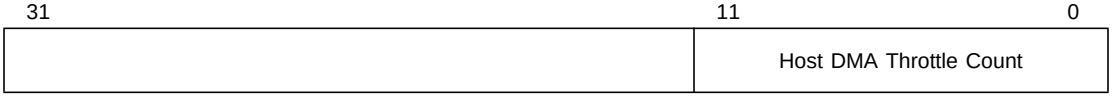
29. Video Subpicture RLE Control input
30. Video Subpicture Output Channel
31. Video SPARC DMA mem req
32. System SPARC DMA mem req
33. DSP DMA
34. Video rescaling write (both buffers not full)

Note:

- The priority of ATAPI and SBP DMA requests are moved to the lowest priority for the number of clocks specified by the ATAPI/SBP DMA throttle register after the previous request.
- The priority of Host DMA requests are moved to the lowest priority for the number of clocks specified by the Host DMA throttle register after the previous request.

SDRAM Host DMA Throttle Register

Cbus Address: 0x3000C



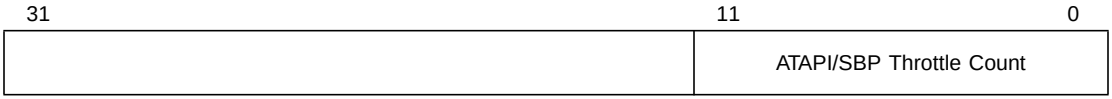
Host DMA Throttle Count **[11:0]**

After a Host DMA request is asserted, the internal host DMA throttle counter is set to the host DMA throttle count. Subsequent host DMA requests are treated as lowest priority until the internal host DMA throttle counter reaches zero. This allows software to spread host DMA DRAM loading uniformly over a field, when the full host DMA bandwidth is not needed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SDRAM ATAPI/SBP DMA Throttle Register

Cbus Address: 0x30018



ATAPI/SBP Throttle Count **[11:0]**

After an ATAPI request is asserted, the internal ATAPI throttle counter is set to the host ATAPI/SBP target throttle count. Subsequent ATAPI requests are treated as lowest priority until the internal ATAPI throttle counter reaches zero. After an SBP request is asserted, the internal SBP throttle counter is set to the host ATAPI/SBP target throttle count. Subsequent SBP requests are treated as lowest priority until the internal SBP throttle counter reaches zero. This allows software to spread ATAPI and SBP DRAM loading uniformly over a field, when the full ATAPI or SBP target bandwidth is not needed.

13.9 SDRAM Timing

The nominal output delay of SDRAM_CLK is matched to the delays of the other processor outputs to the SDRAMs (including SDRAM_DQ). The actual SDRAM_CLK output delay can be adjusted in microcode to lead the other processor outputs to the SDRAMs in increments of the processor CLK period divided by 256. This programmable lead time enables the SDRAM setup and hold time requirements to be met.

The tracking between SDRAM outputs and the accuracy of the lead-time adjustment is such that for a given board layout and SDRAM specification, a single lead-time value allows the SDRAM interface to function over the entire range of processor process variations and operating conditions.

Read data on SDRAM_DQ is latched internally with a read clock whose lead time relative to the other processor outputs to the SDRAMs is specified in microcode. The read clock tracks the other outputs

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

(specifically SDRAM_CLK) so that read data can be captured independent of actual output delay.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 14
Bitstream I/O (Storage) Port

This chapter describes the bitstream I/O port and contains the following sections:

The primary BIO port consists of two component modules that operate independently:

- *ATAPI/SD/CD Controller*: Interfaces to recordable storage devices.
- *1394 Controller*: Transmits and receives MPEG transport and DV streams on up to four channels in each direction.

Note: The secondary BIO port, described in Chapter 9, “Secondary Bitstream Interface,” shares its pins with the ATAPI/SD/CD module.

In addition, the BIO uses a DMA interface for communication with SDRAM. This is similar to other modules in DoMiNo.

The BIO provides read/write pointers, base and limit registers for managing six SDRAM DMA ring buffers for ATAPI/SD sectors, transport channel packets, 1394 isochronous packet input and output, and 1394 asynchronous packet input and output. Through the DMA channels, the BIO writes the input data into ring buffers during the input operations and reads output data from ring buffer during output operations. Two more ring buffers are supported to manage input data for SDRAM-to-SDRAM operations.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

14.1 ATAPI Interface

The ATAP interface supports the following operations:

- Register read
- Register write
- DMA read
- DMA write

The Register read and write operations are used to configure a set of ATAPI device registers. The DMA read and write operations are needed for transferring long sector data.

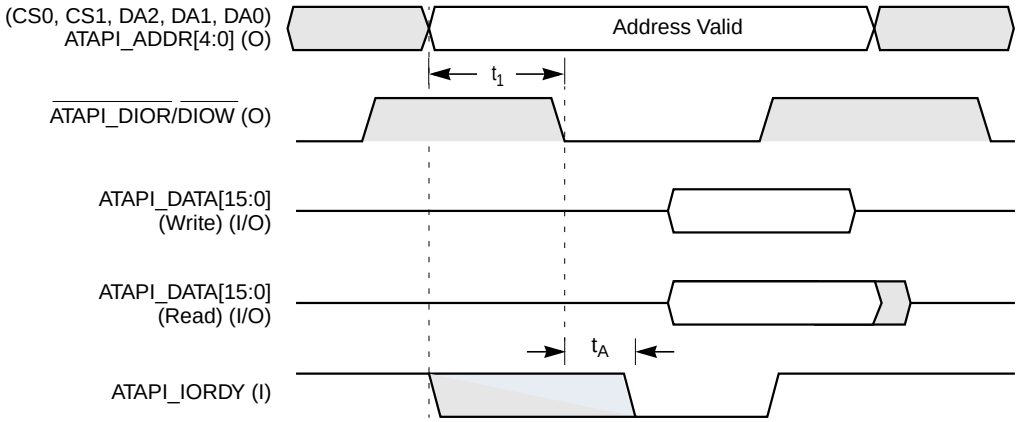
The ATAPI read and write cycles are described in Figure 14.1. The ATAPI register address is defined by the output pins CS0, CS1, DA2, DA1, and DA0, where CS0 is the most significant bit.

14.1.1 Read Cycle

The ATAPI read cycle starts when the ATAPI Interface receives the ATAPI_RD command from the host. The ATAPI Interface subsequently puts the register address on the address bus and asserts the $\overline{\text{ATAPI\_DIOR}}$ signal t1 time later. The 16-bit data ($\overline{\text{AtapiIOCS16}}$ signal is always low) from the ATAPI device is latched by the ATAPI Interface during the rising edge of the $\overline{\text{ATAPI\_DIOR}}$ signal.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 14.1 ATAPI Read and Write Cycle



The wait cycle can be generated by the ATAPI device driving the ATAPI_IORDY signal low during the read cycle. The ATAPI device must drive the ATAPI_IORDY signal low before time tA to initiate a wait cycle. During the wait cycle, the ATAPI Interface keeps the register address and the ATAPI_DIOR signal asserted until the ATAPI_IORDY signal becomes high.

14.1.2 Write Cycle

The ATAPI write cycle starts when the ATAPI Interface receives the ATAPI_WR command from the host. The ATAPI Interface follows by placing the register address on the address bus and asserting the ATAPI_DIOW signal time t1 later.

The ATAPI device can also initiate the wait cycle by driving the ATAPI_IORDY signal low before time tA. During the wait cycle, the ATAPI Interface keeps the register address, write the data, and the ATAPI_DIOW signal asserted until the ATAPI_IORDY signal becomes high.

14.1.3 DMA Operation

The DMA operation is used for transferring long sets of data to or from the ATAPI device. The host must do several ATAPI register read/writes to the ATAPI device using the ATAPI_RD and ATAPI_WR commands before it can initiate the ATAPI DMA transfer from the host.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

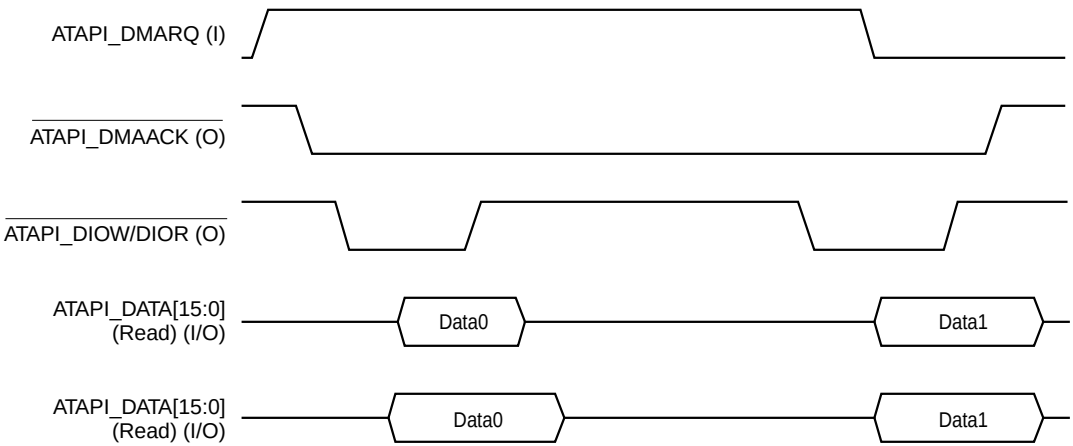
The DMA read/write cycle timing diagram is shown in Figure 14.2.

The DMA transfer starts when the ATAPI device asserts the DMA request signal ATAPI_DMARQ. The ATAPI Interface responds by asserting the DMA acknowledge signal $\overline{\text{ATAPI\_DMAACK}}$. Depending on the DMA read or write operation, the ATAPI Interface asserts the $\overline{\text{ATAPI\_DIOR}}$ or $\overline{\text{ATAPI\_DIOW}}$ signal to transfer the data.

During the DMA read operation, the ATAPI Interface asserts the $\overline{\text{ATAPI\_DIOR}}$ signal to initiate the read operation and latch the data from the ATAPI device on the trailing edge of the $\overline{\text{ATAPI\_DIOR}}$ signal. This process repeats itself until the ATAPI device deasserts the ATAPI_DMARQ signal.

During the DMA write operation, the ATAPI Interface asserts the $\overline{\text{ATAPI\_DIOW}}$ signal and puts the data on the data bus to start the write operation. The ATAPI device must latch the write data during the trailing edge of the $\overline{\text{ATAPI\_DIOW}}$ signal. The DMA write operation completes when the ATAPI device deasserts the ATAPI_DMARQ signal.

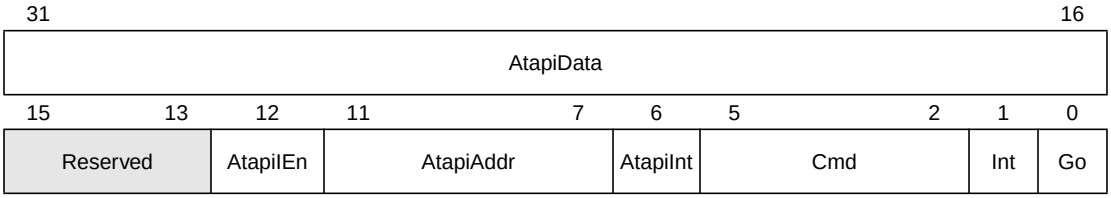
Figure 14.2 ATAPI DMA Read/Write Cycle



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

DVD Control Register

Cbus Address: 0xFB0000



AtapiData **[31:16]**

ATAPI Read/Write data. For the ATAPI_RD command, this field contains the value read from the ATA register indexed by AtapiAddr. For the ATAPI_WR command, this field contains the value to be written to the indexed ATA register.

AtapiIEn **12**

ATAPI Interrupt Enable. When this bit is set and AtapiIntRq changes from 0 to 1, an ATAPI Completion interrupt is generated.

AtapiAddr **[11:7]**

ATAPI Register Address. This field corresponds to pins CS0, CS1, DA2, DA1 and DA0. CS0 is the most significant bit.

AtapiInt **6**

ATAPI Interrupt Request. Software can monitor the status of DoMiNo's ATAPI_INTRQ input pin by reading this read-only bit. External devices will set this bit to indicate the completion of ATAPI operations.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Cmd **[5:2]**

ATAPI Command. Various processing commands to the drive, as shown below. Data written or read from the SDRAM is addressed by a ring buffer pointer.

Command	BIts Used	Description
ATAPI_RD	0000	Read the specified ATAPI device register into AtapiData.
ATAPI_WR	0001	Write the specified ATAPI device register with value AtapiData.
DEC_DK	0010	Decrypt the DK from DVDencDK* registers.
DEC_TK	0011	Decrypt the TK from DVDencTK* registers.
ATAPI_DESC_BS	0100	Use ATAPI interface to read a specified number of DVD sectors, and descramble the data based on bitstream information. The result is written to SDRAM.
ATAPI_DESC_NO	0101	Use ATAPI interface to read a specified number of DVD sectors, and don't descramble the data. The result is written to SDRAM.
SD_DESC_BS	0110	Use SD interface to read a specified number of DVD sectors, and descramble the data based on bitstream information. The result is written to SDRAM.
SD_DESC_NO	0111	Use SD interface to read a specified number of DVD sectors, and don't descramble the data. The result is written to SDRAM.
DESC_SDRAM	1000	Read a specified number of DVD sectors stored in SDRAM, descramble the data based on bitstream information and write back the result into SDRAM. In this mode, the operation is completed normally when the source ring buffer is empty and the target ring buffer is full, and the size of data processed is a multiple of a sector. If any of the conditions does not hold, the results are undefined.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Command	BIts Used	Description
READ_CD	1001	Read a specified number of CD sectors; starts with sector sync search if DVDcfg.SearchCnt is non-zero.
ATAPI_WR_DATA	1010	Write a specified number of DVD sectors in SDRAM to the ATAPI interface.
SD_WR_DATA	1011	Write a specified number of generic sectors in SDRAM to the SD interface with variable sector length.
SD_RD_DATA	1100	Read a specified number of non-DVD sectors with variable sector length. No descrambling is performed.

Int **1**

ATAPI Interrupt. 1 = generate an ATAPI Completion interrupt when the specified command is complete (for transfer commands, completion includes transferring the specified number of sectors). 0 = no interrupt.

Go **0**

1 = start a new command. 0 = previous command is finished. When software sets the Go bit to 1, the command specified in the Cmd field is started. When the BIO completes the command, it resets the Go bit to 0 automatically. If software resets the Go bit to 0 before a command is completed, the command is aborted from the BIO's point of view. When any command is in progress, writing to this register with Go = 1 is ignored. On reset, Go = 0.

14.2 SD Interface

The SD Interface is used to read data from and write data to the DVD drives. The SD interface supports synchronous operations, with the SD synchronous read and write timing diagrams shown in Figure 14.3.

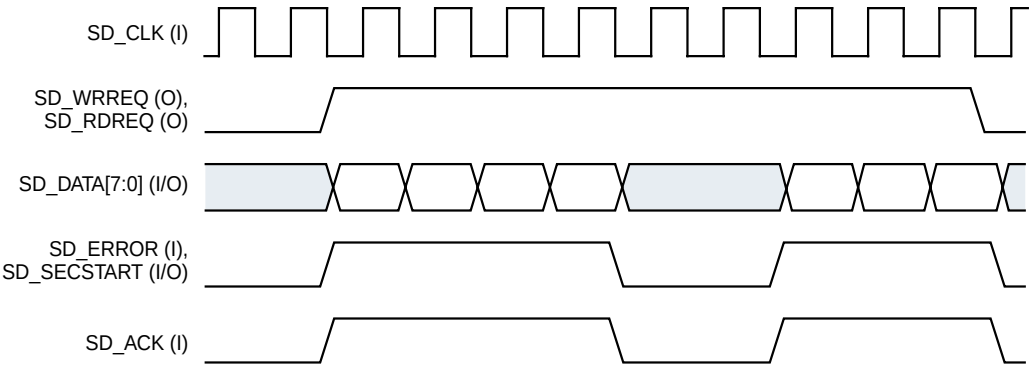
For the read operation, the SD Interface asserts the read request signal SD_RDREQ when it receives the SD_DECS_BS or SD_DESC_NO command from the host. The SD device must respond to the read

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

request by placing the data on the SD_DATA[7:0] bus and asserting the SD_ERROR, SD_SECSTART and SD_ACK status signals. The SD_DATA[7:0] is latched by the SD Interface on the rising edge of SD_CLK when SD_ACK is high. An error occurs when the SD_DATA is latched while SD_ERROR is high.

For the write operation, the SD Interface asserts the write request signal SD_WRREQ when it receives the SD_WR_DATA command from the host. The SD device must respond to the write request by asserting the SD_ACK signal. The SD Interface must place the first write data on the bus when SD_ACK is asserted. The write data is latched by the SD device during the rising edge of the SD_ACK. The SD Interface can place the next write data during the falling edge of the SD_CLK.

Figure 14.3 SD Interface Cycle



14.3 CD Interface

Since the CD Interface supports several CD formats (32-bit, 24-bit, 24-bit IDS, and 16-bit), the host must program the SD/CD Configuration register to configure the CD Interface to the proper CD format. Once all parameters in the configuration register are set, the host programs the READ_CD command and the Go bit in the DVD Control register to enable the CD Interface.

The CD Interface continues to search for the sector sync bytes, once it is enabled and the SrchCnt (sector sync search count) in the SD/CD

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Configuration register is non-zero. The sector sync bytes consist of 12 bytes of data, with the following hexadecimal values: 00, FF, FF, FF, FF, FF, FF, FF, FF, FF, FF, 00. The CD Interface must detect the sector sync consecutively (according to the number specified in the SrchCnt field of the SD/CD Configuration register) in 2352-byte intervals to establish valid sector framing before it can start reading the data from the CD. During data transfer, if the sector sync is not present at the beginning of a 2352-byte sector, the CD Interface must restart the sector sync search, and no CD data is read until sector framing is re-established. If the SrchCnt field in the SD/CD Configuration register is zero, the pass-through (CD-DA) mode is set, and the CD Interface begins reading the CD data without a sector sync search.

14.3.1 SD/CD Configuration Register

This register configures the SD/CD interface. The parallel SD interface can work in synchronous mode. The serial CD interface consists of the following:

- A bit clock pin, CD_BCK, used to sample the data
- A left/right channel clock pin, CD_LRCK, with 16-bit serial data in each phase to frame data at the 32-bit boundary
- A data pin, CD_DATA
- An error pin, CD_C2PO, which indicates errors for each of the 2-byte data

A 2352-byte sector starts from the left channel.

SD/CD Configuration Register

Cbus Address: 0xFB0050

31								23	22	21	20	19	18	17	16
Reserved								SdInt	Atapi Rs	TKDe c	SDWr Re	EnS- eSrt	Mode		Srch- Cnt
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SrchCnt		SDreq	SDack	SDclk	SD Sync	CDG Mode		IDS	No C2po	BCK	LRclk	C2po Ord	DatOrd	Wdlen	

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SdInt **23**

1 = generate interrupt at the beginning of every SD packet (DVD Control register Int bit must also be set). This allows the software to update the DVD Sector Delay register for the next packet; 0= normal mode.

AtapiRs **22**

ATAPI Reset. 1 = Assert ATAPI_RESET pin, 0 = Deassert ATAPI_RESET pin. This happens only when Mode is 0 (ATAPI).

TKDec **21**

Title Key Decryption. 1 = Decrypt the encrypted Title Key on the fly, and use the result for the subsequent descrambling, 0 = Use the previously decrypted Title Key. This bit is used with the ATAPI_DESC_BS, SD_DESC_BS and DESC_SDRAM commands.

SDWrRe **20**

SD Write Request. 0 = An active SD_WRREQ pin indicates a write command (and an active SD_RDREQ pin indicates a read command). 1 = An active SD_WRREQ indicates that there is a valid command, which is a read if SD_RDREQ is active, and a write if SD_RDREQ is not active.

EnSeSrt **19**

1 = Use the SD_SECSTART pin to mark the beginning of a sector, instead of using header search. Valid only for SD mode.

On input, the BIO uses the pin to start reading a sector with length specified in register DVD Sector Length. Between two sector starts, if there is more data than specified in DVD Sector Length, they are ignored; if there is less data than specified in DVD Sector Length, zeroes are padded to fill up the size, and the error masks are set for the padded bytes.

On output, the BIO drives the pin when it sends the first byte.

Mode **[18:17]**

The interface protocol in which the BIO will be working: 0 = ATAPI, 1 = SD, 2 = CD. Software must set this bit

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

properly before the first command is sent. On reset, Mode = 0.

SrchCnt **[16:14]**

The number of times the CD sector sync or DVD Pack Start Code must be located in order to claim a sector is valid, and send it to the SDRAM. A value of 0 means no search is performed.

SDreq **13**

0 = SD_RDREQ/SD_WRREQ pins are active low.

1 = SD_RDREQ/SD_WRREQ pins are active high.

SDack **12**

Used only in synchronous mode: 0 = SD_ACK pin is active low., 1 = SD_ACK pin is active high.

SDclk **11**

0 = SD data is sampled at the falling edge of SD_CLK.

1 = SD data is sampled at the rising edge of SD_CLK.

SD Sync **10**

1 = SD interface is synchronous.

Note: DMN-8600 supports synchronous SD only, regardless of the value of this field. It is still writable.

CDG Mode **[9:8]**

(Obsolete) This field is used when WdLen = 3 (CD-G). Bit 8 indicates either CD-G separate mode (= 0) or CD-G composite mode (= 1) is used. In CD-G separate mode, separate Frame Sync and Block Sync pins are used. In CD-G composite mode, only the Frame Sync pin is used. Bit 1 indicates whether the CdgSclk pin is input (= 0) or output (= 1).

IDS **7**

0 = Data bits are left-adjusted within the left/right channel clock.

1 = Philips mode. The first data bit lags 1 bit clock behind the left/right channel clock edge.

NoC2po **6**

Only used to differentiate the 2 cases where WeLen = 2, DatOrd = 0, Lrck = 1, Bck = 0. (For other cases, this bit

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

is ignored.) When NoC2po = 0, the CD_C2PO pin contains the valid error flag. When NoC2po = 1, CD_C2PO is ignored and no error is assumed.

BCK **5**

0 = Latch data on rising edge of CD_BCK., 1 = Latch data on falling edge of CD_BCK.

LRclk **4**

0 = Left channel active when CD_LRCK pin is low, 1 = Left channel active when CD_LRCK pin is high.

C2poOrd **3**

C2PO error flag ordering: 0 = MSByte first, 1 = LSByte first. Since DMN-8600 uses one bit in the appendix to report errors in each 128-byte block of data, this bit doesn't affect the appendix.

DatOrd **2**

0 = MSbit first (from bit 15 down to bit 0), 1 = LSbit first (from bit 0 up to bit 15).

WdLen **[1:0]**

Number of bitclocks per phase of left-right clock for CD: 0 = 32 bit clocks, 1 = 16 bit clocks, 2 = 24 bit clocks, 3 = CD-G

14.3.2 Valid Signal Formats

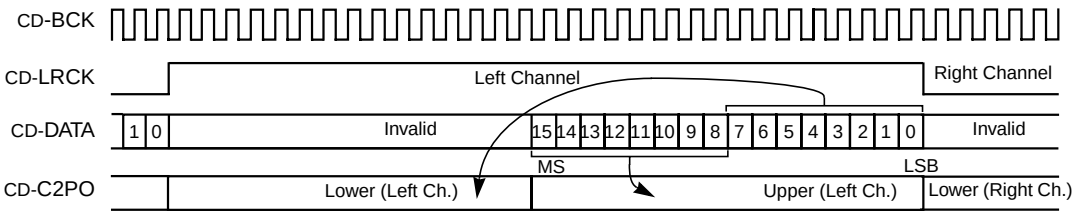
The following signal formats are supported by DMN-8600:

1. IDS = 0, WdLen = 0, DatOrd = 0, Lrck = 1, C2poOrd = 1, Bck = 0
2. IDS = 0, WdLen = 0, DatOrd = 0, Lrck = 0, C2poOrd = 0, Bck = 1
3. IDS = 0, WdLen = 2, DatOrd = 0, Lrck = 1, C2poOrd = 0, Bck = 0, NoC2po = 0
4. IDS = 1, WdLen = 2, DatOrd = 0, Lrck = 0, C2poOrd = 0, Bck = 0 (Philips mode)
5. IDS = 0, WdLen = 2, DatOrd = 1, Lrck = 1, C2poOrd = 0, Bck = 1
6. IDS = 0, WdLen = 2, DatOrd = 0, Lrck = 1, C2poOrd = x, Bck = 0, NoC2po = 1
7. IDS = 0, WdLen = 1, DatOrd = 0, Lrck = 0, C2poOrd = 1, Bck = 0

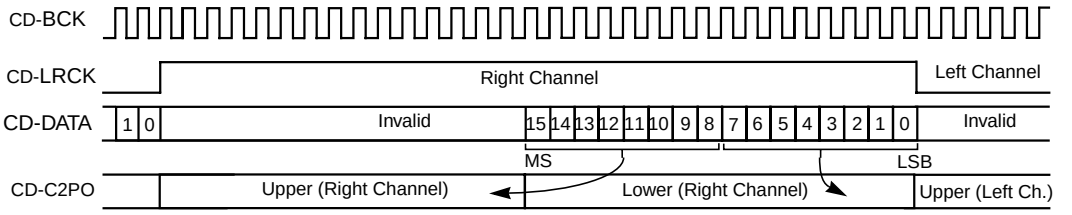
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The numbers in the list correspond to the timing diagrams shown in Figure 14.4 and Figure 14.4, which illustrate the input signal formats of the CD interface's serial modes.

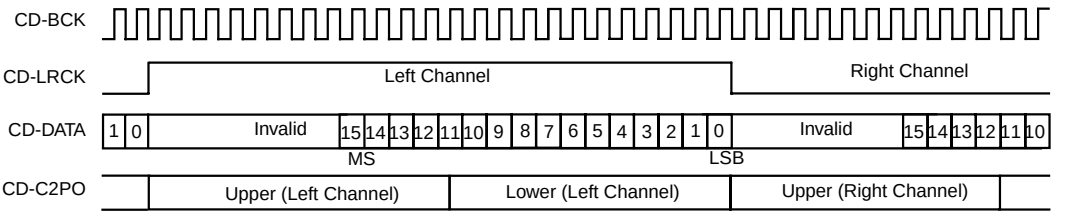
Figure 14.4 CD Interface Input Signal Formats



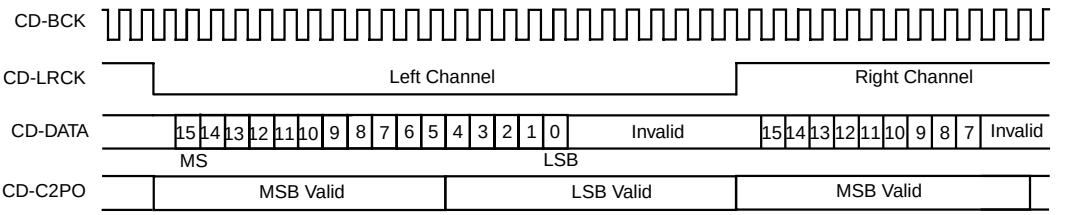
1. 32-bit BCK, MSB First, Right Channel Low, C2PO LSB First, Data Latch Timing High



2. 32-bit BCK, MSB First, Left Channel Low, C2PO MSB First, Data Latch Timing Low



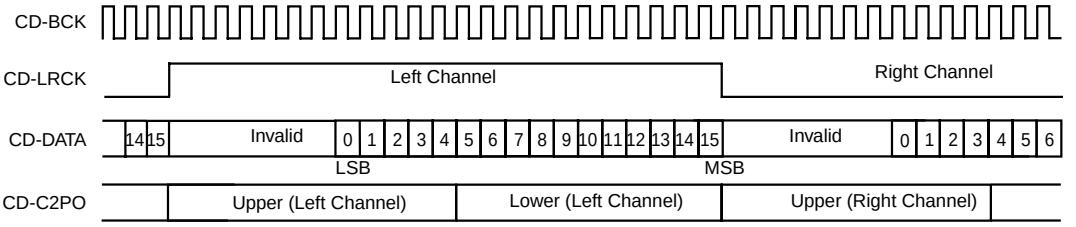
3. 24-bit BCK, MSB First, Right Channel Low, C2PO MSB First, Data Latch Timing High



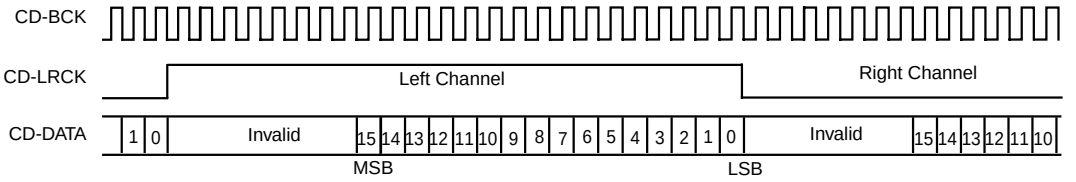
4. 24-bit BCK, MSB First, Left Channel Low, C2PO MSB First, DAta Latch Timing High (IDS)

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

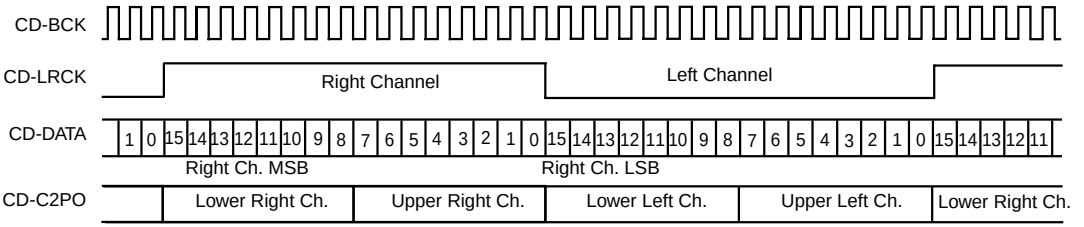
Figure 14.4 CD Interface Input Signal Formats (Cont.)



5. 24-bit BCK, LSB First, Right Channel Low, C2PO MSB First, Data Latch Timing Low



6. 24-bit BCK, MSB First, Right Channel Low, Data latch timing high (Note: no C2PO For This Format)



7. 16-bit BCK, MSB First, Left Channel Low, C2PO LSB First, Data Latch Timing

14.4 1394 Controller

For an MPEG transport stream or for a DV stream data contained in isochronous packets (IPs) received on the 1394 interface, the BIO filters packets by matching channel IDs, and it then performs 1394 descrambling on the scrambled data. In both cases, the BIO sends the raw packets to SDRAM and appends extra information such as an all-zero packet flag, transmitting speed, receive status, and acknowledge code.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

For asynchronous packets (APs) received on the 1394 interface, if the packet is directed to the current DMN-8600 node, the BIO sends the packet to the AP's SDRAM buffer and signals an interrupt at the end of an isochronous cycle. Software is responsible for further input processing, including PCR recovery, PTS handling, DVD navigation and transport section processing, CD descrambling and CRC check, demultiplexing the DVD/CD and transport data in SDRAM, and copying the AV bitstreams to proper VBV buffers during vertical blanking interval (VBI).

For output, DMA channels transfer data from SDRAM to the 1394 and SD/ATAPI interfaces. In the case of 1394 isochronous output, software prepares the proper 1394 time stamp in the packet header and multiplexes packets from different channels. The isochronous packets optionally pass through the 1394 scrambler and channel filter which can turn an output channel on and off instantly.

For 1394 output sending MPEG packets, the BIO supports IPs with one or more source packets, but not with a fraction of a source packet. This simplifies scrambling/descrambling and output timestamp handling.

For 1394 input receiving MPEG packets, the BIO supports IPs with fractions of a source packet as well.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 15
Serial I/O Port

This chapter describes the serial I/O (SIO) port and contains the following sections:

The SIO consists of six peripheral modules: a serial peripheral interface (SPI), an interdevice communications port (IDC), two UARTs, and two infrared interfaces (IR). The following are the major features of the SIO:

- SPI interface with bit granularity transfer support
- Two UART interfaces (one with hardware flow control support)
- Two IR interfaces (one with both IR blaster/IR capture, one with only IR blaster)
- Block data transfer to and from the SDRAM
- DMA transfers, managed by DMA engine/SDRAM double buffers
- Operating range: 81-150 MHz

15.1 SIO IR (Infrared) Interface

The DoMiNo IR modules off load most of the formatting and protocol issues to software; for transmit, the IR module generates waveforms with the programmed pulse length and period characteristics, while for receive, it simply measures the period and duty cycle of incoming pulses.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Since there is no generic IR protocol which can be described, the waveforms shown below illustrate how the values read/written from various IR registers relate to the actual waveforms themselves. (The circled numbers shown in Figure 15.1 correspond to the numbers shown in Table 15.1.)

Waveforms for two common protocols, NCR and Philips RC-5, are shown in the following sections.

Figure 15.1 IR Interface Protocol

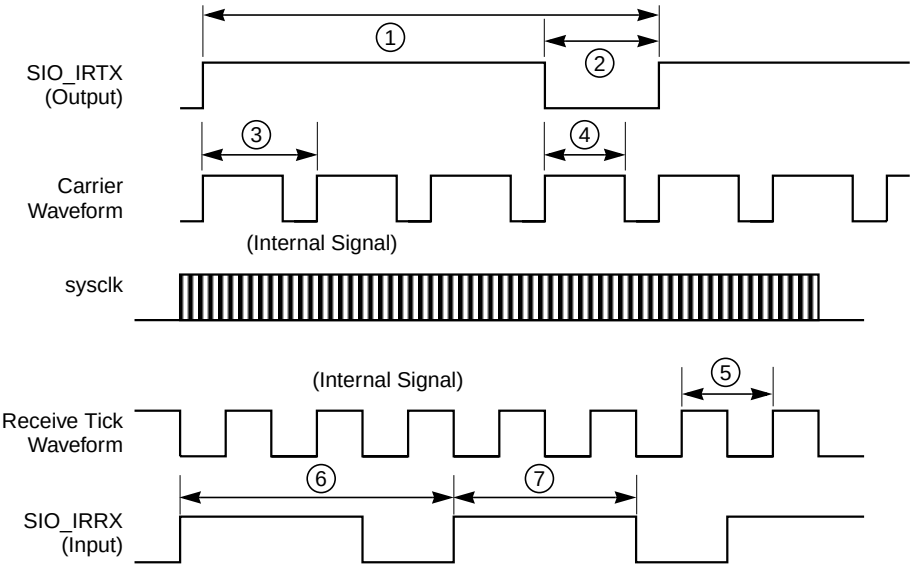


Table 15.1 IR Waveform/Register Value Relationship

Reference	Parameter	Description	Minimum	Maximum
	sysclk	System Clock	81 MHz	150 Mhz
1	MSPR	IRTX Modulated Signal Pulse: User can program overall period of IRTX waveform, in units of carrier wave cycles	0x0 + 1 carrier cycles	0x3ff + 1 carrier cycles
2	MSPL	IRTX Modulated Signal Pulse Low: User can program duty cycle of IRTX waveform, in units of carrier wave cycles	0x0 + 1 carrier cycles	0xff + 1 carrier cycles

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 15.1 IR Waveform/Register Value Relationship

Reference	Parameter	Description	Minimum	Maximum
3	CWP	IRTX Carrier Wave Period: User can program period of carrier waveform, in units of sysclk cycles	0x0 + 1 sysclk cycles	0x3fff + 1 sysck cycles
4	CWPH	IRTX Carrier Wave Period High: User can program duty cycle of carrier waveform, in units of sysclk cycles	0x0 + 1 sysclk cycles	0x1fff + 1 sysclk cycles
5	RTP	IRRX Receive Tick Period: User can program length of an “IR tick”, in terms of sysclk cycles	0x0 + 1 sysclk cycles	0x7fff + 1 sysclk cycles
6	RTC	IRRX Receive Tick Count: IR module counts number of IR ticks between successive falling (or rising) edges of IRRX	0x0 + 1 IR ticks	0xff + 1 IR ticks
7	RPH	IRRX Receive High Tick Count: IR module counts number of IR ticks that IRRX is high	0x0 + 1 IR ticks	0xff + 1 IR ticks

15.1.1 IR Transmit Functionality

The IR interface block controls the output to the IR “blaster” pin SIO_IRTX*. The IR transmit circuit allows the host to program the period and the pulse-high width of the carrier signal. The IR registers allow carriers in the range of 4.58 kHz to 10 MHz with full control of the duty cycle.

The IR interface controls the carrier frequency and the envelope of the modulated pulse. Communication with an external device requires the carrier frequency of the IR interface to match the IR receiver in the target device. The IR interface can then send control messages by generating sequences of pulses with varying envelopes.

The period and duty cycle of the carrier and modulated signal are set via the CWP/CWPH and MSPL/MSPR Registers, respectively.

To program the modulated signal’s pulse shape, the pulse’s period and pulse-low width are programmed with the period programmed last. Bit 26 of the Transmit Period register (IR_MSPR) is the Transmit Enable bit. When programmed to one, this bit indicates that the pulse shape values should be loaded immediately after the completion of the current pulse

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

and used to generate the next pulse. This bit is automatically cleared after the registers and should be reset by the host during the IR transmit interrupt handling routine if any further bits are to be sent. The IR transmit interrupt executes immediately after the Pulse Shape Registers have been loaded, requiring an interrupt latency of one pulse time from the host.

15.1.2 IR Receive Functionality

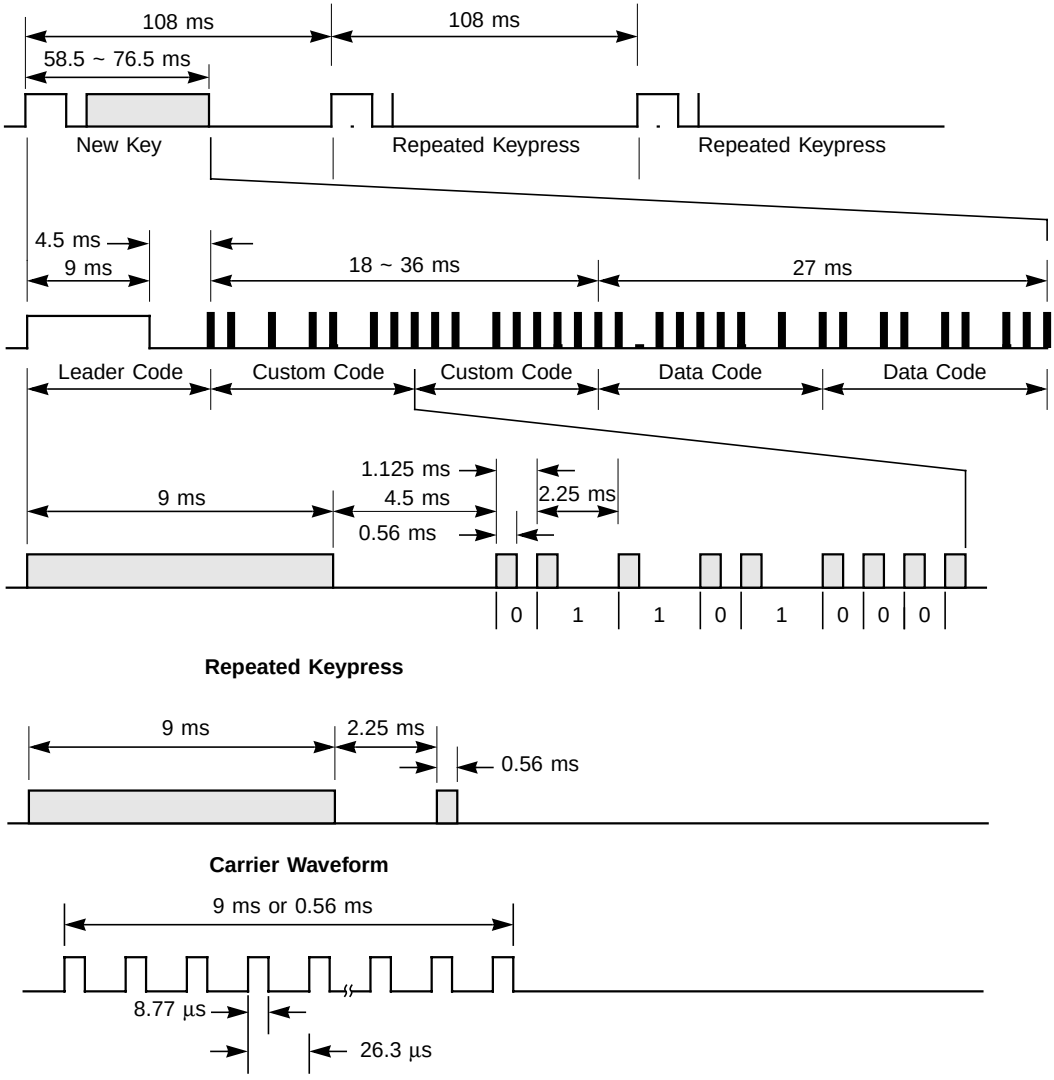
The IR interface block is responsible for interrupting the processor on the receipt of pulses on the SIO_IRRX pin. The IR receive functionality consists of measuring the period and duty cycle of demodulated incoming pulses and generating a processor interrupt for the IR software. The signal period is measured by counting the number of ticks between falling edges of the input signal; the duty cycle is determined by counting the number of ticks during which the incoming signal is high. Note that the tick value is programmable from 1 to 32768 sysclk periods via the IR_RTP register.

Some protocols, such as the NEC protocol, allow for optimized processing of noisy inputs by prepending a start pulse of defined length to all valid transmissions. Other protocols, such as Philips, do not. The IR interface provides the IR_RFR register for detecting valid start pulse lengths between 16 and 240 ticks. Filtering is enabled via the IR_RTC register and is automatically disabled upon receipt of a valid start pulse, so that the pulses that follow are properly processed.

The IR receive interrupt executes after the tick count value has been loaded into the Tick Count register (IR_RTC) on each falling edge of the SIO_IRRX signal. Again, this provides a minimum of 1.125 ms between interrupts for the NCR protocol (shown in Figure 15.2) and 1.778 ms for the RC-5 protocol (shown in Figure 15.3).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 15.2 NCR IR Protocol



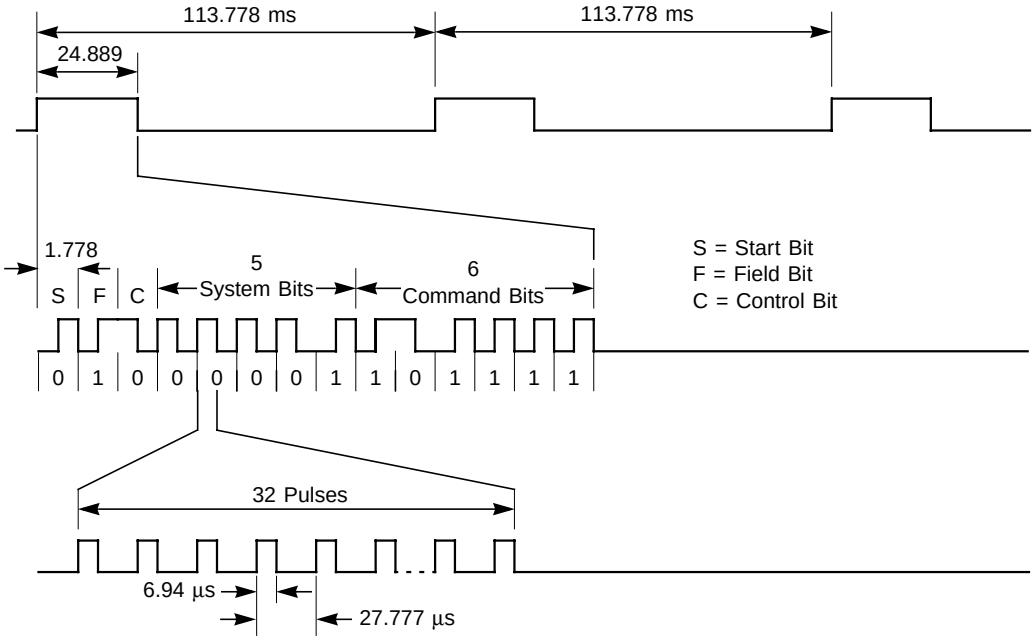
In Figure 15.2, successive line traces show increasing detail in the IRTX waveform (as if zooming in to reveal more detail).

SIO_IRTX and SIO_IRRX are asynchronous. The timing parameter values are under program control, e.g., application-specific.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Note: Demodulated waveforms are expected.

Figure 15.3 Philips RC-5 Protocol



In Figure 15.3, successive line traces show increasing detail in the IRTX waveform (as if zooming in to reveal more detail).

SIO_IRTX and SIO_IRRX are asynchronous. The timing parameter values are under program control, e.g., application-specific.

Demodulated waveforms are expected.

15.1.3 IR Programming Guidelines

Using the DMA Engine with IR requires that data in SDRAM be word-aligned and that it follow a particular format. This is necessary because the DMA Engine actually reads/writes two IR registers at once.

In non-DMA mode, in order to “write” an IR datagram, two registers must be written sequentially: MSPL and MSPR. However, the DMA Engine writes these two registers simultaneously by writing a single 32-bit word with the upper bits of MSPR and MSPL concatenated. The TX data to

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

be sent out must be aligned in SDRAM, as illustrated in Table 15.2 and Table 15.3.

Similarly, in non-DMA mode, in order to “read” an IR datagram, the processor must read two registers: RPH and RTC. Again, here the DMA Engine reads both of these registers simultaneously and places the data in SDRAM according to these tables.

Table 15.2 Alignment of Transmit Data in SDRAM

IR TX Data	Address; Data	Address; Data	Address; Data	Address; Data
1st Chunk	x8000; MSPL[31:24]	x8001; MSPL[23:16]	x8002; MSPR[31:24]	x8003; MSPR[23:16]
2nd Chunk	x8004; MSPL[31:24]	x8005; MSPL[23:16]	x8006; MSPR[31:24]	x8007; MSPR[23:16]
3rd Chunk	x8008; MSPL[31:24]	x8009; MSPL[23:16]	x800a; MSPR[31:24]	x800b; MSPR[23:16]
4th Chunk	x800c; MSPL[31:24]	x800d; MSPL[23:16]	x800e; MSPR[31:24]	x800f; MSPR[23:16]

Table 15.3 Alignment of Receive Data in SDRAM

IR RX Data	Address; Data	Address; Data	Address; Data	Address; Data
1st Chunk	x8800; RPH[31:24]	x8801; RPH[23:16]	x8802; RTC[31:24]	x8803; RTC[23:16]
2nd Chunk	x8804; RPH[31:24]	x8805; RPH[23:16]	x8806; RTC[31:24]	x8807; RTC[23:16]
3rd Chunk	x8808; RPH[31:24]	x8809; RPH[23:16]	x880a; RTC[31:24]	x880b; RTC[23:16]
4th Chunk	x880c; RPH[31:24]	x880d; RPH[23:16]	x880e; RTC[31:24]	x880f; RTC[23:16]

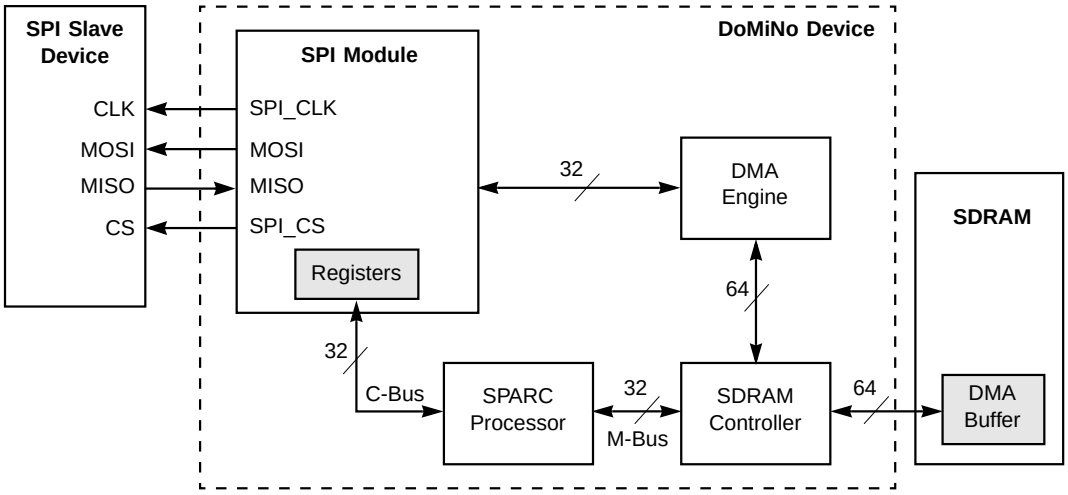
15.2 SIO SPI (Serial Peripheral Interface)

The SPI interface is a generic 3-wire interface suitable for use with the various types of SPI (for example, Sony and Motorola).

A typical SPI interface application is shown in Figure 15.4, where the SPI controller is the bus master.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 15.4 SPI Block Diagram



The SIO_SPI_MOSI and SIO_SPI_MISO lines are used for full-duplex data streams. As one byte of “write” data is serially shifted out the SIO_SPI_MOSI pin, a byte of “read” data is shifted in on SIO_SPI_MISO. Both “read” and “write” data are strobed by the SIO_SPI_CLK signal, and the SIO_SPI_CS line is used as an enable signal to the slave device. When SIO_SPI_CS is asserted, all devices drive data on one edge of SIO_SPI_CLK and latch data on the following opposite edge.

The SPI module can support CPHA=0 and CPHA=1 Motorola SPI transfers, as well as generic 3-wire interfaces that require bit transfers that are not an integer number of bytes. In DMA modes, buffers containing read and write data exist in SDRAM and are accessed via the DMA Engine. The size of the buffer is user-configurable.

The SPI module provides the following functionality:

- Motorola SPI support in master mode for both CPHA=0 and CPHA=1
- Sony SPI support
- SPI clock frequency range from 4.58 kHz to 37.5 MHz (150 MHz system clock)
- Four SIO_SPI_CS[n] signals (16 when used with an external decoder)

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- Programmable inter-byte delay (for use with CPHA=0 modes)
- Programmable serial bit ordering (MS or LS bit shifted first of each byte)
- Programmable polarity of SIO_SPI_CLK and SIO_SPI_CS[n] signals
- Programmable timing of SIO_SPI_CS[n] setup/hold time before start of cycle
- SPI transfer cycles that are not an integer number of bytes
- Blocks of up to 2 Kbytes between interrupts
- Various transfer modes that trade off efficiency vs. application programming complexity
- Polled or interrupt-driven interface
- Bit granularity support: the last “byte” transferred in a cycle can have 1-8 bits.
- DMA Engine Registers for two channels (TX and RX); the SPI uses the DMA Engine to facilitate data transfer to and from SDRAM.

15.2.1 SPI Interface Signals

The SPI module is designed to be a SPI master mode device compatible with the Motorola SPI specification. In addition, it is intended to be a generic 3-wire interface for other 3-wire devices.

The SPI interface consists of the following signals:

SIO_SPI_CS[3:0]: These are the chip select signals to the individual slave devices. Used by themselves, they provide selects to 4 devices. With an external decoder, the number of devices increases to 16.

SIO_SPI_CLK: Data strobe for the interface.

SIO_SPI_MOSI: SPI Master Out Slave In (Serial Data Out). This is the write data (output) port for the SPI interface.

SIO_SPI_MISO: SPI Master In Slave Out (Serial Data In). This is the read data (input) port for the interface.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.2.2 SPI Interface: Protocol Description

DoMiNo always acts as a master in SPI transfers. Figure 15.5 and Figure 15.6 the format of a typical 32-bit SPI data transfer, and the timing relationships between successive bytes transferred. The circled numbers in Figure 15.5 and Figure 15.6 refer to the steps that follow.

Figure 15.5 32-Bit SPI Data Transfer Format

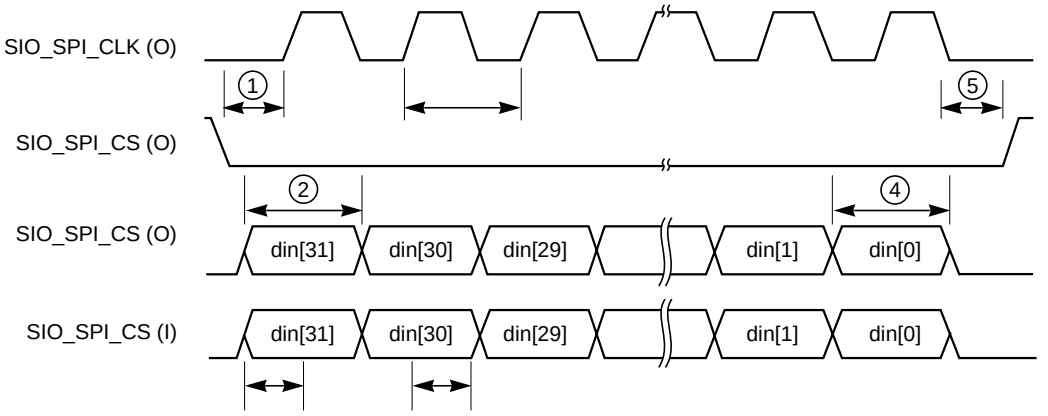
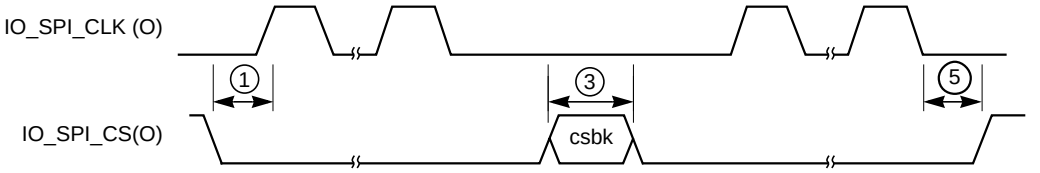


Figure 15.6 Inter-Byte Timing Relationship



1. SIO_SPI_CS asserts: Chip select signal asserts in preparation for data transfer. The CSDL field in the SPI_CONFIG register allows adjustment of the setup and hold times of the SIO_SPI_CS[n] signals about SIO_SPI_CLK. CSDL is the number of half SIO_SPI_CLK cycles between SIO_SPI_CS assertion and SIO_SPI_CLK ungating.
2. First Bit of Data Transfer: Data simultaneously gets shifted out the SIO_SPI_MOSI pin and shifted in the SIO_SPI_MISO pin. Depending on the LSBF setting in the SPI_CONFIG register, this

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

may be either the MSB or the LSB of the value written into SPI_TEMP.

3. Inter-Byte Spacing: SIO_SPI_CLK is gated in-between bytes. The amount of time between successive bytes transferred can be adjusted by programming the CSDL and IBBK fields in the SPI_CONFIG register. Depending on the value of CSBK, SIO_SPI_CS may or may not be asserted during this inter-byte spacing time.
4. Last Bit of Data Transfer: The final bit of the value written into SPI_TEMP is shifted out the SIO_SPI_MOSI pin and is shifted in the SIO_SPI_MISO pin. This may be either the MSB or the LSB, depending on the value of LSBF in the SPI_CONFIG register.
5. SIO_SPI_CS de-asserts: Chip select signal deasserts, data transfer is terminated. The CSDL field again determines the number of 1/2 SIO_SPI_CLK cycles between SIO_SPI_CS negation and SIO_SPI_CLK gating.

15.2.3 SPI Clocking

The SPI module is synchronous to the system clock (sysclk), and all flops in the module are clocked by this system clock. The module uses an internal Clock Enable to generate SIO_SPI_CLK. This enable is programmable by the host, and is used to set the SIO_SPI_CLK frequency at one half the Clock Enable rate.

The SPI module can support a SIO_SPI_CLK rate from 4.58 kHz to 37.5 MHz. Note that most SPI slave devices ignore gaps of time between active bytes on the interface, with the bus operations being paced by active edges of the SIO_SPI_CLK.

The programmable value for SIO_SPI_CLK is given by the following equation:

$$f_{spiclk} = \frac{f_{sysclk}}{2 \times (progdiv + 2)}$$

where *progdiv* is the value written to the programmable divider (HDIV,SPED).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.2.4 SPI Transfer: Host-Polled Mode

In Host-Polled Mode, all read/write data for the SPI cycle is handled by updates of the SPI_TEMP register: either the external host, SPARC processor, or the DMA Engine can update this register.

In non-DMA mode, as the cycle progresses, the host polls for a set HAEN (Host Access Enable) bit indicating that the SPI_TEMP register is ready to be read/write updated. A host write SPI_TEMP clears the HAEN bit, indicating to the SPI module that the host has updated SPI_TEMP by either reading the current read data or by writing the next write data. If HAEN is set after the contents of SPI_TEMP are transferred, the SPI cycle will stall until HAEN becomes clear. If the DMA Engine is being used, no polling is necessary, as the DMA Engine automatically takes care of reading and writing the SPI_TEMP register when needed.

15.2.5 SPI Programming Examples

Typical SPI programming sequences for both DMA and non-DMA operation are listed below. For this example, assume a transfer of 16 bytes out and a receive of 16 bytes in over the SPI pins. For DMA, assume that TX data comes from an allocated SDRAM buffer starting at 0x8000, and that RX data is placed in an allocated SDRAM buffer starting at 0x9000.

1. Wait until the ENAB bit is cleared, indicating that no current SPI cycle is running.

   Note: The SPI_CONFIG register values must not be changed mid-cycle as the values in this register are read while the cycle is running.

2. If the module isn't configured already, configure it as follows:

   a. In the SPI_CONFIG register, configure the SPI cycle characteristics (polarity, SPI_CS[n] setup/hold, inter-byte blanking, and so forth) and clear bits [15:14] of the SPI_CONFIG register.

   b. Set the desired SPI_CLK speed by programming the divider (HDIV, SPED).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

c. Set the CSSL[3:0] bits to select the SPI_CS[3:0] line to enable. If desired, enable the SPI_cycle_done module interrupt in the SIO top-level module interrupt registers.

d. Set BSIZ to the desired number of bytes, minus one, to be transferred. For this example, since the transfer is 16 bytes over the SPI pins, 0xf is programmed into BSIZ.

3. If the DMA engine is to be used, set up SDRAM and the appropriate registers in both the SPI TX and RX channels. For this example, use the following steps:

   a. Place the TX data in the following SDRAM address range: [0x8000 - 0x8018]. This region would have 16 bytes of “real data” (0x8000-0x8010) to be transferred and 8 bytes of dummy data (0x8010-0x8018) which does not get sent out the SPI pins. The 8 bytes of dummy data are needed for the ENAB bit in the SPI_CONTROL to deassert when the SPI cycle has completed.

   b. Set the TX and RX address pointers as follows:

      TX_addrptr1=0x8004,

      TX_addrptr2=0x8018

      RX_addrptr1=0x9000

      RX_addrptr2=0x9010. (The RX data region does not need the eight extra bytes of dummy data.)

   c. Set the RX_control register to 0xd,

   d. Set the TX_control register to 0xd.

      Note: If the number of bytes to be transferred over the pins is not a multiple of four—that is, if SPI_SIZE % 4 != 0, and the DMA Engine is used, the size of the receive buffer must increased slightly, to the next multiple of four. For example, to transfer 18 bytes, set SPI_SIZE to 17, but set the gap between RX_addrptr1 and RX_addrptr2 to 20. (20 is next multiple after 16).

4. Write SPI_TEMP with the initial write data. If using DMA, this data is present at @0x8000; the DMA engine will handle all remaining updates of SPI_TEMP, starting from 0x8004. The order of the bytes sent is always from most significant byte to least. The order of sending the bits in each byte is controlled by the LSBF bit.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

5. Set the SPI ENAB bit to start the SPI cycle. Software may poll this bit to check for cycle completion.

   Note: The ENAB bit is set-only, as it can not be cleared by the host/DMA engine. It is cleared by the SPI module at cycle completion.

6. When the HAEN bit in the SPI_CONTROL register is high, the SPI has finished shifting out the last piece of transmit data or shifting in the newest piece of receive data. In non-DMA mode, SPI_TEMP can now be read, if desired, to obtain the data just shifted in from SIO_SPI_MISO. In DMA mode, SPI_TEMP is automatically read and stored in a FIFO for transfer to SDRAM, since both transmit and receive must be active.

7. At this point, the SPI_TEMP register must be written with the next chunk of data. After this data is written, this sequence of events repeats after HAEN goes active.

   If non-DMA mode is used, the host must poll HAEN and explicitly update the SPI_TEMP register with new data (and thus clear the HAEN bit) BSIZ[10:2] + 2 times. These two extra 32-bit writes are the equivalent of the eight “dummy data” bytes described in step 3a, and must be done before ENAB goes low and the spi_cycle_done interrupt signals.

   If DMA mode is used, these updates are handled automatically by the DMA engine, and no polling needs to be done.

   For a 16-byte example, in non-DMA mode, the user must explicitly write SPI_TEMP six times: first, the initial 4-byte write in step 3 (data 1-4), then three subsequent 4-byte writes (data 5-8, 9-12, 13-16) then two additional times with dummy data. If DMA mode is used, the user only needs to write SPI_TEMP explicitly once, with the initial data. The first byte read back during the cycle is not valid data and should be discarded.

   Note: If the last “data word” to be transferred has < 32 bits—that is, bit granularity is used or only 1-3 bytes of the final SPI_TEMP write need to be transferred, then the corresponding “word” received from SIO_SPI_MISO needs special consideration. Software must mask the value read from SPI_TEMP to get the real data bytes that are valid, since reading this register (either explicitly, or having the

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

DMA engine do it) returns 32-bits of data, only some of which are valid. For example, if the last “word” has only one “real byte”, then data is present in SPI_TEMP[7:0]. If the last “word” has two bytes, then data is present in SPI_TEMP[15:0], and so on. This consideration is required in both DMA and non-DMA modes. If bit granularity is used, the user may also need to have software mask out bits within a particular byte to get the valid data.

8. After the BSIZ+1 bytes are written or read, the cycle terminates. The ENAB bit is cleared and a SPI_cycle_done interrupt is sent to the host.

15.2.6 Other Applications

The SPI module is very flexible. Various “ad hoc” modes of operation can be used beyond the ones described above, such as generic 3-wire interfaces; for example:

- Write only N bits to a device by using the SPI_BITGRAN register
- Sony SPI interface by setting the LSBF bit in the SPI_CONFIG register.
- Motorola CPHA=0 mode by using the IBBK, CSBK, and CKBK inter-byte blanking fields in the SPI_CONFIG register.
- General Instruments SPI interface by using inter-byte blanking and setting the GIMD bit in the SPI_CONFIG register.

See the register descriptions for more information.

15.2.7 SPI Programming Guidelines

The SPI module should be programmed in host-polled mode. Host-polled mode allows full duplex, and the functional characteristics of the SIO_SPI_CLK and SIO_SPI_CS[n] signals (polarity, SIO_SPI_CS[n] setup/hold times, inter-byte blanking time, etc.) are programmable. SIO_SPI_CLK runs only during a SPI transfer cycle. When a cycle is not occurring, SIO_SPI_CLK is at a static level determined by the CPOL bit. During a SPI cycle, data is transferred only when SPI_CS[n] is asserted and SIO_SPI_CLK is active.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

In host-polled mode, all read/write data is taken care of via update of the SPI_TEMP register by either the host, SPARC processor, or the DMA Engine. The HAEN bit in the SPI_CONTROL register is used in this mode as a semaphore for host updates. The SPI bus will stall if the host has not updated the SPI_TEMP register before the 4 bytes in the SPI_TEMP register have been transmitted.

The DMA Engine of the Bus Bridge is required for programming the SPI module for DMA.

When programming the SPI module for DMA, note that the DoMiNo device differs from previous AViA@TV devices, such as the CL9311 and CL9315.

Setting up SPI registers for DMA is similar to programming a host-polled mode style transfer, with two additional requirements:

- Both the TX and RX DMA channels must always be programmed (no true unidirectional transfer).
- The TX size must equal the RX size plus eight bytes.l

The reason for these requirements is that in a host-polled mode transfer, the SPI_TEMP register is both written and read with data in a “ping-pong” fashion. First, SPI_TEMP is written; as its bits are shifted out on the pins, input data bits are shifted in. Once the entire word has been sent out, SPI_TEMP can be read.

If the user wants to perform a unidirectional transfer (TX / RX), a dummy buffer for the other channel (RX / TX) must be allocated, and the channel must be enabled/serviced.

One valid example of matching TX/RX data sizes is a single TX buffer of 72 bytes, and 3 RX buffers of 8, 32, and 24 bytes spaced apart across the SDRAM memory space (TX: x8000-x8048, RX: x8100-x8108, x8110-x8130, x8150-x8168): the amount of data in the TX buffers must be eight bytes more than is in the RX buffers (72 bytes TX, 64 bytes RX).

15.3 IDC Interface

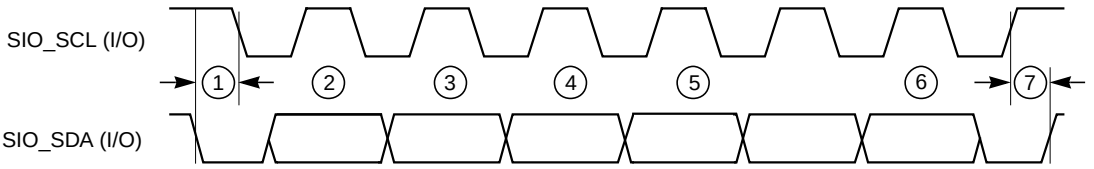
The SIO_SDA (IDC Data Bus) and SIO_SCL (IDC Clock) signals are used to interface the DMN-8600 to other devices. The IDC has DMA

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

registers for two channels (TX and RX). The DMA Engine takes care of filling data into (TX) and draining data out of (RX) the IDC's internal FIFOs.

IDC devices can act as either master or slave; masters initiate and terminate a data transfer, as well as generate the clock signal (SIO_SCL). Figure 15.7 illustrates the format of a typical IDC data transfer. The circled numbers in the figure refer to the steps that follow.

Figure 15.7 IDC Interface Data Transfer Protocol



1. START Condition: Master initiates transaction by making a HIGH to LOW transition on SIO_SDA while SIO_SCL is high.
2. Slave Address Bits 1-7: Master sends out Slave Address, MSB first, on SIO_SDA. Data on SIO_SDA must be stable when SIO_SCL is HIGH; only when SIO_SCL is LOW can SIO_SDA change.
3. Read/Write Bit: This bit determines direction of the data transfer. If SIO_SDA is HIGH, then the master will read from the slave (master-receiver, slave-transmitter). If SIO_SDA is LOW, then the master will write to the slave (master-transmitter, slave-receiver).
4. Master Acknowledge Bit: Receiver pulls SIO_SDA LOW while SIO_SCL is HIGH to acknowledge receipt of address.
5. Data Bits 1–8: Data is sent out, MSB first, on SIO_SDA. Data on SIO_SDA must be stable when SIO_SCL is HIGH; only when SIO_SCL is LOW can SIO_SDA change.
6. Acknowledge Bit: Receiver pulls SIO_SDA LOW while SIO_SCL is HIGH to acknowledge receipt of data.
7. STOP Condition: Master terminates transaction by making a LOW-to-HIGH transition on SIO_SDA while SIO_SCL is HIGH.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.4 SIO UART Interface

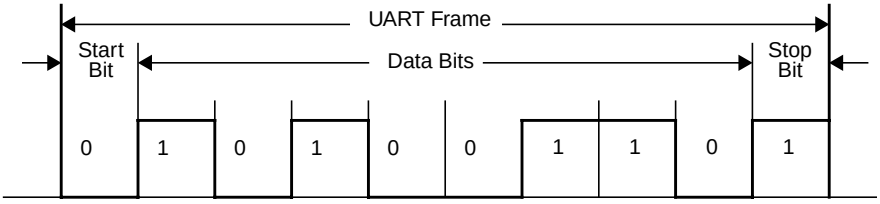
DoMiNo has two 16550-compatible UARTs. The UARTs have the following features:

- Communication speed is up to 256 Kbaud
- An internal FIFO allows both transmitter and receiver to hold up to 16 bytes of data
- Hardware flow control (UART1 only)
- Prescaler
- DMA registers for four channels (TX and RX for each UART instance); the DMA Engine takes care of filling data into (TX) and draining data out of (RX) the UART's internal FIFOs.

15.4.1 UART Data Frame

Figure 15.8 shows the composition of the basic UART data frame: one start bit, eight data bits, one stop bit.

Figure 15.8 UART Data Frame



15.4.2 Baud Rate Generator

The baud rate generator implements two 8-bit registers called Divisor Latch Registers (DLM and DLL) that can be write-only accessed by the internal SPARC processor or the external host processor. The processor can load 16-bit data into DLM and DLL (DLM holds the eight most significant bits and DLL holds the eight least significant bits) to obtain an output (nbdout) from the baud rate generator that is 16 times the desired baud rate. This block implements a 16-bit counter to generate the nbdout from the input clock. The Divisor value can be calculated as follows:

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

$$Divisor = \frac{InputClock}{16(BaudRate)}$$

Where Input Clock is the SYSCLK divided by two plus the Prescale value stored in the UARTn External Clock / Prescaler Register (UARTn_PRESCALER), that is—

$$InputClock = \frac{SYSCLK}{2 + DIV}$$

15.5 SIO Register Descriptions

This section describes the registers of the Serial Input/Output (SIO) module in the DoMiNo chip. The SIO consists of the following modules:

- One Serial Peripheral Interface (SPI) module
- One Inter-Device Communications (IDC) module
- Two UART modules
- Two infrared interface (IR) modules

In this section the register addresses are shown as offsets from the module base address. The module base address of all SIO registers is 0xBE0000.

The module base address itself is an offset from the Control Space base address within the 16MB register address space selected by the host, as explained in the Product Overview / General Description / Memory Mapping section of this manual. Therefore, the physical address of a specific register is the sum of the Control Space Base Address, the module offset, and the register offset.

All SIO registers are readable and writable, unless mentioned otherwise. All registers are implemented as 32-bit registers so that they fall on word (32-bit) boundaries. They must be accessed through 32-bit CBus loads and stores only.

Note: Reserved (Rsvd) bits must always be written as 0 and are undefined when read.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The SIO Reset register allows users to “software reset” each of the peripheral modules.

15.5.1 Interrupt Hierarchy

The SIO module presents a single signal to the SPARC processor to indicate that an interrupt is pending and needs to be serviced. This signal is generated by OR-ing the register bits of both the SIO Top Level Module Interrupt Status Register (SIO_IRQ_STATUS) and the SIO Top Level DMA Interrupt Status Register (SIO_DMA_IRQ).

The SIO Top Level Module Interrupt Status Register (SIO_IRQ_STATUS) stores the state of each peripheral module’s interrupt output pins. When servicing an interrupt, the appropriate module’s register must be serviced before clearing SIO_IRQ_STATUS. Otherwise, if the module generates a level interrupt (interrupt output pin stays high until serviced) and SIO_IRQ_STATUS is cleared, on the following cycle, the appropriate bit in SIO_IRQ_STATUS is set again. This indicates that a new interrupt is now pending when it really is not.

The SIO Top Level DMA Interrupt Status Register (SIO_DMA_IRQ) stores the bits of the SIO DMA Engine Interrupt Status Register (INTR_STATUS_ADDR). It is this register, INTR_STATUS_ADDR, which actually stores the interrupt output pins of all of the various SIO DMA channels. In contrast to SIO_IRQ_STATUS, SIO_DMA_IRQ is two levels away from each of the DMA channel’s interrupt output pins. When servicing an interrupt, INTR_STATUS_ADDR must be cleared before clearing SIO_DMA_IRQ. Otherwise, on the following cycle, the appropriate bit in SIO_DMA_IRQ is still set. This indicates that a new interrupt is now pending when it really is not.

By writing the Module Interrupt Enable/DMA Interrupt Mask registers, the user can control which interrupts are latched in the Module Interrupt Status/DMA Interrupt Status registers, respectively. The register bits of the Module Interrupt Status register and the DMA Interrupt Status register are OR’d together to signal to the SPARC processor that the SIO has an interrupt pending.

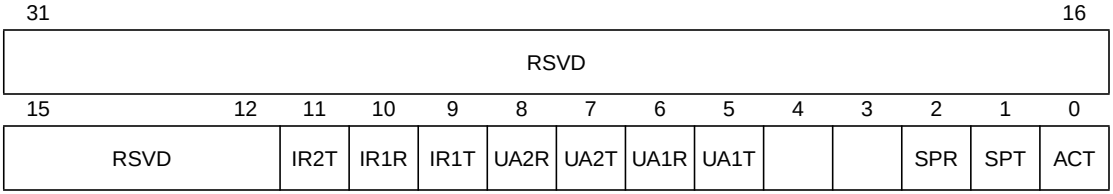
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.5.2 SIO Top Level & DMA Engine Registers

The registers described below handle DMA interrupt/reset issues associated with each of the peripheral modules: SPI, the two UARTs, and the two IRs.

SIO DMA Engine Interrupt Mask Register (INTR_MASK_ADDR)

Offset = 0xBF0100 Read/Write Default = 0x0000 0000



Each DMA channel has one bit to mask interrupts that are recorded in the SIO DMA Engine Interrupt Status register.

CH_MASK **Channel Interrupt Mask** **11:1**

Each DMA channel has one mask bit associated with it, as shown below; that is, bits in the INTR_MASK_ADDR register correspond to similarly named bits in the INTR_STATUS_ADDR register.

1 = Any interrupts (of type 1 or 2) that occur are recorded in the channel's bits in the SIO DMA Engine Interrupt Status register.

0 = The channel's interrupts are ignored, and the corresponding bit value in the SIO DMA Engine Interrupt Status register does not change (that is, if the DMA operation is complete, no interrupt is generated).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR2T	**IR2 Transmit Mask**	**11**
IR1R	**IR1 Receive Mask**	**10**
IR1T	**IR1 Transmit Mask**	**9**
UA2R	**UART2 Receive Mask**	**8**
UA2T	**UART2 Transmit Mask**	**7**
UA1R	**UART1 Receive Mask**	**6**
UA1T	**UART1 Transmit Mask**	**5**
SPR	**SPI Receive Mask**	**2**
SPT	**SPI Transmit Mask**	**1**
ACT	**Action Bit**	**0**

This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

This bit always reads back as 0.

SIO DMA Engine Interrupt Status Register (INTR_STATUS_ADDR)

Offset = 0xBF0104 Read/Write Default = 0x0000 0000

31								23	22	21	20	19	18	17	16
RSVD									R2T2	R2T1	R1R2	R1R1	R1T2	R1T1	U2R2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U2R1	U2T2	U2T1	U1R2	U1R1	U1T2	U1T1					SPR2	SPR1	SPT2	SPT1	ACT

Each DMA channel has 2 bits, one for each of the possible interrupts (ch_intr1 or ch_intr2) for that DMA channel.

CH_INTR1/2 **Channel Interrupt 1/2** **22:1**

Each DMA channel has two interrupt bits associated with it:

Bit 1 indicates completion of the entire DMA operation—that is, no DMA buffers are pending.
1 = Completed

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Bit 2 is for indicating when each programmed double buffer (set in Addr_Ptr1/2, propagated to Addr_Ptr3/4) has completed:
1 = Completed

On transmit: All data in programmed memory space has drained out of the channel FIFO to the module;

On receive: Sufficient data to fill memory space has been collected from the module.

R2T2	**IR2 Transmit Interrupt 2 Status**	**22**
R2T1	**IR2 Transmit Interrupt 1 Status**	**21**
R1R2	**IR1 Receive Interrupt 2 Status**	**20**
R1R1	**IR1 Receive Interrupt 1 Status**	**19**
R1T2	**IR1 Transmit Interrupt 2 Status**	**18**
R1T1	**IR1 Transmit Interrupt 1 Status**	**17**
U2R2	**UART2 Receive Interrupt 2 Status**	**16**
U2R1	**UART2 Receive Interrupt 1 Status**	**15**
U2T2	**UART2 Transmit Interrupt 2 Status**	**14**
U2T1	**UART2 Transmit Interrupt 1 Status**	**13**
U1R2	**UART1 Receive Interrupt 2 Status**	**12**
U1R1	**UART1 Receive Interrupt 1 Status**	**11**
U1T2	**UART1 Transmit Interrupt 2 Status**	**10**
U1T1	**UART1 Transmit Interrupt 1 Status**	**9**
SPR2	**SPI Receive Interrupt 2 Status**	**4**
SPR1	**SPI Receive Interrupt 1 Status**	**3**
SPT2	**SPI Transmit Interrupt 2 Status**	**2**
SPT1	**SPI Transmit Interrupt 1 Status**	**1**
ACT	**Action Bit** This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written	**0**

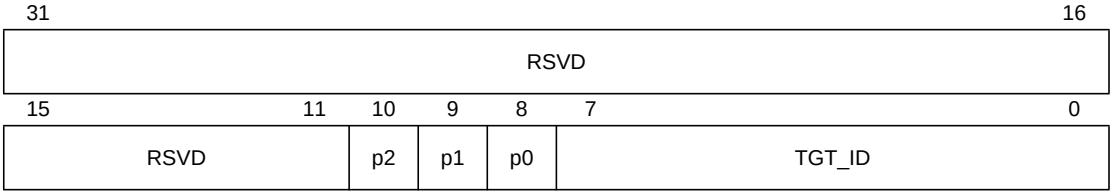
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

This bit always reads back as 0.

SIO DMA Engine Priority Selection Register (PRI_SEL_ADDR)

Offset = 0xBF0108 Read/Write Default = 0x0000 0000



The DMA Engine provides a facility to fix or alter the priority of the DMA channels requesting a transaction with SDRAM.

P2, P1, P0 **Priority Mode** **10:8**
These bits set the priority mode as shown in Table 15.4.

Table 15.4 Priority Mode Select

P2 [10]	P1[9]	P0[8]	Operation
0	0	0	No Operation
0	0	1	Programmable Priority
0	1	X	Rotating Priority
1	X	X	Fixed Priority

Programmable Priority

This setting ([P2:P0])=001) is used to alter the priority of the DMA channels, in case a particular target channel requires the highest priority, whatever the arbitration scheme that is selected. The programming of these bits overrides the default behavior.

The highest priority is assigned to the target module channel whose ID is programmed in TGT_ID.

Rotating Priority

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

This setting ([P2:P0])=01x) applies a rotating priority to the DMA Channels. In this mode, the DMA channel that is currently being serviced is assigned the lowest priority, and the immediate next channel is assigned the highest priority.

Fixed Priority

In fixed priority mode ([P2:P0])=1xx), the target module with the lowest ID number is always given the highest priority.

Initially, DMA channel 1 (SPI transmit, SPI_TX) has the highest priority, and DMA channel 11 (infrared “blaster” 2, IR2_TX) has the lowest (Table 15.5 explains the channel numbers). This can be changed to programmable priority, fixed priority, or rotating priority.

TGT_ID **Target Module ID** **7:0**

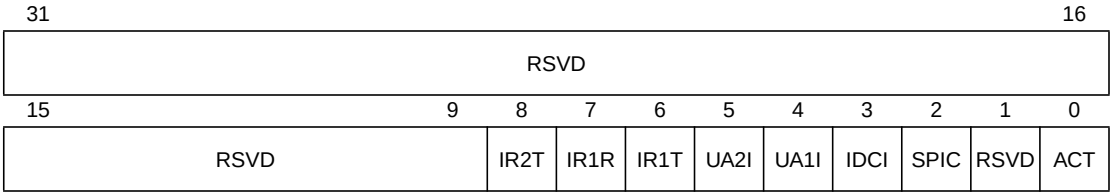
Legal values for the target ID are shown in Table 15.5.

Table 15.5 Legal Target ID Values

SPI_TX = 0x1	UART1_TX = 0x5	IR1_TX = 0x9
SPI_RX = 0x2	UART1_RX = 0x6	IR1_RX = 0xA
	UART2_TX = 0x7	IR2_TX = 0xB
	UART2_RX = 0x8	

SIO Top Level Module Interrupt Status Register (SIO_IRQ_STATUS)

Offset = 0xBF0140 Read/Write Default = 0x0000 0000



This register, depending on the value of the corresponding bit in the SIO Top Level Module Interrupt Enable register, records each of the interrupt signals generated by the various peripheral modules in the SIO.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Bits [8:1] of the SIO Top Level Module Interrupt Status register are ORed together with bits [11:1] of the SIO Top Level DMA Interrupt Status register to create a single interrupt signal, which is sent to the SPARC processor. The SPARC processor must read both of these Interrupt Status registers to determine which module/DMA channel is the source of the interrupt, and act accordingly.

Note: Each of the following Interrupt Status bits signals whether the corresponding SIO module (the module indicated in the bit's name) has an interrupt waiting to be serviced.
1 = Waiting
0 = Done

Note: Before clearing an Interrupt Status bit, software must resolve the corresponding module's interrupt, as described in the particular module's section in this chapter. Note that the procedure to resolve the interrupt varies from module to module. Refer to Section 15.5.1, "Interrupt Hierarchy," page 15-20 for more details.

IR2T **IR2 Transmit Interrupt Status** **8**

IR1R **IR1 Receive Interrupt Status** **7**

IR1T **IR1 Transmit Interrupt Status** **6**

UA2I **UART2 Interrupt Status** **5**

UA1I **UART1 Interrupt Status** **4**

IDCI **IDC Interrupt Status** **3**

SPIC **SPI Cycle Done Interrupt Status** **2**

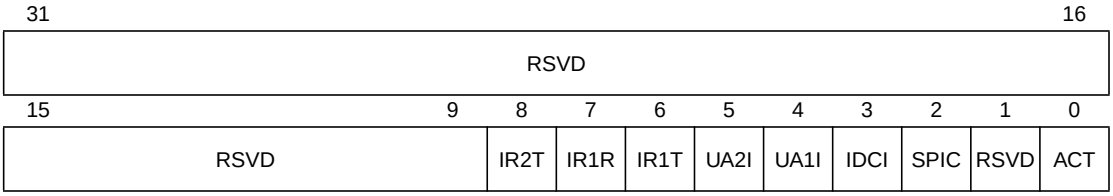
ACT **Action Bit** **0**
This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

This bit always reads back as 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SIO Top Level Module Interrupt Enable Register (SIO_IRQ_ENABLE)

Offset = 0xBF0144 Read/Write Default = 0x0000 0000



This register controls which SIO modules can have their interrupts recorded in the SIO Top Level Module Interrupt Status register.

Interrupt Enable **8:1**

1 = The corresponding bit in the SIO Top Level Module Interrupt Status register records the interrupts signaled by the target module.

0 = The corresponding interrupt is ignored.

IR2T **IR2 Transmit Interrupt Enable** **8**

IR1R **IR1 Receive Interrupt Enable** **7**

IR1T **IR1 Transmit Interrupt Enable** **6**

UA2I **UART2 Interrupt Enable** **5**

UA1I **UART1 Interrupt Enable** **4**

IDCI **IDC Interrupt Enable** **3**

SPIC **SPI Cycle Done Interrupt Enable** **2**

ACT **Action Bit** **0**

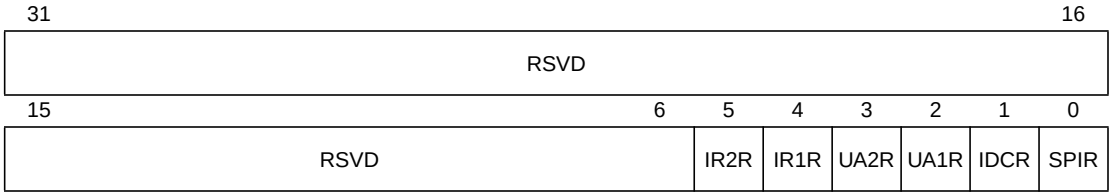
This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

This bit always reads back as 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SIO Top Level Reset Register (SIO_RESET)

Offset = 0xBF0148 Read/Write Default = 0x0000 0000



This register allows the user to “software-reset” each of the SIO target modules by explicitly setting then clearing the corresponding bit in this register. Modules must be explicitly reset to 1, then released to 0 before use.

Reset **5:0**

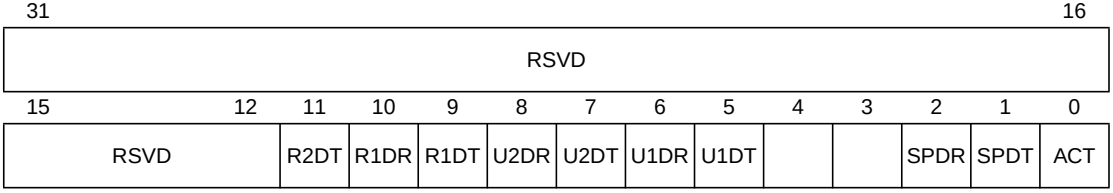
Each of the bits shown below is tied to the reset pin of the corresponding module.

1 = Reset

IR2R	**IR2 Reset**	**5**
IR1R	**IR1 Reset**	**4**
UA2R	**UART2 Reset**	**3**
UA1R	**UART1 Reset**	**2**
IDCR	**IDC Reset**	**1**
SPIR	**SPI Reset**	**0**

SIO Top Level DMA Interrupt Status Register (SIO_DMA_IRQ)

Offset = 0xBF014C Read/Write Default = 0x0000 0000



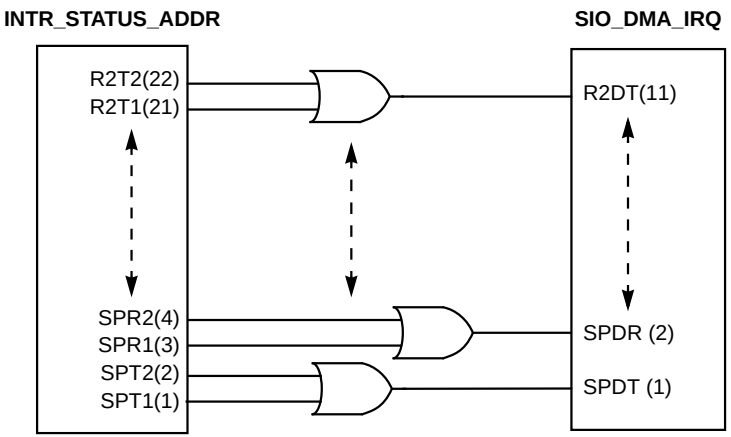
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

This register, depending on the value of the corresponding bit in the SIO Top Level DMA Interrupt Mask register, registers each of the interrupt signals generated by the SIO's DMA channels.

Bits [11:1] of the SIO Top Level Module Interrupt Status register are ORed together with bits [8:1] of the SIO Top Level DMA Interrupt Status register to create a single interrupt signal which is sent to the SPARC processor. The SPARC processor must read both of these Interrupt Status registers to determine which module/DMA channel is the source of the interrupt, and act accordingly.

Since the bits in this register are simply recorded versions of the OR-ing of the DMA channel interrupt bits (that is, bits [2:1] of INTR_STATUS_ADDR OR-ed and recorded in bit 1, bits [4:3] of INTR_STATUS_ADDR OR-ed and recorded in bit 2, etc.), the user must service the interrupt at the DMA channel level before explicitly clearing the bits in this register. Figure 15.9 shows these connections.

Figure 15.9 Connections Between INTR_STATUS_ADDR and SIO_DMA_IRQ Registers



Interrupt Status **11:1**

Note: Each of the Interrupt Status bits shown below signals that the corresponding DMA channel has an interrupt waiting to be serviced:
1 = Waiting
0 = Done

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The DMA Engine's Interrupt Status register must be read to determine which of two events the interrupt is signaling:

One of the double buffers has been filled/drained.

The entire DMA transfer is complete.

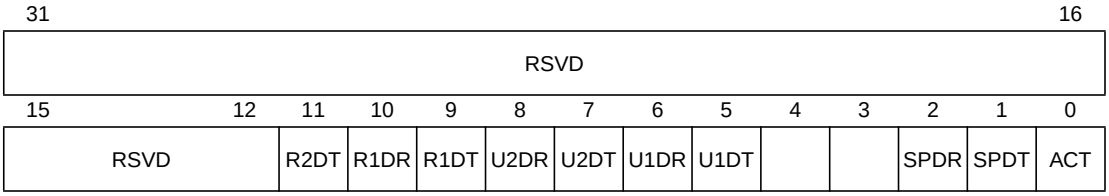
Note: Before an Interrupt Status bit can be cleared, the appropriate bits (that is, the bits corresponding to the interrupt that the software is processing currently) in the DMA Engine Interrupt Status register must be cleared (by software). Refer to Section 15.5.1, "Interrupt Hierarchy," page 15-20 for more details.

R2DT	**IR2 DMA Transmit Interrupt Status**	**11**
R1DR	**IR1 DMA Receive Interrupt Status**	**10**
R1DT	**IR1 DMA Transmit Interrupt Status**	**9**
U2DR	**UART2 DMA Receive Interrupt Status**	**8**
U2DT	**UART2 DMA Transmit Interrupt Status**	**7**
U1DR	**UART1 DMA Receive Interrupt Status**	**6**
U1DT	**UART1 DMA Transmit Interrupt Status**	**5**
SPDR	**SPI DMA Receive Interrupt Status**	**2**
SPDT	**SPI DMA Transmit Interrupt Status**	**1**
ACT	**Action Bit** This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged. This bit always reads back as 0.	**0**

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SIO Top Level DMA Interrupt Mask Register (SIO_DMA_MASK)

Offset = 0xBF0150 Read/Write Default = 0x0000 0000



This register controls which DMA channels can have their interrupts recorded in the SIO Top Level DMA Interrupt Status register.

INT_MASK **Interrupt Mask** **11:1**

The interrupt mask bits in this register work as follows:

1 = The corresponding channel's interrupts are ignored.

0 = The corresponding channel's interrupts are recorded in the SIO Top Level DMA Interrupt Status register.

R2DT **IR2 DMA Transmit Interrupt Mask** **11**

R1DR **IR1 DMA Receive Interrupt Mask** **10**

R1DT **IR1 DMA Transmit Interrupt Mask** **9**

U2DR **UART2 DMA Receive Interrupt Mask** **8**

U2DT **UART2 DMA Transmit Interrupt Mask** **7**

U1DR **UART1 DMA Receive Interrupt Mask** **6**

U1DT **UART1 DMA Transmit Interrupt Mask** **5**

SPDR **SPI DMA Receive Interrupt Mask** **2**

SPDT **SPI DMA Transmit Interrupt Mask** **1**

ACT **Action Bit** **0**

This bit is an action bit. The value of this bit is written to any selected bits during register writes. Bits to be written are selected by placing a 1 in each desired location as the register is written. Any bit positions that contain a 0 during register writes remain unchanged.

This bit always reads back as 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.5.3 SIO IDC Registers

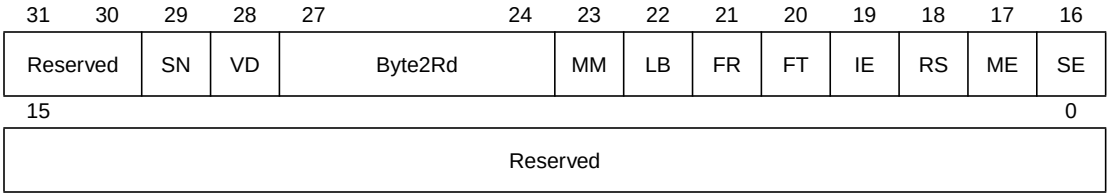
DoMiNo supports both read and write in master and slave mode on the IDC interface. The data rate is 100 kHz to 400 kHz.

Ten different registers control the IDC operation, and 12 registers are devoted to DMA management (6 for TX, 6 for RX).

Note: RSVD (Reserved) bits must always be written as 0 and are undefined when read.

IDC Control Register 1 (IDC_CONTROL1)

Offset = 0xBE0080 Read/Write Default = 0x0000 0000



SN **Send No-Acknowledge in Slave Receiver Mode 29**
1 = A Nack is generated after the current byte is received in slave receiver mode.

0 = A stop, start, or repeat start condition is detected. This bit reads back as 0.

VD **Byte2Rd field has valid data 28**
1 = Transfer data from bits [27:23] to the internal state machine.

This bit clears itself and reads back as 0.

Byte2Rd **# of Bytes to read when Master Receiver [27:24]**
This field tells the internal state machine how many bytes to read. The VD bit must be 1 to validate the data.

These bits read back as 0.

MM **Master Mode 23**
1 = The transaction is a master read from the external device.

0 = The transaction is a master write to the external device.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

LB **Last Byte** **22**
1 = A stop or repeat start condition is generated after the current operation completes (all bytes in the TX FIFO are transferred in master transmitter mode, or all bytes received in the master receiver mode).

Software can set this bit before or after all bytes are received or transmitted. However, if this bit is not set when data transfer is complete, the IDC bus is held until either the bit is set or more data is transferred.

This bit is used only in master mode.

FR **Flush RX Data Fifo** **21**
1 = Reset the receive data FIFO.

This bit clears itself and reads back as 0.

FT **Flush TX Data Fifo** **20**
1 = Reset the transmit data FIFO. This bit clears itself and reads back as 0.

IE **Interrupt Enable** **19**
1 = Interrupts will be generated.

RS **Repeat Start** **18**
1 = Generate a Repeat Start at the end of the current master mode operation (all bytes transmitted or all bytes received).

ME **Master Enable** **17**
1 = Master mode reads/writes are enabled. Setting this bit triggers the state machine to start a master mode operation.

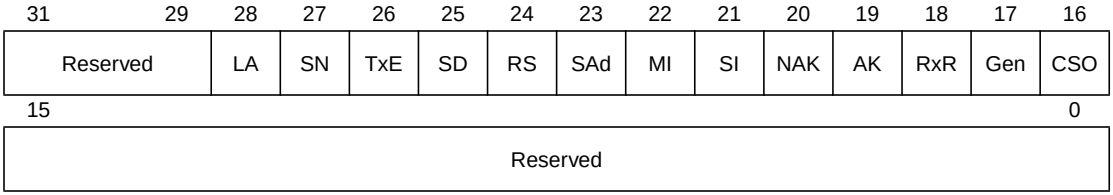
SE **Slave Enable** **16**
1 = Slave receive mode is enabled.

Note: IDC master and slave modes are not mutually exclusive. In master mode, DMN-8600 can initiate transfers. In slave mode, it responds to transfers.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC Status Register (IDC_STATUS1)

Offset = 0xBE0084 Read Only Default = 0x0000 0000



This register is cleared when read. During the IDC interrupt service routine, the host must read this register. If this register is not read, subsequent IDC events cannot set it. Table 15.6 lists the conditions that affect this register.

Table 15.6 Status Register Events

Mode	Condition
Master-Transmitter	Tx FIFO is empty and the LB bit of IDC_Control1 is clear.
	Tx FIFO is empty, the LB bit of IDC_Control1 is set, and the STOP condition has been generated.
	No acknowledge was received after transmitting the slave address on the bus.
Master-Receiver	The number of bytes in the Byte2Rd field of IDC_Control1 has been read and the LB bit is cleared.
	The number of bytes in the Byte2Rd field of IDC_Control1 has been read, the LB bit is set, and the STOP condition has been generated.
	Rx FIFO is full.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 15.6 Status Register Events (Cont.)

Mode	Condition
Slave-Transmitter	An address is recognized on the bus that matches the value in IIDC_Status1.
	Tx FIFO is empty.
	The STOP condition has been detected.
	The Repeat Start condition has been detected.
Slave-Receiver	An address is recognized on the bus that matches the value in IIDC_Status1.
	The Rx FIFO is half full (when a byte is received and already there are three bytes in the RxFIFO).
	The STOP condition has been detected.
	A General Call condition has been detected (slave address 0x00 has been recognized on the bus).

LA **Lost Arbitration** **28**
1 = The IDC lost arbitration during data phase. All data in the transmit FIFO is cleared. Software must re-program the operation.

If the IDC device loses arbitration during the address phase, it automatically retries in order to complete the operation.

SN **Send nACK** **27**
This bit reflects the state of the internal SEND_NAK bit.

TxE **Tx Register Empty** **26**
1 = The transmit FIFO count is zero.

SD **Stop Detected** **25**
1 = A stop condition was detected or generated.

RS **Repeat Start** **24**
1 = A Repeat Start was detected with the IDC device address active, or a Repeat Start was generated.

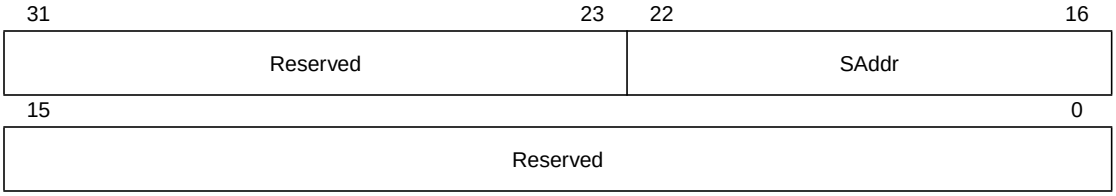
SAd **Slave Address** **23**
1 = The slave state machine detected a start condition, and the address is the IDC device address.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

MI	**Master Interrupt** 1 = An event associated with master mode operation occurred.	**22**
SI	**Slave Interrupt** 1 = An event associated with slave mode operation occurred.	**21**
NAK	**No Acknowledge** 1 = Either no slave responded after the IDC device sent out an address, or else the receiver did not acknowledge after the IDC device sent out data when the IDC was in master/slave transmitter mode.	**20**
AK	**Acknowledge** 1 = The addressed slave returned an acknowledge during the address phase.	**19**
RxR	**Rx Data Ready** 1 = The receive FIFO is not empty.	**18**
Gen	**General Call** 1 = The slave detected a general call address.	**17**
CSO	**Current Slave Operation** Meaningful in slave mode only. 1 = External master is reading DMN-8600 0 = External master is writing to DMN-8600	**16**

IDC Slave Address Register (IDC_SLAVEADDR)

Offset = 0xBE0088 Read/Write Default = 0x0000 0000

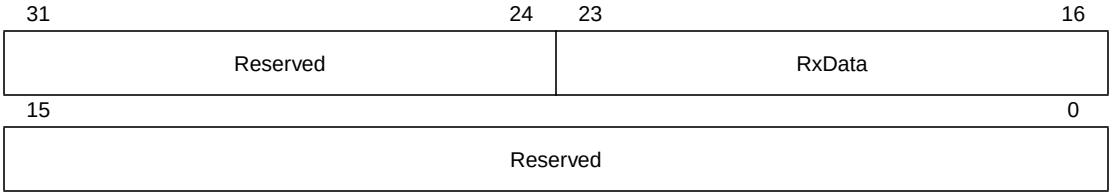


SAddr	**Slave Address** The IDC device responds to this address when in slave mode.	**[22:16]**

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC Receive Data Register (IDC_RX_DATA)

Offset = 0xBE008C Read Only Default = 0x0000 0000



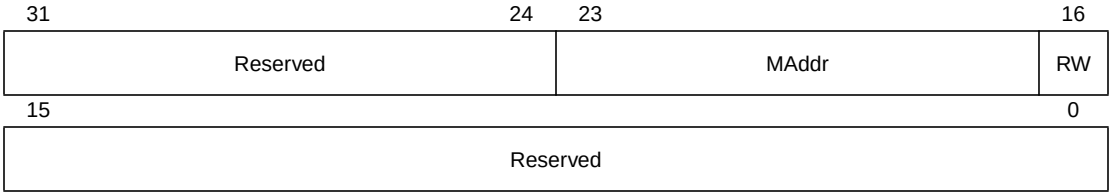
RxData **Receive Data** **[23:16]**
Reading this location reads data from the 8-byte RX FIFO. Data is placed in the RX FIFO when the IDC device is either the master of a valid read, or the slave of a valid write.

Caution: Do not read more bytes from this register than the number indicated in the RxCnt field of the IDC FIFO Fullness register. Attempting to read more bytes than are available will corrupt the RxCnt value, requiring a software reset.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC Master Address Register (IDC_MASTERADDR)

Offset = 0xBE0090 Read/Write Default = 0x0000 0000



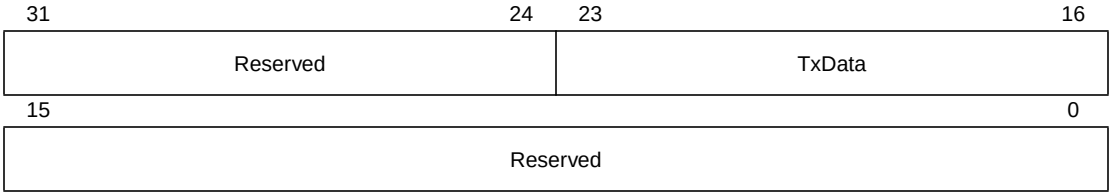
MAddr **Master Address** **[23:17]**
The IDC device uses this address when in master mode. It is the destination address for writes from the IDC device, and the address from which the IDC device reads.

RW **Read/Write** **16**
1 = Read operation

0 = Write operation

IDC Transmit Data Register (IDC_TX_DATA)

Offset = 0xBE0094 Write Only Default = 0x0000 0000



TxData **Transmit Data** **[23:16]**
Writing to this location writes data to the 8-byte transmit FIFO. The data in the transmit FIFO is transmitted over the IDC lines (SIO_SDA and SIO_SCL) when the IDC device is either the master during a write, or the slave during a read. This is a write-only register.

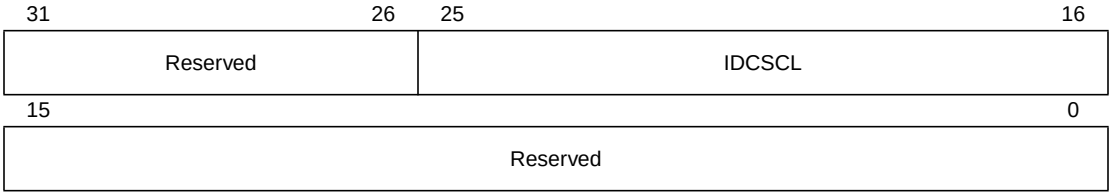
Caution: In the master transmitter mode, the transmit FIFO must not be filled with data until the IDC device wins arbitration. Otherwise, if the winning master tries to read from the IDC device, the IDC device sends the data intended for its own

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

master write, not the data requested by the winning master. This caution applies only in a multi-master environment.

IDC Clock Register (IDC_CLOCK)

Offset = 0xBE0098 Read/Write Default = 0x007D 0000

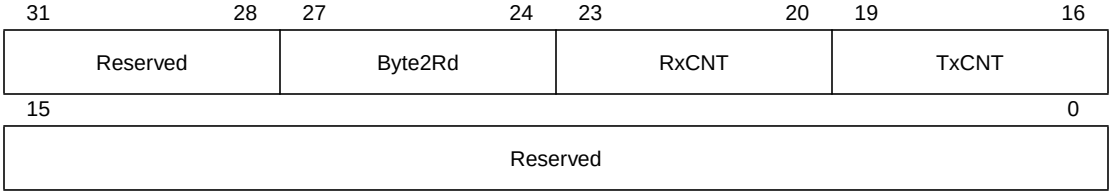


IDCSCL **IDC Clock Frequency** **[25:16]**
This field sets an upper bound on the IDC clock frequency. This upper bound is given by (sysclk / 4*IDCSCL). The actual IDC clock frequency is also affected by the amount of filtering applied by the IDC Receive Filter register. Since this filtering internally delays the rising edges of the SIO_SCL line, it effectively slows down the bus. With filtering employed, the IDC clock frequency is on the order of:

```
frequency = sysclk / [(4*IDCSCL) + (IDCSAMP+1)(IDCRISE+1)]
```

IDC FIFO Fullness Register (IDC_FIFO_STATUS)

Offset = 0xBE009C Read Only Default = 0x0000 0000



Byte2Rd **Bytes To Read** **[27:24]**
The actual number of bytes left to receive in master receive mode.

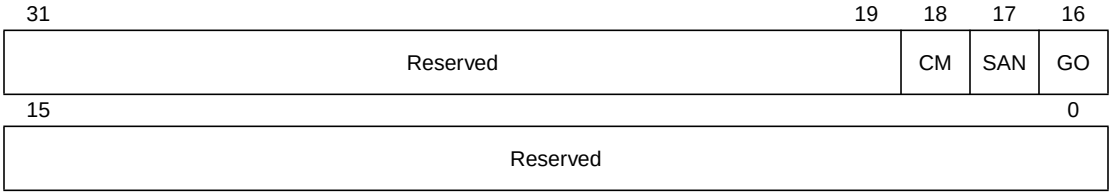
RxCNT **Received Byte Count** **[23:20]**
Number of bytes in (“fullness” of) the received data FIFO.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

TxCNT **Transmit Byte Count** **[19:16]**
Fullness of the transmit data FIFO.

IDC Control Register 2 (IDC_CONTROL2)

Offset = 0xBE00A0 Read/Write Default = 0x0000 0000



CM **Compatibility Mode** **18**
1 = IDC Mode. If the slave is enabled, and if either its address or a general call is recognized on the bus, the clock line SIO_SCL is held low and an interrupt is generated.

0 = Compatibility Mode. If the slave is enabled, and if either its address or a general call is recognized on the bus, an ACK is immediately generated.

SAN **Slave Address NAK** **17**
This bit is ignored in compatibility mode. In other modes:

1 = The slave address is NACKed.

0 = The slave address is ACKed.

Go **Hold/Go (autoclear)** **16**
This bit is ignored in compatibility mode. In other modes:

1 = The SIO_SDA line transmits ACK or NAK according to the value in the SAN bit, SIO_SCL is released, and the Go bit is cleared.

0 = The SIO_SCL line is pulled down when the slave address or a general call is recognized on the SIO_SDA line. The processor is interrupted, and the bus is stalled until the processor writes 1 into this bit.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC Receive Filter Register (IDC_RX_FILTER)

Offset = 0xBE00A4 Read/Write Default = 0x0003 0000

Two types of filtering are available on the SIO_SCL and SIO_SDA inputs to improve noise immunity during slow signal transitions:

- The first filter reduces the input sampling rate to once every (IDCSAMP+1) sysclk cycles.
- The second filter processes these sub-sampled inputs and delays internal rising or falling transitions until a given number of consecutive high or low states has been sampled.

IDCFALL **IDC Falling Edge** **[31:28]**
This field sets the number of consecutive low samples required to pass a falling edge. The actual value used is IDCFALL+1.

IDCRISE **IDC Rising Edge** **[23:20]**
This field sets the number of consecutive high samples required to pass a rising edge. The actual value used is IDCRISE+1.

Note: Typically, only the rising edge filter is used, since IDC systems tend to have slow rise times.

IDCSAMP **IDC Samples** **[19:16]**
This field determines the number of sysclk cycles between successive input samples. The actual value used is IDCSAMP+1.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC DMA Transmit Control Register (IDC_TX_CONTROL REG_ADDR)

Offset = 0xBE00C0 Read/Write Default = 0x0000 0000

GO **Load Current Address Buffer Registers** **3**

This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

Note: DMN-8600 always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

MODE **Transfer Mode Select** **2**

This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**

This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

When FLUS is set for transmit channels, all data currently in the channel FIFOs is dropped.

When FLUS is set for receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set

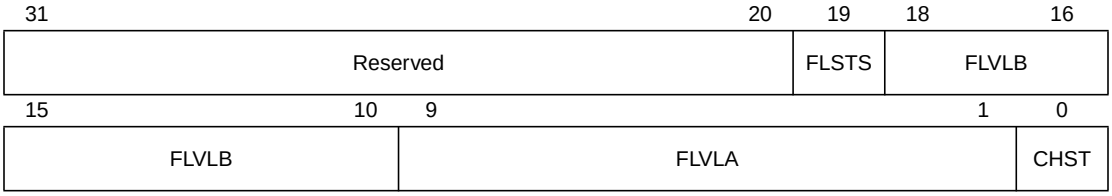
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

IDC DMA Transmit Status Register (IDC_TX_STATUS_REG_ADDR)

Offset = 0xBE00C4 Read/Write Default = 0x0000 0000



FLSTS **FIFO Flush Status** **19**
1 = The flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

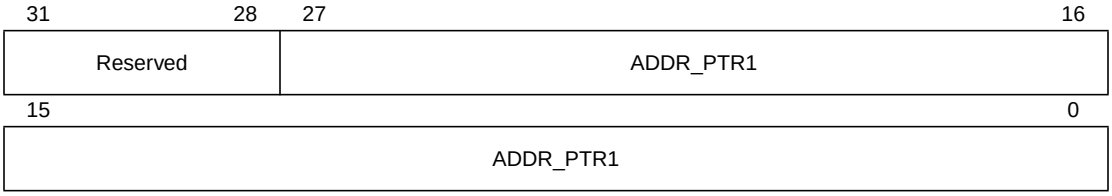
FLVLB **FIFO B Byte Count** **[18:10]**
This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **[9:1]**
This field shows the number of bytes currently held in FIFO A.

CHST **DMA Channel Status** **0**
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

IDC DMA Transmit Address Pointer 1 Register (IDC_TX_ADDR_PTR1_ADDR)

Offset = 0xBE00C8 Read/Write Default = 0x0000 0000



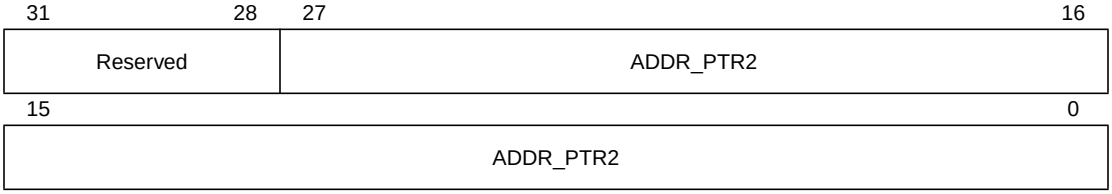
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ADDR_PTR1 Address Pointer 1 [27:0]
In double-buffer mode, this register indicates the Base Address for the next SDRAM Buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IDC DMA Transmit Address Pointer 2 Register (IDC_TX_ADDR_PTR2_ADDR)

Offset = 0xBE00CC Read/Write Default = 0x0000 0000

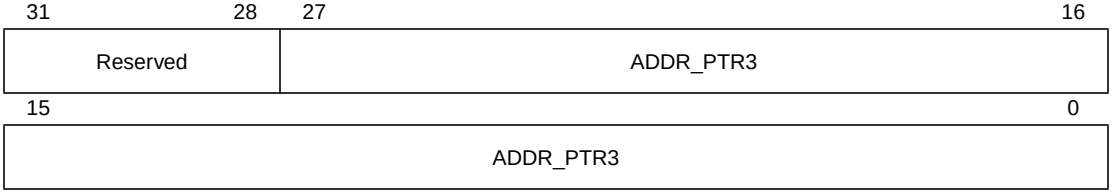


ADDR_PTR2 Address Pointer 2 [27:0]
In double-buffer mode, this register indicates the End Address for the next SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IDC DMA Transmit Address Pointer 3 Register (IDC_TX_ADDR_PTR3_ADDR)

Offset = 0xBE00D0 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 [27:0]
This register is updated by hardware during a DMA operation.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

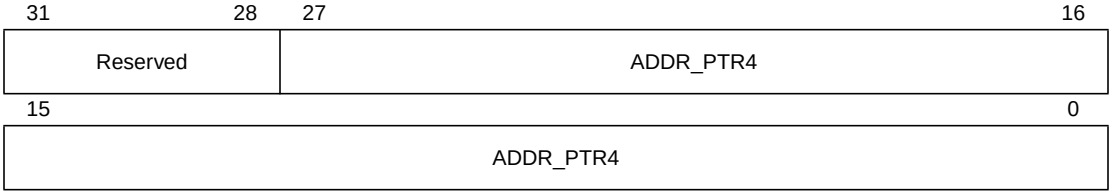
For read channels (receive), it indicates the current value of the read pointer.

In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

IDC DMA Transmit Address Pointer 4 Register (IDC_TX_ADDR_PTR4_ADDR)

Offset = 0xBE00D4 Read/Write Default = 0x0FFF FFFF



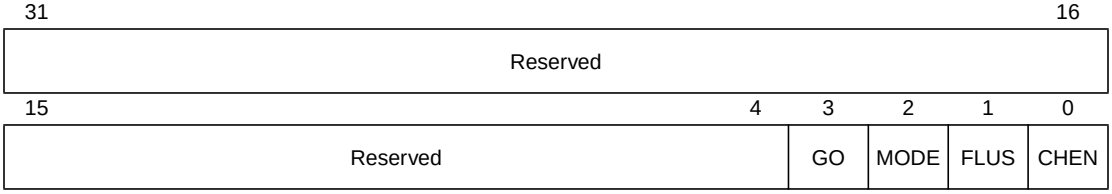
ADDR_PTR4 Address Pointer 4 [27:0]

In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

IDC DMA Receive Control Register (IDC_RX_CONTROL_REG_ADDR)

Offset = 0xBE00E0 Read/Write Default = 0x0000 0000



GO Load Current Address Buffer Registers 3

This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation. (See Note on <Hyperactive>page 42.)

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**

This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**

This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

When FLUS is set for transmit channels, all data currently in the channel FIFOs is dropped.

When FLUS is set for receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

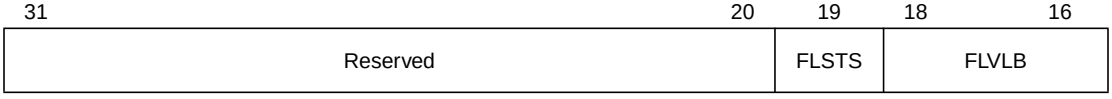
CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

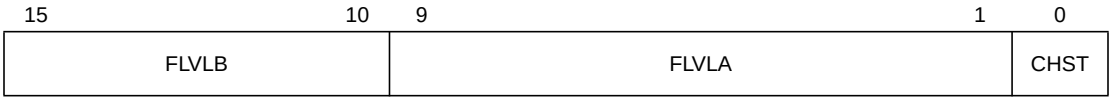
CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

IDC DMA Receive Status Register (IDC_RX_STATUS_REG_ADDR)

Offset = 0xBE00E4 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.



FLSTS **FIFO Flush Status** **19**
1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

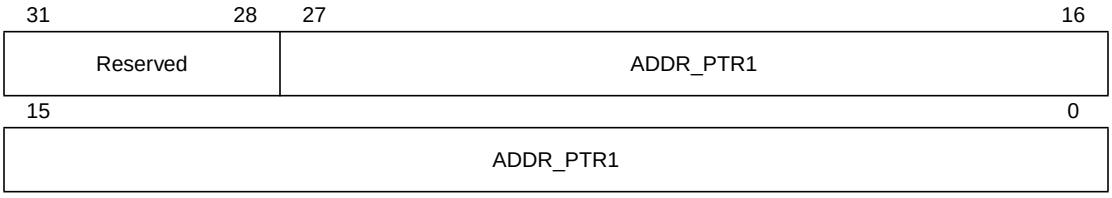
FLVLB **FIFO B Byte Count** **[18:10]**
This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **[9:1]**
This field shows the number of bytes currently held in FIFO A.

CHST **DMA Channel Status** **0**
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

IDC DMA Receive Address Pointer 1 Register (IDC_RX_ADDR_PTR1_ADDR)

Offset = 0xBE00E8 Read/Write Default = 0x0000 0000



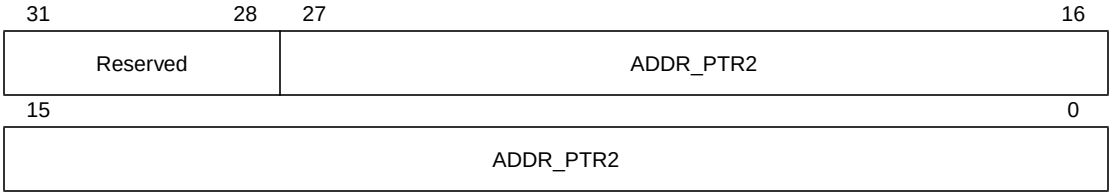
ADDR_PTR1 **Address Pointer 1** **[27:0]**
In double-buffer mode, this register indicates the Base Address for the next SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IDC DMA Receive Address Pointer 2 Register (IDC_RX_ADDR_PTR2_ADDR)

Offset = 0xBE00EC Read/Write Default = 0x0000 0000



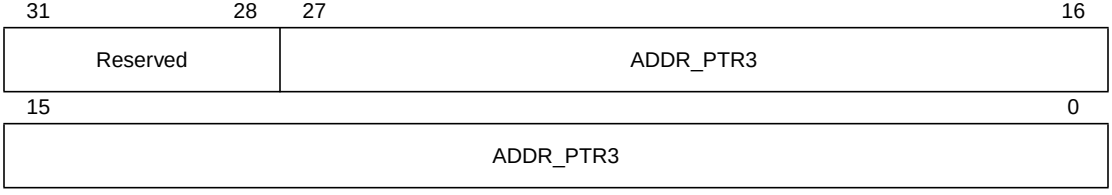
ADDR_PTR2 Address Pointer 2 [27:0]

In double-buffer mode, this register indicates the End Address for the next SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IDC DMA Receive Address Pointer 3 Register (IDC_RX_ADDR_PTR3_ADDR)

Offset = 0xBE00F0 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 [27:0]

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

For read channels (receive), it indicates the current value of the read pointer.

In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

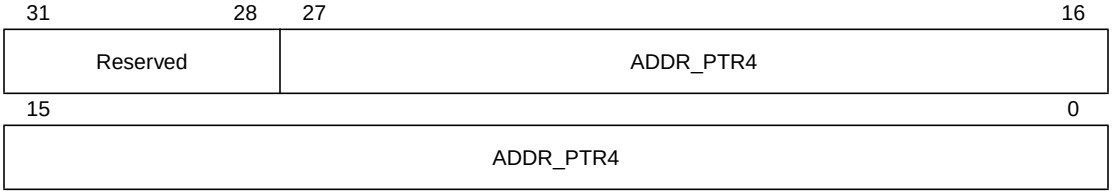
- The DMA channel is idle.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- The previous SDRAM buffer has completed.

IDC DMA Receive Address Pointer 4 Register (IDC_RX_ADDR_PTR4_ADDR)

Offset = 0xBE00F4 Read/Write Default = 0x0FFF FFFF



ADDR_PTR4 Address Pointer 4 [27:0]
In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

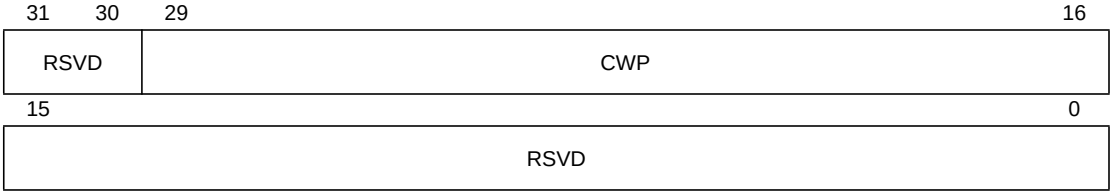
- The DMA channel is idle.
- The previous SDRAM buffer has completed.

15.5.4 SIO IR1 / IR2 Registers

IR1 Transmit Carrier Wave Period Register (IR1_CWP)

IR2 Transmit Carrier Wave Period Register (IR2_CWP)

Offset = 0xBF0000 / 0xBF0080 Read/Write Default = 0x3FFF0000



CWP Carrier Wave Period 29:16
This register encodes the period of the IR carrier wave in units of sysclk cycles. This register allows carrier frequencies as low as 4.58 kHz.

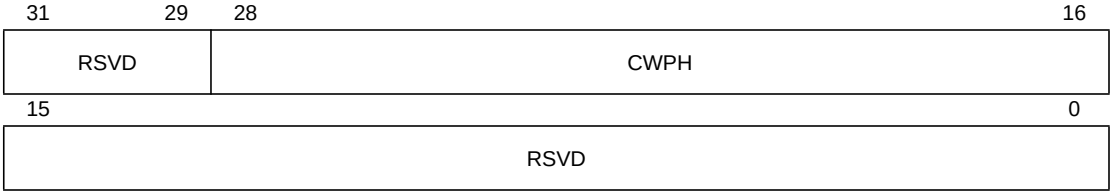
The actual carrier wave period is CWP[13:0] + 1.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 Transmit Carrier Wave Pulse High Register (IR1_CWPH)

IR2 Transmit Carrier Wave Pulse High Register (IR2_CWPH)

Offset = 0xBF0004 / 0xBF0084 Read/Write Default = 0x1FFF0000



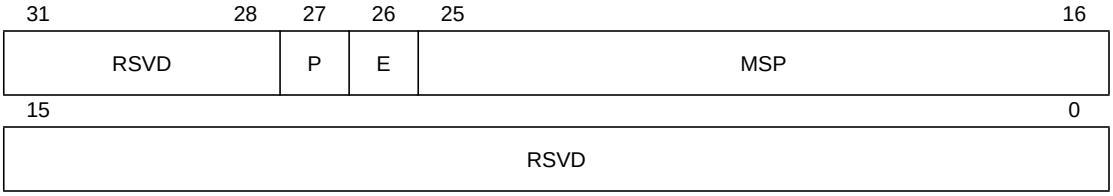
CWPH **Carrier Wave Pulse High** **28:16**
This register encodes the high pulse width of the IR carrier wave in units of sysclk cycles.

The actual carrier wave pulse high period is CWPH[12:0] + 1.

IR1 Modulated Signal Period Register (IR1_MSPR)

IR2 Modulated Signal Period Register (IR2_MSPR)

Offset = 0xBF0008 / 0xBF0088 Read/Write Default = 0x0000 0000



This is one of two registers written to produce an IR output. The other one is the IR Modulated Signal Pulse Low (IR_MSPL) register. Since this register contains the enable bit that initiates the transmission, it must be written last.

P **Output Polarity** **27**
1 = The SIO_IRTX* output is active high, normally low.

0 = The SIO_IRTX* output is active low, normally high.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

E **Transmit Enable** **26**

1 = Load the current transmit modulated waveform, the period high value, and the pulse high value into the output counters after the current waveform has completed.

This bit is automatically reset to zero after the waveform is loaded.

MSP **Modulated Signal Period** **25:16**

The MSP value specifies the modulated signal period in units of carrier pulses. The period is given by:

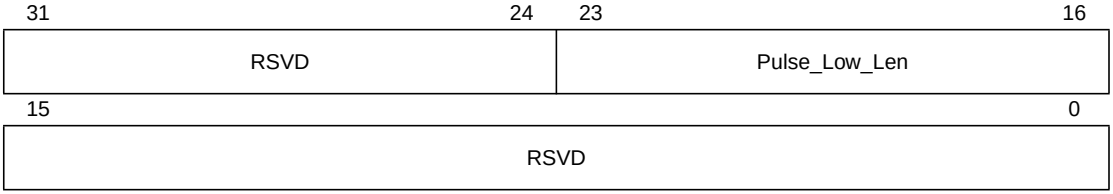
```
period = (MSPR[9:0]+1)(carrier wave period)
```

This register is sufficiently large to support the 9 ms pulse required for the start bit in the NEC format.

IR1 Modulated Signal Pulse Low Register (IR1_MSPL)

IR2 Modulated Signal Pulse Low Register (IR2_MSPL)

Offset = 0xBF000C / 0xBF008C Read/Write Default = 0x0000 0000



This register is one of two registers that is written to produce a bit stream output. Since the IR1_MSPR contains the enable bit that initiates the transmission, that register must be written last.

Pulse_Low_Len **23:16**

This register specifies the low pulse width in units of carrier pulses. The period is given by:

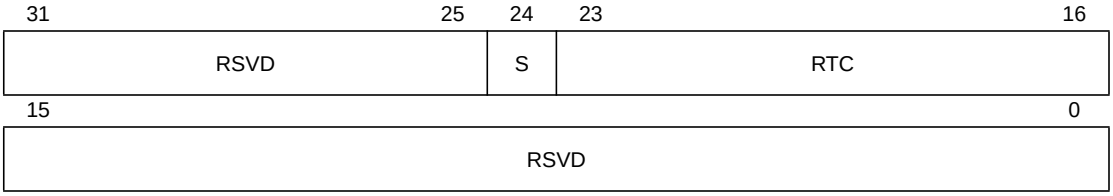
```
period = (MSPL[7:0]+1)(carrier wave period).
```

This register is sufficient to provide a 4.5 ms pulse, the largest low pulse width required by existing IR standards.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 Receive Tick Count Register (IR1_RTC)

Offset = 0xBF0010 Read/Write Default = 0x0000 0000



This register consists of two parts: an enable bit to start noise pulse filtering, and a receive tick count register, which records the time between deassertions of the IR input in terms of IR ticks. The tick period can be modified by programming the Receive Tick Period register.

Note: Deassertion of the IR input can be on either the rising edge or falling edge of SIO_IRRX, depending on the settings of bit 24 in the IR1 Receive Filter Register (IR1_RFR)

S **Filter Start Pulse** **24**

1 = The IR receiver is disabled until IRRX is asserted for a period specified by the IR Receive Filter register.

0 = The IR receiver is always enabled.

This value is cleared after the reception of a valid start pulse.

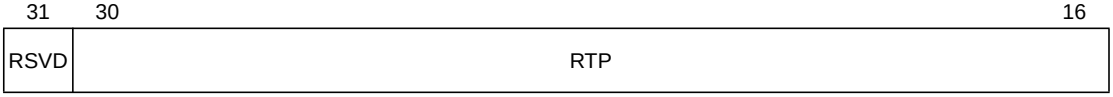
RTC **Receive Tick Count** **23:16**

The tick count register records the tick count from the last deassertion of SIO_IRRX to the current deassertion of SIO_IRRX. This allows software to determine the validity of the bits, as well as the value.

This field is read-only.

IR1 Receive Tick Period Register (IR1_RTP)

Offset = 0xBF0014 Read/Write Default = 0x1B770000



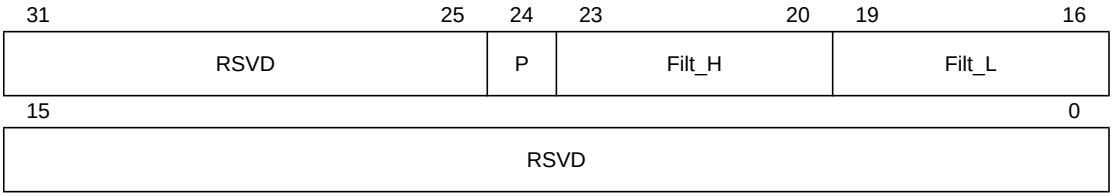
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

RTP **Receive Tick Period** **30:16**
This register determines the tick period used by the IR receiver. The tick period provides the basic time unit for the receive tick counter and is reset to 7032 sysclk cycles.

The actual tick period is given by RTP[14:0] + 1.

IR1 Receive Filter Register (IR1_RFR)

Offset = 0xBF0018 Read/Write Default = 0x0053 0000



This register holds the upper and lower limits for the receive pulse high width filter as well as the polarity select for the IR receive signal SIO_IRRX.

P **IR RX Signal Polarity** **24**
This bit sets the input signal (SIO_IRRX) polarity.

1 = SIO_IRRX is active low and RTC reports the time between rising edges of SIO_IRRX, while RPH reports the time that IRRX is low.

0 = SIO_IRRX is active high and RTC reports the time between falling edges of SIO_IRRX, while RPH reports the time that SIO_IRRX is high.

This bit is set to zero on reset.

Filt_H **IR RX Start Pulse Filter Upper Limit** **23:20**
This 4-bit value is matched against the upper 4 bits of the 8-bit tick count value to determine when a start pulse is too large, and therefore is not considered valid. This value has no effect if the Filter Start Pulse bit in the IR Receive Tick Count register is set to zero.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

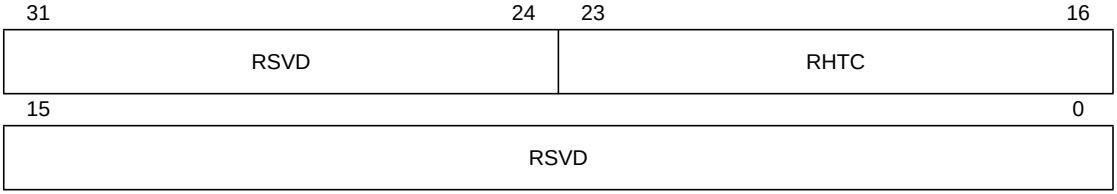
The default value is 5.

Filt_L **IR RX Start Pulse Filter Lower Limit** **19:16**
This 4-bit value is matched against the upper 4 bits of the 8-bit tick count value to determine when a start pulse is too small, and therefore is not considered valid. This value has no effect if the Filter Start Pulse bit in the IR Receive Tick Count register is set to zero.

The default value is 3.

IR1 Receive Pulse High Tick Count Register (IR1_RPH)

Offset = 0xBF001C Read/Write Default = 0x0000 0000

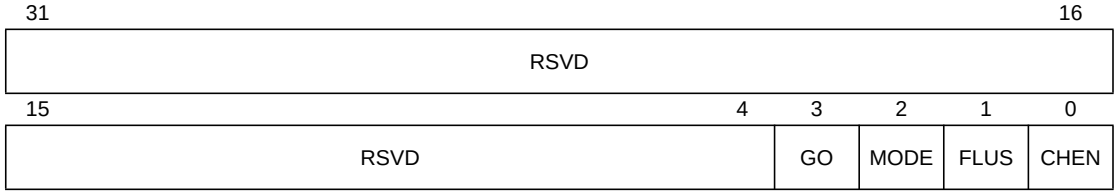


RHTC **Receive Pulse High Tick Count** **23:16**
This register contains the high pulse width for the previously received input pulse. This register is loaded after each deassertion of the SIO_IRRX input. Note that the default value of each tick is 7032 cycles. The tick period can be modified by programming the Receive Tick Period register.

IR1 DMA Transmit Control Register (IR_TX_CONTROL_REG_ADDR)

IR2 DMA Transmit Control Register (IR2_TX_CONTROL_REG_ADDR)

Offset = 0xBF0040 / 0xBF00C0 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

GO **Load Current Address Buffer Registers** **3**
This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in
double-buffer mode operation.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**
This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**
This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

1 = For transmit channels, all data currently in the channel FIFOs is dropped.

1 = For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

CHEN **DMA Channel Enable** **0**
This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

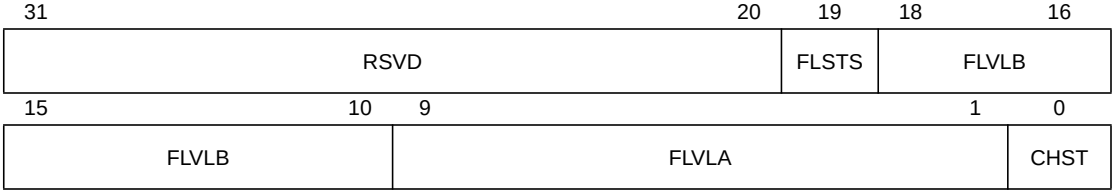
CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 DMA Transmit Status Register (IR_TX_STATUS_REG_ADDR)

IR2 DMA Transmit Status Register (IR2_TX_STATUS_REG_ADDR)

Offset = 0xBF0044 / 0xBF00C4 Read/Write Default = 0x0000 0000



FLSTS **FIFO Flush Status** **19**
This bit is set when the flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**
This field shows the number of bytes currently held in FIFO B.

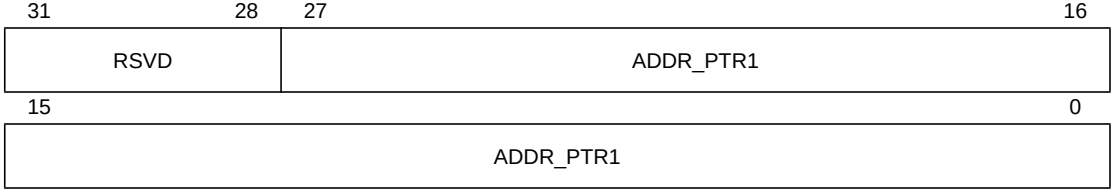
FLVLA **FIFO A Byte Count** **9:1**
This field shows the number of bytes currently held in FIFO A.

CHST **DMA Channel Status** **0**
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

IR1 DMA Transmit Address Pointer1 Register (IR_TX_ADDR_PTR1_ADDR)

IR2 DMA Transmit Address Pointer1 Register (IR2_TX_ADDR_PTR1_ADDR)

Offset = 0xBF0048 / 0xBF00C8 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ADDR_PTR1 Address Pointer 1 27:0

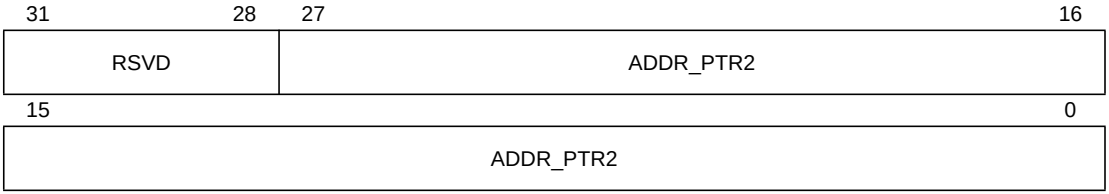
In double-buffer mode, this register indicates the Base Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IR1 DMA Transmit Address Pointer2 Register (IR_TX_ADDR_PTR2_ADDR)

IR2 DMA Transmit Address Pointer2 Register (IR2_TX_ADDR_PTR2_ADDR)

Offset = 0xBF004C / 0xBF00CC Read/Write Default = 0x0000 0000



ADDR_PTR2 Address Pointer 2 27:0

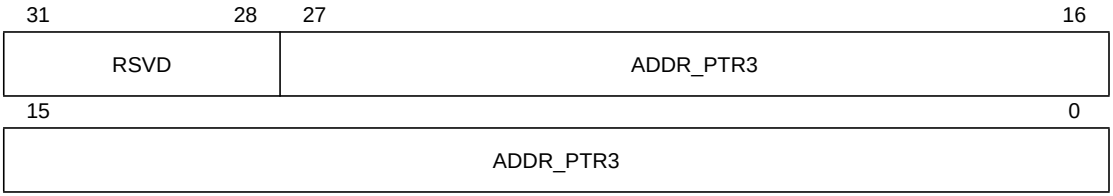
In double-buffer mode, this register indicates the End Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IR1 DMA Transmit Address Pointer3 Register (IR_TX_ADDR_PTR3_ADDR)

IR2 DMA Transmit Address Pointer3 Register (IR2_TX_ADDR_PTR3_ADDR)

Offset = 0xBF0050 / 0xBF00D0 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ADDR_PTR3 Address Pointer 3 **27:0**

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

For read channels (receive), it indicates the current value of the read pointer.

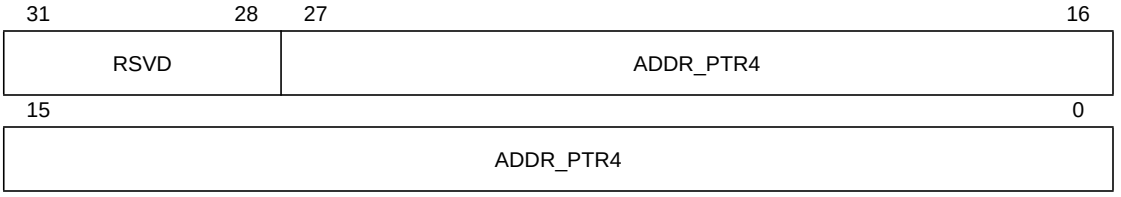
In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

IR1 DMA Transmit Address Pointer4 Register (IR_TX_ADDR_PTR4_ADDR)

IR2 DMA Transmit Address Pointer4 Register (IR2_TX_ADDR_PTR4_ADDR)

Offset = 0xBF0054 / 0xBF00D4 Read/Write Default = 0x0FFF FFFF



ADDR_PTR4 Address Pointer 4 **27:0**

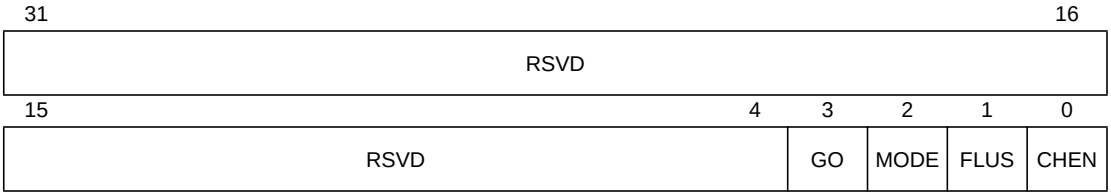
In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 DMA Receive Control Register (IR_RX_CONTROL_REG_ADDR)

Offset = 0xBF0060 Read/Write Default = 0x0000 0000



GO **Load Current Address Buffer Registers** **3**

This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**

This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**

This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

For transmit channels, when FLUS is set, all data currently in the channel FIFOs is dropped.

For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

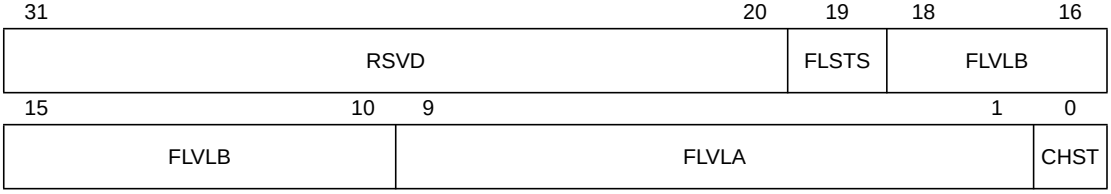
CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

IR1 DMA Receive Status Register (IR_RX_STATUS_REG_ADDR)

Offset = 0xBF0064 Read/Write Default = 0x0000 0000



FLSTS **FIFO Flush Status** **19**

1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**

This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **9:1**

This field shows the number of bytes currently held in FIFO A.

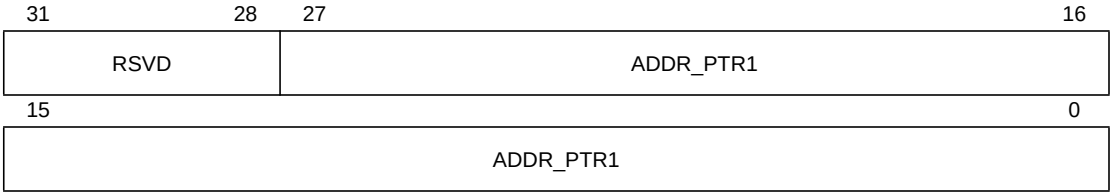
CHST **DMA Channel Status** **0**

1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 DMA Receive Address Pointer1 Register (IR_RX_ADDR_PTR1_ADDR)

Offset = 0xBF0068 Read/Write Default = 0x0000 0000



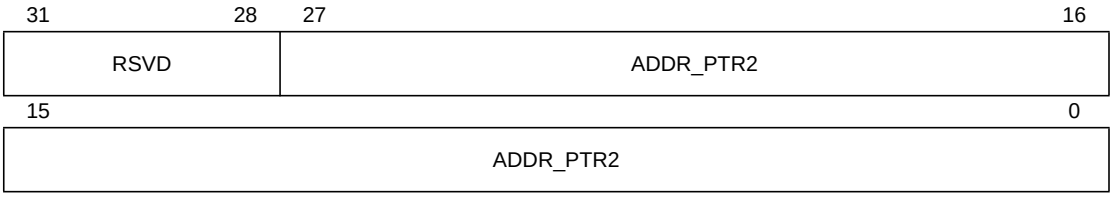
ADDR_PTR1 Address Pointer 1 27:0

In double-buffer mode, this register indicates the Base Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

IR1 DMA Receive Address Pointer2 Register (IR_RX_ADDR_PTR2_ADDR)

Offset = 0xBF006C Read/Write Default = 0x0000 0000



ADDR_PTR2 Address Pointer 2 27:0

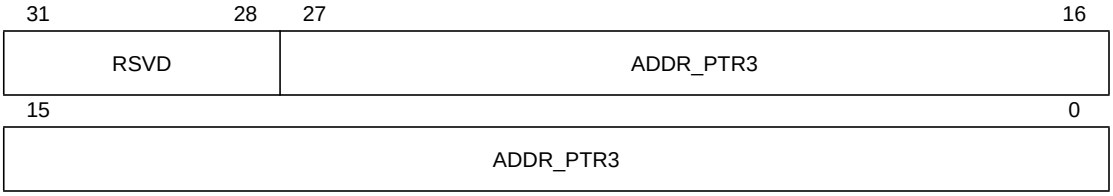
In double-buffer mode, this register indicates the End Address for the “next” SDRAM Buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

IR1 DMA Receive Address Pointer3 Register (IR_RX_ADDR_PTR3_ADDR)

Offset = 0xBF0070 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 27:0

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

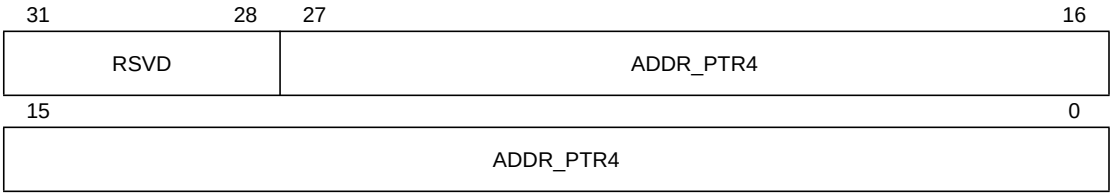
For read channels (receive), it indicates the current value of the read pointer.

In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

IR1 DMA Receive Address Pointer4 Register (IR_RX_ADDR_PTR4_ADDR)

Offset = 0xBF0074 Read/Write Default = 0x0FFF FFFF



ADDR_PTR4 Address Pointer 4 27:0

In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

15.5.5 SIO SPI Registers

SPI Configuration Register (SPI_CONFIG)

Offset = 0xBE0000 Read/Write Default = 0x0000 0000

BBMD **Bang Mode Enabled** **31**

1 = The low-frequency bit-bang mode is enabled. When in this mode, all SPI signal lines are mapped to bytes in the SPI_TEMP register.

For each byte time, the value of byte [n] in the SPI_TEMP register is held static and the bits are mapped to the SPI signal lines as shown in Table 15.7.

Table 15.7 Bit-bang Mode Bit to SPI Signal Mapping

Byte/Bit	Signal	In/Out
BYTE[n].7	Don't Care	N/A
BYTE[n].6	SIO_SPI_CS[3]	Out
BYTE[n].5	SIO_SPI_CS[2]	Out
BYTE[n].4	SIO_SPI_CS[1]	Out
BYTE[n].3	SIO_SPI_CS[0]	Out
BYTE[n].2	SIO_SPI_CLK	Out
BYTE[n].1	SIO_SPI_MOSI	Out
BYTE[n].0	SIO_SPI_MISO	In

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

The length of a byte time is a function of the current SIO_SPI_CLK frequency, the setting of inter-byte blanking, and SIO_SPI_CS setup/hold (CSDL[2:0]).

SIO_SPI_MOSI is set to the value of the last bit sent by the slave SPI device for each byte time. In bit-bang mode, a minimum of two bytes are needed for one cycle of SIO_SPI_CLK. Added to this is the time needed to hold the bit stable for the entire byte time/interbyte blank, SIO_SPI_CS[n] setup, and SIO_SPI_CS[n] hold. Therefore, this mode is suitable only for low-frequency applications.

SPED **Clock Divider, LSB** **30:24**

The SIO_SPI_CLK frequency is determined by the following equation:

```
SIO_SPI_CLK = sysclk / 2(progdiv + 2)
```

where progdiv = {HDIV, SPED}

(That is, progdiv is the concatenation of HDIV bits in the SPI Clock Divider register with the SPED bits the SPI Configuration register.)

HUEN **Host Update Enable** **23**

1 = Set the update mode to either single word mode or host-polled mode. Values are given in Table 15.8.

Table 15.8 SPI Transfer Modes

HUEN	ODW	ODR	Description
0	0	0	Single-word mode. The SPI interface outputs BSIZ[1:0] + 1 bytes sourced from SPI_TEMP and then completes the cycle. The BSIZ[1:0] + 1 bytes are found in SPI_TEMP at the end of the cycle. The data in SPI_SHIFT is undefined.
1	0	0	Host-polled mode. All updates of BSIZ[1:0] + 1 bytes are done by the host. Access to the SPI_TEMP register is indicated by the HAEN bit. After the current word is transferred, the cycle stalls until the host writes the SPI_TEMP register which clears the HAEN bit and allows the cycle to continue. If the read data is needed, the host reads the SPI_TEMP register before doing the write.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

OIN **22**

This bit must always be written as 0.

CSPL[3:0] **SPI Chip Select Level [n]** **21:18**

1 = The corresponding SIO_SPI_CS[n] signal is active high.

0 = The corresponding SIO_SPI_CS[n] signal is active low.

CPOL **SPI Idle Clock Level** **17**

1 = The SIO_SPI_CLK line is pulled high when the SPI interface is idle.

0 = The SIO_SPI_CLK line is held low when the SPI interface is idle.

CPHA **SPI Active Clock Level** **16**

1 = The SIO_SPI_CLK line on the first half of the first bit period of a SPI transaction is inverted from the value of SIO_SPI_CLK when the SPI interface is idle.

0 = The SIO_SPI_CLK line on the first half of the first bit period of a SPI transaction is the same as the value of SIO_SPI_CLK when the SPI interface is idle.

ODW **15**

This bit must always be written as 0.

ODR **14**

This bit must always be written as 0.

IBBK **Inter-Byte Blanking Period** **13:10**

This is the inter-byte blanking period (in bits) where, depending on the settings of the CSBK, SIO_SPI_CS is negated and SIO_SPI_CLK is gated between successive bytes. SIO_SPI_CLK is always gated during this interval.

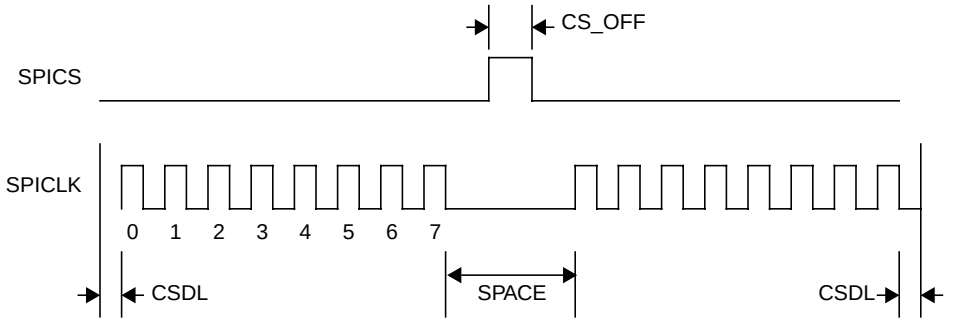
When IBBK=0, no inter-byte blanking occurs.

If inter-byte blanking is used, SIO_SPI_CS has a setup/hold period approximately equal to the blanking time determined by the CSDL setting.

Figure 15.10 shows the relationship between the CSBK, CSDL, and IBBK fields.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 15.10 SPI Inter-Byte Blanking



WHERE:

CSBK	CSDL	IBBK	SPACE	CS_OFF	NOTES
x	x	0	1	0	
0	n>0	m>0	(2*CSDL)+IBBK	0	General Case
1	n>0	m>0	(2*CSDL)+IBBK	IBBK	General Case

CSBK **SPI Chip Select Blanking Level** **9**

1 = SIO_SPI_CS[n] is negated during the inter-byte blanking period.

IBBM **Inter-Byte Blanking Mode** **8**

1 = Inter-byte blanking is allowed between the first and second bytes only.

0 = Inter-byte blanking is allowed between every byte.

LSBF **Least Significant Bit First** **7**

1 = The least significant bit of each write byte is shifted out first, and each read byte in the SPI_SHIFT register is swapped before writing it to the SPI_TEMP register. Since the swapping occurs between the SPI_SHIFT register and the data registers, software must shift bits for bytes in the SPI_SHIFT register.

0 = The most significant bit of each write byte is shifted out first.

CSDL[2:0] **SPI Chip Select Delay** **6:4**

These bits allow adjustment of the setup and hold times of the SIO_SPI_CS[n] about SPI_CLK. CSDL is the number of half SIO_SPI_CLK cycles between the following events:

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- SIO_SPI_CS[n] assertion and SIO_SPI_CLK ungating
- SIO_SPI_CS[n] negation and SIO_SPI_CLK gating

Figure 15.10 shows the relationship between the CSBK, CSDL, and IBBK fields.

CSSL[3:0]- SPI Chip Select Enable **3:0**

These bits select which of the SPI chip select lines is used for the current SPI cycle. The bit encoding is shown in Table 15.9:

Table 15.9 CSSL Bit to Chip Select Mapping

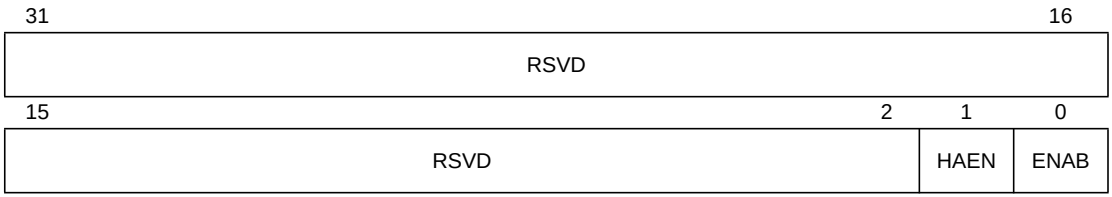
Bit	Signal
CSSL: 0001	SIO_SPI_CS[0] enabled
CSSL: 0010	SIO_SPI_CS[1] enabled
CSSL: 0100	SIO_SPI_CS[2] enabled
CSSL: 1000	SIO_SPI_CS[3] enabled

When the SPI cycle is idle, all four chip select lines are in the negated state set by CSPL.

Note: Additional selects can be decoded through external logic, if required.

SPI Control Register (SPI_CONTROL)

Offset = 0xBE0004 Read/Write Default = 0x0000 0000



HAEN **Host Access Enable** **1**

1 = Host access of SPI_TEMP is enabled.

HAEN is set when the SPI_TEMP register is loaded with the contents of SPI_SHIFT.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

HAEN is cleared when the host writes SPI_TEMP.

If the ENAB bit is set (that is, the SPI cycle is running) and HAEN=0, host writes to SPI_TEMP are discarded and do not affect the state of the HAEN bit. HAEN=0 whenever ENAB=0, HUEN=0.

ENAB **SPI Enable** **0**

Set by the host to begin the SPI transfer. When set, host writes to SPI_CONTROL and SPI_CONFIG are discarded.

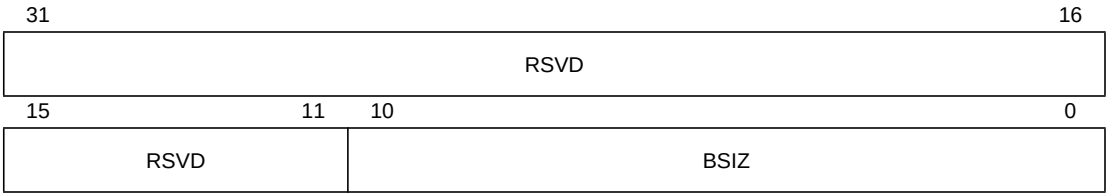
Cleared by the SPI module when the transfer completes.

This bit can be polled by software to check for cycle completion.

Note: Reset interrupts the current cycle and clears this bit.

SPI DMA Size Register (SPI_DMASIZE)

Offset = 0xBE0010 Read/Write Default = 0x0000 0000



BSIZ **SPI Data Transfer** **10:0**

This field defines the size of the data block to be transferred by the SPI interface over the pins.

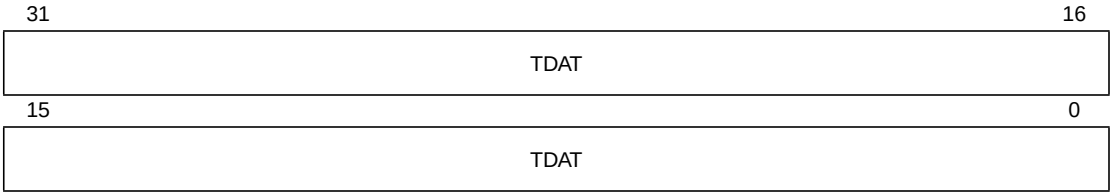
In host-polled mode, the host or SPARC processor must write/read the SPI_TEMP register BSIZ+1 times. Afterwards, the SPI_TEMP register must be written twice with dummy data.

With the DMA engine, these write/read updates are taken care of automatically. See Section 15.2.5, “SPI Programming Examples,” page 15-12 for more details.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SPI Temp Register (SPI_TEMP)

Offset = 0xBE0014 Read/Write Default = 0x0000 0000



TDAT **Temporary Data** **31:0**

This register holds a 32-bit word to write out on the SIO_SPI_MOSI pin and to read in a 32-bit word on the SIO_SPI_MISO pin.

In host update (single-word, host-polled) modes, the host accesses this register directly.

In DMA mode, the DMA engine writes/reads this register automatically in a ping-pong fashion (first write, then read, then write, ...).

If the LSBF bit is set, then, for each byte, the bits are sent out LSB first, and the bit ordering of each byte is reversed from the order in which the bits were received on the SIO_SPI_MISO pin.

SPI Shift Register (SPI_SHIFT)

Offset = 0xBE0018 Read/Write Default = 0x0000 0000

SREG **Shift Register** **31:0**

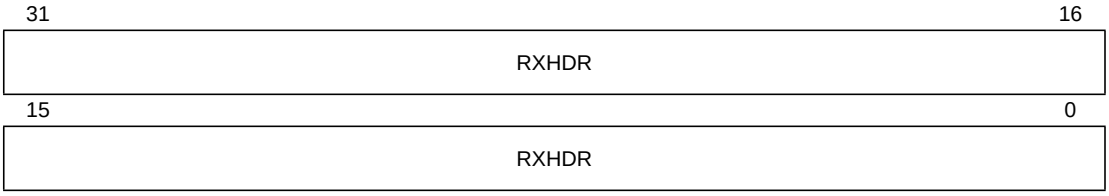
This is a shift register for the SIO_SPI_MOSI and SIO_SPI_MISO pins. When reading this register after the completion of a SPI cycle, bit 1 is the last bit read from SIO_SPI_MISO. The order of the bits is unaffected by the LSBF bit, which causes the swap of the bits when transferring to/from the SPI_TEMP register. When a SPI cycle

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

completes, SPI_SHIFT contains the last four bytes read from SIO_SPI_MISO.

SPI Receive Header Register (SPI_RXHDR)

Offset = 0xFFBE001C Read Only Default = 0x0000 0000

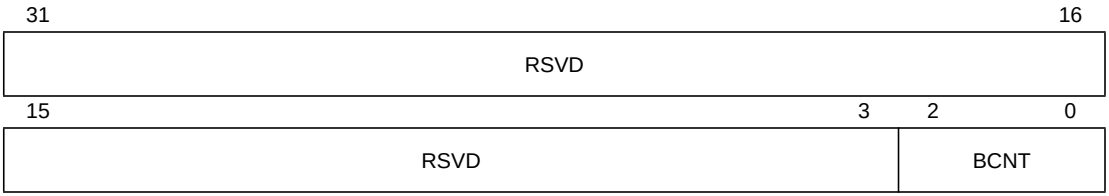


RXHDR - SPI Receive Header [31:0]

This register contains the first four bytes received through the SIO_SPI_MISO pin as part of a SPI cycle.

SPI Bit Granularity Register (SPI_BITGRAN)

Offset=0x 0xBE0020 Read/Write Default=0x0000 0000



BCNT **SPI Bit Granularity** **2:0**

BCNT, together with the BSIZ field in the SPI_SIZE register, allows the SPI interface to have bit granularity on the last 32-bit word in a SPI cycle to be transferred. After SPI_TEMP is written with the final word of data, anywhere from 8 to 32 of these bits can be sent out on the pins. Sending out less than 8 bits is not allowed. Table 15.10 shows how many bits of the final 32-bit word are sent out for the allowed values of BSIZ and BCNT.

```
If (BCNT != 0)
NumBits Transferred = (BSIZ)* 8 + BCNT
else
NumBits Transferred = (BSIZ + 1) * 8
```

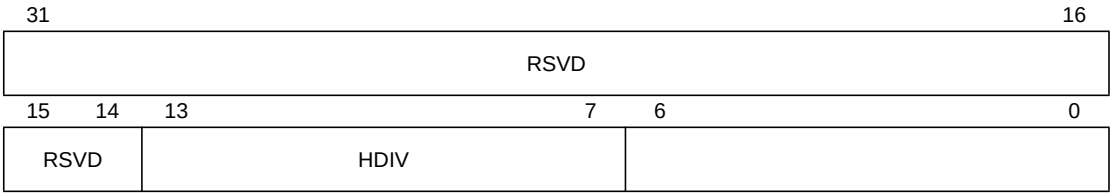

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 15.10 Number of Bits Sent Depending on Programmed Values of BSIZ and BCNT

BSIZ	BCNT	# Bits Sent	BSIZ	BCNT	# Bits Sent
0	0	8	2	0	24
0	1	N/A	2	1	17
0	2	N/A	2	2	18
0	3	N/A	2	3	19
0	4	N/A	2	4	20
0	5	N/A	2	5	21
0	6	N/A	2	6	22
0	7	N/A	2	7	23
1	0	16	..	..	..
1	1	9	..	..	..
1	2	10	4	0	40=32+8
1	3	11	4	5	37=32+5
1	4	12	14	0	120=32+32+32+24
1	5	13	14	7	119=32+32+32+23
1	6	14	17	0	144=32+32+32+32+16
1	7	15	17	6	142=32+32+32+32+14

SPI Clock Divider Register, MSB (SPI_SPED_MSB)

Offset = 0xBE0028 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

HDIV **Clock Divider, MSB** **13:7**

The SPI_CLK frequency is determined by the following equation:

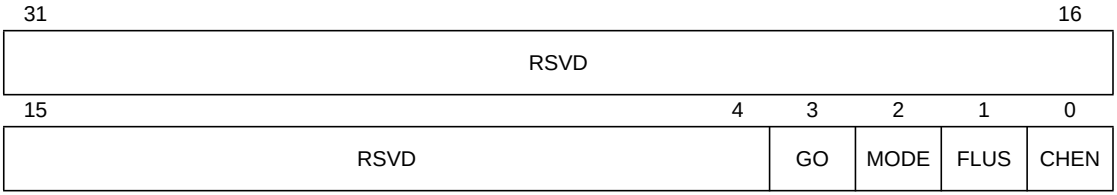
```
SIO_SPI_CLK = sysclk / 2(progdiv + 2)
```

where progdiv = {HDIV, SPED}

(That is, progdiv is the concatenation of HDIV bits in the SPI Clock Divider register with the SPED bits in the SPI Configuration register.)

SPI DMA Transmit Control Register (SPI_TX_CONTROL_REG_ADDR)

Offset = 0xBE0040 Read/Write Default = 0x0000 0000



GO **Load Current Address Buffer Registers** **3**

This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**

This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

FLUS **FIFO Flush** **1**

This is a self-clearing bit for flushing the DMA channel's two internal FIFOs.

For transmit channels, when FLUS is set, all data currently in the channel FIFOs is dropped.

For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

SPI DMA Transmit Status Register (SPI_TX_STATUS_REG_ADDR)

Offset = 0xBE0044 Read/Write Default = 0x0000 0000

FLSTS **FIFO Flush Status** **19**

1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**

This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **9:1**

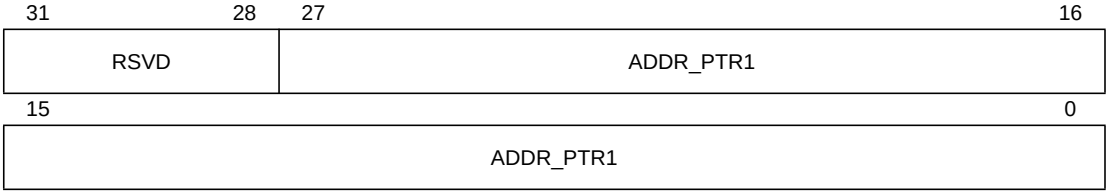
This field shows the number of bytes currently held in FIFO A.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

CHST **DMA Channel Status** **0**
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

SPI DMA Transmit Address Pointer1 Register (SPI_TX_ADDR_PTR1_ADDR)

Offset = 0xBE0048 Read/Write Default = 0x0000 0000

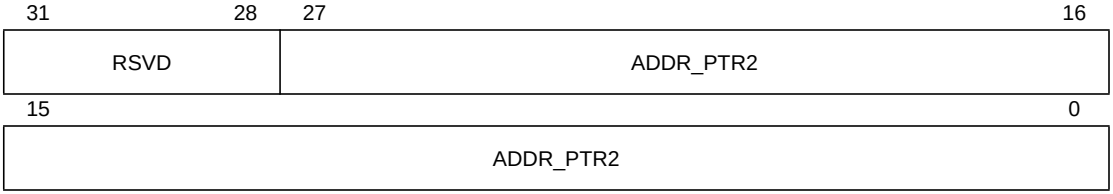


ADDR_PTR1 **Address Pointer 1** **27:0**
In double-buffer mode, this register indicates the Base Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

SPI DMA Transmit Address Pointer2 Register (SPI_TX_ADDR_PTR2_ADDR)

Offset = 0xBE004C Read/Write Default = 0x0000 0000



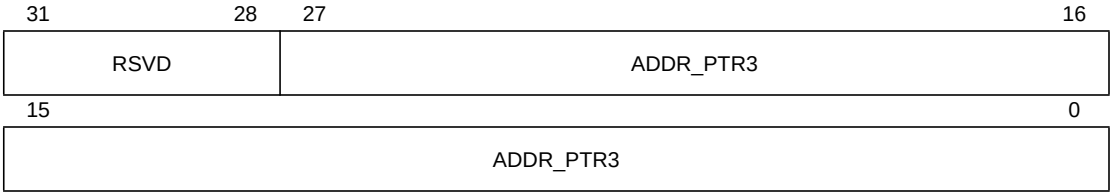
ADDR_PTR2 **Address Pointer 2** **27:0**
In double-buffer mode, this register indicates the End Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SPI DMA Transmit Address Pointer3 Register (SPI_TX_ADDR_PTR3_ADDR)

Offset = 0xBE0050 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 27:0

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

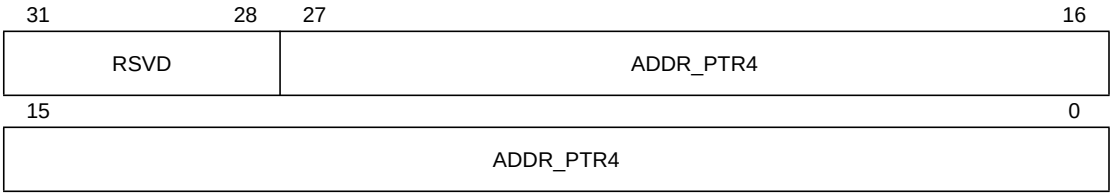
For read channels (receive), it indicates the current value of the read pointer.

In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

SPI DMA Transmit Address Pointer4 Register (SPI_TX_ADDR_PTR4_ADDR)

Offset = 0xBE0054 Read/Write Default = 0x0FFF FFFF



ADDR_PTR4 Address Pointer 4 27:0

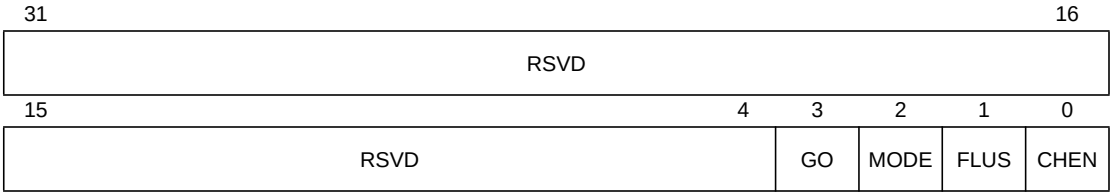
In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SPI DMA Receive Control Register (SPI_RX_CONTROL_REG_ADDR)

Offset = 0xBE0060 Read/Write Default = 0x0000 0000



GO **Load Current Address Buffer Registers** **3**
This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**
This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**
This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

For transmit channels, when FLUS is set, all data currently in the channel FIFOs is dropped.

For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

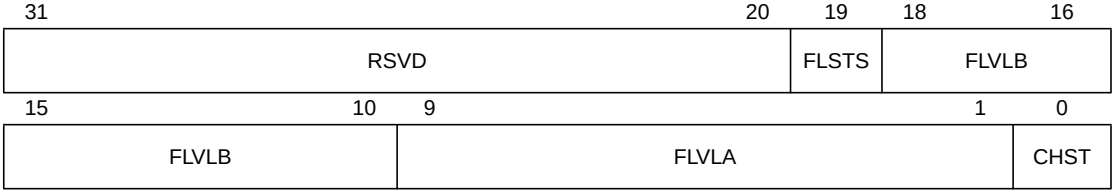
CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

SPI DMA Receive Status Register (SPI_RX_STATUS_REG_ADDR)

Offset = 0xBE0064 Read/Write Default = 0x0000 0000



FLSTS **FIFO Flush Status** **19**

1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**

This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **9:1**

This field shows the number of bytes currently held in FIFO A.

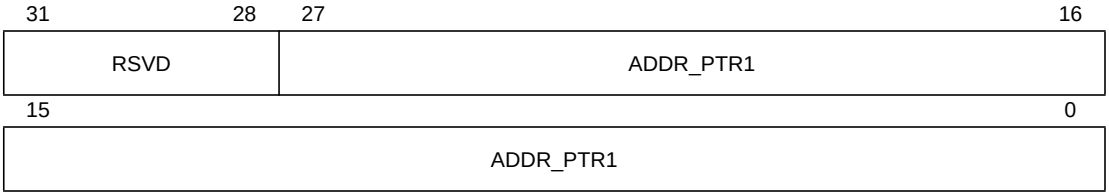
CHST **DMA Channel Status** **0**

1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SPI DMA Receive Address Pointer1 Register (SPI_RX_ADDR_PTR1_ADDR)

Offset = 0xBE0068 Read/Write Default = 0x0000 0000

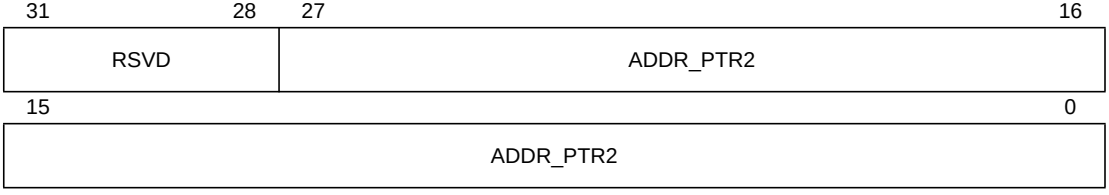


ADDR_PTR1 Address Pointer 1 27:0
In double-buffer mode, this register indicates the Base Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

SPI DMA Receive Address Pointer2 Register (SPI_RX_ADDR_PTR2_ADDR)

Offset = 0xBE006C Read/Write Default = 0x0000 0000



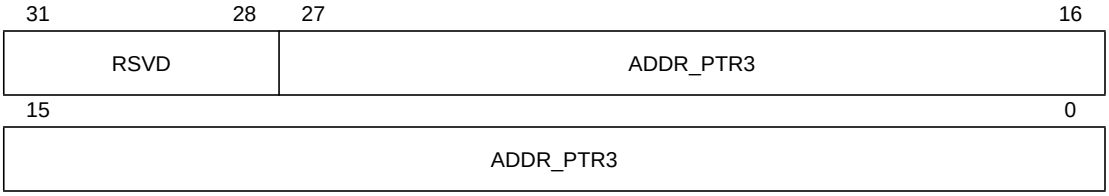
ADDR_PTR2 Address Pointer 2 27:0
In double-buffer mode, this register indicates the End Address for the “next” SDRAM buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SPI DMA Receive Address Pointer3 Register (SPI_RX_ADDR_PTR3_ADDR)

Offset = 0xBE0070 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 27:0

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

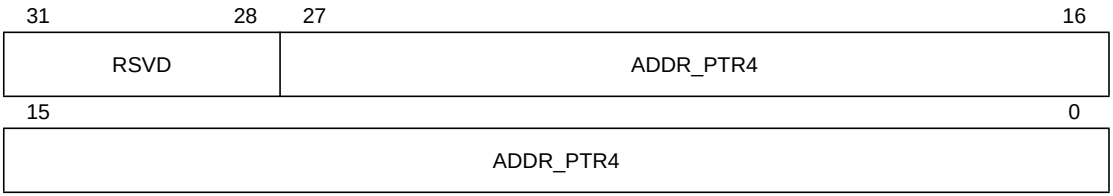
For read channels (receive), it indicates the current value of the read pointer.

In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

SPI DMA Receive Address Pointer4 Register (SPI_RX_ADDR_PTR4_ADDR)

Offset = 0xBE0074 Read/Write Default = 0x0FF FFFFF



ADDR_PTR4 Address Pointer 4 27:0

In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

15.5.6 SIO UART1/UART2 Registers

UART1 Receive Buffer / Transmit Holding / Divisor Latch LSB Register (UART1_RBR0_THR0_DLL1)

UART2 Receive Buffer / Transmit Holding / Divisor Latch LSB Register (UART2_RBR0_THR0_DLL1)

In effect, three registers share the offset address 0x100: UART1_DLL, UART1_THR, and UART1_RBR.

Similarly, three registers share the offset address 0x180: UART2_DLL, UART2_THR, and UART2_RBR.

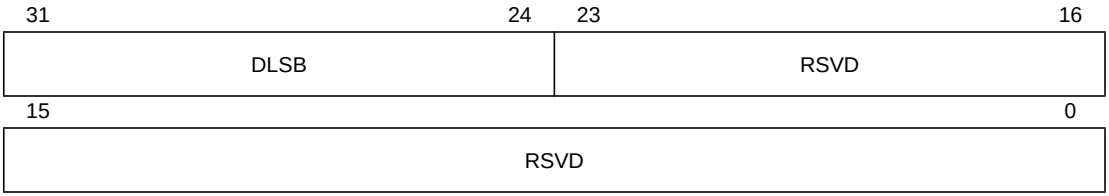
The value of the DLAB bit in the UART* Line Control register determines which register is physically written or read, as shown below:

- Write UART*_DLL when DLAB = 1
- Write UART*_THR when DLAB = 0)
- Read UART*_RBR when DLAB = 0

In any case, each of these three registers is always accessed via the same software name: UART1_RBR0_THR0_DLL1 or UART2_RBR0_THR0_DLL1.

UART1 / UART2 Divisor Latch LSB Register (DLL)

Offset = 0xBE0100 / 0xBE0180 Write only Default = 0x0000 0000



DLSB **Divisor LSB** **31:24**
The baud rate generator implements two 8-bit divisor latches that can be write only accessed by the external host processor or SPARC processor. The processor can load 16-bit data into DLM and DLL (DLM holds the eight most significant bits, DMSB, and DLL holds the eight least significant bits, DLSB) to obtain an output (nBD-

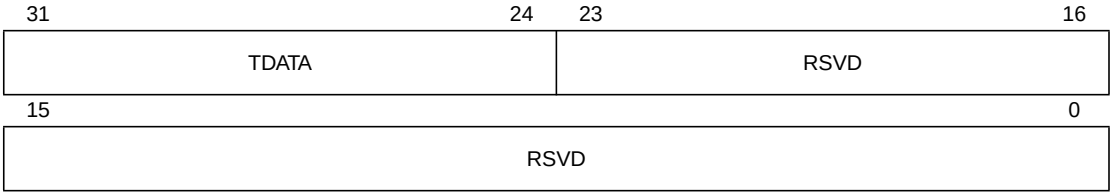
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

OUT) from the baud rate generator that is 16 times the desired baud rate.

Refer to Section 15.4.2, “Baud Rate Generator,” page 15-18 for more information on generating a desired baud rate from a given system clock.

UART1 / UART2 Transmit Holding Register (THR)

Offset = 0xBE0100 / 0xBE0180 Write only Default = 0x0000 0000



TDATA **Transmit Data** **31:24**

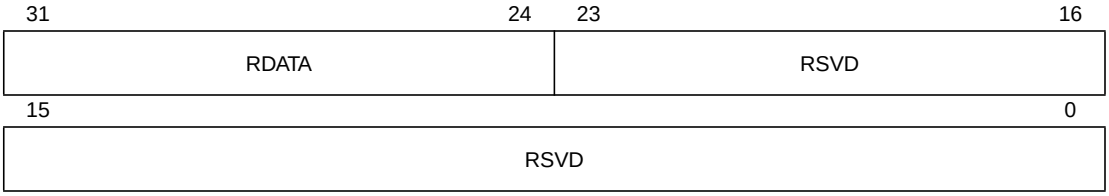
The host processor writes into this register the data byte to be transmitted. In FIFO mode, data bytes to be transmitted are written into the transmit FIFO, which can hold 16 bytes. Depending upon the mode, the data byte from the Transmit Holding Register or from the transmit FIFO is transferred to the Transmit Shift Register at an appropriate time, and is transmitted on the SOUT pin.

After module reset, the Transmit Holding Register is empty, so the THRE bit in the Line Status Register is set. This causes an interrupt to the SPARC processor if the interrupts of the UART are enabled. The SPARC processor can reset this interrupt by either loading data into the Transmit Holding Register, or by reading the Interrupt Identification Register. In the second case, the interrupt is reset only if the highest priority interrupt awaiting service in the UART is the THRE interrupt. Once reset, the THRE bit is set again only when the SPARC processor loads data into the Transmit Holding Register, and then the data is transferred to the Transmit Shift Register. In FIFO mode, the THRE bit is reset when the data is loaded into the transmit FIFO; once reset, it will be set only after the transmit FIFO is empty.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 / UART2 Receive Buffer Register (RBR)

Offset = 0xBE0100 / 0xBE0180 Read only Default = 0x0000 0000



RDATA **Receive Data** **31:24**

The host processor can read received data from this register.

In FIFO mode, received data is stored in the receive FIFO and is made available whenever the SPARC processor reads the Receive Buffer Register. The DR (Data Ready) bit in the Line Status Register monitors the condition of the receive buffer register. When a data byte has been completely received and transferred to the Receive Buffer Register from the Receive Shift Register, the DR bit is set. This can cause an interrupt to the SPARC processor if interrupts are enabled.

The DR bit is reset only when the SPARC processor reads the Receive Buffer Register, or when all the data from the receive FIFO has been drained (in FIFO mode).

UART1 Interrupt Enable Register / Divisor Latch MSB Register (UART1_IER0_DLM1)

UART2 Interrupt Enable Register / Divisor Latch MSB Register (UART2_IER0_DLM1)

In effect, two registers share the offset address 0x104: UART1_DLM and UART1_IER. Similarly, two registers share the offset address 0x184: UART2_DLM and UART2_IER.

The value of the DLAB bit in the UART* Line Control register determines which register is physically written or read, as shown below:

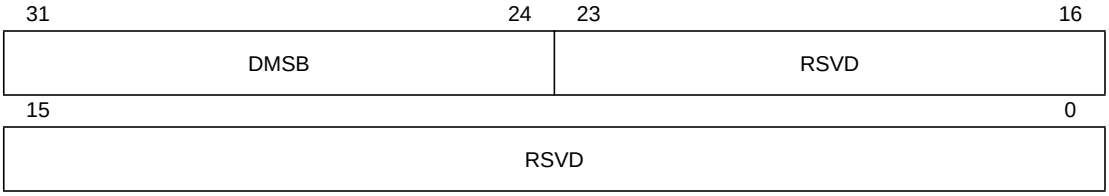
- Write UART*_DLM when DLAB = 1
- Read/Write UART*_IER when DLAB = 0

In either case, these registers are always accessed via the same software name: UART1_IER0_DLM1 or UART2_IER0_DLM1.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 / UART2 Divisor Latch MSB Register (DLM)

Offset = 0xBE0104 / 0xBE0184 Write only Default = 0x0000 0000



DMSB **Divisor MSB** **31:24**

The baud rate generator implements two 8-bit divisor latches that can be write-only accessed by the external host processor or SPARC processor. The processor can load 16-bit data into DLM and DLL (DLM holds the eight most significant bits, DMSB, and DLL holds the eight least significant bits, DLSB) to obtain an output (nBD-OUT) from the baud rate generator that is 16 times the desired baud rate.

Refer to Section 15.4.2, “Baud Rate Generator,” page 15-18 for more information on generating a desired baud rate from a given system clock.

UART1 / UART2 Interrupt Enable Register (IER)

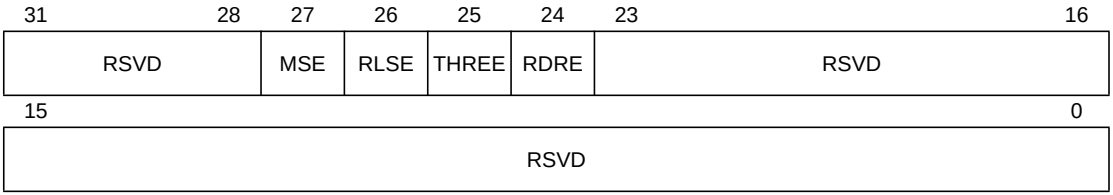
Note: The value of the DLAB bit in the UARTn Line Control Register determines which register gets physically read/written:

UARTn_DLM	when DLAB = 1
UARTn_IER	when DLAB = 0

In either case, these registers are always accessed via the same software name, UART1_IER0_DLM1 or UART2_IER0_DLM1.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Offset = 0xBE0104 / 0xBE0184 Read/Write Default = 0x0000 0000



This register enables the sources of interrupts in the UART to interrupt the SPARC processor. Each of the sources of interrupts can be individually enabled/disabled. Disabling an interrupt does not raise the interrupt signal to the SPARC processor from that particular source of interrupt. The setting of interrupt source bits is independent of the Interrupt Enable Register's bits.

MSE **Modem Status Interrupt Enable** **27**
1 = Enable the modem status interrupt source to activate the interrupt signal of the UART module. This source is the DCTS bit in the Modem Status Register.

RLSE **Receiver Line Status Interrupt Enable** **26**
1 = Enable the receiver line status interrupt sources to activate the interrupt signal of the UART module. These sources are the OE, PE, FE, and BI bits in the Line Status Register.

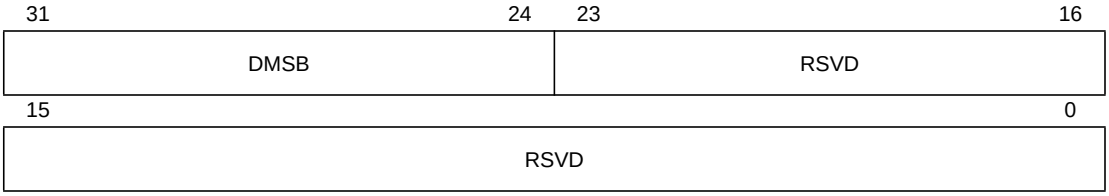
THREE **Transmit Holding Register Empty Interrupt Enable 25**
1 = Enable the THRE bit in the Line Status Register to activate the interrupt signal of the UART.

RDRE **Receiver Data Ready Interrupt Enable** **24**
1 = Enable the DR bit in the Line Status Register to activate the interrupt signal of the UART. The RDRE bit also enables the trigger level interrupt and the time-out interrupt.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 / UART2 Divisor Latch MSB Register (DLM)

Offset = 0xBE0104 / 0xBE0184 Read/Write Default = 0x0000 0000



DMSB **Divisor MSB** **31:24**

The baud rate generator implements two 8-bit latches, called divisor latches, that can be accessed by the host processor for read/write. The host processor can load 16-bit data in these divisor latches to obtain 16*baud frequency output from the baud rate generator. This block implements a 16-bit counter, together with the DLL, to generate the NBDOUT from the input clock.

UART1 Interrupt Identification / FIFO Control Register (UART1_IIR_FCR)

UART2 Interrupt Identification / FIFO Control Register (UART2_IIR_FCR)

In effect, two registers share the offset address 0x108: UART1_IIR, and UART1_FCR.

Similarly, two registers share the offset address 0x188: UART2_IIR, and UART2_FCR.

The read or write operation determines which register is accessed:

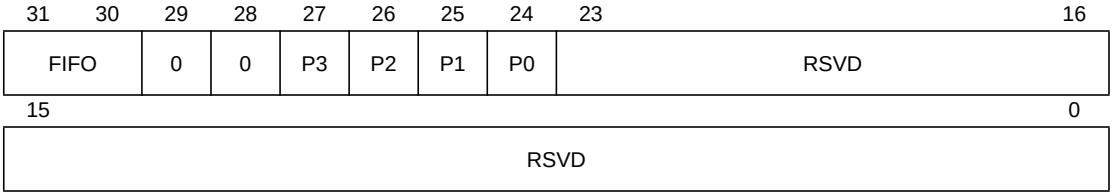
- UART*_IIR is read-only; that is, reading this address returns the contents of UART*_IIR.
- UART*_FCR is write-only; that is, writing to this address changes the contents of UART*_FCR.

In either case, each of these two registers is always accessed via the same software name, UART1_IIR_FCR or UART2_IIR_FCR.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 Interrupt Identification Register (IIR)

Offset = 0xBE0108 / 0xBE0188 Read only Default = 0x0100 0000



The data in this register indicates to the host processor if an interrupt is pending, and identifies the highest priority interrupt in the UART that is requesting service.

FIFO **31:30**
A value of 11 in this field indicates the FIFO mode of operation. Otherwise, these two bits are set to 0.

P3, P2, P1, P0 Priority **27:24**
These bits give the interrupt type and priority, as shown in Table 15.11.

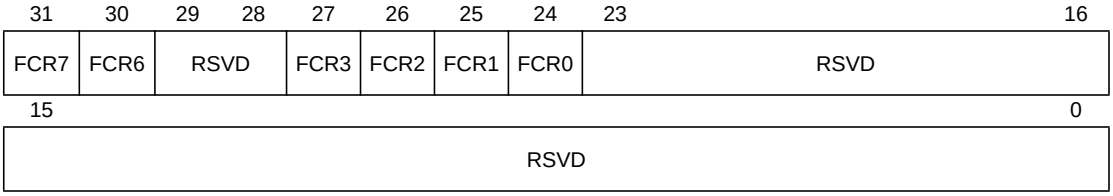
Table 15.11 Interrupt Identification Register Details

P3	P2	P1	P0	Priority Level	Interrupt Type
0	0	0	1	-	None
0	1	1	0	Highest	Receiver Line Status
0	1	0	0	Second	Receiver Data Available or Trigger Level Reached
1	1	0	0	Second	Character Time-out Indication
0	0	1	0	Third	THR Empty
0	0	0	0	Lowest	Modem Status

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 FIFO Control Register (FCR)

Offset = 0xBE0108 / 0xBE0188 Write only Default = 0x0000 0000



FCR[7:6] **Trigger Level** **31:30**
These bits are used to set the trigger levels for the receive FIFO interrupt, as shown in Table 15.12:

Table 15.12 FCR Trigger Levels

FCR7	FCR6	Interrupt Type
0	0	Trigger Level = 01 (default)
0	1	Trigger Level = 04
1	0	Trigger Level = 08
1	1	Trigger Level = 14

FCR[3] **Mode Control / DMA Enable** **27**
1= The internal UART signals nrxrdy and ntxrdy change from mode 0 to mode1 if FCR[0]=1. This bit must be set high if the user wishes to use the DMA Engine to feed/drain the UARTs with data.

FCR[2] **Clear Transmit FIFO** **26**
1 = Clear all bytes in the transmit FIFO and reset its counter to 0. The shift register is not cleared. The 1 that is written to this bit is self-clearing.

FCR[1] **Clear Receive FIFO** **25**
1 = Clear all bytes in the receive FIFO and resets its counter to 0. The shift register is not cleared. The 1 that is written to this bit is self-clearing.

FCR[0] **Enable FIFOs** **24**
1= Enable both the transmit and receive FIFOs.

0 = Clear all bytes in both FIFOs.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

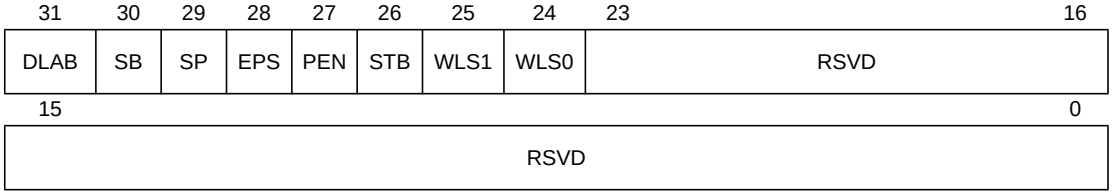
When changing from FIFO mode to NS16450 mode and vice versa, data is automatically cleared from the FIFOs. This bit must be 1 when other FCR bits are written to, or they will not be programmed. If the user wishes to have the DMA Engine automatically fill/drain the UART FIFOs with data, then this bit must be set to 1 to have the FIFOs enabled in the first place.

By default, non-FIFO mode is enabled.

UART1 Line Control Register (UART1_LCR)

UART2 Line Control Register (UART2_LCR)

Offset = 0xBE010C / 0xBE018C Read/Write Default = 0x0000 0000



The UART is configured to operate as follows:

- Transmit 5 bits/character and 1 stop bit
- Parity disabled
- Odd parity selected
- SP is 0, SB is disabled, DLAB is 0

which means that the first access to the DLL-THR-RBR and the DLM-IER registers will not be to baud generator registers.

DLAB **Divisor Latch Select** **31**
1 = Select the divisor latches to be accessed during read/write operations to the DLL-THR-RBR and the DLM-IER registers.

SB **Set Break** **30**
1= The SOUT pin goes low.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SP **Stick Parity** **29**

1 = When in the stick parity mode (SP = high and PEN = high), change the parity bit transmitted to the inverse of the EPS bit programmed in the Line Control Register.

EPS **Even Parity Select** **28**

1 = The total number of 1's in the transmitted data bits plus the parity bit is even.

0 = The total number of 1's in the transmitted data bits plus the parity bit is odd.

PEN **Parity Enable** **27**

1 = Enable the generation and transmission of parity bit in the transmitter and receiver, and the checking of parity bits in the receiver block of the UART.

STB **Stop Bits** **26**

This bit, along with WLS[1:0], sets the number of bits to be transmitted/received per character, and the number of stop bits to be transmitted per character. The receiver checks for the first stop bit only. Details are shown in Table 15.13.

Table 15.13 Character Length and Stop Bits of LCR

WLS1	WLS0	STB	CharLength	Stop Bits
0	0	0	5 Bits	1
0	1	0	6 Bits	1
1	0	0	7 Bits	1
1	1	0	8 Bits	1
0	0	1	5 Bits	1.5
0	1	1	6 Bits	2
1	0	1	7 Bits	2
1	1	1	8 Bits	2

WLS[1:0] **Word Length Select** **25:24**

These bits, along with STB, set the number of bits to be transmitted/received per character, and the number of

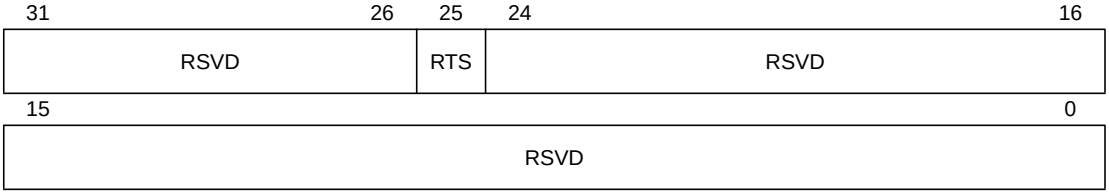
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

stop bits to be transmitted per character. The receiver checks for the first stop bit only. Details are shown in Table 15.13.

UART1 Modem Control Register (UART1_MCR)

UART2 Modem Control Register (UART2_MCR)

Offset = 0xBE0110 / 0xBE0190 Read/Write Default = 0x0000 0000



This register controls the interface with the modem, or with the data set for a peripheral device emulating the modem. After module reset, this register is set to no loopback, and all modem control output pins are high.

RTS **Request To Send** **25**

If hardware flow control is enabled, then the SIO_UART*_RTS pin (and this bit) acts as a function of the receive FIFO's fullness; the software-programmed value is ignored. The output RTS strobe is deasserted (pin high, bit low) if the receive FIFO has only two byte slots "open"; the strobe is reasserted (pin low, bit high) when it is three bytes from full.

If hardware flow control is disabled, this bit controls the SIO_UART*_RTS (request to send) output signal:

1 = The RTS pin is low.

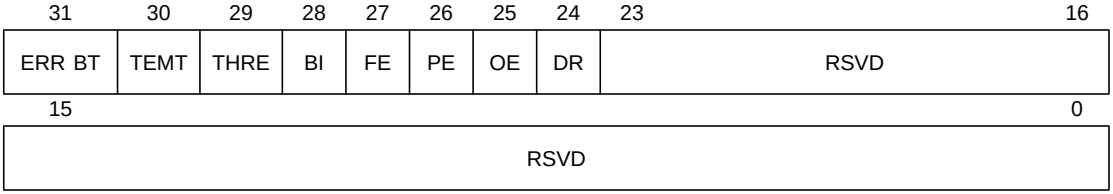
0 = The RTS pin is high.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 Line Status Register (UART1_LSR)

UART2 Line Status Register (UART2_LSR)

Offset = 0xBE0114 / 0xBE0194 Read only Default = 0x60000000



This register provides status information about the UART transmitter and receiver to the host processor. After module reset, the TEMT and THRE bits are high (meaning the transmitter is idle) and all other bits are low (meaning no receiver error conditions exist).

ERRBT **Error Bit** **31**

This bit is always low in non-FIFO mode. In FIFO mode, it is set when there is at least one parity, framing, or break indication in the FIFO.

This bit is reset when the Line Status Register is read, and there are no subsequent errors in the FIFO.

TEMT **Transmitter Empty** **30**

1 = Both the Transmitter Holding Register and the Transmitter Shift Register are empty (transmitter idle condition).

THRE **Transmit Holding Register Empty** **29**

1 = The UART is ready to accept a new character for transmission. This bit causes the UART to issue an interrupt to the host processor when the THRE bit is set in the Interrupt Enable register.

The bit is set by the UART internal logic when the data to be transmitted is shifted from the Transmit Holding Register to the Transmit Shift Register.

This bit is cleared when the system loads data into the Transmit Holding Register.

In FIFO mode, this bit is reset when at least one byte is written to the FIFO, and it is set when the transmit FIFO is empty.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

BI **Break Indicator** **28**

1 = The duration of break transmission is longer than one word transmission time on all word boundaries. The receiver checks only one stop bit, regardless of the number of stop bits programmed. In FIFO mode, only one null character is loaded into the receive FIFO with the framing error. The break error is revealed to the host processor whenever that particular data is read from the receive FIFO.

FE **Framing Error** **27**

1 = The stop bit following the data/parity bit is 0.

This bit is reset whenever the host processor reads the Line Status Register.

In FIFO mode, the framing error is stored for the particular byte received. This error is revealed to the host processor when the associated character is at the top of the FIFO. The UART will try to resynchronize after the framing error. To do this, it assumes that the framing error was due to the next start bit, so it samples this start bit twice and then it takes in the data.

PE **Parity Error** **26**

1 = The received character does not have correct parity. The PE bit is set upon detection of parity error, and is reset when the host processor reads the contents of the Line Status Register. In FIFO mode, this error is associated with the particular character received, and it is revealed to the host processor when the byte is at the top of the FIFO.

OE **Overrun Error** **25**

1 = Data in the Receive Buffer Register was not read by the host processor before the next character was transferred into the register. The OE bit is set when an overrun condition occurs, and is reset when host processor reads the Line Status Register. In FIFO mode, this bit is set when the FIFO is full and the next character is received completely in the Receive Shift Register. The character in the Receive Shift Register is not put into the receive FIFO.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

DR **Data Ready** **24**

1 = A complete incoming character is transferred from the Receive Shift Register into the Receive Buffer Register.

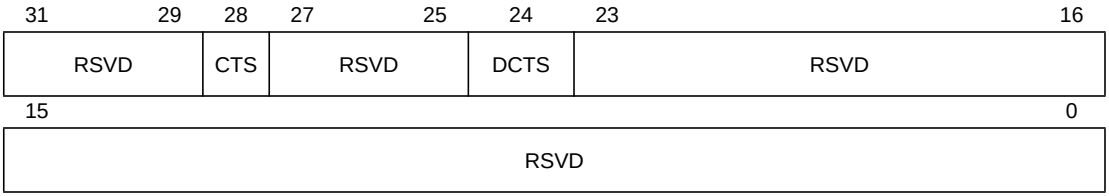
This bit is reset by reading the Receive Buffer Register.

In FIFO mode, this bit is set whenever a character is received and transferred to the receive FIFO. It is reset after reading all the bytes from the receive FIFO.

UART1 Modem Status Register (UART1_MSR)

UART2 Modem Status Register (UART2_MSR)

Offset = 0xBE0118 / 0xBE0198 Read/Write Default = 0x0000 0000



CTS **Clear To Send** **28**

This bit reflects the state of the CTS input of the UART. The CTS bit is the complement of the value coming into the SIO_UART*_CTS pin.

1 = CTS pin is active low

0 = CTS pin is active high

DCTS **Delta CTS** **24**

1= The CTS has changed state.

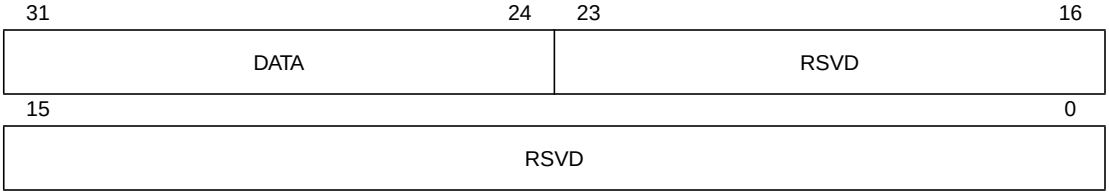
0 = Cleared after the Modem Status Register has been read.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 Scratch Pad Register (UART1_SPR)

UART2 Scratch Pad Register (UART2_SPR)

Offset = 0xBE011C / 0xBE019C Read/Write Default = 0x0000 0000



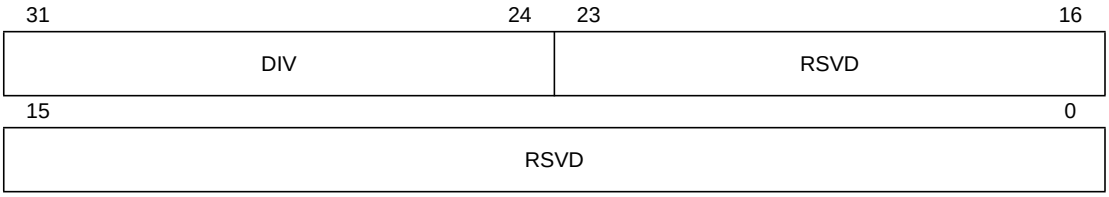
DATA **31:24**

This 8-bit general purpose register does not control the UART operation in any way. It can be used to hold data temporarily.

UART1 External Clock / Prescaler Register (UART1_PRESCALER)

UART2 External Clock / Prescaler Register (UART2_PRESCALER)

Offset = 0xBE0120 / 0xBE01A0 Read/Write Default = 0x0A00 0000



DIV **Clock Divider** **31:24**

This register can alter the frequency of the external input clock, which is fed back into the UART.

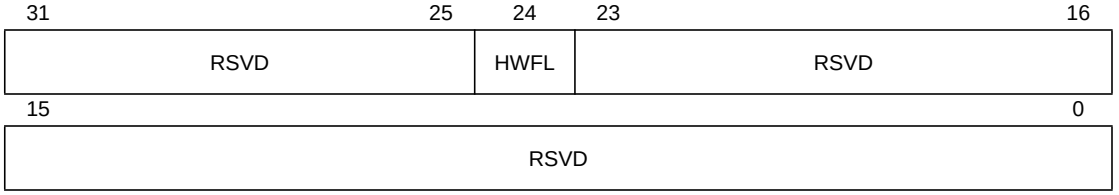
Note: These registers are known as UART1_EXT_CLOCK and UART2_EXT_CLOCK in global.h

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 Hardware Flow Control Register (UART1_HW_FLOW_CTRL)

UART2 Hardware Flow Control Register (UART2_HW_FLOW_CTRL)

Offset = 0xBE0124 / 0xBE01A4 Read/Write Default = 0x0000 0000



HWFL **Hardware Flow Control** **24**

1 = The UART enables hardware flow control. Under hardware flow control, the UART transmit and receive operations operate as follows:

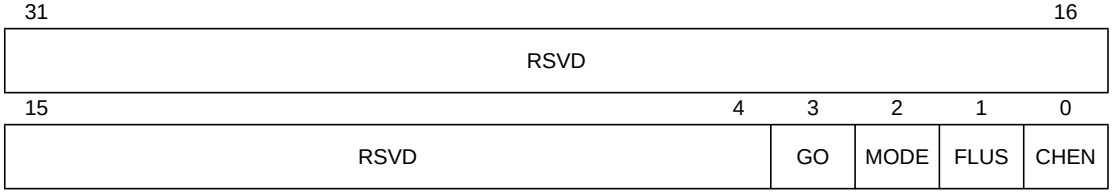
- TX = The input CTS strobe is used as an output enable for a data byte (if CTS is high, then do not send data out on the pins). Race conditions result in data going out.
- RX = The output RTS strobe is deasserted if the receive FIFO is at a level of 2 bytes from full, and is reasserted when it is 3 bytes from full. The software-programmed value in the Modem Control Register is ignored, and the register bit reflects the output on the SIO_UART*_RTS pin.

0 = Hardware flow control is disabled, and the UART operation is normal.

UART1 DMA Transmit Control Register (UART1_TX_CONTROL_REG_ADDR)

UART2 DMA Transmit Control Register (UART2_TX_CONTROL_REG_ADDR)

Offset = 0xBE0140 / 0xBE01C0 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

GO **Load Current Address Buffer Registers** **3**
This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**
This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**
This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

1 = For transmit channels, all data currently in the channel FIFOs is dropped.

1 = For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

CHEN **DMA Channel Enable** **0**
This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

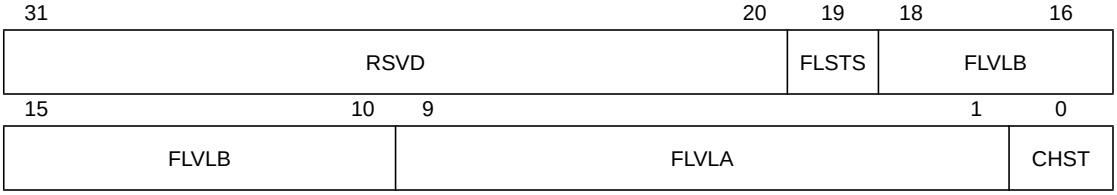
CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 DMA Transmit Status Register (UART1_TX_STATUS_REG_ADDR)

UART2 DMA Transmit Status Register (UART2_TX_STATUS_REG_ADDR)

Offset = 0xBE0144 / 0xBE01C4 Read/Write Default = 0x0000 0000



FLSTS **FIFO Flush Status** **19**
1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**
This field shows the number of bytes currently held in FIFO B.

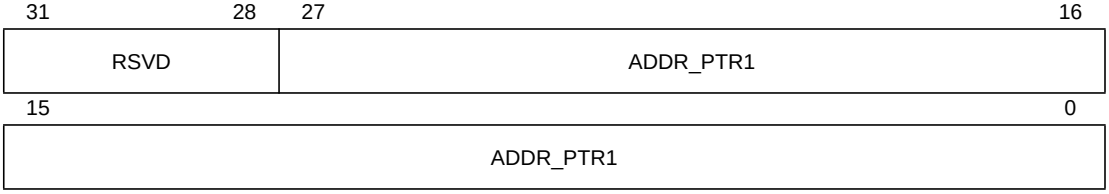
FLVLA **FIFO A Byte Count** **9:1**
This field shows the number of bytes currently held in FIFO A.

CHST **DMA Channel Status** **0**
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

UART1 DMA Transmit Address Pointer1 Register (UART1_TX_ADDR_PTR1_ADDR)

UART2 DMA Transmit Address Pointer1 Register (UART2_TX_ADDR_PTR1_ADDR)

Offset = 0xBE0148 / 0xBE01C8 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

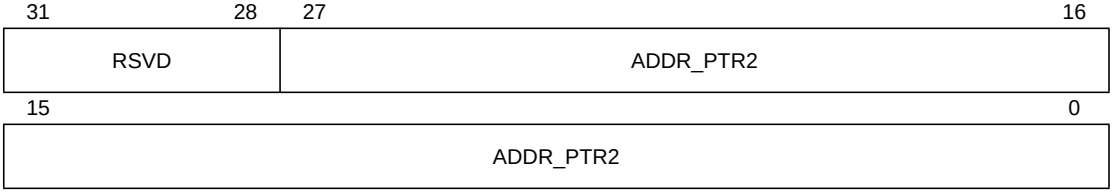
ADDR_PTR1 Address Pointer 1 27:0
In double-buffer mode, this register indicates the Base Address for the “next” SDRAM Buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

UART1 DMA Transmit Address Pointer2 Register (UART1_TX_ADDR_PTR2_ADDR)

UART2 DMA Transmit Address Pointer2 Register (UART2_TX_ADDR_PTR2_ADDR)

Offset = 0xBE014C / 0xBE01CC Read/Write Default = 0x0000 0000



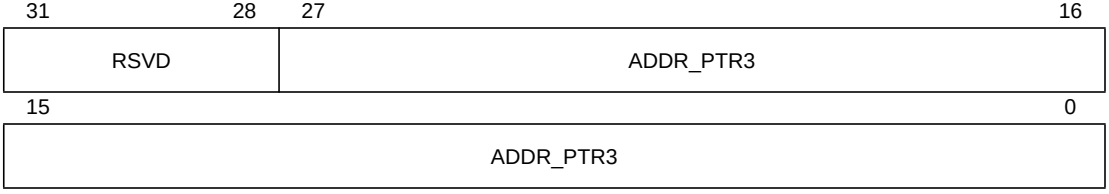
ADDR_PTR2 Address Pointer 2 27:0
In double-buffer mode, this register indicates the End Address for the “next” SDRAM Buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

UART1 DMA Transmit Address Pointer3 Register (UART1_TX_ADDR_PTR3_ADDR)

UART2 DMA Transmit Address Pointer3 Register (UART2_TX_ADDR_PTR3_ADDR)

Offset = 0xBE0150 / 0xBE01D0 Read/Write Default = 0x0000 0000



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ADDR_PTR3 Address Pointer 3 27:0

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

For read channels (receive), it indicates the current value of the read pointer.

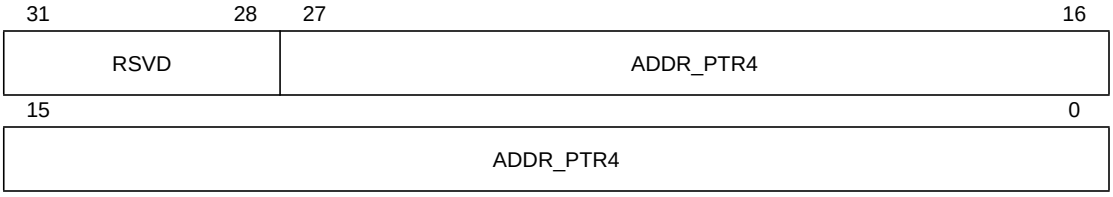
In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

UART1 DMA Transmit Address Pointer4 Register (UART1_TX_ADDR_PTR4_ADDR)

UART2 DMA Transmit Address Pointer4 Register (UART2_TX_ADDR_PTR4_ADDR)

Offset = 0xBE0154 / 0xBE01D4 Read/Write Default = 0x0FFF FFFF



ADDR_PTR4 Address Pointer 4 27:0

In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

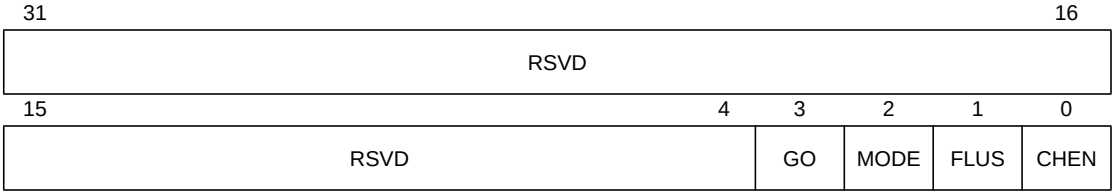
- The DMA channel is idle.
- The previous SDRAM buffer has completed.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 DMA Receive Control Register (UART1_RX_CONTROL_REG_ADDR)

UART2 DMA Receive Control Register (UART2_RX_CONTROL_REG_ADDR)

Offset = 0xBE0160 / 0xBE01E0 Read/Write Default = 0x0000 0000



GO **Load Current Address Buffer Registers** **3**

This bit is used to control when ADDR_PTR1 and ADDR_PTR2 for a particular DMA channel are loaded into the “current buffer” address registers (ADDR_PTR3 and ADDR_PTR4) in double-buffer mode operation.

Note: DoMiNo always runs in double-buffer mode. Single-buffer (ring buffer) mode is not supported.

Go is set by software after the address range for the “next buffer” is programmed (ADDR_PTR1/ADDR_PTR2).

Go is cleared by the hardware after these “next values” are loaded. It is also cleared on reset.

MODE **Transfer Mode Select** **2**

This bit must always be set to ‘1’ so that the channel operates in double-buffer mode.

FLUS **FIFO Flush** **1**

This is a self-clearing bit for flushing the DMA channel’s two internal FIFOs.

1 = For transmit channels, all data currently in the channel FIFOs is dropped.

1 = For receive channels, all data currently in the channel FIFOs is sent to the SDRAM address indicated by ADDR_PTR3.

Setting FLUS does not terminate a transfer; it merely dumps data (if transmit) or sends whatever data remains in the FIFOs up to the SDRAM (if receive).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

CHEN **DMA Channel Enable** **0**

This bit enables/disables the DMA channel. If CHEN is cleared during a DMA operation, DMA is paused. Set CHEN again to resume the DMA operation. All state information is preserved when DMA is paused, and the DMA operation continues from where it left off.

CHEN is controlled by software. It must be cleared once the DMA operation has finished. On reset, CHEN is 0.

UART1 DMA Receive Status Register (UART1_RX_STATUS_REG_ADDR)

UART2 DMA Receive Status Register (UART2_RX_STATUS_REG_ADDR)

Offset = 0xBE0164 / 0xBE01E4 Read/Write Default = 0x0000 0000

FLSTS **FIFO Flush Status** **19**

1 = Flushing of a particular DMA channel is complete.

Software must reset FLSTS by writing 0.

FLVLB **FIFO B Byte Count** **18:10**

This field shows the number of bytes currently held in FIFO B.

FLVLA **FIFO A Byte Count** **9:1**

This field shows the number of bytes currently held in FIFO A.

CHST **DMA Channel Status** **0**

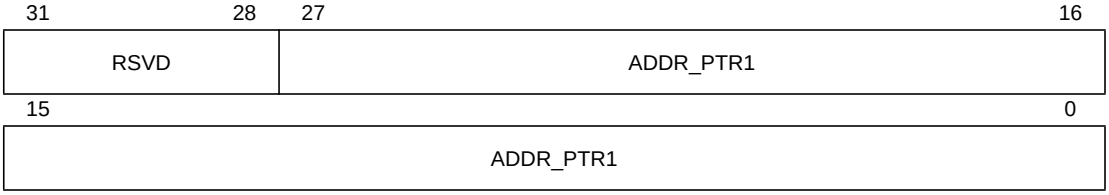
1 = The DMA transfer operation has finished. CHST can be cleared by writing 1 to it.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 DMA Receive Address Pointer1 Register (UART1_RX_ADDR_PTR1_ADDR)

UART2 DMA Receive Address Pointer1 Register (UART2_RX_ADDR_PTR1_ADDR)

Offset = 0xBE0168 / 0xBE01E8 Read/Write Default = 0x0000 0000



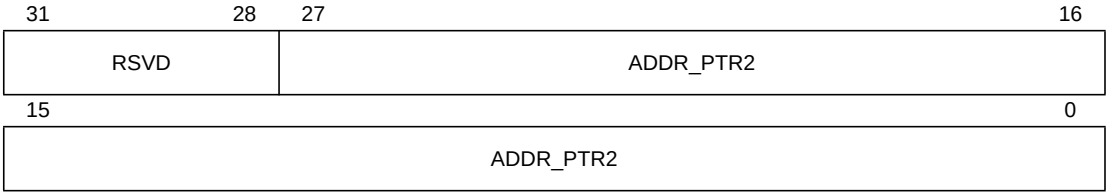
ADDR_PTR1 Address Pointer 1 27:0
In double-buffer mode, this register indicates the Base Address for the “next” SDRAM Buffer about to be transferred.

Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

UART1 DMA Receive Address Pointer2 Register (UART1_RX_ADDR_PTR2_ADDR)

UART2 DMA Receive Address Pointer2 Register (UART2_RX_ADDR_PTR2_ADDR)

Offset = 0xBE016C / 0xBE01EC Read/Write Default = 0x0000 0000



ADDR_PTR2 Address Pointer 2 27:0
In double-buffer mode, this register indicates the End Address for the “next” SDRAM Buffer about to be transferred.

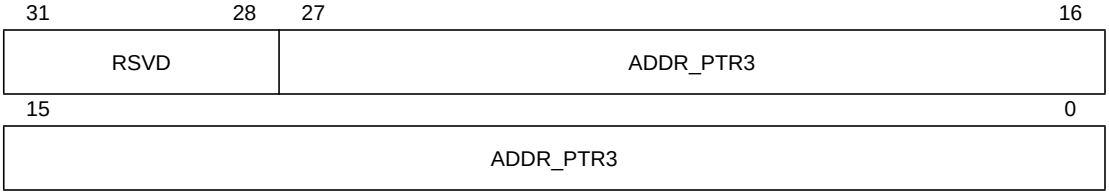
Note: The maximum size for each SDRAM buffer transfer is 511 bytes. Therefore, the difference between ADDR_PTR2 and ADDR_PTR1 should not exceed 511.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

UART1 DMA Receive Address Pointer3 Register (UART1_RX_ADDR_PTR3_ADDR)

UART2 DMA Receive Address Pointer3 Register (UART2_RX_ADDR_PTR3_ADDR)

Offset = 0xBE0170 / 0xBE01F0 Read/Write Default = 0x0000 0000



ADDR_PTR3 Address Pointer 3 27:0

This register is updated by hardware during a DMA operation.

For write channels (transmit), this register indicates the current value of the write pointer in SDRAM.

For read channels (receive), it indicates the current value of the read pointer.

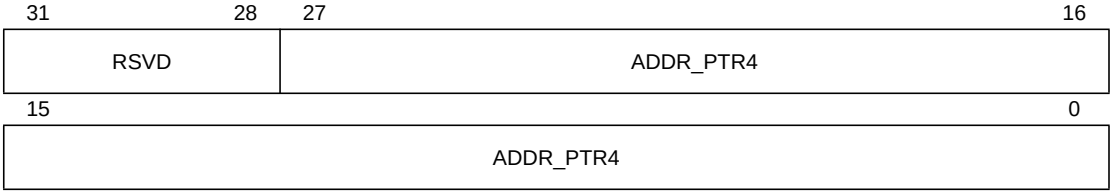
In double-buffer mode, this register is loaded with the contents of ADDR_PTR1 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

UART1 DMA Receive Address Pointer4 (UART1_RX_ADDR_PTR4_ADDR)

UART2 DMA Receive Address Pointer4 (UART2_RX_ADDR_PTR4_ADDR)

Offset = 0xBE0174 / 0xBE01F4 Read/Write Default = 0x0FFF FFFF



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ADDR_PTR4 Address Pointer 4 27:0

In double-buffer mode, this register is loaded with the contents of ADDR_PTR2 if Go is high and if either of the following is true:

- The DMA channel is idle.
- The previous SDRAM buffer has completed.

15.5.7 Reserved Registers (SIO UART2)

UART2 Modem Control Register (UART2_MCR)

Offset = 0xBE0190 Read/Write Default = 0x0000 0000

UART2 Modem Status Register (UART2_MSR)

Offset = 0xBE0198 Read/Write Default = 0x0000 0000

UART2 Hardware Flow Control Register (UART2_HW_FLOW_CTRL)

Offset = 0xBE01A4 Read/Write Default = 0x0000 0000

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 16
Clock Control and Power Management

This chapter describes the clock control and power management for the DMN-8600 and contains the following sections:

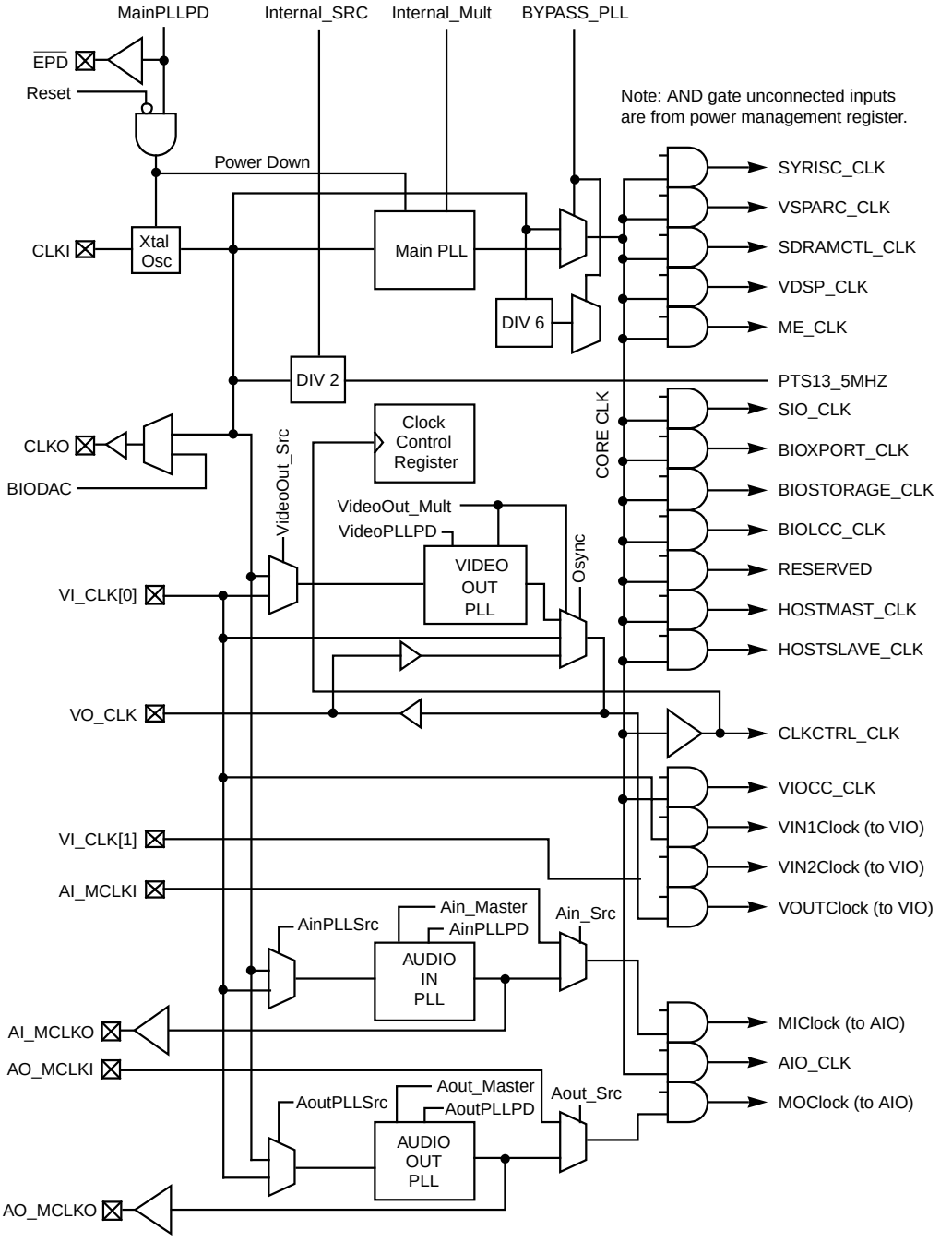
- Section 16.1, “Clock and Power Registers”
- Section 16.2, “Main PLL Power Down and Wake-Up Sequence”

As shown in Figure 16.1, the DMN-8600 has four clock generation PLLs: main PLL, Video Out PLL, Audio Out PLL and Audio In PLL. An on-chip software adjustable 13.5 MHz crystal oscillator or externally generated 13.5 or 27 MHz system clocks are used as the reference frequency for the PLLs. The 13.5 MHz reference frequency is also used as the clock for the PTS time stamp counters in the AV I/O and BIO units.

The DMN-8600 has a full power-down mode which shuts down all PLLs and logic operating from the internally generated clocks. Power down mode is exited by hardware reset. If SDRAM is in standby-refresh power down mode, then the contents are preserved; otherwise, the contents are lost. External wake-up events may be selected to automatically generate a wake up reset. In addition, 14 submodules may be individually powered down by gating off their clock tree. The submodules can be powered up without resetting the entire chip.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 16.1 Clock and Power Block Diagram



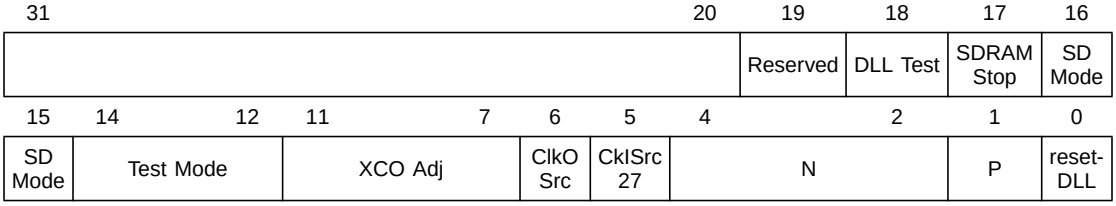
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

16.1 Clock and Power Registers

All registers shown in this section are clocked by the CLKCTRL_CLK and are NOT disabled unless the Main PLL is powered down.

Internal Clock Control Register

Cbus Address: 0xC20008



DLL Test **18**

Enables DLL test mode when set. Should not be set by software. This field is reset to zero.

SDRAM Stop **17**

SDRAM Stop. When software writes a one to this bit, pending SDRAM transactions are completed and the sdrams are placed in standby-refresh power down mode. Before setting this bit software must stop all internal DMA operations or the chip will hang. After setting this bit, the SPARC processors must only access the master or CBus address space. Software should poll this register until this bit reads as one to be sure that the sdram is actually stopped. When software writes zero to this bit (and the bit is set), the sdrams are taken out of standby refresh mode. Software should poll this register until this bit reads as zero to be sure that the sdram is actually enabled. The SDRAMs should not be enabled until the the internal clock is stable (1 ms after reset or frequency change) and the DLL has been reset (10 ms after reset-DLL is set). This bit is reset to one (SDRAM disabled). Before performing normal accesses after a power on reset, the drams must be initialized by setting the DRAM clock control register, followed by the DRAM configuration register 1 msec later (see DRAM interface specification). This field is reset to one

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

SD Mode **[16:15]**

Sigma-Delta Mode. Selects sigma delta noise shaping algorithm for the audio PLLs. This field should normally be set to zero. Default value equals 0x0.

Test Mode **[14:12]**

Enables crystal oscillator, system PLL, video PLL, audioIn PLL and audioOut PLL outputs to be sent to the TDO pin for testing purposes as shown below. This field is reset to zero by software. Whenever this field is not 0, the TDO output enable would be asserted also. Default value equals 000

Test Mode Value	MIclockO Output
0	Normal audio input master clock
1	Crystal oscillator out
2	System PLL VCO out
3	VideoOut PLL VCO out
4	AudioIn PLL VCO out
5	AudioOut PLL VCO out

XCO Adj **[11:7]**

The frequency offset of the internal crystal oscillator from its nominal 13.5 MHz operating point in sign magnitude format in the range of –15 to +15. The maximum positive and negative offsets correspond to a frequency adjustment of ± 100 ppm. The frequency adjustment can only be applied when a 13.5 MHz crystal is used. This field is reset to 01111. For proper operation of the crystal oscillator, software should not change the value of Adj[4:0] by more than one count every 100 ms. For example, if the current value of Adj[4:0] is 00000, but the desired final value is 3, the following update sequence should be issued:

Change Adj[4:0] to 1; Wait at least 100 ms

Change Adj[4:0] to 2; Wait at least 100 ms

Change Adj[4:0] to 3

Default value equals 0x0F

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

ClkO Src **6**

If clear, then the clkO/DAC pin is driven by the output of the internal adjustable crystal oscillator to provide a system clock reference. If set, the clkO/DAC pin is driven by the BIO sigma/delta DAC as specified by the TCdacCtl register to control an external VCXO. This bit is reset to zero.

ClkISrc27 **5**

If set, the clock input pin (CLKI) is divided by two before being applied to the PLLs, PTS and transport stream time stamp counters. This bit should be set if the external clock input or crystal frequency is 27 MHz, otherwise it is assumed to be 13.5 MHz. The 13.5 MHz clock is divided by 150 before being accumulated in the video PTS counter register. This bit is reset to one.

The internal clock operating frequency is determined by N and P as follows:

$$\text{internalClock} = \frac{\text{SysclkPLL(N)}}{P + 1}$$

SysclkPLL(N) is determined by the table below.

N	SysclkPLL (Mhz)
0	108
1	121.5
2	135
3	148.5
4	162
5	175.5
6	189
7	202.5

After reset, N is set to 4 and P is set to 1. Software must change N and P to correct values after reset. Changing the internal clock frequency will cause DRAM timing violations particularly when increasing the frequency. Before changing the internal clock frequency, SDRAM activity must be stopped by setting the SDRAM stop bit and reading it until it reads as one. DRAM activity including cache misses and DMA transfers should be avoided until the DLLs are reset. The resetDLL bit is a

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

self clearing bit which must be set by software at least 1 ms after any internal clock frequency change (including starting the internal clock PLL on reset). The SDRAM stop bit can be cleared 10 us after the resetDLL bit is set. SDRAM activity can resume after the SDRAM stop bit is read as zero. The internal clock frequency should not be set below 81 MHz, or DRAM timing violations may occur.

N **[4:2]**

See above. This field is reset to 0x4.

P **1**

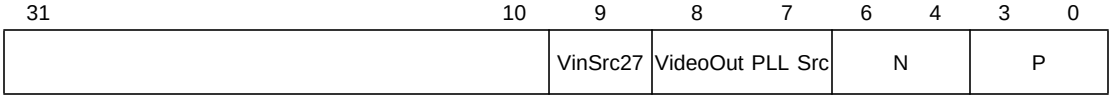
See above. Default value equals 1.

resetDLL **0**

See above. Default value equals 0.

Video Output Clock Control Register

Memory Space Address: 0xC2000C



VinSrc27 **9**

If set, the Video Input 1 Clock pin is divided by two before being applied to the PLLs. This bit should be set if the external Video Input 1 Clock pin is 27 MHz, otherwise it is assumed to be 13.5 MHz. After power up, this bit is 1.

VideoOut PLL Src **[8:7]**

Specifies the video output PLL clock source as shown below. If the source is internal or video input 1 clock, then the video out clock pin becomes an output of the video output clock. This field is reset to two.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

VideoOut PLL Src Value	PLL Source Clock	Application
0	If Osync clear, Clk pin; else Video Input 1 Clock pin	Internally generated video output timing
1	Video Input 1 Clock pin	Video Out locked to video input stream 1. Both video input clocks should be the same. Implementation note: the video output clock pin must have minimal skew to the input clock pin.
2	N/A	Externally generated video output clock

When Osync is set, the video output PLL is bypassed, and the video output clock is the Video Input 1 clock. When Osync is clear and the video out PLL src field is two, the video output PLL is bypassed and the video output clock is supplied by the video out clock pin. When the video out PLL src field is zero or one and Osync is clear, the video output clock is internally generated with an operating frequency is determined by N and P as follows:

$$VideoOutClock = \frac{VidOutPLL(N)}{P+1}$$

VidOutPLL(N) is determined by the table below.

N	Output Frequency
0	74.25 MHz
1	74.25/1.001 MHz
2	216 MHz

After reset, N and P are set to 0.

N **[6:4]**

See above. This field is reset to 0.

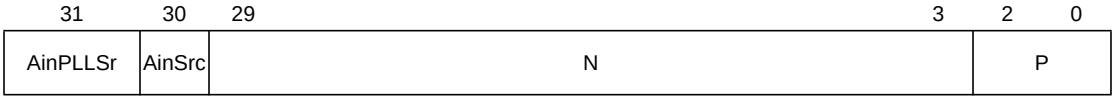
P **[3:2]**

See above. This field is reset to 0.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Audio Input Clock Control Register

Cbus Address: 0xC20010



AinPLLSr **31**

If unset, the Ain PLL uses the internal CLKI as reference clock. If set, uses Video Input 1 Clock pin as reference clock. After power up, it is reset to ‘0’

AinSrc **30**

If set, the internally generated master input clock with the clock frequency specified by the Ain Master field is used. If clear, then an external master input clock is used, as supplied on the AI_MCLKI pin. In either case, the internally generated master input audio clock is output on the AI_MCLKO pin. At reset, this bit set if the PLL_BYPASS pin is zero, otherwise it is reset.

The internal audio input master clock operating frequency is determined by N and P as follows:

$$\text{AudioInClock} = 13.5\text{MHz}\frac{N+1}{2(P+1)}$$

N is interpreted as a 3.24 bit fractional number. In DoMiNo, N values are restricted so that 13.5 MHz * (N + 1)/2 is within ± 100 ppm of one of the frequencies shown below:

88.2 * 384 kHz

96 * 384 kHz

176.4 * 256 kHz

192 * 256 kHz

After reset, N is set to 4.0176 and P is set to 3.

N **[29:3]**

See above. This field is reset to 0x404816F.

P **[2:0]**

See above. This field is reset to 0x3.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Audio Output Clock Control Register

Cbus Address: 0xC20014

31	30	29 ... 3	2 ... 0
AoPLLSr	AoutSrc	N	P

AoPLLSr **31**

If unset, the Ain PLL uses the internal CLKI as reference clock. If set, uses Video Input 1 Clock pin as reference clock. After power up, it is reset to ‘0’

AoutSrc **30**

If set, the internally generated master output clock with the clock frequency specified by the Aout Master field is used for IEC-958 output timing and internally derived output bit clock and output frame clock timing. If clear, then an external master output clock is used, as supplied on the MOclockI pin. In either case, the internally generated master audio clock is output on the MOclockO pin. At reset, this bit set if the PLL_BYPASS pin is zero, otherwise it is reset.

The internal audio input master clock operating frequency is determined by N and P as follows:

$$AudioInClock = 13.5\ \text{MHz}\ \frac{N+1}{2(P+1)}$$

N is interpreted as a 3.24 bit fractional number. In DoMiNo, N values are restricted so that 13.5 MHz * (N + 1)/2 is within ± 100 ppm of one of the frequencies shown below:

88.2 * 384 kHz

96 * 384 kHz

176.4 * 256 kHz

192 * 256 kHz

After reset, N is set to 4.0176 and P is set to 3.

N **[29:3]**

See above. This field is reset to 0x404816F.

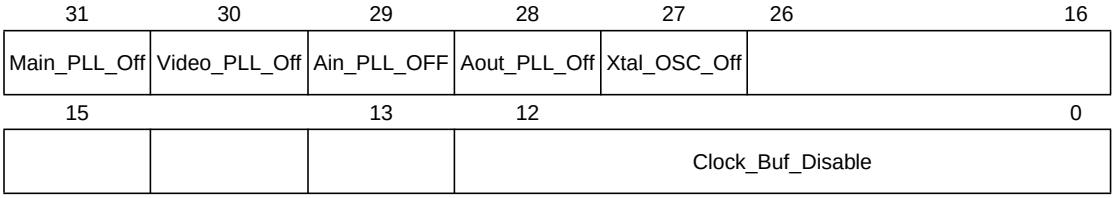
P **[2:0]**

See above. This field is reset to 0x3.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Power Management Register

Cbus Address: 0xC20000



Note: The Power Management Register at control bus address 0xC20000 is used to selectively disable DMN-8600 modules as well as power down all internally clocked logic. After reset, all PLLs and modules are enabled so that all subblocks can be reset. After the power up sequence is completed, individual units can be disabled by software to save power. All unused bits should be written with 0 and are read as 0.

Main_PLL_Off **31**

Power down main PLL, crystal oscillator and internally clocked logic when set. Can only be cleared by Reset. Since Main PLL may not be running during reset, the power down signal to the Main PLL is overridden by reset. Programming note: An external host, rather than the internal SPARCs, should set this bit. After the host sets this bit, it should not perform a host access for at least 10 microseconds.

Video_PLL_Off **30**

Power down video out PLL, and internally clocked video output processing when set. When cleared, video output timing will not be stable for 1 ms.

Ain_PLL_Off **29**

Power down Audio In PLL when set. When cleared, audio input PLL timing will not be stable for 1 ms.

Aout_PLL_Off **28**

Power down Audio Out PLL when set. When cleared, audio output PLL timing will not be stable for 1 ms.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Xtal_OSC_Off **27**

Power down the internal crystal oscillator when set. This bit can be set when an external clock source is used. If there is no external clock source and this bit is set, the chip will hang.

Clock_Buf_Disable **[12:0]**

When set, corresponds to the clock disable for SMARTCARD_CLK, AIO_CLK, VIO_CLK, HOSTSLAVE_CLK, ASYNCMAST_CLK, BIOLCC_CLK, BIOSTORAGE_CLK, BIOXPORT_CLK, SIO_CLK, ME_CLK, VDSP_CLK, VSPARC_CLK, and SYRISC_CLK. Clocks should only be disabled when the associated unit is idle – i.e., no outstanding Cbus/memory transactions or pending operations. Clock_Buf_Disable[11] also disables the MIclock, and MOclock buffer inside AIO. Clock_Buf_Disable[10] also disables VIN1Clock, VIN2Clock VOUTClock (the PTS clock is not disabled). This field is reset to 0x0000.

Wake Up Source Register

Memory Space Address: 0xC20004

31 — 4	3	2	1	0
Reserved	GPIO[0] WE	UART WE	IR WE	1394 WE

Note: The Wake Up Source register at control bus address 0xC20004 is used to enable or disable wake-up events. It is reset to zero.

GPIO[0] WE **3**

WakeUp Enable. If set, an interrupt event on GPIO[0] will wake up DoMiNo; otherwise, it is ignored in power down state.

0UART WE **2**

UART WakeUp Enable. If set, a falling edge on SIO_UART1_RX will wake up DoMiNo, otherwise it is ignored in power down state.

IR WE **1**

IR WakeUp Enable. If set, a falling edge on SIO_IRRX will wake up DoMiNo, otherwise it is ignored in power down state.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

1394 WE	**0**
1394 WakeUp Enable. If set, the 1394 link on signal (BIO_LINK_ON) will wake up DoMiNo; otherwise, the link on signal is ignored.	

16.2 Main PLL Power Down and Wake-Up Sequence

Prior to placing the DMN-8600 in a power-down state, all internal units should be idle (other than async host or async master/system RISC, which are writing to the power management register). DRAM should be placed in the power-down state by setting the DRAM stop bit in the Internal Clock Control register after all DRAM transactions have been suspended. The MAIN PLL should be the last thing to be powered down. When in the power-down state, the only way to wake up the DMN-8600 is through reset, which may be generated in one of the following ways:

- The DMN-8600 reset pin asserted.
- Slave Host wake-up event. When the DMN-8600 is in power-down mode, the DMN-8600 asynchronously monitors the $\overline{\text{H\_CS}}$ pin and generates a reset. If the slave host interface is not selected by the Mode pins, then this function is disabled. The toggling of $\overline{\text{H\_CS}}$ must be part of a legal DoMiNo access cycle; otherwise, the chip may hang after wake up.
- 1394 wake-up event. When the DMN-8600 is in power-down mode and 1394 wake up is enabled, the DMN-8600 asynchronously monitors the LinkOn signal from the 1394 Phy chip and generates a reset on a rising edge.
- IR receive wake-up event. When the DMN-8600 is in power-down mode and IR wake up is enabled, the DMN-8600 asynchronously monitors SIO_IRRX and generates a reset on a falling edge.
- UART1 input wake-up event. When the DMN-8600 is in power-down mode and UART1 wake up is enabled, the DMN-8600 asynchronously monitors the SIO_UART1_RX signal and generates a reset on a falling edge.
- GPIO[0] input wake-up event. When the DMN-8600 is in power-down mode and GPIO[0] wakeup is enabled, the DMN-8600 asynchronously monitors the GPIO[0] signal and generates a reset on an interrupt event. The type of interrupt event is selected by the GPIO[0] field in the Interrupt/GPIO Configuration register.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 17
JTAG Boundary Scan

this chapter describes the JTAG boundary scan and contains the following sections:

The DMN-8600 processor implements a JTAG boundary scan interface in accordance with IEEE 1149.1. The optional TRST pin is provided to simplify resetting the tap controller in systems that do not use the JTAG port. The tap controller must be reset after power-up to enable normal processor operation.

Note: The Tap Controller is not reset by RST.

DMN-8600 boundary scan supports only board-level testing (for example, SAMPLE/PRELOAD and EXTEST instructions); component testing (for example, the INTEST instruction) is not supported.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

17.1 JTAG Instruction Set

The DMN-8600 processor JTAG instructions are three bits in length, encoded as shown in Table 17.1.

Table 17.1 JTAG Instruction Set

Opcode[2:0]	Instruction
000	EXTEST
001	SAMPLE/PRELOAD
010–1110	Private
111	BYPASS

The Private instructions are “hazardous” as defined by 1149.1 and should not be used.

Note: The DMN-8600 processor does not load the BYPASS instruction into the tap controller's instruction register when the controller is in the test-logic-reset state. The instruction loaded instead selects a 32-bit data register (set to 0x1FFFFFFF when exiting the Capture-DR state) between TDI and TDO.

17.2 Boundary Scan Chain Cells

The Boundary Scan Chain Cells are listed in Table 17.2. Cell #1 is closest to TDO, cell #595 is closest to TDI. Note: There are two flip-flops for each pad on the chain: one control flip-flop and one data flip-flop. Not all pads are on the boundary scan chain. Open-drain outputs can be driven HIGH via the EXTEST instruction. If this is hazardous to system

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

operation, the enable cell and corresponding data cell should never both be set to “1” during EXTEST.

Table 17.2 Boundary Scan Chain Cells

Cell	Pin	Name	Type
0	–	Control	[CTRL]
1	U1	$\overline{\text{ATAPI\_DMAACK}}$	[IN/OUT]
2	–	Control	[CTRL]
3	R4	ATAPI_DATA[0]	[IN/OUT]
4	–	Control	[CTRL]
5	T2	ATAPI_DATA[13]	[IN/OUT]
6	–	Control	[CTRL]
7	R3	ATAPI_DATA[1]	[IN/OUT]
8	–	Control	[CTRL]
9	T1	$\overline{\text{ATAPI\_DIOW}}$	[OUT]
10	–	Control	[CTRL]
11	R2	ATAPI_DATA[12]	[IN/OUT]
12	–	Control	[CTRL]
13	P4	ATAPI_DATA[2]	[IN/OUT]
14	–	Control	[CTRL]
15	R1	ATAPI_DATA[11]	[IN/OUT]
16	–	Control	[CTRL]
17	P3	ATAPI_DATA[3]	[IN/OUT]
18	–	Control	[CTRL]
19	P2	ATAPI_DATA[10]	[IN/OUT]
20	–	Control	[CTRL]
21	P1	ATAPI_DATA[4]	[IN/OUT]
22	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
23	N4	ATAPI_DATA[5]	[IN/OUT]
24	–	Control	[CTRL]
25	N3	ATAPI_DATA[9]	[IN/OUT]
26	–	Control	[CTRL]
27	N2	ATAPI_DATA[6]	[IN/OUT]
28	–	Control	[CTRL]
29	N1	ATAPI_DATA[7]	[IN/OUT]
30	–	Control	[CTRL]
31	M4	ATAPI_DATA[8]	[IN/OUT]
32	–	DUMMY	[N/A]
33	–	DUMMY	[N/A]
34	–	DUMMY	[N/A]
35	–	DUMMY	[N/A]
36	–	DUMMY	[N/A]
37	–	DUMMY	[N/A]
38	–	DUMMY	[N/A]
39	–	DUMMY	[N/A]
40	–	DUMMY	[N/A]
41	–	DUMMY	[N/A]
42	–	DUMMY	[N/A]
43	–	DUMMY	[N/A]
44	–	DUMMY	[N/A]
45	–	DUMMY	[N/A]
46	–	DUMMY	[N/A]
47	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
48	–	DUMMY	[N/A]
49	–	DUMMY	[N/A]
50	–	Control	[CTRL]
51	M3	BIO_PHY_DATA[2]	[IN/OUT]
52	–	Control	[CTRL]
53	M2	BIO_PHY_DATA[3]	[IN/OUT]
54	–	DUMMY	[N/A]
55	–	DUMMY	[N/A]
56	–	Control	[CTRL]
57	M1	BIO_LREQ	[OUT]
58	–	DUMMY	[N/A]
59	–	DUMMY	[N/A]
60	–	Control	[CTRL]
61	L4	BIO_LPS	[OUT]
62	–	Control	[CTRL]
63	L3	BIO_PHY_DATA[1]	[IN/OUT]
64	–	Control	[CTRL]
65	L1	BIO_PHY_CLK	[IN]
66	–	Control	[CTRL]
67	L2	BIO_PHY_CTL[1]	[IN/OUT]
68	–	Control	[CTRL]
69	K4	BIO_LINK_ON	[IN]
70	–	Control	[CTRL]
71	K3	BIO_PHY_DATA[7]	[IN/OUT]
72	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
73	K2	BIO_PHY_DATA[6]	[IN/OUT]
74	–	Control	[CTRL]
75	K1	BIO_PHY_DATA[5]	[IN/OUT]
76	–	Control	[CTRL]
77	J1	BIO_PHY_DATA[4]	[IN/OUT]
78	–	Control	[CTRL]
79	J2	BIO_PHY_DATA[0]	[IN/OUT]
80	–	Control	[CTRL]
81	J3	BIO_PHY_CTL[0]	[IN/OUT]
82	–	DUMMY	[N/A]
83	–	DUMMY	[N/A]
84	–	DUMMY	[N/A]
85	–	DUMMY	[N/A]
86	–	DUMMY	[N/A]
87	–	DUMMY	[N/A]
88	–	DUMMY	[N/A]
89	–	DUMMY	[N/A]
90	–	DUMMY	[N/A]
91	–	DUMMY	[N/A]
92	–	DUMMY	[N/A]
93	–	DUMMY	[N/A]
94	–	DUMMY	[N/A]
95	–	DUMMY	[N/A]
96	–	DUMMY	[N/A]
97	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
98	–	DUMMY	[N/A]
99	–	DUMMY	[N/A]
100	–	DUMMY	[N/A]
101	–	DUMMY	[N/A]
102	–	DUMMY	[N/A]
103	–	DUMMY	[N/A]
104	–	Control	[CTRL]
105	H1	VO_CLK	[IN/OUT]
106	–	DUMMY	[N/A]
107	–	DUMMY	[N/A]
108	–	DUMMY	[N/A]
109	–	DUMMY	[N/A]
110	–	Control	[CTRL]
111	J4	VO_D[0]	[OUT]
112	–	DUMMY	[N/A]
113	–	DUMMY	[N/A]
114	–	DUMMY	[N/A]
115	–	DUMMY	[N/A]
116	–	Control	[CTRL]
117	H2	VO_D[1]	[OUT]
118	–	Control	[CTRL]
119	H3	VO_D[2]	[OUT]
120	–	Control	[CTRL]
121	H4	VO_D[3]	[OUT]
122	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
123	G1	VO_D[4]	[OUT]
124	–	Control	[CTRL]
125	G2	VO_D[5]	[OUT]
126	–	Control	[CTRL]
127	G3	VO_D[6]	[OUT]
128	–	Control	[CTRL]
129	F1	VO_D[7]	[OUT]
130	–	Control	[CTRL]
131	F2	VO_D[8]	[OUT]
132	–	Control	[CTRL]
133	G4	VO_D[9]	[OUT]
134	–	Control	[CTRL]
135	E1	VO_D[10]	[OUT]
136	–	Control	[CTRL]
137	F3	VO_D[11]	[OUT]
138	–	Control	[CTRL]
139	E2	VO_D[12]	[OUT]
140	–	Control	[CTRL]
141	D1	VO_D[13]	[OUT]
142	–	Control	[CTRL]
143	F4	VO_D[15]	[OUT]
144	–	DUMMY	[N/A]
145	–	DUMMY	[N/A]
146	–	Control	[CTRL]
147	E3	VO_D[14]	[OUT]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
148	–	DUMMY	[N/A]
149	–	DUMMY	[N/A]
150	–	DUMMY	[N/A]
151	–	DUMMY	[N/A]
152	–	DUMMY	[N/A]
153	–	DUMMY	[N/A]
154	–	DUMMY	[N/A]
155	–	DUMMY	[N/A]
156	–	DUMMY	[N/A]
157	–	DUMMY	[N/A]
158	–	DUMMY	[N/A]
159	–	DUMMY	[N/A]
160	–	DUMMY	[N/A]
161	–	DUMMY	[N/A]
162	–	DUMMY	[N/A]
163	–	DUMMY	[N/A]
164	–	DUMMY	[N/A]
165	–	DUMMY	[N/A]
166	–	DUMMY	[N/A]
167	–	DUMMY	[N/A]
168	–	DUMMY	[N/A]
169	–	DUMMY	[N/A]
170	–	DUMMY	[N/A]
171	–	DUMMY	[N/A]
172	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
173	C1	VI_VSYNC[0]	[IN]
174	–	Control	[CTRL]
175	D3	VI_D[5]	[IN]
176	–	Control	[CTRL]
177	C2	VI_D[4]	[IN]
178	–	DUMMY	[N/A]]
179	–	DUMMY	[N/A]]
180	–	Control	[CTRL]
181	C3	VI_D[3]	[IN]
182	–	Control	[CTRL]
183	B2	VI_D[2]	[IN]
184	–	Control	[CTRL]
185	A1	VI_D[9]	[IN]
186	–	Control	[CTRL]
187	A2	VI_D[8]	[IN]
188	–	Control	[CTRL]
189	B3	VI_D[7]	[IN]
190	–	Control	[CTRL]
191	C4	VI_D[6]	[IN]
192	–	DUMMY	[N/A]
193	–	DUMMY	[N/A]
194	–	DUMMY	[N/A]
195	–	DUMMY	[N/A]
196	–	DUMMY	[N/A]
197	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
198	–	DUMMY	[N/A]
199	–	DUMMY	[N/A]
200	–	DUMMY	[N/A]
201	–	DUMMY	[N/A]
202	–	DUMMY	[N/A]
203	–	DUMMY	[N/A]
204	–	DUMMY	[N/A]
205	–	DUMMY	[N/A]
206	–	DUMMY	[N/A]
207	–	DUMMY	[N/A]
208	–	DUMMY	[N/A]
209	–	DUMMY	[N/A]
210	–	Control	[CTRL]
211	A5	VI_CLK[0]	[IN]
212	–	DUMMY	[N/A]
213	–	DUMMY	[N/A]
214	–	Control	[CTRL]
215	A7	PLL_BYPASS	[IN]
216	–	Control	[CTRL]
217	A12	CLKO	[OUT]
218	–	Control	[CTRL]
219	B12	AI_SCLK	[IN/OUT]
220	–	DUMMY	[N/A]
221	–	DUMMY	[N/A]
222	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
223	C12	AI_FSYNC	[IN/OUT]
224	–	DUMMY	[N/A]
225	–	DUMMY	[N/A]
226	–	DUMMY	[N/A]
227	–	DUMMY	[N/A]
228	–	Control	[CTRL]
229	D12	AI_D[1]	[IN]
230	–	Control	[CTRL]
231	A13	AI_MCLKO	[OUT]
232	–	Control	[CTRL]
233	B13	AO_MCLKO	[OUT]
234	–	DUMMY	[N/A]
235	–	DUMMY	[N/A]
236	–	Control	[CTRL]
237	C13	AI_D[0]	[IN]
238	–	Control	[CTRL]
239	B14	AO_IEC958	[OUT]
240	–	Control	[CTRL]
241	D13	AO_FSYNC	[OUT]
242	–	DUMMY	[N/A]
243	–	DUMMY	[N/A]
244	–	Control	[CTRL]
245	A14	AO_SCLK	[OUT]
246	–	DUMMY	[N/A]
247	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
248	–	Control	[CTRL]
249	C14	AO_D[1]	[OUT]
250	–	DUMMY	[CTRL]
251	–	DUMMY	[N/A]
252	–	DUMMY	[N/A]
253	–	DUMMY	[N/A]
254	–	Control	[CTRL]
255	D14	AO_D[0]	[OUT]
256	–	DUMMY	[N/A]
257	–	DUMMY	[N/A]
258	–	DUMMY	[N/A]
259	–	DUMMY	[N/A]
260	–	DUMMY	[N/A]
261	–	DUMMY	[N/A]
262	–	Control	[CTRL]
263	A15	AO_D[3]	[OUT]
264	–	DUMMY	[N/A]
265	–	DUMMY	[N/A]
266	–	DUMMY	[N/A]
267	–	DUMMY	[N/A]
268	–	DUMMY	[N/A]
269	–	DUMMY	[N/A]
270	–	DUMMY	[N/A]
271	–	DUMMY	[N/A]
272	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
273	–	DUMMY	[N/A]
274	–	DUMMY	[N/A]
275	–	DUMMY	[N/A]
276	–	DUMMY	[N/A]
277	–	DUMMY	[N/A]
278	–	DUMMY	[N/A]
279	–	DUMMY	[N/A]
280	–	DUMMY	[N/A]
281	–	DUMMY	[N/A]
282	–	DUMMY	[N/A]
283	–	DUMMY	[N/A]
284	–	DUMMY	[N/A]
285	–	DUMMY	[N/A]
286	–	DUMMY	[N/A]
287	–	DUMMY	[N/A]
288	–	DUMMY	[N/A]
289	–	DUMMY	[N/A]
290	–	Control	[CTRL]
291	A16	$\overline{\text{SDRAM\_WE}}$	[OUT]
292	–	Control	[CTRL]
293	C15	SDRAM_CKE	[OUT]
294	–	Control	[CTRL]
295	B16	$\overline{\text{SDRAM\_RAS}}$	[OUT]
296	–	Control	[CTRL]
297	B17	$\overline{\text{SDRAM\_CAS}}$	[OUT]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
298	–	Control	[CTRL]
299	A17	SDRAM_A[2]	[OUT]
300	–	Control	[CTRL]
301	A18	SDRAM_A[15]	[OUT]
302	–	Control	[CTRL]
303	C17	SDRAM_A[3]	[OUT]
304	–	Control	[CTRL]
305	B18	SDRAM_A[4]	[OUT]
306	–	Control	[CTRL]
307	A19	SDRAM_A[0]	[OUT]
308	–	Control	[CTRL]
309	C18	SDRAM_A[6]	[OUT]
310	–	Control	[CTRL]
311	B19	SDRAM_A[1]	[OUT]
312	–	Control	[CTRL]
313	A20	SDRAM_A[5]	[OUT]
314	–	Control	[CTRL]
315	E17	SDRAM_A[12]	[OUT]
316	–	Control	[CTRL]
317	D18	SDRAM_A[8]	[OUT]
318	–	Control	[CTRL]
319	C19	SDRAM_A[10]	[OUT]
320	–	Control	[CTRL]
321	B20	SDRAM_A[7]	[OUT]
322	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
323	F17	SDRAM_A[11]	[OUT]
324	–	Control	[CTRL]
325	E18	SDRAM_A[13]	[OUT]
326	–	Control	[CTRL]
327	D19	SDRAM_A[9]	[OUT]
328	–	Control	[CTRL]
329	C20	SDRAM_A[14]	[OUT]
330	–	Control	[CTRL]
331	D20	SDRAM_DQ[24]	[IN/OUT]
332	–	Control	[CTRL]
333	E19	SDRAM_DQ[25]	[IN/OUT]
334	–	Control	[CTRL]
335	F18	SDRAM_DQ[26]	[IN/OUT]
336	–	Control	[CTRL]
337	G17	SDRAM_DQ[27]	[IN/OUT]
338	–	Control	[CTRL]
339	E20	SDRAM_DQS[3]	[IN/OUT]
340	–	Control	[CTRL]
341	F19	SDRAM_DQ[30]	[IN/OUT]
342	–	Control	[CTRL]
343	G18	SDRAM_DQ[28]	[IN/OUT]
344	–	Control	[CTRL]
345	H17	SDRAM_DQ[29]	[IN/OUT]
346	–	Control	[CTRL]
347	F20	SDRAM_DQ[31]	[IN/OUT]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
348	–	Control	[CTRL]
349	G19	SDRAM_DQM[3]	[OUT]
350	–	Control	[CTRL]
351	H20	$\overline{\text{SDRAM\_CLK}}$[1]	[OUT]
352	–	Control	[CTRL]
353	G20	SDRAM_CLK[1]	[OUT]
354	–	Control	[CTRL]
355	H19	SDRAM_DQ[23]	[IN/OUT]
356	–	Control	[CTRL]
357	H18	SDRAM_DQ[22]	[IN/OUT]
358	–	Control	[CTRL]
359	J17	SDRAM_DQ[21]	[IN/OUT]
360	–	Control	[CTRL]
361	J18	SDRAM_DQ[20]	[IN/OUT]
362	–	Control	[CTRL]
363	J19	SDRAM_DQS[2]	[IN/OUT]
364	–	Control	[CTRL]
365	J20	SDRAM_DQ[19]	[IN/OUT]
366	–	Control	[CTRL]
367	K17	SDRAM_DQ[18]	[IN/OUT]
368	–	Control	[CTRL]
369	K18	SDRAM_DQ[17]	[IN/OUT]
370	–	Control	[CTRL]
371	K19	SDRAM_DQ[16]	[IN/OUT]
372	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
373	K20	SDRAM_DQM[2]	[OUT]
374	–	Control	[CTRL]
375	L17	SDRAM_DQM[1]	[OUT]
376	–	Control	[CTRL]
377	L18	SDRAM_DQ[8]	[IN/OUT]
378	–	Control	[CTRL]
379	L19	SDRAM_DQ[11]	[IN/OUT]
380	–	Control	[CTRL]
381	L20	SDRAM_DQ[13]	[IN/OUT]
382	–	Control	[CTRL]
383	M20	SDRAM_DQ[10]	[IN/OUT]
384	–	Control	[CTRL]
385	M19	SDRAM_DQS[1]	[IN/OUT]
386	–	Control	[CTRL]
387	M18	SDRAM_DQ[14]	[IN/OUT]
388	–	Control	[CTRL]
389	M17	SDRAM_DQ[12]	[IN/OUT]
390	–	Control	[CTRL]
391	N19	SDRAM_DQ[9]	[IN/OUT]
392	–	Control	[CTRL]
393	N18	SDRAM_DQ[15]	[IN/OUT]
394	–	Control	[CTRL]
395	N17	SDRAM_DQM[0]	[OUT]
396	–	Control	[CTRL]
397	N20	$\overline{\text{SDRAM\_CLK}}$[0]	[OUT]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
398	–	Control	[CTRL]
399	P20	SDRAM_CLK[0]	[OUT]
400	–	Control	[CTRL]
401	P19	SDRAM_DQ[7]	[IN/OUT]
402	–	Control	[CTRL]
403	P18	SDRAM_DQ[6]	[IN/OUT]
404	–	Control	[CTRL]
405	R20	SDRAM_DQ[5]	[IN/OUT]
406	–	Control	[CTRL]
407	P17	SDRAM_DQ[4]	[IN/OUT]
408	–	Control	[CTRL]
409	R19	SDRAM_DQS[0]	[IN/OUT]
410	–	Control	[CTRL]
411	T20	SDRAM_DQ[3]	[IN/OUT]
412	–	Control	[CTRL]
413	R18	SDRAM_DQ[2]	[IN/OUT]
414	–	Control	[CTRL]
415	T19	SDRAM_DQ[1]	[IN/OUT]
416	–	Control	[CTRL]
417	R17	SDRAM_DQ[0]	[IN/OUT]
418	–	DUMMY	[N/A]
419	–	DUMMY	[N/A]
420	–	DUMMY	[N/A]
421	–	DUMMY	[N/A]
422	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
423	U20	SIO_UART1_TX	[OUT]
424	–	Control	[CTRL]
425	T18	SIO_SPI_MISO	[IN/OUT]
426	–	Control	[CTRL]
427	U19	SIO_UART2_TX	[OUT]
428	–	DUMMY	[N/A]
429	–	DUMMY	[N/A]
430	–	DUMMY	[N/A]
431	–	DUMMY	[N/A]
432	–	Control	[CTRL]
433	V20	SIO_UART1_RTS	[OUT]
434	–	Control	[CTRL]
435	U18	SIO_SPI_MOSI	[OUT]
436	–	Control	[CTRL]
437	V19	SIO_SCL	[IN/OUT]
438	–	Control	[CTRL]
439	W20	SIO_UART2_RX	[IN]
440	–	Control	[CTRL]
441	V18	SIO_SPI_CS[1]	[OUT]
442	–	Control	[CTRL]
443	W19	SIO_UART1_RX	[IN/OUT]
444	–	Control	[CTRL]
445	Y20	SIO_IRRX	[IN]
446	–	Control	[CTRL]
447	Y19	SIO_SPI_CS[0]	[OUT]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
448	–	Control	[CTRL]
449	W18	SIO_SPI_CS[3]	[OUT]
450	–	Control	[CTRL]
451	V17	SIO_UART1_CTS	[IN/OUT]
452	–	Control	[CTRL]
453	Y18	SIO_SDA	[IN/OUT]
454	–	Control	[CTRL]
455	W17	SIO_IRTX2	[OUT]
456	–	Control	[CTRL]
457	V16	SIO_SPI_CS[2]	[OUT]
458	–	Control	[CTRL]
459	Y17	SIO_SPI_CLK	[OUT]
460	–	Control	[CTRL]
461	U14	SIO_IRTX1	[OUT]
462	–	DUMMY	[N/A]
463	–	DUMMY	[N/A]
464	–	DUMMY	[N/A]
465	–	DUMMY	[N/A]
466	–	DUMMY	[N/A]
467	–	DUMMY	[N/A]
468	–	DUMMY	[N/A]
469	–	DUMMY	[N/A]
470	–	DUMMY	[N/A]
471	–	DUMMY	[N/A]
472	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
473	U13	M_ALE	[OUT]
474	–	DUMMY	[N/A]
475	–	DUMMY	[N/A]
476	–	DUMMY	[N/A]
477	–	DUMMY	[N/A]
478	–	DUMMY	[N/A]
479	–	DUMMY	[N/A]
480	–	DUMMY	[N/A]
481	–	DUMMY	[N/A]
482	–	DUMMY	[N/A]
483	–	DUMMY	[N/A]
484	–	DUMMY	[N/A]
485	–	DUMMY	[N/A]
486	–	DUMMY	[N/A]
487	–	DUMMY	[N/A]
488	–	DUMMY	[N/A]
489	–	DUMMY	[N/A]
490	–	DUMMY	[N/A]
491	–	DUMMY	[N/A]
492	–	DUMMY	[N/A]
493	–	DUMMY	[N/A]
494	–	DUMMY	[N/A]
495	–	DUMMY	[N/A]
496	–	DUMMY	[N/A]
497	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
498	–	DUMMY	[N/A]
499	–	DUMMY	[N/A]
500	–	DUMMY	[N/A]
501	–	DUMMY	[N/A]
502	–	DUMMY	[N/A]
503	–	DUMMY	[N/A]
504	–	DUMMY	[N/A]
505	–	DUMMY	[N/A]
506	–	DUMMY	[N/A]
507	–	DUMMY	[N/A]
508	–	DUMMY	[N/A]
509	–	DUMMY	[N/A]
510	–	DUMMY	[N/A]
511	–	DUMMY	[N/A]
512	–	DUMMY	[N/A]
513	–	DUMMY	[N/A]
514	–	DUMMY	[N/A]
515	–	DUMMY	[N/A]
516	–	DUMMY	[N/A]
517	–	DUMMY	[N/A]
518	–	DUMMY	[N/A]
519	–	DUMMY	[N/A]
520	–	DUMMY	[N/A]
521	–	DUMMY	[N/A]
522	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
523	–	DUMMY	[N/A]
524	–	DUMMY	[N/A]
525	–	DUMMY	[N/A]
526	–	DUMMY	[N/A]
527	–	DUMMY	[N/A]
528	–	DUMMY	[N/A]
529	–	DUMMY	[N/A]
530	–	DUMMY	[N/A]
531	–	DUMMY	[N/A]
532	–	DUMMY	[N/A]
533	–	DUMMY	[N/A]
534	–	DUMMY	[N/A]
535	–	DUMMY	[N/A]
536	–	DUMMY	[N/A]
537	–	DUMMY	[N/A]
538	–	DUMMY	[N/A]
539	–	DUMMY	[N/A]
540	–	DUMMY	[N/A]
541	–	DUMMY	[N/A]
542	–	DUMMY	[N/A]
543	–	DUMMY	[N/A]
544	–	DUMMY	[N/A]
545	–	DUMMY	[N/A]
546	–	DUMMY	[N/A]
547	–	DUMMY	[N/A]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
548	–	DUMMY	[N/A]
549	–	DUMMY	[N/A]
550	–	DUMMY	[N/A]
551	–	DUMMY	[N/A]
552	–	DUMMY	[N/A]
553	–	DUMMY	[N/A]
554	–	DUMMY	[N/A]
555	–	DUMMY	[N/A]
556	–	DUMMY	[N/A]
557	–	DUMMY	[N/A]
558	–	DUMMY	[N/A]
559	–	DUMMY	[N/A]
560	–	DUMMY	[N/A]
561	–	DUMMY	[N/A]
562	–	Control	[CTRL]
563	M16	MCONFIG[0]	[IN]
564	–	DUMMY	[N/A]
565	–	DUMMY	[N/A]
566	–	DUMMY	[N/A]
567	–	DUMMY	[N/A]
568	–	DUMMY	[N/A]
569	–	DUMMY	[N/A]
570	–	DUMMY	[N/A]
571	–	DUMMY	[N/A]
572	–	Control	[CTRL]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 17.2 Boundary Scan Chain Cells (Cont.)

Cell	Pin	Name	Type
573	V4	ATAPI_ADDR[4]	[IN/OUT]
574	–	Control	[CTRL]
575	W3	ATAPI_ADDR[0]	[IN/OUT]
576	–	Control	[CTRL]
577	Y2	ATAPI_DATA[15]	[IN/OUT]
578	–	Control	[CTRL]
579	Y1	ATAPI_RESET	[OUT]
580	–	Control	[CTRL]
581	W2	ATAPI_INTRQ	[IN]
582	–	Control	[CTRL]
583	V3	$\overline{\text{ATAPI\_DIOR}}$	[OUT]
584	–	Control	[CTRL]
585	W1	ATAPI_DMARQ	[IN]
586	–	Control	[CTRL]
587	V2	ATAPI_ADDR[1]	[IN/OUT]
588	–	Control	[CTRL]
589	U3	ATAPI_ADDR[3]	[OUT]
590	–	Control	[CTRL]
591	V1	ATAPI_ADDR[2]	[OUT]
592	–	Control	[CTRL]
593	U2	ATAPI_DATA[14]	[IN/OUT]
594	–	Control	[CTRL]
595	T3	ATAPI_IORDY	[IN]

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter 18
Specifications

This chapter specifies the electrical, mechanical and AC timing characteristics of the DMN-8600. It also provides a table listing each pin in alpha-numeric order, with corresponding voltage levels, in these sections:

18.1 Electrical Specifications

This section specifies the electrical requirements for the DMN-8600 processor.

Table 18.1 Absolute Maximum Ratings[1]

Parameters			Value	Units
Supply Voltages	VDD_5	For 5V tolerant signals	-0.5 to 5.5	V
	VDD_2.5	DDR SDRAM (DDR only)	-0.25 to 2.75	V
	VDD_2.5	SDR SDRAM (SDR only)	-0.3 to 3.6	V
	VDD_3.3	I/O supply voltage	-0.3 to 3.6	V
	VDD_1.8	Core power supply	-0.18 to 1.98	V
	VDD_A	Analog supply voltage	-0.3 to 3.6	V
	VDD_DLL	Digital Voltage - Internal clock DLL	-0.18 to 1.98	V
	VDD_RREF	Analog power (isolated)	-0.3 to 3.6	V
Input Voltage			VDD_{IO} 0.4	V

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.1 Absolute Maximum Ratings[1] (Cont.)

Parameters		Value	Units
Output Voltage		VDD_{IO} 0	V
Temperatures	Storage Temperature Range	–55 to +150	˚C
	Operating Temperature Range (amb)	0 to 70	˚C
	Operating Temperature Range (case)	-10 to 110	˚C
	Reflow peak solder temperature	220 ˚C for 10 seconds max	˚C

1. Exposure to stresses beyond those listed in this table may result in device unreliability, permanent damage, or both.

Caution: The ambient and case operating temperature range values specified in Table 18.1 must be adhered to for any system design.

Table 18.2 Operating Conditions

Parameters			Min	Typical	Max	Unit
Supply Voltages	VDD_5	For 5V tolerant signals	4.75	5.0	5.25	V
	VDD_2.5	DDR SDRAM (DDR only)	2.375	2.5	2.625	V
	VDD_2.5	SDR SDRAM (SDR only)	3.135	3.3	3.465	V
	VDD_3.3	Supply Voltage for I/O	3.135	3.3	3.465	V
	VDD_1.8	Core supply voltage	1.71	1.8	1.89	V
	VDD_A	Analog Voltage	3.135	3.3	3.465	V
Temperatures	T_J	Junction Temperature	0	70	125	˚C
	θ_{JA}	Thermal resistance (4-layer PCB)		14.0		˚C

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.3 DC Characteristics

Sym	Parameter	Condition	Min	Typ	Max	Unit
V_{IH}	High-level input voltage	$V_{DD\_3.3}$ = Max and $V_{DD\_2.5}$ = Max	2.4	–	3.45	V
V_{IL}	Low-level input voltage	$V_{DD\_3.3}$ = Min and $V_{DD\_2.5}$ = Min	–	–	0.8	V
V_{IHG}	High-level input voltage for VCLK	$V_{DD\_3.3}$ = Max and $V_{DD\_2.5}$ = Max	2.4	–	3.45	V
V_{ILG}	Low-level input voltage for VCLK	$V_{DD\_3.3}$ = Min and $V_{DD\_2.5}$ = Min	–	–	0.8	V
V_{OH}	High-level output voltage	$V_{DD\_3.3}$ = Min, I_{OH} = drive level of individual signal	2.4	–		V
V_{OL}	Low-level output voltage	$V_{DD\_3.3}$ = Min, I_{OL} = drive level of individual signal	–	–	0.5	V
I_{IH}	High-level input current	$V_{DD\_3.3}$ and $V_{DD\_2.5}$ = Max, V_{IN} = V_{DD}	–	–	10	µA
I_{IL}	Low-level input current	$V_{DD\_3.3}$ and $V_{DD\_2.5}$ = Max, V_{IN} = 0 V	-10	–	–	µA
I_{OZ}	Output leakage current	Hi-Z output driven to 0 V and 3.15V	-10	–	10	µA
I_{OZM}	Output leakage current, SDRAM pins	Hi-Z output driven to 0 V and V_{DD}	-10	–	10	µA
$I_{DD\_2.5}$	Supply current @V_{DD} = 2.5 V	T_A = 25 ˚C, V_{IN} = 0 or 2.5 V, C_L = 50 pF	–	TBD	–	A
$I_{DD\_3.3}$	Supply current @V_{DD} = 3.3 V	T_A = 25 ˚C, V_{IN} = 0 or 3.3 V, C_L = 50 pF	–	TBD	–	A
C_{IN}	Input capacitance[1]		–	10	–	pF
C_{OUT}	Output Capacitance[1]		–	12	–	pF
$C_{I/O}$	I/O Pin capacitance[1]		–	12	–	pF
P_D	Power dissipation	VDD Nominal @ 25 ˚C at 148.5 MHz	–	2.8	–	W

1. Not 100% tested, guaranteed by design characteristics.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2 AC Timing

This section provides the AC timing for the DMN-8600 processor's various interfaces and is divided into these sections:

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.1 Miscellaneous Timing

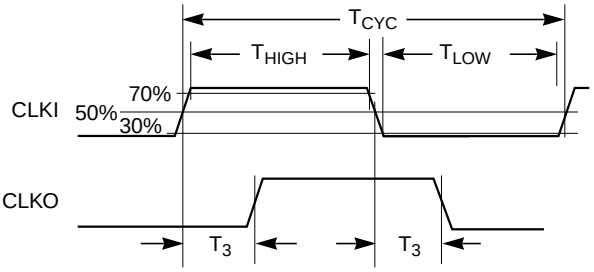
Table 18.4, Figure 18.1 and Figure 18.2 show miscellaneous AC timing parameters.

Table 18.4 Miscellaneous Timing Values

Symbol	Description	Timing Value (13.5 MHz Crystal)		Unit	Timing Value (27 MHz Crystal)		Unit
		Min	Max		Min	Max	
T_{CYC}	CLKI cycle time	74.07	–	ns	37.04		ns
T_{HIGH}	CLKI high time	33	–	ns	16.5		ns
T_{LOW}	CLKI low time	33	–	ns	16.5		ns
T3[1]	CLKO output delay with respect to CLKI	3	20	ns	3	20	ns
T_{GPW}	GPIO input pulse width	2	–	cycles	2		cycles

1. T3 minimum delay is @ load = 0 pF. T3 maximum delay is @ load = 50 pF.

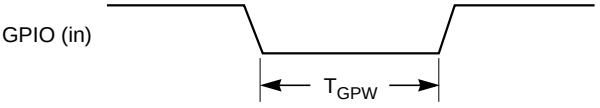
Figure 18.1 Timing Diagram for CLKI and CLKO



The DMN-8600 has eight general-purpose I/O pins, four of which can be used to interrupt the SPARC. They can be configured to be edge or level sensitive and must be asserted for at least two clock cycles, as shown in Figure 18.2.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.2 General-Purpose I/O Timing

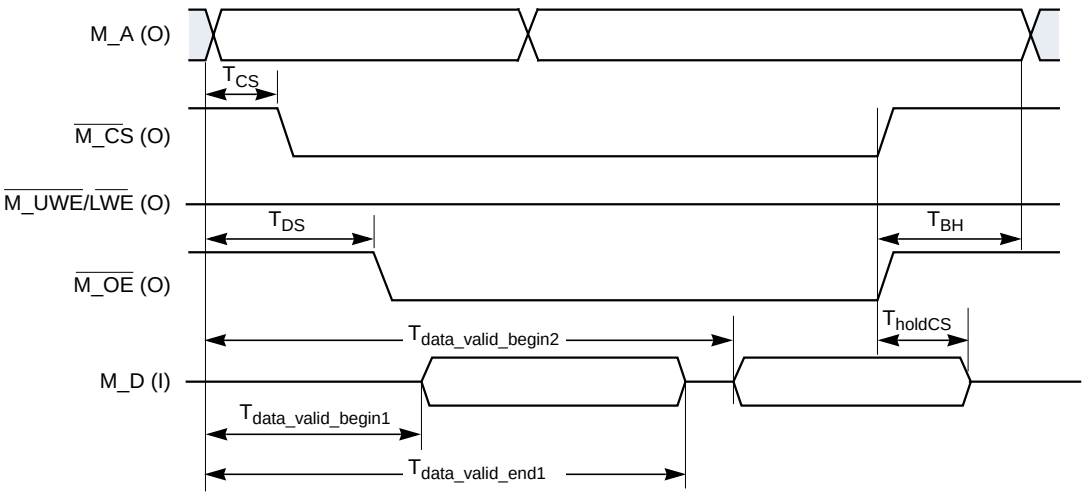


18.2.2 Host Master Timing

When in Master mode, the DMN-8600 controls an external slave device through use of an async master interface. The DMN-8600 master interface supports 68K and SRAM data strobe modes with either a self-paced or device-paced protocol. The master interface can also multiplex address and data lines.

Unless otherwise noted, cycle type variations are independent of each other and can be mixed and matched as the system designer sees fit. This accounts for a total of nine possible variations, as shown in Figure 18.3 through Figure 18.11 and described in Table 18.5.

Figure 18.3 Self-Paced Async Master Read Cycle in SRAM Mode



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.4 Self-Paced Async Master Write Cycle in SRAM Mode

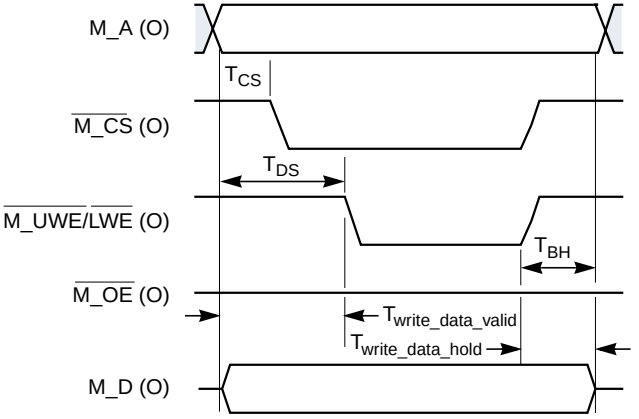
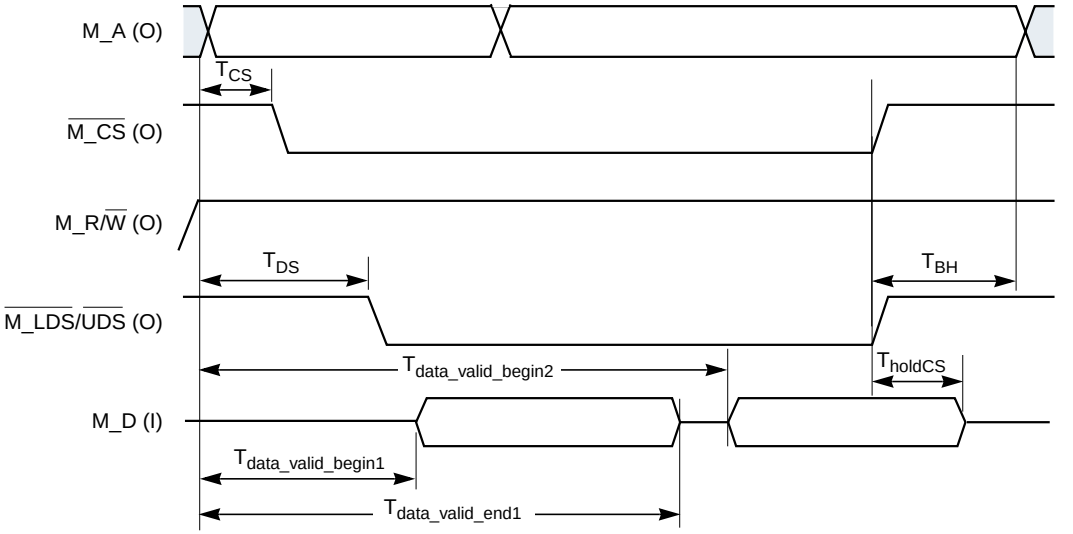


Figure 18.5 Self-Paced Async Master Read Cycle in 68K Mode



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.6 Self-Paced Async Master Write Cycle in 68K Mode

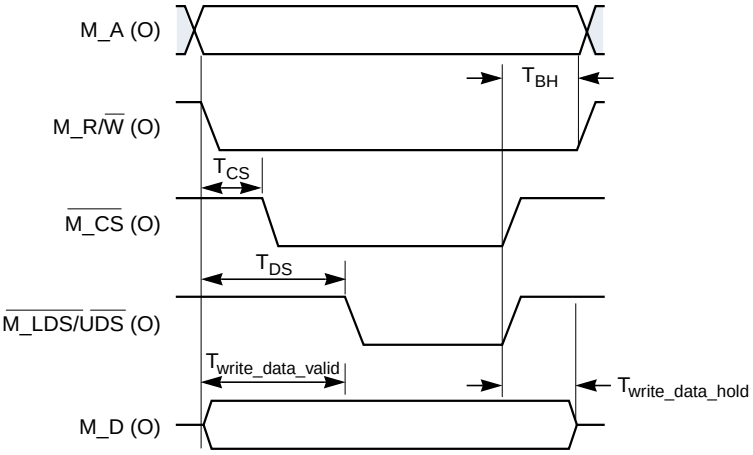
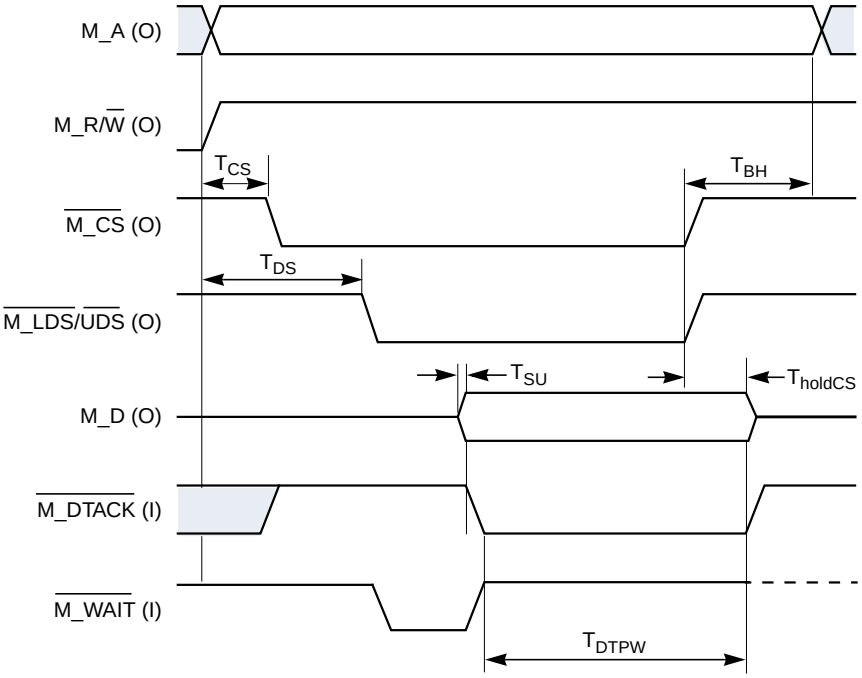
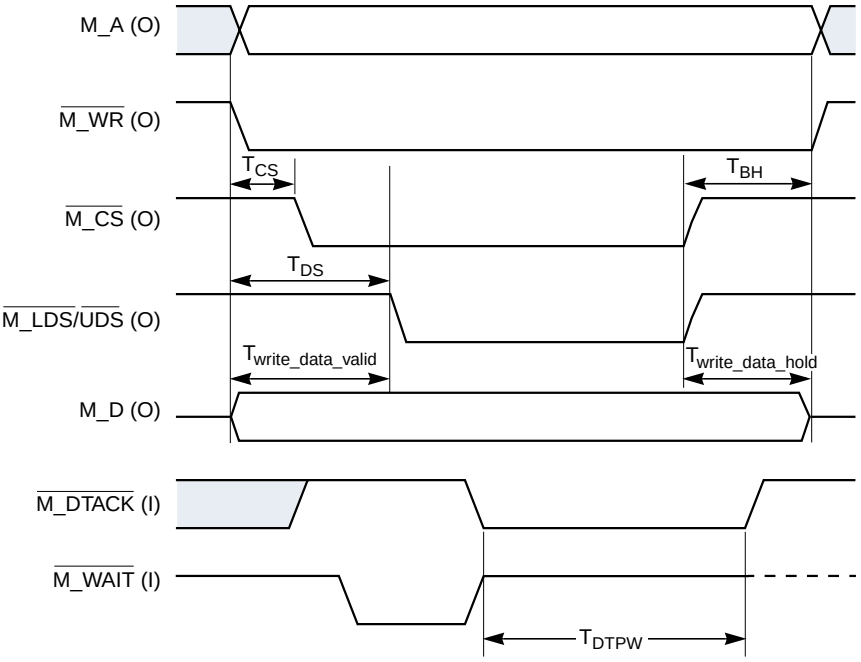


Figure 18.7 Device-Paced Async Master Read Cycle in 68K Mode



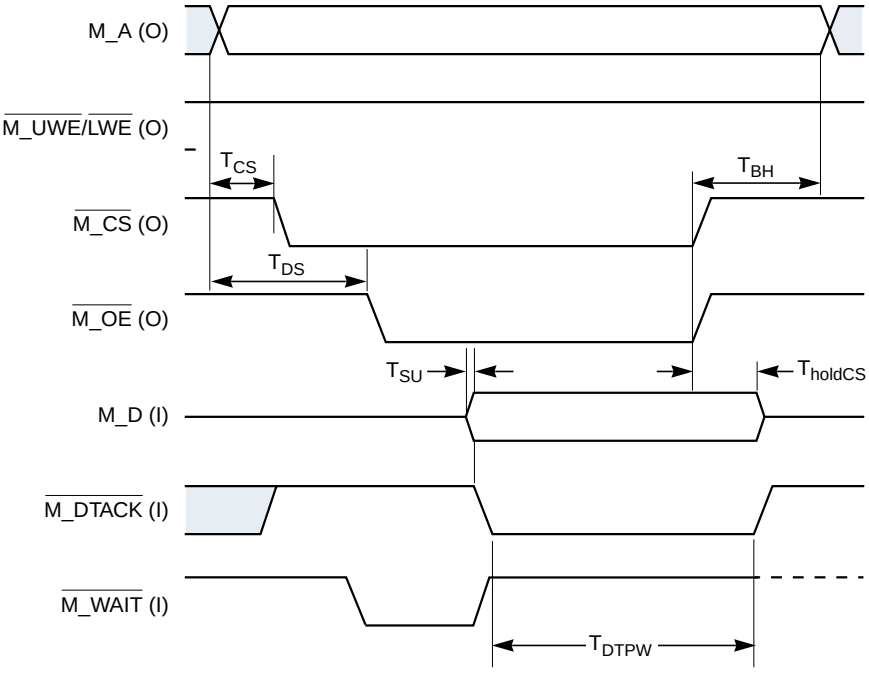
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.8 Device-Paced Async Master Write Cycle in 68K Mode



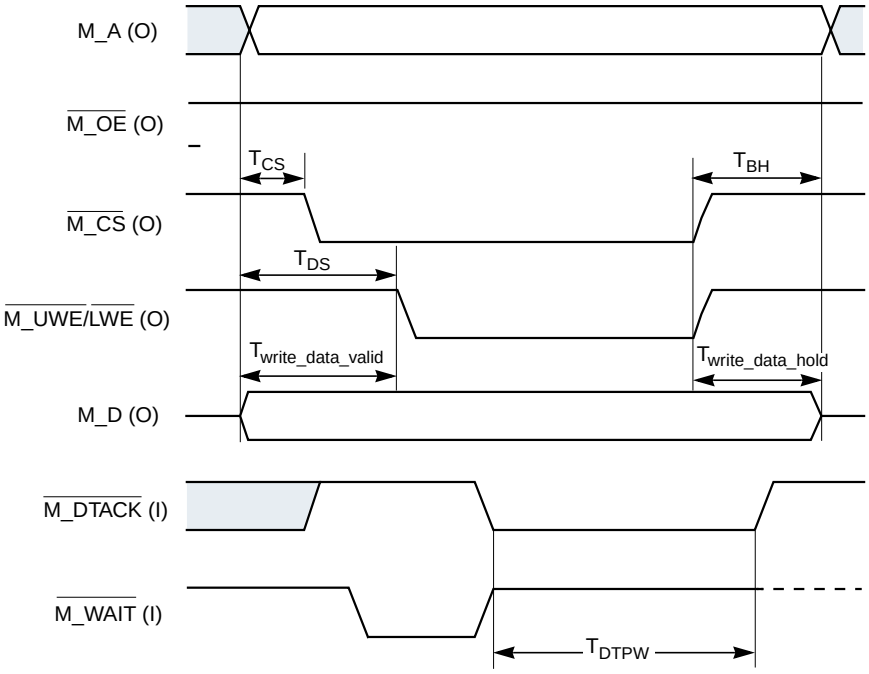
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.9 Device-Paced Async Master Read Cycle in SRAM Mode



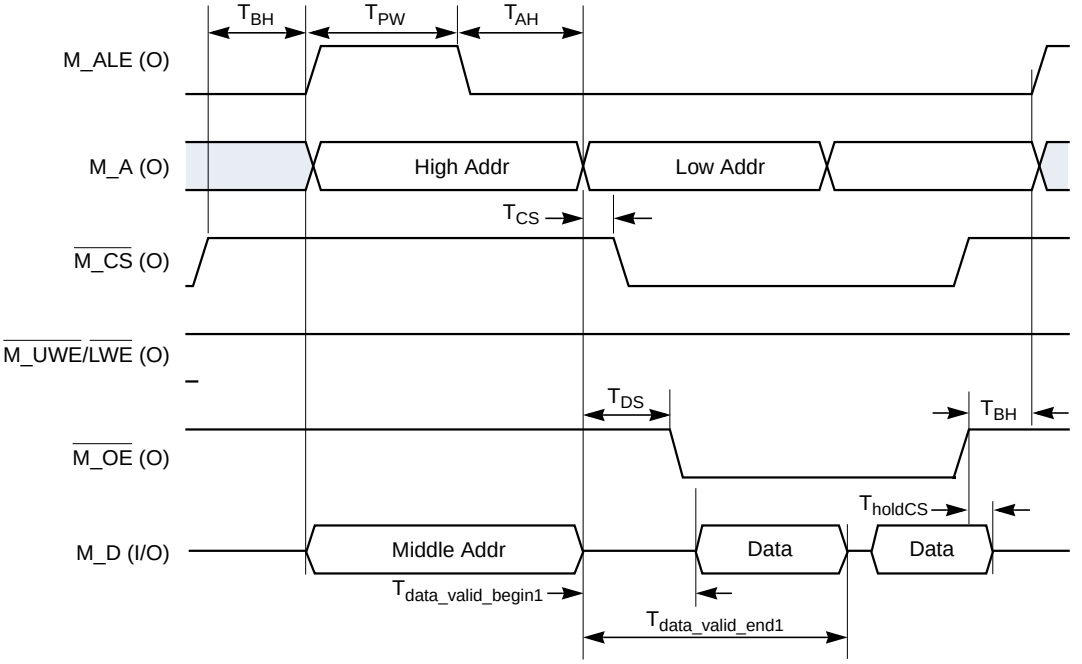
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.10 Device-Paced Async Master Write Cycle in SRAM Mode



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.11 Multiplexed Address Async Master Read Cycle



Note: In Table 18.5, TCS, TDS, TAH, TDT, TBDT, and TBH are programmable parameter delays. They represent delay in the number of clock cycles. They can be programmed in the personality dependent register of each chip select.

Table 18.5 Async Host Master Timing Parameters - Master Mode Only (Master replaces Slave)

Parameters	Description	Min	Max
T_{PW}	M_ALE pulse width from M_A and M_ALE asserted to fall of M_ALE. This can be treated as setup time.	AS clock cycles – 2 ns	AS clock cycles + 2 ns
T_{AH}	Time from fall of M_ALE to M_A/M_D change. This can be treated as hold time by an external device.	AH clock cycles – 4 ns	AH clock cycles + 2 ns
T_{CS}	Delay from M_A and R/$\overline{W}$ to fall of $\overline{M\_CS}$.	CS clock cycles – 5 ns	CS clock cycles + 3 ns

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.5 Async Host Master Timing Parameters - Master Mode Only (Master replaces Slave) (Cont.)

Parameters	Description	Min	Max
T_{DS}	Delay from M_A and R/$\overline{W}$ to fall of data strobes.	DS clock cycles – 6 ns	DS clock cycles + 2 ns
T_{GPW}	GPIO pulse width.	2 clock cycles	
$T_{data\_valid\_begin(n)}$[1]	Delay from M_A and M_RD/$\overline{WR}$ stable to stable nth read data beginning (applicable for bursts and single access).		DT + (n – 1) (BDT + 1) clock cycles – 4 ns
T_{holdCS}	Read data hold time with respect to CS rising.	1 clock cycle	BH – 1 clock cycle
T_{SU}	Setup time of input data with respect to M_$\overline{DTACK}$/$\overline{WAIT}$.	0 ns	
T_{DTPW}	M_$\overline{DTACK}$/$\overline{WAIT}$ pulse width.	2 clock cycles	
$T_{data\_valid\_end(n)}$[1]	Delay for M_A and R/$\overline{W}$ stable to end of nth read data during bursts (for single access, T_{holdCS} is applicable).	DT + (n – 1) (BDT + 1) + 3 clock cycles.	
$T_{write\_data\_valid}$	Delay from write strobes M_$\overline{UWE}$, M_$\overline{LWE}$, M_$\overline{UDS}$, M_$\overline{LDS}$ falling to write data valid.	– DS clock cycles – 2 ns	– DS clock cycles + 6 ns
$T_{write\_data\_hold}$	Write data hold time after data strobes pull up	BH clock cycles – 2 ns	BH clock cycles + 6 ns
T_{BH}	Delay from end of cycle to start of next cycle.	BH clock cycles – 3 ns	

1. The parameters $T_{data\_valid\_begin(n)}$ and $T_{data\_valid\_end(n)}$ represent the minimum possible window during which the data pins must be stable.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.6 Async Host Master Timing Parameters - Host Slave and Slave plus Limited Master Modes Only

Parameters	Description	Min	Max
T_{PW}	M_ALE pulse width from M_A and M_ALE asserted to fall of M_ALE. This can be treated as setup time by external slave.	AS clock cycles – 2 ns	AS clock cycles + 2 ns
T_{AH}	Time from fall of M_ALE to M_A/M_D change. This can be treated as hold time by an external slave.	AH clock cycles – 4 ns	AH clock cycles + 3 ns
T_{CS}	Delay from M_A to fall of $\overline{M\_CS}$.	CS clock cycles – 7 ns	CS clock cycles + 2 ns
T_{DS}	Delay from M_A to fall of data strobes.	DS clock cycles – 4 ns	DS clock cycles + 4 ns
$T_{data\_valid\_begin(n)}$[1]	Delay from M_A and M_RD/$\overline{WR}$ stable to stable nth read data (applicable for bursts and single access).		DT + (n-1) (BDT + 1) clock cycles – 6 ns
T_{holdCS}	Read data hold time with respect to CS rising.	1 clock cycle	BH – 1 clock cycle
$T_{data\_valid\_end(n)}$[1]	Delay for stable address to end of nth read data during bursts (for single access, T_{holdCS} is applicable).	DT + (n – 1) (BDT + 1) + 3 clock cycles.	
$T_{write\_data\_valid}$	Delay from write strobe falling to write data valid.	– DS clock cycles – 4 ns	– DS clock cycles + 2 ns
$T_{write\_data\_hold}$	Write data hold time after strobe pulls up	BH clock cycles – 4 ns	BH clock cycles + 2 ns
T_{BH}	Delay from end of cycle to start of next cycle.	BH clock cycles – 2 ns	

1. The parameters $T_{data\_valid\_begin(n)}$ and $T_{data\_valid\_end(n)}$ represent the minimum possible window during which the data pins must be stable.

18.2.3 Host Slave Timing

The diagrams shown in Figures 18.12, 18.13, 18.14 and 18.15 give the slave mode read-from and write-to timing.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.3.1 I-Mode Write AC Timing

Figure 18.12 I-Mode Write AC Timing Diagram

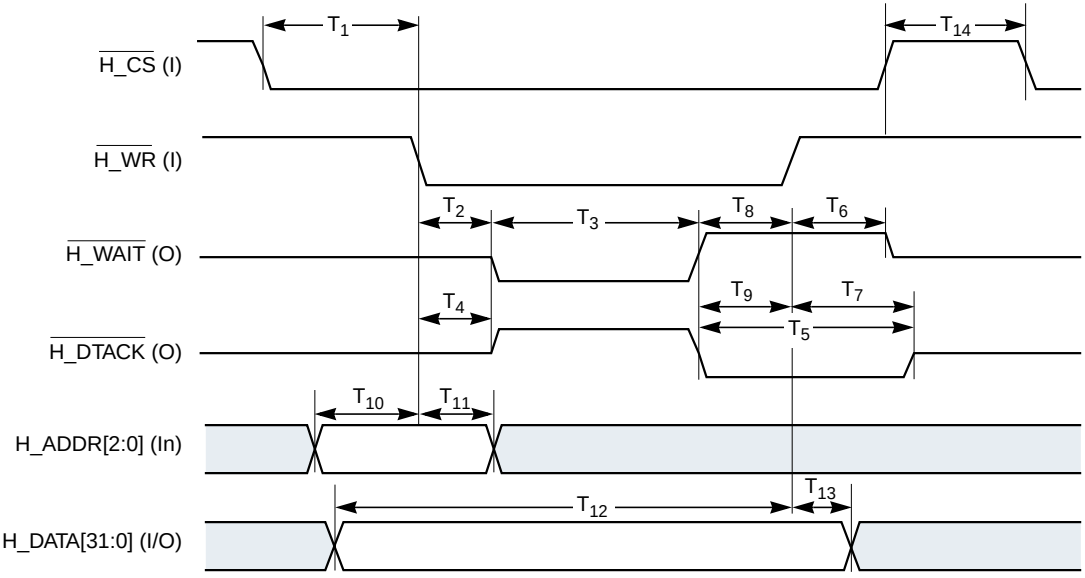


Table 18.7 I-Mode Write AC Timing Parameters[1]

Symbol	Description	Timing Value	
		Min	Max
T_1	$\overline{H\_CS}$ setup time with respect to $\overline{WR}$ falling.	3.0	–
T_2	$\overline{H\_WAIT}$ output delay time with respect to $\overline{WR}$ falling.		3 cycles + 14 ns
T_3	$\overline{H\_WAIT}$ assertion period.	2 cycles	–
T_4	$\overline{H\_DTACK}$ output delay time with respect to $\overline{WR}$ falling	–	3 cycles + 14 ns
T_5	$\overline{H\_DTACK}$ assertion period.	2 cycles	–
T_6	Output delay from $\overline{WR}$ rising to $\overline{H\_WAIT}$ 3-state.	2 cycles	3 cycles
T_7	Output delay from $\overline{WR}$ rising to $\overline{H\_DTACK}$ 3-state.	2 cycles	3 cycles
T_8	$\overline{WR}$ hold time with respect to WAIT rising.	3.0 ns	–
T_9	$\overline{WR}$ hold time with respect to DTACK falling.	3.0 ns	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

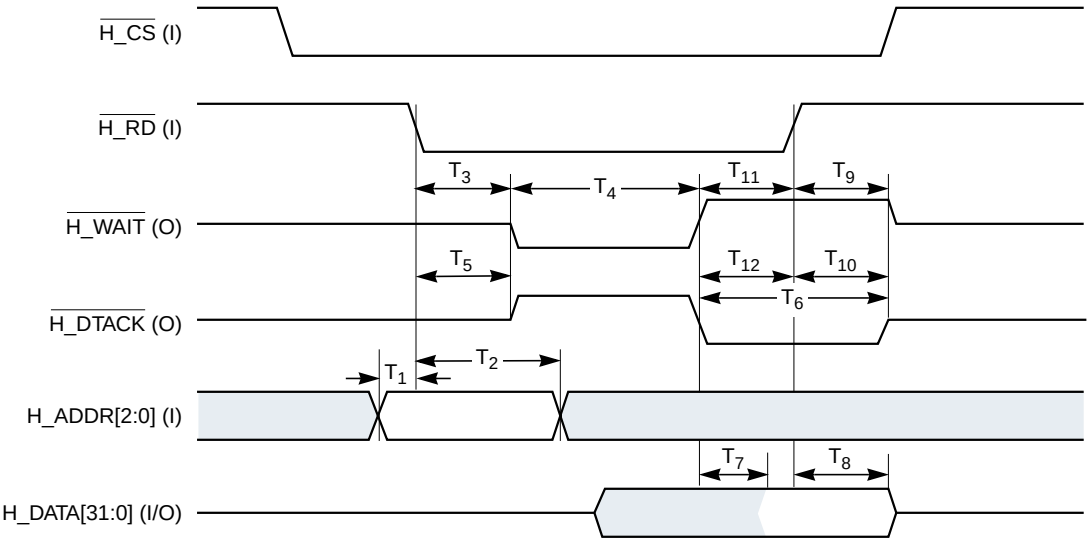
Table 18.7 I-Mode Write AC Timing Parameters[1] (Cont.)

Symbol	Description	Timing Value	
		Min	Max
T_{10}	H_ADDR[2:0] setup time with respect to $\overline{WR}$ falling.	3.0 ns	–
T_{11}	H_ADDR[2:0] hold time with respect to $\overline{WR}$ falling.	2.0 ns	–
T_{12}	H_DATA[31:0] setup time with respect to $\overline{WR}$ rising.	3.0 ns	–
T_{13}	H_DATA[31:0] hold time with respect to $\overline{WR}$ rising.	2.0 ns	–
T_{14}	$\overline{H\_CS}$ high period.	2 cycles	

1. $\overline{H\_WAIT}$ and $\overline{H\_DTACK}$ are pulled up by an internal pull-up on 3-state.

18.2.3.2 I-Mode Read AC Timing

Figure 18.13 I-Mode Read AC Timing Diagram



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.8 I-Mode Read AC Timing Parameters[1]

Symbol	Description	Timing Value	
		Min	Max
T_1	H_ADDR[2:0] input setup with respect to $\overline{RD}$ falling.	3.0 ns	–
T_2	H_ADDR[2:0] input hold time with respect to $\overline{RD}$ falling.	2.0 ns	–
T_3	$\overline{H\_WAIT}$ output delay time with respect to $\overline{RD}$ falling.	–	3 cycles + 14 ns
T_4	$\overline{H\_WAIT}$ assertion period.	2 cycles	–
T_5	$\overline{H\_DTACK}$ output delay time with respect to $\overline{RD}$ falling.	–	3 cycles + 14 ns
T_6	$\overline{H\_DTACK}$ assertion period.	2 cycles	–
T_7	Delay from $\overline{H\_WAIT}$ rising to data valid.	–	1 cycle
T_8	Delay from $\overline{RD}$ rising to data float.	2 cycles	–
T_9	Output delay from $\overline{RD}$ rising to $\overline{H\_WAIT}$ 3-stated.	2 cycles	3 cycles
T_{10}	Output delay from $\overline{RD}$ rising to $\overline{H\_DTACK}$ 3-stated.	2 cycles	3 cycles
T_{11}	$\overline{RD}$ holdtime with respect to $\overline{H\_WAIT}$ rising.	2.0 ns	–
T_{12}	$\overline{RD}$ holdtime with respect to $\overline{H\_DTACK}$ falling.	2.0 ns	–

1. $\overline{H\_WAIT}$ and $\overline{H\_DTACK}$ are pulled up by an internal pull-up on 3-state.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.3.3 M-Mode Write Timing

Figure 18.14 M-Mode Write AC Timing Diagram

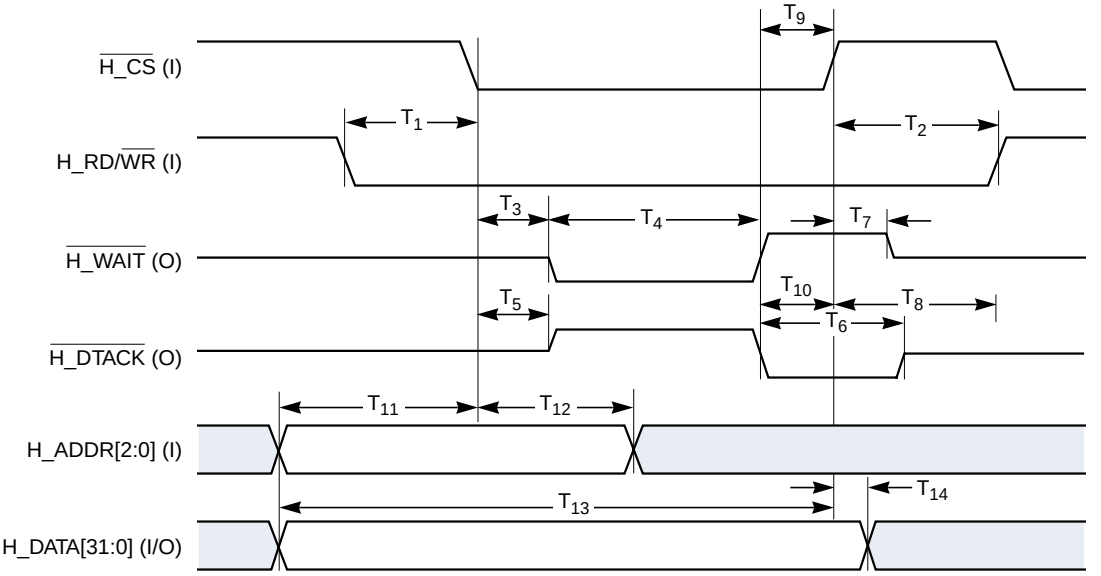


Table 18.9 M-Mode Write AC Timing Parameters[1]

Symbol	Description	Timing Value	
		Min	Max
T_1	$\overline{WR}$ setup time with respect to $\overline{H\_CS}$ falling.	3.0 ns	–
T_2	$\overline{WR}$ hold time with respect to $\overline{H\_CS}$ rising.	2.0 ns	–
T_3	$\overline{H\_WAIT}$ output delay time with respect to $\overline{H\_CS}$ falling	–	3 cycles + 14 ns
T_4	$\overline{H\_WAIT}$ assertion period.	2 cycles	–
T_5	$\overline{DTACK}$ output delay time with respect to $\overline{WR}$ falling	–	3 cycles + 14 ns
T_6	$\overline{DTACK}$ assertion period.	2 cycles	–
T_7	Output delay from $\overline{H\_CS}$ rising to $\overline{H\_WAIT}$ 3-state.	2 cycles	3 cycles
T_8	Output delay from $\overline{H\_CS}$ rising to $\overline{H\_DTACK}$ 3-state.	2 cycles	3 cycles
T_9	$\overline{H\_CS}$ hold time with respect to $\overline{H\_WAIT}$ rising.	2.0 ns	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

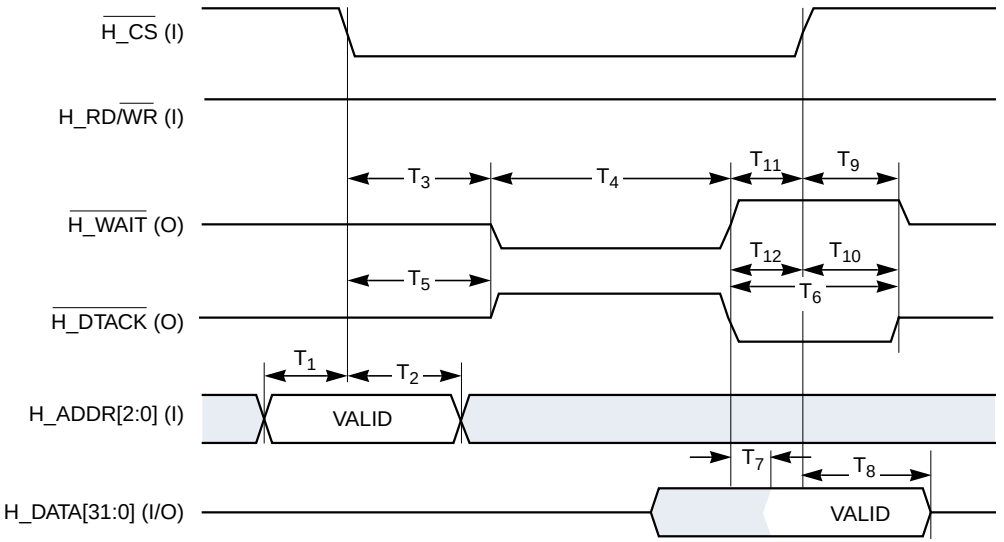
Table 18.9 M-Mode Write AC Timing Parameters[1] (Cont.)

Symbol	Description	Timing Value	
		Min	Max
T_{10}	$\overline{H\_CS}$ hold time with respect to $\overline{H\_DTACK}$ falling.	2.0 ns	–
T_{11}	H_ADDR[2:0] setup time with respect to $\overline{H\_CS}$ falling.	3.0 ns	–
T_{12}	H_ADDR[2:0] hold time with respect to $\overline{H\_CS}$ falling.	2.0 ns	–
T_{13}	H_DATA[31:0] setup time with respect to $\overline{H\_CS}$ rising.	3.0 ns	–
T_{14}	H_DATA[31:0] hold time with respect to $\overline{H\_CS}$ rising.	2.0 ns	–

1. $\overline{H\_WAIT}$ and $\overline{H\_DTACK}$ are pulled up by an internal pull-up on 3-state.

18.2.3.4 M-Mode Read AC Timing

Figure 18.15 M-Mode Read AC Timing Diagram



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.10 M-Mode Read AC Timing Parameters[1]

Symbol	Description	Timing Value	
		Min	Max
T_1	H_ADDR[2:0] input setup time with respect to $\overline{\text{H\_CS}}$ falling.	3.0 ns	–
T_2	H_ADDR[2:0] input hold time with respect to $\overline{\text{H\_CS}}$ falling.	2.0 ns	–
T_3	$\overline{\text{H\_WAIT}}$ output delay time with respect to $\overline{\text{H\_CS}}$ falling.	–	3 cycles + 14 ns
T_4	$\overline{\text{H\_WAIT}}$ assertion period.	2 cycles	–
T_5	$\overline{\text{H\_DTACK}}$ output delay time with respect to $\overline{\text{H\_CS}}$ falling.	–	3 cycles + 14 ns
T_6	$\overline{\text{H\_DTACK}}$ assertion period.	2 cycles	–
T_7	Delay from $\overline{\text{H\_WAIT}}$ rising to data valid.	–	1 cycle
T_8	Delay from $\overline{\text{RD}}$ rising to data float.	2 cycles	–
T_9	Output delay from $\overline{\text{H\_CS}}$ rising to $\overline{\text{H\_WAIT}}$ 3-stated.	2 cycles	3 cycles
T_{10}	Output delay from $\overline{\text{H\_CS}}$ rising to $\overline{\text{H\_DTACK}}$ 3-stated.	2 cycles	3 cycles
T_{11}	$\overline{\text{H\_CS}}$ hold time with respect to $\overline{\text{H\_WAIT}}$ rising.	2.0 ns	–
T_{12}	$\overline{\text{H\_CS}}$ hold time with respect to $\overline{\text{DTACK}}$ falling.	2.0 ns	–

1. $\overline{\text{H\_WAIT}}$ and $\overline{\text{H\_DTACK}}$ are pulled up by an internal pull-up on 3-state.

18.2.4 SDRAM Interface AC Timing

SDRAM AC Timing diagrams are described in the sections that follow.

18.2.4.1 Clock Signals to SDRAM

The clock signal to the SDRAM is driven from the DMN-8600. For SDR SDRAM, only the SDRAM_CLK output is used to connect to the clock input of the DRAM. For DDR SDRAM, the differential clock signal is used, and both SDRAM_CLK and $\overline{\text{SDRAM\_CLK}}$ are connected to CLK and $\overline{\text{CLK}}$ of the DRAM, respectively.

For SDR mode operation, the Clock high (low) period is defined as the period when the clock signal is above (below) the defined Vih (Vil) voltage level. For DDR mode, the crossing of CLK and $\overline{\text{CLK}}$ is used as the reference.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Note: According to the specification, the HIGH/LOW period for the clock signal should be a minimum of 40% of the cycle time in SDR SDRAM, and in the range of 45–55% of the cycle time in DDR SDRAM.

Figure 18.16 SDRAM Clock LOW and HIGH Period Definition

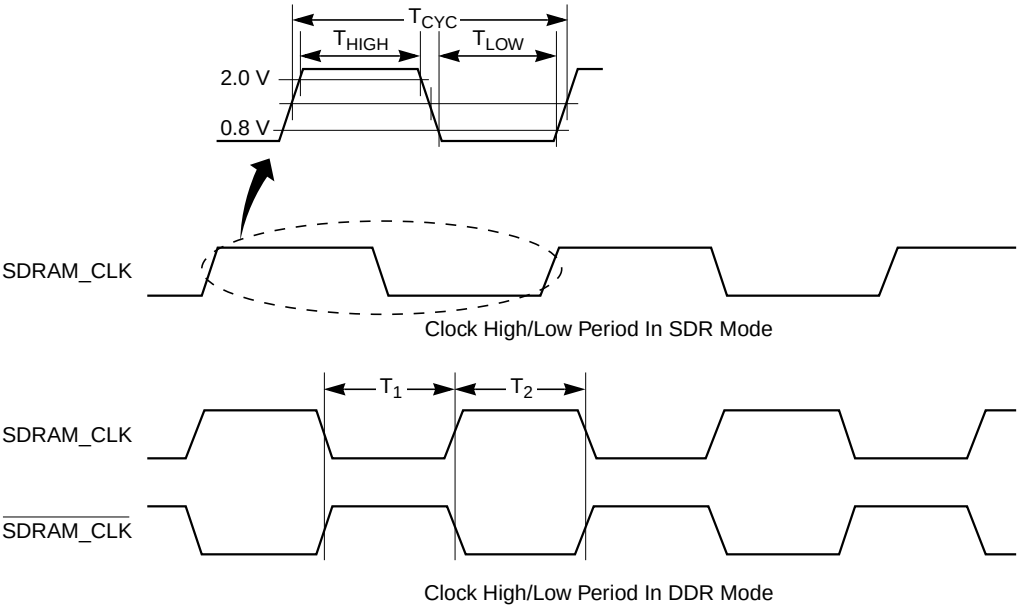


Table 18.11 Clock Signals to SDRAM Timing

Symbol	Description	Min	Max	Units
T_{CYC}	Clock cycle period in SDR or DDR mode	81	148.5	MHz
T_{HIGH}	Clock HIGH period in SDR mode	0.4	0.6	T_{CYC}
T_{LOW}	Clock LOW period in SDR mode	0.4	0.6	T_{CYC}
T_1	Clock HIGH period in DDR mode	0.45	0.55	T_{CYC}
T_2	Clock LOW period in DDR mode	0.45	0.55	T_{CYC}

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.4.2 DMN-8600 Writing to SDRAM in SDR Mode

SDRAM writes are in burst mode. A burst length of 8 is programmed during initialization. The Burst Stop command is issued to terminate writes when needed. Valid data is driven in the cycle when a write command is issued.

Figure 18.17 DMN-8600 Writing to SDRAM in SDR Mode

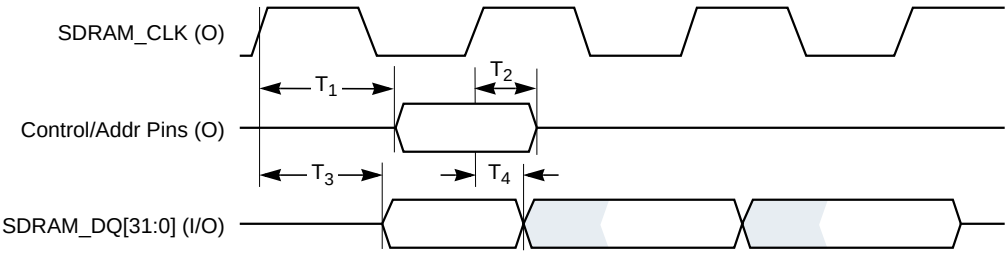


Table 18.12 DMN-8600 Write to SDRAM Parameters

Symbol	Description	Min	Max	Units
T_1	Control/Addr pins output delay		4.5	ns
T_2	Control/Addr pins hold time	1.0		
T_3	Output Data delay		4.5	ns
T_4	Output Data hold time	1.0		ns

Please note the following information:

- Control pins include $\overline{\text{SDRAM\_CAS}}$, $\overline{\text{SDRAM\_RAS}}$, $\overline{\text{SDRAM\_WE}}$, and address pins include SDRAM_A[15:0], all of which output of the DMN-8600.
- SDRAM_DQ[31:0] is driven from DMN-8600 for writes to SDRAM.

18.2.4.3 DMN-8600 Reading from SDRAM in SDR Mode

SDRAM reading, like writing, is also in burst mode. CAS latency for a DRAM read is programmed once during initialization of the memory chip. For SDR mode, it can be 2 or 3.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

A read command is issued to read data from SDRAM. DRAM returns data on the 2nd/3rd cycle after the command, depending on the CAS latency.

Figure 18.18 DMN-8600 Reading from SDRAM in SDR Mode

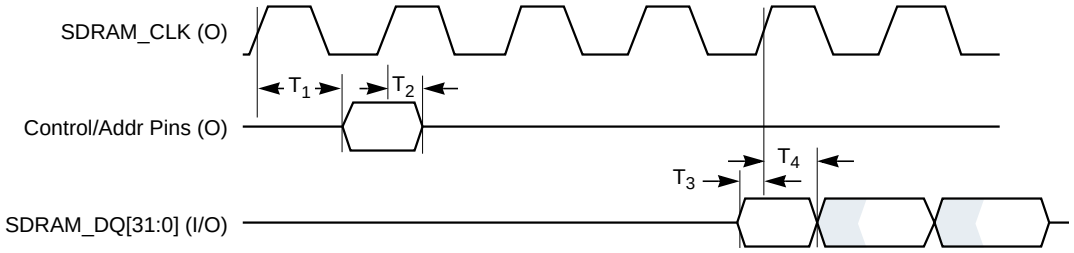


Table 18.13 DMN-8600 Read from SDRAM (SDR Mode) Parameters

Symbol	Description	Min	Max	Unit
T1	Control/Addr pins output delay		4.5	ns
T2	Control/Addr pins hold time	1.0		ns
T3	Input data setup time	2.0		ns
T4	Input data hold time	1.0		ns

Please note the following related information:

- Control pins include $\overline{\text{SDRAM\_CAS}}$, $\overline{\text{SDRAM\_RAS}}$, $\overline{\text{SDRAM\_WE}}$, and address pins include SDRAM_A[15:0], all of which are outputs of DoMiNo
- SDRAM_DQ[31:0] is driven from the external DRAM chip.
- A clock (with programmable phase shift) is utilized to capture the input data on SDRAM_DQ. So, the value of T3 can be negative–that is, data can appear on the cycle after the one indicated by the CAS latency value. The internal programmable clock will be able to capture the data up to the specified delay.

18.2.4.4 DMN-8600 Writing to SDRAM in DDR Mode

For DDR mode, when writing into the SDRAM, DMN-8600 drives SDRAM_DQS signals together with the SDRAM_DQ pin. SDRAM_DQS

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

should be used as the sampling edge for the data on SDRAM_DQ. DQS is generally placed in the center of the valid data window.

Figure 18.19 DMN-8600 Write to SDRAM in DDR Mode

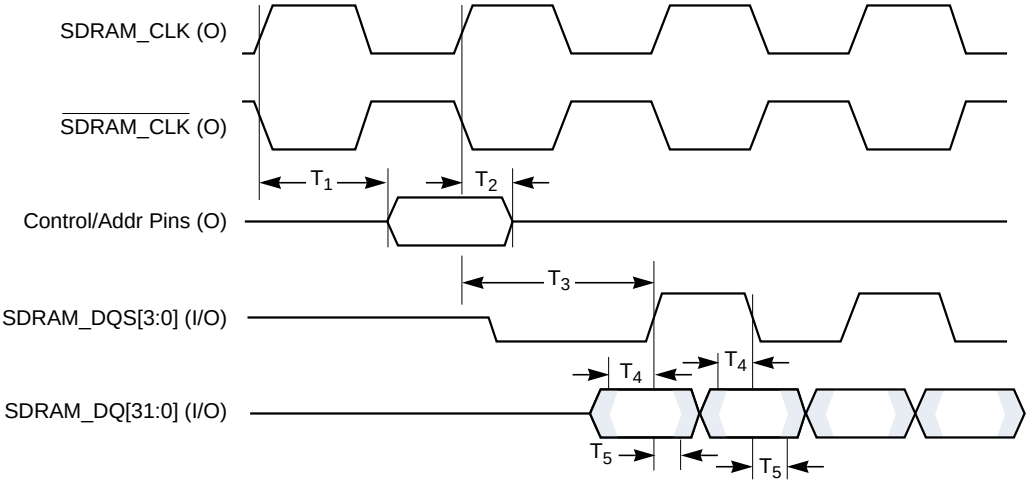


Table 18.14 DMN-8600 Write to SDRAM (DDR Mode) Parameters

Symbol	Description	Min	Max	Unit
T_1	Control/Addr pins output delay		4.5	ns
T_2	Control/Addr pins hold time	1.0		ns
T_3	Output data strobe delay	0.75	1.25	T_{CYC}
T_4	Output data setup time	1.5		ns
T_5	Output data hold time	1.0		ns

Please note the following related information:

- Control pins include $\overline{\text{SDRAM\_CAS}}$, $\overline{\text{SDRAM\_RAS}}$, $\overline{\text{SDRAM\_WE}}$, and address pins include SDRAM_A[15:0], all of which are outputs of DoMiNo
- SDRAM_DQ[31:0] and SDRAM_DQS[3:0] are driven from DMN-8600 for writes to SDRAM.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.4.5 DMN-8600 Read from SDRAM in DDR Mode

For DDR mode when reading from SDRAM, the SDRAM drives the SDRAM_DQS signals together with the SDRAM_DQ pin. DQS pins are generally edge-aligned with the data on SDRAM_DQ. The strobe is delayed within DMN-8600 so that the edge can be used to sample data.

Figure 18.20 DMN-8600 Read from SDRAM in DDR Mode

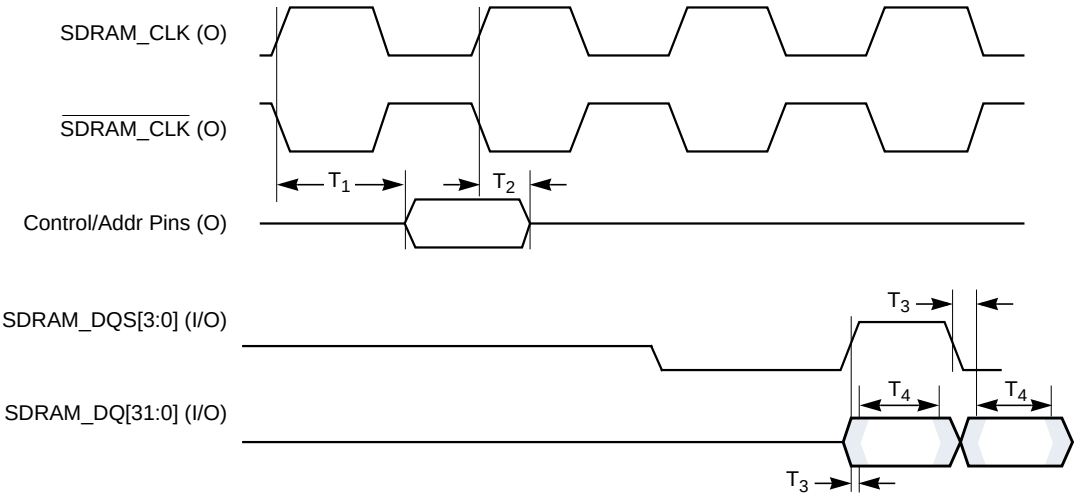


Table 18.15 DMN-8600 Read from SDRAM (DDR mode) Parameters

Symbol	Description	Min	Max	Unit
T_1	Control/Addr pins output delay	–	4.5	ns
T_2	Control/Addr pins hold time	1.0	–	ns
T_3	Data strobe edge to output data edge	–	0.6	ns
T_4	Input data valid time	0.3	–	T_{CYC}

Please note the following related information:

- Control pins include $\overline{\text{SDRAM\_CAS}}$, $\overline{\text{SDRAM\_RAS}}$, $\overline{\text{SDRAM\_WE}}$, and address pins include SDRAM_A[15:0], all of which are outputs of DoMiNo.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- SDRAM_DQ[31:0] and SDRAM_DQS[3:0] are driven from the SDRAM for reads.

18.2.5 CD Interface Timing

Figure 18.21 and Table 18.16 show the timing for the CD signal input.

Figure 18.21 CD Interface Timing

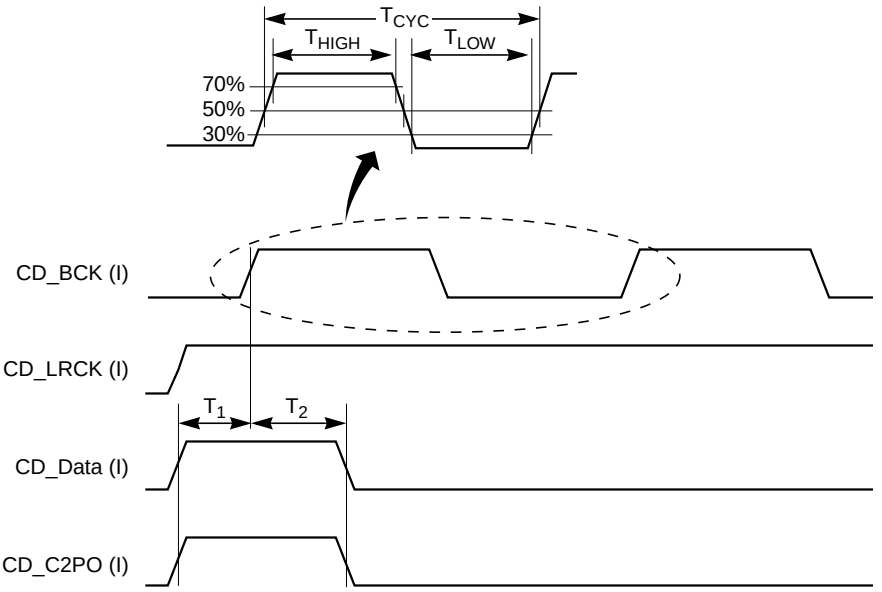


Table 18.16 CD Input Timing

Symbol	Description	Min	Max	Units
T_{CYC}	Cycle clock period	100		
T_{HIGH}	CD_BCK HIGH pulse width	50		ns
T_{LOW}	CD_BCK LOW pulse width	50		ns
T_1	CD_DATA, CD_LRCK, CD_C2PO setup	10		ns
T_2	CD_DATA, CD_LRCK, CD_C2PO hold	10		ns

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.6 IDC Interface Timing

IDC Interface timing is divided into the following two groups:

- IDC Slave Timing. See Figure 18.22 and Table 18.17.
- IDC Master Timing. See Figure 18.23 and Table 18.18.

Figure 18.22 IDC Interface AC Slave Timing

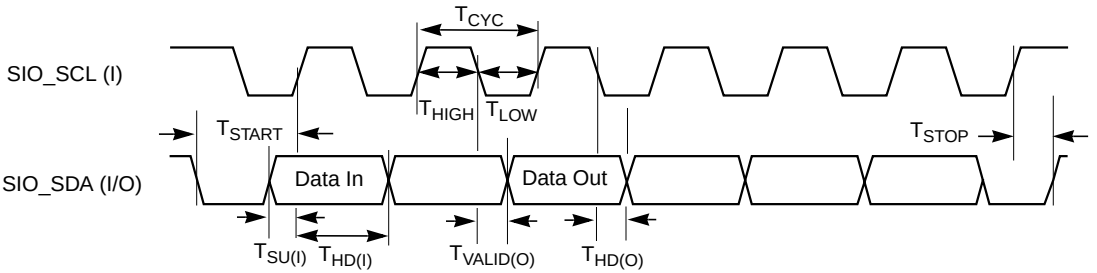


Table 18.17 IDC Interface Slave Timing Parameter

Param	Description	IDC Design Values (in sysclk cycles)	
		Min	Max
T_{HIGH}	High period of SIO_SCL(I)	M{(S+1)(R+1), 3}	------
T_{LOW}	Low period of SIO_SCL(I)	M{(S+1)(F+1), 2}	------
T_{CYC}	Clock period of SIO_SCL(I)	M{(S+1)[(R+1)+(F+1)], 4}	------

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.17 IDC Interface Slave Timing Parameter

Param	Description	IDC Design Values (in sysclk cycles)	
		Min	Max
T_{START}	Delay from SIO_SDA(I) falling to next SIO_SCL(I) rising	M{(S+1)(F- R + 1), 2}	------
T_{STOP}	Delay from SIO_SCL(I) rising to SIO_SDA(I) rising	M{(S+1), 2}	------
$T_{SU(I)}$	SIO_SDA(I) set up time before SIO_SCL(I) rising	M{(S+1)(F-R) for SIO_SDA(I) falling only, 0}	------
$T_{HD(I)}$	SIO_SDA(I) hold time after SIO_SCL(I) rising	M{(S+1) for SIO_SDA(I) rising after, (S+1)(R - F+1) for SIO_SDA(I) falling after, 3}	------
$T_{VALID(O)}$	Delay from SIO_SCL(I) falling to SIO_SDA(O) valid	9 + (S+1)(F)	8 + (S+1)(F+1)
$T_{HD(O)}$	SIO_SDA(O) hold time after SIO_SCL(I) falling	9 + (S+1)(F)	8 + (S+1)(F+1)

Notes:

- The S, R, F parameters are programmable (from 0 - 15) register fields for RX sampling and filtering located in the IDC Receive Filter (IDC_RX_FILTER) register (see Section 15.5.3, “SIO IDC Registers”):

  S = IDCSAMP[3:0]

  R = IDCRISE[3:0]

  F = IDCFALL[3:0]

- M{a,b} = Max{a,b} = greater of a and b

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.23 IDC Interface AC Master Timing

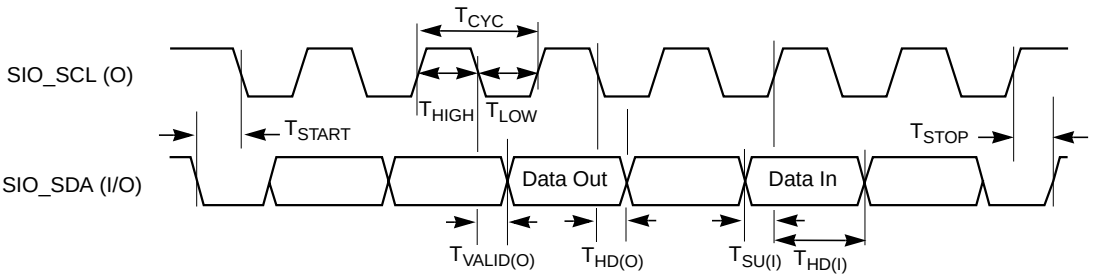


Table 18.18 IDC Interface Master Timing Parameter

Param	Description	IDC Design Values (in sysclk cycles)	
		Min	Max
T_{HIGH}	High period of SIO_SCL(O)	8 + 2(C) + (S+1)(R)	7+2(C) + (S+1)(R+1)
T_{LOW}	Low period of SIO_SCL(O)	2(C)	
T_{CYC}	Clock period of SIO_SCL(O)	8 + 4(C) + (S+1)(R)	7 + 4(C) + (S+1)(R+1)
T_{START}	Delay from SIO_SDA(O) falling to next SIO_SCL(O) falling	(C)	
T_{STOP}	Delay from SIO_SCL(O) rising to SIO_SDA(O) rising	10 + 2(C) + (S+1)(R)	9 + 2(C) + (S+1)(R+1)
$T_{SU(I)}$	SIO_SDA(I) set up time before SIO_SCL(O) rising	M{(S+1)(F-R) for SIO_SDA(I) falling only, 0}	------
$T_{HD(I)}$	SIO_SDA(I) hold time after SIO_SCL(O) rising	M{(S+1) for SIO_SDA(I) rising after, (S+1)(R - F+1) for SIO_SDA(I) falling after, 3}	------
$T_{VALID(O)}$	Delay from SIO_SCL(O) falling to SIO_SDA(O) valid	C + 2	
$T_{HD(O)}$	SIO_SDA(O) hold time after SIO_SCL(O) falling	C + 2	

Notes:

- The C parameter is a programmable (from 2 - 1023) register field for output clock frequency located in the IDC Clock (IDC_CLOCK) register (see Section 15.5.3, “SIO IDC Registers”):

  C = IDCSCL[9:0]

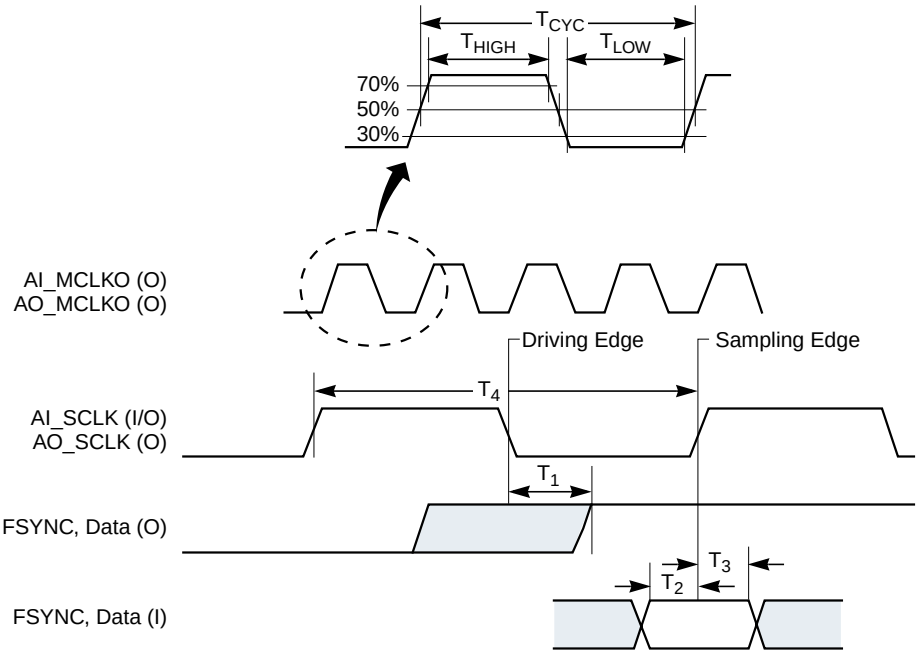
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

- For legend for S, R, F, and M{a,b}, see Notes following Table 18.17.

18.2.7 Audio Timing

Audio AC timing is shown in Figure 18.24 and is referenced in Table 18.19.

Figure 18.24 Audio Input/Output AC Timing



Note: Diagram shown is with falling driving edge/rising sampling edge. The same parameters are valid for rising driving edge/falling sample edge.

Table 18.19 Audio Input/Output AC Timing Parameters

Symbol	Description	Min[1]	Typ	Max	Unit
T_{CYC}	MCLK cycle clock period	19.2			ns
T_{HIGH}	MCLK high pulse width		8.5		
T_{LOW}	MCLK low pulse width		8.5		

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.19 Audio Input/Output AC Timing Parameters

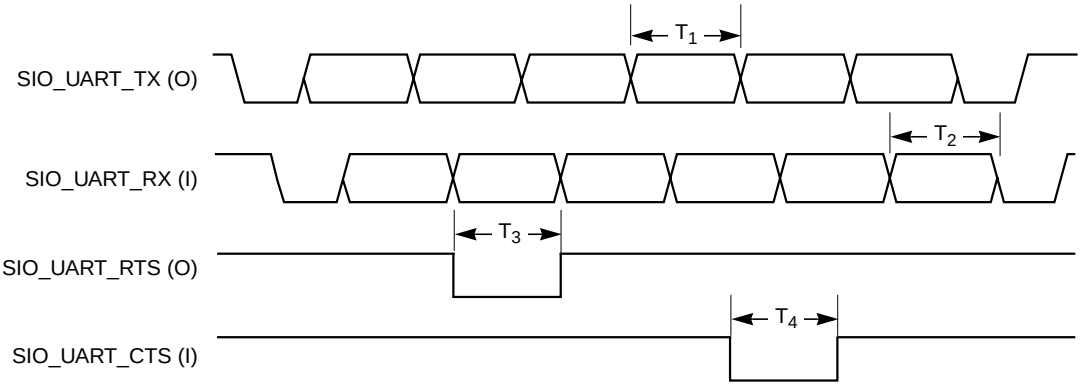
Symbol	Description	Min[1]	Typ	Max	Unit
T1	Driving edge SCLK to FSYNC/data output delay	–10		10	ns
T2	FSYNC/data input to sampling edge SCLK setup	10			ns
T3	FSYNC/data input to sampling edge SCLK hold	10			ns
T4[2]	SCLK period (Absolute)	40[3]			ns
T4[2]	SCLK period (Relative): - Less than four data pins active, IEC958 active, R958=0, and OTim =0 - Four data pins active - Less than four data pins active	 6 cycles 9 cycles 5 cycles			ns

1. “Cycles” refers to internal system clock cycles.
2. Four data pins are active if ChCnt=3, or FrForm=1, or (FrForm=0 && ChCnt=4). See Audio Output Control register and Audio Input Control register for bit descriptions.
3. The minimum SCLK period is the greater of the absolute or relative minimum period

18.2.8 UART Interface Timing

UART is an asynchronous, two-wire protocol: one for transmitting data, and one for receiving data. Figure 18.25 shows timing parameters for the UART interface. Table 18.20 lists timing values for UART parameters.

Figure 18.25 UART Interface AC Timing



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

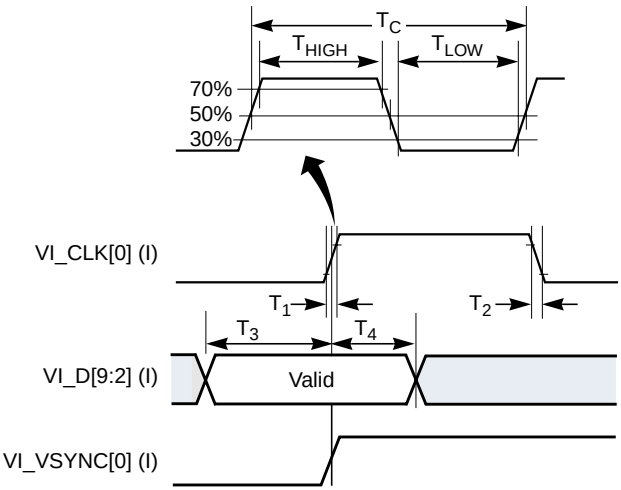
Table 18.20 UART Interface AC Timing

Symbol	Description	Value
T_1, T_2, T_3, T_4	Minimum data valid time for signals on SIO_UART pins.	3 sysclk cycles

18.2.9 Video Interface Timing

Figure 18.26 and Table 18.21 show the AC timing for the DMN-8600 device Video interface.

Figure 18.26 AC Timing for Video Input Stream at VI_CLK[0]



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.21 Video Input Stream AC Timing Parameters at VI_CLK[0]

Symbol	Description	Timing Value			Unit
		Min	Typ	Max	
T_C	Cycle time	13.46 (74.25 MHz)		37.03 (27 MHz)	ns
T_1	Rise time	0.5		5.0 (27 MHz) 2.0 (74.25 MHz)	ns
T_2	Fall time	0.5		5.0 (27 MHz) 2.0 (74.25 MHz)	ns
T_3	Input data setup time (VI_D[9:2] and VI_VSYNC[0]) before the rising edge of VI_CLK[0]	3.0			
T_4	Input data hold time for VI_D[9:2], VI_VSYNC[0] after the rising edge of VI_CLK[0]	0			ns

18.2.9.1 Video Output

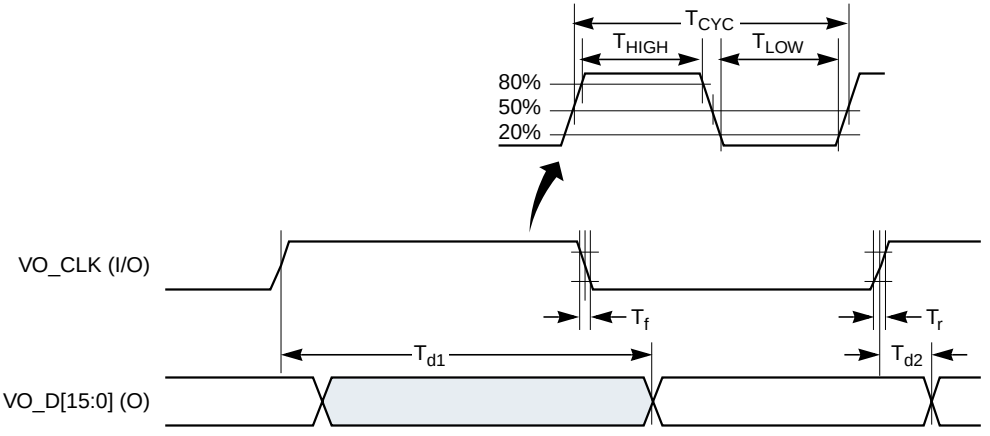
For video output, Table 18.22 defines the video output clock source. The timing of all video I/O ports of the video output channel is defined with respect to the VO_CLK pin and the OSync register bit setting of the Video Status register.

Table 18.22 Video Out Clock Source

OSync	Video Out PLL Source	VO_CLK
0	0	Video out clock is internally generated; DMN-8600 drives out VO_CLK pin
1	Don't care	VI_CLK[0]; DMN-8600 drived out VO_CLK pin
0	1	VI_CLK[0]; DMN-8600 drived out VO_CLK pin
0	2	Externally generated video output clock; VO_CLK pin in input

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.27 AC Timing of Video Output at VO_CLK



Note: If OSync is set (or VoutPLL SRC value is 1), then VO_D timing is specified relative to the video clock input (VI_CLK[0]) and video output clock should not be used by the system.

Table 18.23 Video Output AC Timing Parameters at VO_CLK

Symbol	Description	Timing Value			Unit
		Min	Typ	Max	
t_C	Cycle time	13.46		37.03	ns
t_r	Rise time	0.5		5.0 (27 MHz) 2.0 (74.25 MHz)	ns
t_f	Fall time	0.5		5.0 (27 MHz) 2.0 (74.25 MHz)	ns
t_{d1}	Output data valid time VO_D[15:0] after the rising edge of VO_CLK			8.0	ns
t_{d2}	Output data hold time for VO_D[15:0] after the rising edge of VO_CLK	0.5			ns

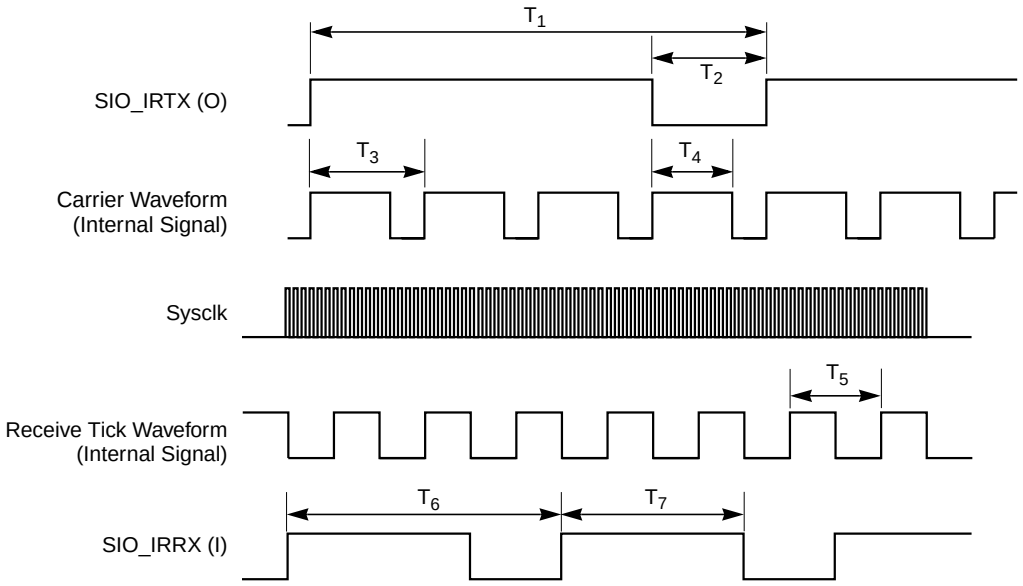
Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.10 IR Interface Timing

DMN-8600 has two IR modules: one with full-duplex operation (TX and RX), and one with TX-only functionality. The DMN-8600 IR modules offload most of the formatting and protocol issues to software; for transmit, the IR module generates waveforms with the programmed pulse length and period characteristics, while for receive, it simply measures the period and duty cycle of incoming pulses.

Since there is no generic IR protocol which can be described, the waveforms shown below illustrate how the values read/written from various IR registers relate to the actual waveforms themselves. Waveforms for two common protocols, NCR and Philips RC-5, are also shown below.

Figure 18.28 IR Interface Timing



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.24 IR AC Timing Parameters

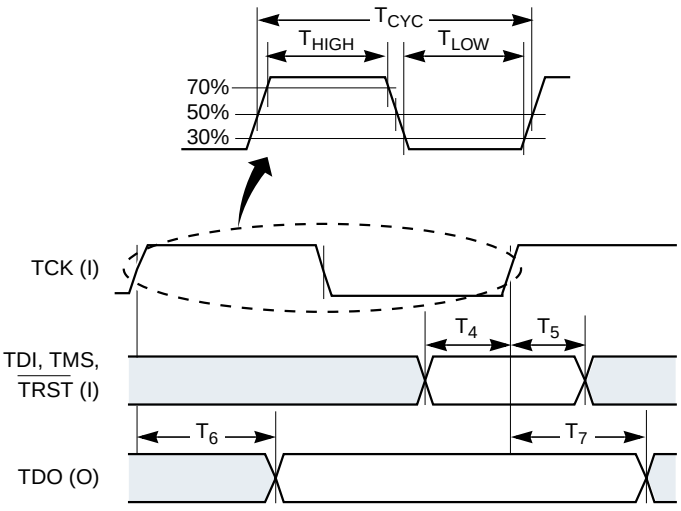
Symbol	Description	Min	Max
T_1	IRTX modulated signal pulse: user can program overall period of IRTX waveform in units of carrier wave cycles.	0x0 + 1 carrier cycles	0x3ff + 1 carrier cycles
T_2	IRTX modulated signal pulse Low: user can program duty cycle of IRTX waveform in units of carrier wave cycles.	0x0 + 1 carrier cycles	0xff + 1 carrier cycles
T_3	IRTX carrier wave period: user can program period of carrier waveform in units of system clock cycles.	0x0 + 1 sys clock cycles	0x3fff + 1 sys clock cycles
T_4	IRTX modulated signal pulse High: user can program duty cycle of carrier waveform in units of system clock cycles.	0x0 + 1 sys clock cycles	0x1fff + 1 sys clock cycles
T_5	IRRX receive tick period: User can program length of an “IR tick” in terms of system clock cycles.	0x0 + 1 sys clock cycles	0x7fff + 1 sys clock cycles
T_6	IRRX receive tick count: IR module counts number of IR ticks between successive falling (or rising) edges of IRRX.	0x0 + 1 IR ticks	0xff + 1 IR ticks
T_7	IRRX receive high tick count: IR module counts number of IR ticks that IRRX is high.	0x0 + 1 IR ticks	0xff + 1 IR ticks

18.2.11 JTAG Interface Signal Timing

The timing parameters for the JTAG interface are shown in Figure 18.29 and described in Table 18.25.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.29 JTAG Interface Timing Diagram



Note: $\overline{\text{TRST}}$ is an asynchronous reset.

Table 18.25 JTAG Interface AC Timing Values

Symbol	Description	Timing Value (ns)	
		Min	Max
T_{CYC}	TCK period	100.0	–
T_{HIGH}	TCK HIGH time	40.0	–
T_{LOW}	TCK LOW time	40.0	–
T_4	TDI, TMS, $\overline{\text{RST}}$ setup time to TCK	–	10.0
T_5	TDI, TMS, $\overline{\text{RST}}$ hold time from TCK	5.0	–
T_6	TDO delay time from TCK	–	2.0
T_7	TDO hold time from CLK	1.0	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.12 ATAPI AC Timing

18.2.12.1 ATAPI DMA Protocol

Figure 18.30 ATAPI DMA AC Timing

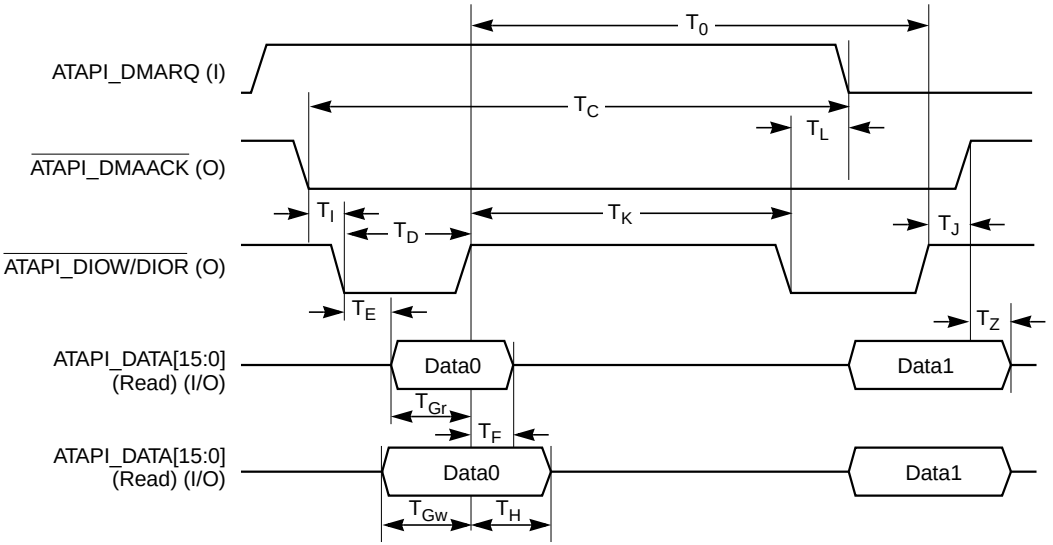


Table 18.26 ATAPI DMA Protocol Timing

Symbol	Parameters	Mode0 ns	Mode1 ns	Mode2 ns	Comment
t_0	Cycle time (min)	480	150	120	
t_C	$\overline{\text{ATAPI\_DMAACK}}$ to ATAPI_DMARQ delay (max)[1]	200	100	80	
t_D	$\overline{\text{ATAPI\_DIOR}}$/$\overline{\text{ATAPI\_DIOW}}$ 16-bit (min)	215	80	70	RWTime * C[2]
t_E	$\overline{\text{ATAPI\_DIOR}}$ data access (max)	250	150	60	
t_F	$\overline{\text{ATAPI\_DIOR}}$ data hold (min)	5	5	5	
t_{Gr}	$\overline{\text{ATAPI\_DIOR}}$ data setup (min)	100	30	20	
t_{Gw}	$\overline{\text{ATAPI\_DIOW}}$ data setup (min)	100	30	20	
t_H	$\overline{\text{ATAPI\_DIOW}}$ data hold (min)	20	15	10	RWHold * C

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.26 ATAPI DMA Protocol Timing (Cont.)

Symbol	Parameters	Mode0 ns	Mode1 ns	Mode2 ns	Comment
t_I	$\overline{\text{ATAPI\_DMAACK}}$ to $\overline{\text{ATAPI\_DIOR}}$/$\overline{\text{ATAPI\_DIOW}}$ setup (min)	0	0	0	
t_J	$\overline{\text{ATAPI\_DIOR}}$/$\overline{\text{ATAPI\_DIOW}}$ to $\overline{\text{ATAPI\_DMAACK}}$ hold (min)	20	15	10	RWHold * C
t_{Kr}	$\overline{\text{ATAPI\_DIOR}}$ negated pulse width (min)	50	50	25	RWWidth * C
t_{Kw}	$\overline{\text{ATAPI\_DIOW}}$ negated pulse width (min)	215	50	25	RWWidth * C
t_{Lr}	$\overline{\text{ATAPI\_DIOR}}$ to ATAPI_DMARQ delay (max)	120	40	35	
t_{Lw}	$\overline{\text{ATAPI\_DIOW}}$ to ATAPI_DMARQ delay (max)	40	40	35	
t_Z	$\overline{\text{ATAPI\_DMAACK}}$ to 3-state (max)	20	25	25	

1. This timing only applies to a single DMA transfer.
2. “C” is the system clock cycle time.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.12.2 ATAPI PIO Read and Write Protocol

Figure 18.31 ATAPI PIO Read and Write Timing

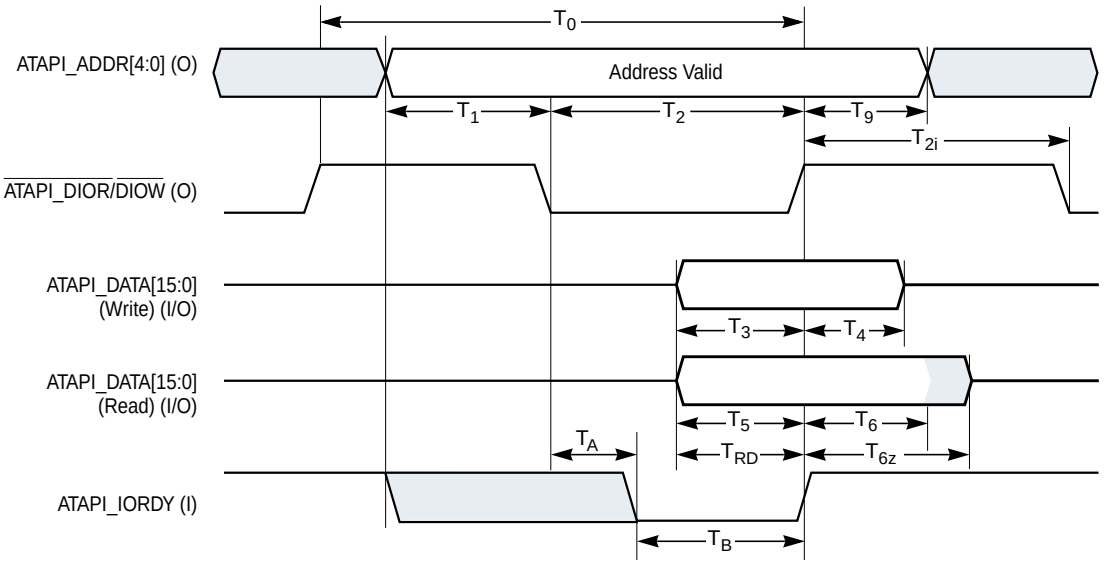


Table 18.27 ATAPI PIO Read and Write Timing Parameters

Symbol	Timing Parameters	Mode0 ns	Mode1 ns	Mode2 ns	Mode3 ns	Mode4 ns
t0	Cycle time (min)	600	383	240	180	120
t1	Address valid to $\overline{\text{ATAPI\_DIOR}}$/ $\overline{\text{ATAPI\_DIOW}}$ setup (min)	70 (RWSetup * C)	50 (RWSetup * C)	30 (RWSetup * C)	30 (RWSetup * C)	25 (RWSetup * C)
t2	$\overline{\text{ATAPI\_DIOR}}$/ $\overline{\text{ATAPI\_DIOW}}$ (16-bit) (min)	165 (RWTime * C)	125 (RWTime * C)	100 (RWTime * C)	80 (RWTime * C)	70 (RWTime * C)
t2i	$\overline{\text{ATAPI\_DIOR}}$/ $\overline{\text{ATAPI\_DIOW}}$ recovery time (min)	- (RWRcv * C)	- (RWRcv * C)	- (RWRcv * C)	70 (RWRcv * C)	25 (RWRcv * C)
t3	$\overline{\text{ATAPI\_DIOW}}$ data setup (min)	60	45	30	30	20

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.27 ATAPI PIO Read and Write Timing Parameters (Cont.)

Symbol	Timing Parameters	Mode0 ns	Mode1 ns	Mode2 ns	Mode3 ns	Mode4 ns
t4	$\overline{\text{ATAPI\_DIOW}}$ data hold (min)	30 (RWHold * C)	20 (RWHold * C)	15 (RWHold * C)	10 (RWHold * C)	10 (RWHold * C)
t5	$\overline{\text{ATAPI\_DIOR}}$ data setup (min)	50	35	20	20	20
t6	$\overline{\text{ATAPI\_DIOR}}$ data hold (min)	5	5	5	5	5
t6z	$\overline{\text{ATAPI\_DIOR}}$ data 3-state (max)	30	30	30	30	30
t9	$\overline{\text{ATAPI\_DIOR}}$/ $\overline{\text{ATAPI\_DIOW}}$ to address valid hold (min)	20 (RWHold * C)	15 (RWHold * C)	10 (RWHold * C)	10 (RWHold * C)	10 (RWHold * C)
tRd	Read data valid to ATAPI_IORDY active (if ATAPI_IORDY initially low after tA) (min)	0	0	0	0	0
tA	ATAPI_IORDY setup time (min)	35	35	35	35	35
tB	ATAPI_IORDY pulse width (max)	1250	1250	1250	1250	1250

18.2.13 SD Interface Timing

The SD interface is used to read or write the data from or to the DVD drives. The SD synchronous read and write timing diagram is shown in Figure 18.32.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.32 SD Interface Timing

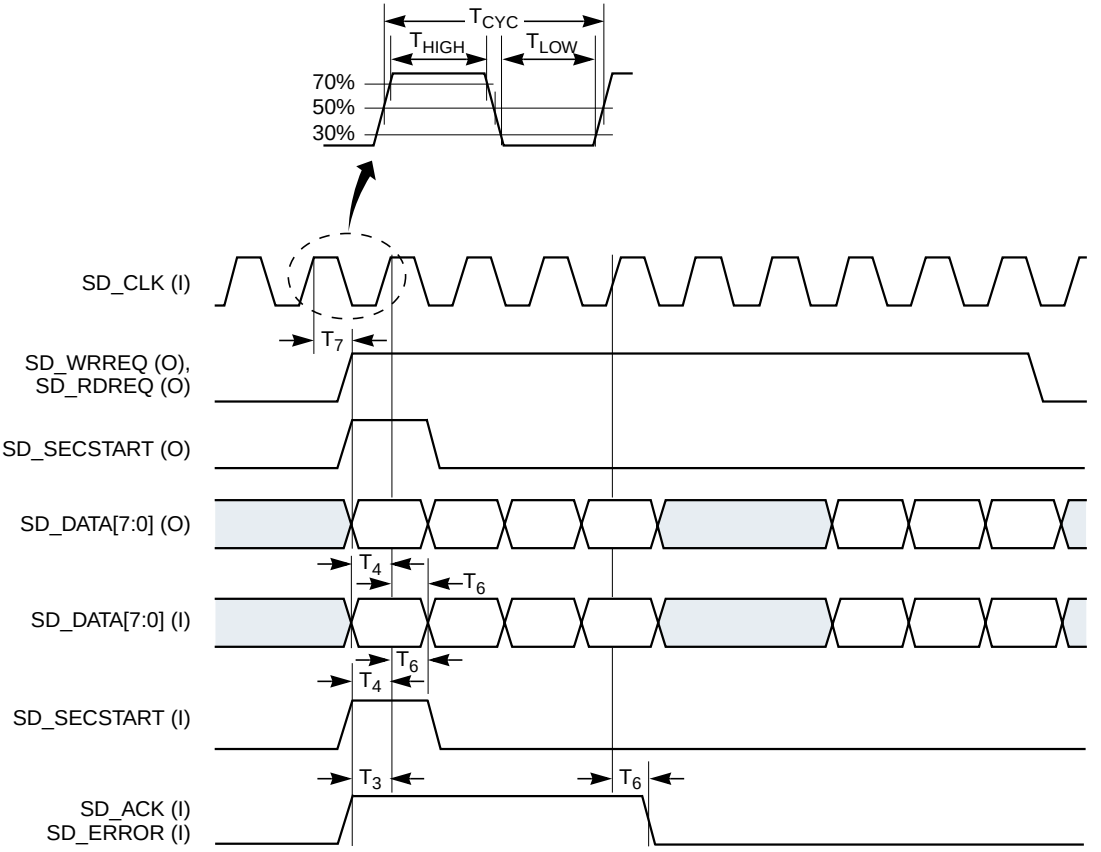


Table 18.28 SD Input Timing

Symbol	Description	Light Loading (50 MHz)[1]		Heavy Loading (27/16 MHz)	
		Min	Max	Min	Max
	Input Transition Time		2.0 ns		5.0 ns
	Ouput Load		10 pF		60 pF
T_{CYC}	SD_CLK period	20 ns		37 ns	
T_{HIGH}	SD_CLK high	9 ns		16 ns	

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.28 SD Input Timing (Cont.)

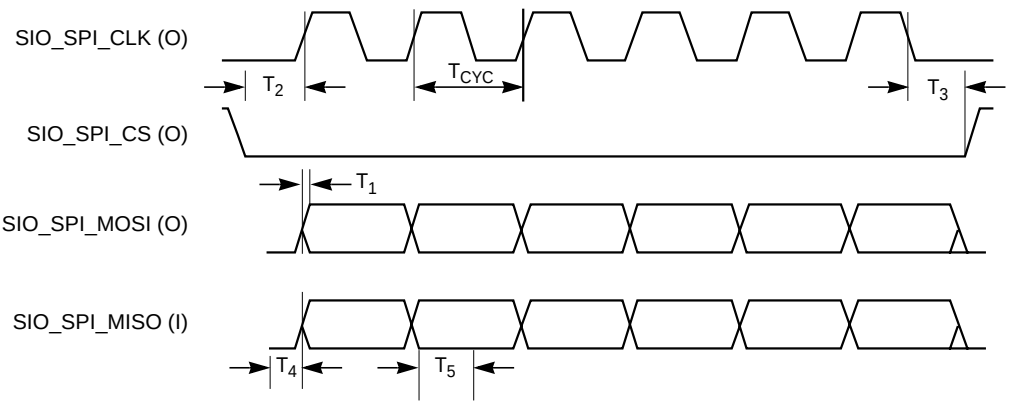
Symbol	Description	Light Loading (50 MHz)[1]		Heavy Loading (27/16 MHz)	
		Min	Max	Min	Max
T_{LOW}	SD_CLK low	9 ns		16 ns	
T_3	SD_ERROR, SD_SECSTART, SD_ACK setup	6 ns		6 ns	
T_4	SD_DATA setup	4 ns		3 ns	
T_6	SD_ERROR, SD_SECSTART, SD_DATA, SD_ACK hold	1.25 ns		1.25 ns	
T_7	SD_WRREQ, SD_RDREQ, SD_DATA and SD_SECSTART output delay	3 ns	11 ns	4 ns	18 ns

1. Note: Light loading (50 MHz) is for point-to-point connections in a multi-DMN-8600 configuration.

18.2.14 SPI Interface Timing

Figure 18.33 shows typical AC timing parameters for DMN-8600 SPI transfers. The parameter values are listed in Table 18.29.

Figure 18.33 32-Bit SPI Data Transfer Format



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.29 SPI Interface Timing Parameters

Param	Description	Min	Max
T_{CYC}	SIO_SPI_CLK cycle time (frequency)	0.2 ms (37.5 MHz)	26 ns(4.6 kHz)
T_1	Delay time, SIO_SPI_CLK edge to SIO_SPI_MOSI active	2 cycles + 10 ns	2 cycles + 30 ns
T_2	Assertion, SIO_SPI_CS to SPI_CLK edge[1]	0 ns	91 ns
T_3	Deassertion, SPI_CLK edge to SIO_SPI_CS[1]	0 ns	91 ns
T_4	Setup time, SIO_SPI_MISO to SPI_CLK edge	3 cycles	
T_5	Hold time, SIO_SPI_MISO to SPI_CLK edge	3 cycles	

1. These values are programmed in the SPI Configuration register.

18.2.15 1394 Timing

1394 signal level parameters are shown in Figure 18.34 and described in Table 18.30.

Figure 18.34 SBP Signal Level Parameters

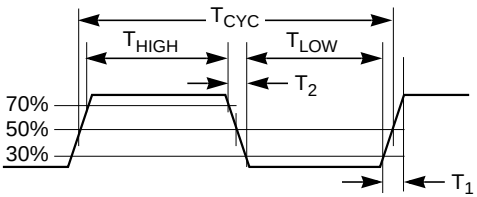


Table 18.30 1394 AC Timing Parameters

Param	Description	Min	Max	Unit
T_{CYC}	BIO_PHY_CLK frequency	49.152 ± 100 ppm	49.152 ± 100 ppm	MHz
T_{HIGH}	BIO_PHY_CLK High time			
T_{LOW}	BIO_PHY_CLK Low time			
	BIO_PHY_CLK duty cycle	45	55	%
T_1	BIO_PHY_CLK Rise time	0.7	2.4	ns
T_2	BIO_PHY_CLK Fall time	0.7	2.4	ns

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.35 1394 PHY to Link Transfer Waveform at the PHY

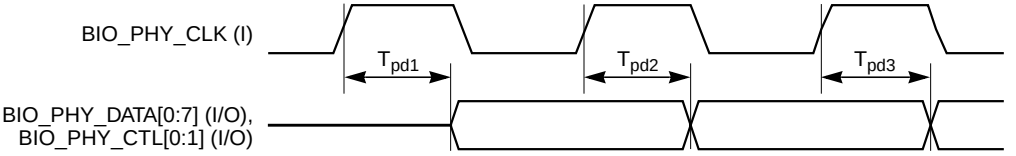


Figure 18.36 1394 Link to PHY Transfer Waveform at the PHY

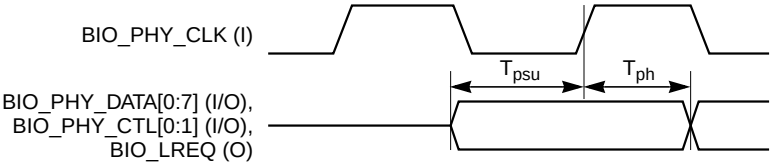


Table 18.31 1394 AC Timing Parameters at the PHY

Symbol	Description	Min	Max	Unit
t_{pd1}	Delay time: BIO_PHY_CLK input high to initial instance of BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] outputs valid	0.5	13.5	ns
t_{pd2}	Delay time: BIO_PHY_CLK input high to subsequent instance(s) of BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] outputs valid	0.5	13.5	ns
t_{pd3}	Delay time: BIO_PHY_CLK input high to BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] invalid (high-impedance)	0.5	13.5	ns
t_{psu}	Setup time: BIO_PHY_DATA[0:7], BIO_PHY_CTL[0:1], and BIO_LREQ inputs before BIO_PHY_CLK	6.0		ns
t_{ph}	Hold time: BIO_PHY_DATA[0:7], BIO_PHY_CTL[0:1], and BIO_LREQ inputs after BIO_PHY_CLK	0		ns

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.37 1394 Link to PHY Transfer Waveform at the Link

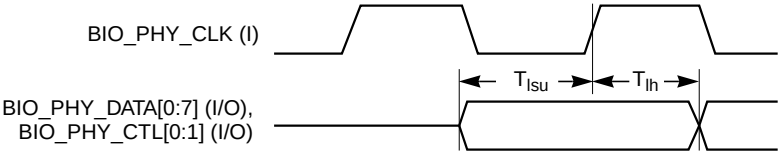


Figure 18.38 1394 PHY to Link Transfer Waveform at the Link

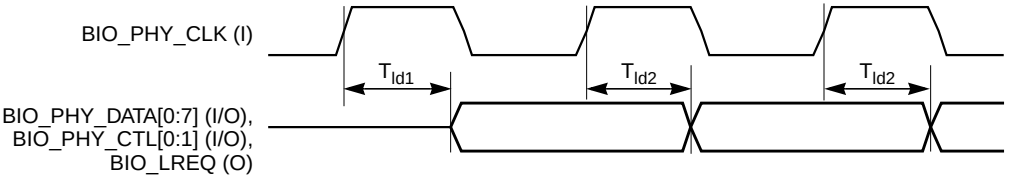


Table 18.32 1394 AC Timing Parameters at the Link

Symbol	Description	Timing Value (ns)	
		Min	Max
t_{ld1}	Delay time, BIO_PHY_CLK input high to initial instance of BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] outputs valid	1	10
t_{ld2}	Delay time, BIO_PHY_CLK input high to subsequent instance(s) of BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] outputs valid	1	10
t_{ld3}	Delay time, BIO_PHY_CLK input high to BIO_PHY_DATA[0:7] and BIO_PHY_CTL[0:1] invalid (high-impedance)	1	10
t_{lsu}	Setup time BIO_PHY_DATA[0:7] , BIO_PHY_CTL[0:1] , and BIO_LREQ inputs before BIO_PHY_CLK	6	
t_{lh}	Hold time BIO_PHY_DATA[0:7] , BIO_PHY_CTL[0:1] , and BIO_LREQ inputs after BIO_PHY_CLK	0	

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.2.16 SBP Interface Timing

SBP timing diagrams are shown in figures Figure 18.39 through Figure 18.43, and are described in Table 18.33

Figure 18.39 SBP Clock Timing

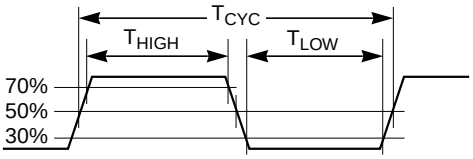
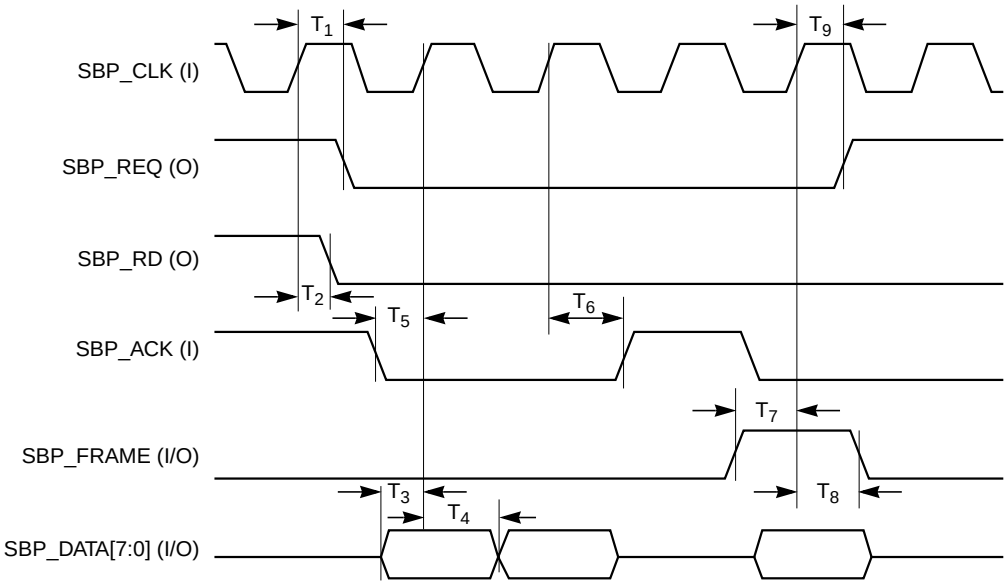


Figure 18.40 SBP Incoming Transfer (POL = 1, WRREQ = 0)



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.41 SBP Outgoing Transfer (POL = 1, WRREQ = 0)

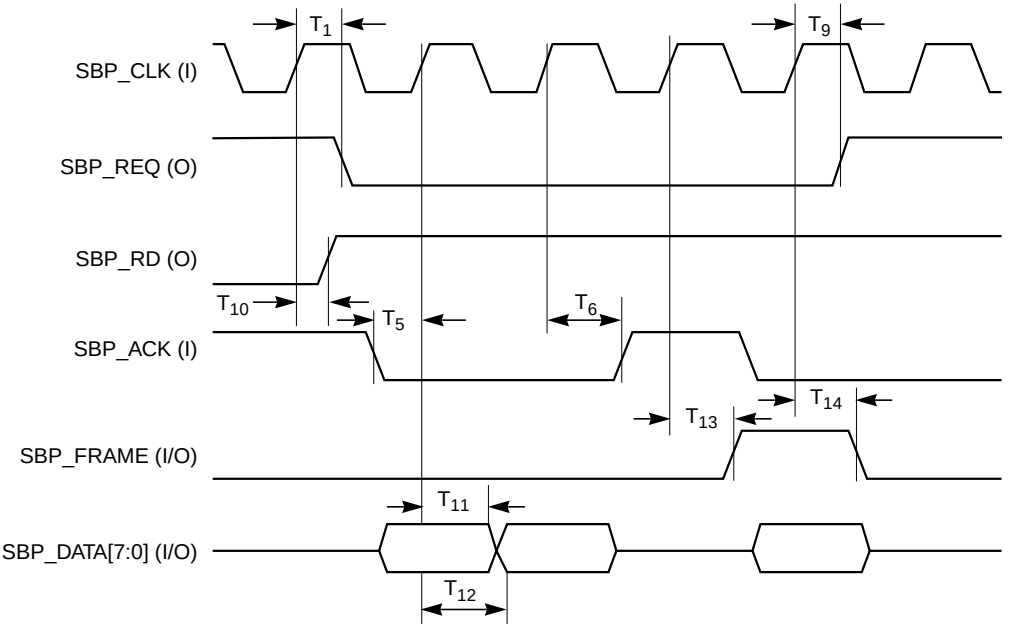
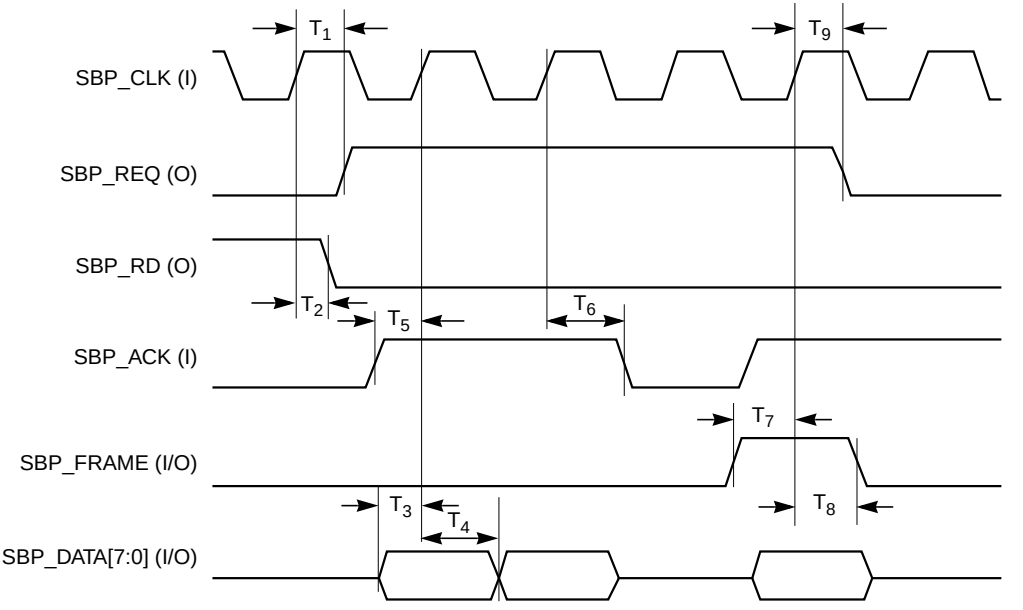


Figure 18.42 SBP Incoming Transfer (POL = 0,WRREQ = 0)



Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.43 SBP Outgoing Transfer (POL = 0, WRREQ = 0)

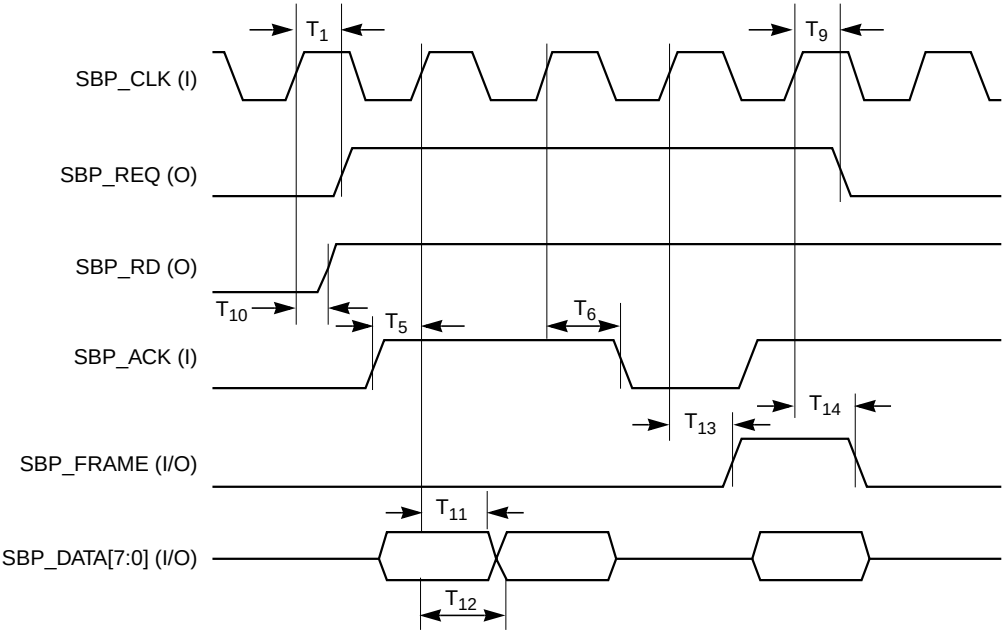


Table 18.33 SBP Timing Parameters

Symbol	Description	Light Loading ns (50 MHz)		Heavy Loading ns (27/16 MHz)	
		Min	Max	Min	Max
T_{CLK}	Clock period	20		37	ns
T_{HIGH}	Clock high time	9		16	ns
T_{LOW}	Clock low time	9		16	ns
T1	SBP_REQ output delay time		12.0		18.0
T2	SBP_RD output delay time		12.0		18.0
T3	SBP_DATA input setup time	3.0		3.0	
T4	SBP_DATA input hold time	1.25		1.25	
T5	SBP_ACK input setup time	5.0		5.0	
T6	SBP_ACK input hold time	1.25		1.25	

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.33 SBP Timing Parameters

Symbol	Description	Light Loading ns (50 MHz)		Heavy Loading ns (27/16 MHz)	
		Min	Max	Min	Max
T7	SBP_FRAME input setup time	5.0		5.0	
T8	SBP_FRAME input hold time	1.25		1.25	
T9	SBP_REQ output hold time	3.0		4.0	
T10	SBP_RD output hold time	3.0		4.0	
T11	SBP_DATA output hold time	3.0		4.0	
T12	SBP_DATA output delay time		12.0		18.0
T13	SBP_FRAME output delay time		12.0		18.0
T14	SBP_FRAME output hold time	3.0		4.0	

Note: Light loading (50 MHz) is for point-to-point connections only (in a multi-DMN-8600 configuration).

18.3 Pin Description

The DMN-8600 is packaged in a 308-pin Ball Grid Array (BGA) package and is drop-in pin compatible, presenting OEMs with unprecedented flexibility and cost reduction in their product lines.

Table 18.34 lists the pin numbers in sequence, including, pin name, I/O voltage and I/O type and Figure 18.44 shows the pinout. Note that some pin numbers are multiplexed and therefore have multiple pin names (signal names).

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List

Number	Pin Name	Voltage[1]	I/O Type
A1	VI_D[9]	3.3/5	I
A2	VI_D[8]	3.3/5	I
A3	NC	NO CONNECT	I
A4	NC	NO CONNECT	I
A5	VI_CLK[0]	3.3/5	I
A6	NC	NO CONNECT	I
A7	PLL_BYPASS	3.3	I
A8	VSS_A	GROUND	–
A9	VSS_A	GROUND	–
A10	CLKI	3.3	I
A11	XVDD	3.3	O
A12	CLKO	3.3	O
A13	AI_MCLKO	3.3	O
A14	AO_SCLK	3.3	O
A15	AO_D[3]	3.3	O
A16	$\overline{\text{SDRAM\_WE}}$	2.5/3.3	O
A17	SDRAM_A[2]	2.5/3.3	O
A18	SDRAM_A[15]	2.5/3.3	O
A19	SDRAM_A[0]	2.5/3.3	O
A20	SDRAM_A[5]	2.5/3.3	O
B1	NC	NO CONNECT	I
B2	VI_D[2]	3.3/5	I
B3	VI_D[7]	3.3/5	I
B4	NC	NO CONNECT	I
B5	NC	NO CONNECT	I

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
B6	TMS	3.3	I
B7	TDO	3.3	O
B8	VDD_A	3.3 ANALOG	–
B9	VDD_A	3.3 ANALOG	–
B10	CLKX	3.3	O
B11	VSS_RREF	GROUND	–
B12	AI_SCLK	3.3	I/O
B13	AO_MCLKO	3.3	O
B14	AO_IEC958	3.3	O
B15	AO_D[2]	3.3	O
B16	$\overline{\text{SDRAM\_RAS}}$	2.5/3.3	O
B17	$\overline{\text{SDRAM\_CAS}}$	2.5/3.3	O
B18	SDRAM_A[4]	2.5/3.3	O
B19	SDRAM_A[1]	2.5/3.3	O
B20	SDRAM_A[7]	2.5/3.3	O
C1	VI_VSYNC[0]	3.3/5	I
C2	VI_D[4]	3.3/5	I
C3	VI_D[3]	3.3/5	I
C4	VI_D[6]	3.3/5	I
C5	NC	NO CONNECT	I
C6	NC	NO CONNECT	I
C7	TDI	3.3	I
C8	VSS_A	GROUND	–
C9	VSS_A	GROUND	–
C10	VSS_X	GROUND	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
C11	RREF	resistor connect	I
C12	AI_FSYNC	3.3	I/O
C13	AI_D[0]	3.3	I
C14	AO_D[1]	3.3	O
C15	SDRAM_CKE	2.5/3.3	O
C16	VDD_2.5	2.5/3.3	–
C17	SDRAM_A[3]	2.5/3.3	O
C18	SDRAM_A[6]	2.5/3.3	O
C19	SDRAM_A[10]	2.5/3.3	O
C20	SDRAM_A[14]	2.5/3.3	O
D1	VO_D[13]	3.3	O
D2	NC	NO CONNECT	I
D3	VI_D[5]	3.3/5	I
D4	VDD_1.8	1.8	–
D5	VDD_3.3	3.3	–
D6	NC	NO CONNECT	I
D7	$\overline{\text{TRST}}$	3.3	I
D8	TCK	3.3	I
D9	VDD_A	3.3	–
D10	VDD_A	3.3	–
D11	VDD_RREF	3.3	–
D12	AI_D[1]	3.3	I
D13	AO_FSYNC	3.3	O
D14	AO_D[0]	3.3	O
D15	VDD_2.5	2.5/3.3	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
D16	SDRAM_VREF (For DDR only) [2]	VDD_2.5 2	I
D17	VDD_2.5	2.5/3.3	–
D18	SDRAM_A[8]	2.5/3.3	O
D19	SDRAM_A[9]	2.5/3.3	O
D20	SDRAM_DQ[24]	2.5/3.3	I/O
E1	VO_D[10]	3.3	O
E2	VO_D[12]	3.3	O
E3	VO_D[14]	3.3	O
E4	VDD_3.3	3.3	–
E9	VDD_3.3	3.3	–
E10	VDD_1.8	1.8	–
E11	VDD_3.3	3.3	–
E12	VDD_1.8	1.8	–
E17	SDRAM_A[12]	2.5/3.3	O
E18	SDRAM_A[13]	2.5/3.3	O
E19	SDRAM_DQ[25]	2.5/3.3	I/O
E20	SDRAM_DQS[3] (For DDR parts only)	2.5/3.3	I/O
F1	VO_D[7]	3.3	O
F2	VO_D[8]	3.3	O
F3	VO_D[11]	3.3	O
F4	VO_D[15]	3.3	O
F17	SDRAM_A[11]	2.5/3.3	O
F18	SDRAM_DQ[26]	2.5/3.3	I/O
F19	SDRAM_DQ[30]	2.5/3.3	I/O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
F20	SDRAM_DQ[31]	2.5/3.3	I/O
G1	VO_D[4]	3.3	O
G2	VO_D[5]	3.3	O
G3	VO_D[6]	3.3	O
G4	VO_D[9]	3.3	O
G17	SDRAM_DQ[27]	2.5/3.3	I/O
G18	SDRAM_DQ[28]	2.5/3.3	I/O
G19	SDRAM_DQM[3]	2.5/3.3	O
G20	SDRAM_CLK[1]	2.5/3.3	O
H1	VO_CLK	3.3	I/O
H2	VO_D[1]	3.3	O
H3	VO_D[2]	3.3	O
H4	VO_D[3]	3.3	O
H8	VSS	GND	–
H9	VSS	GND	–
H10	VSS	GND	–
H11	VSS	GND	–
H12	VSS	GND	–
H13	VSS	GND	–
H17	SDRAM_DQ[29]	2.5/3.3	I/O
H18	SDRAM_DQ[22]	2.5/3.3	I/O
H19	SDRAM_DQ[23]	2.5/3.3	I/O
H20	$\overline{\text{SDRAM\_CLK}}$[1] (For DDR parts only)	2.5/3.3	O
J1	BIO_PHY_DATA[4]	3.3	I/O
J2	BIO_PHY_DATA[0]	3.3	I/O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
J3	BIO_PHY_CTL[0]	3.3	I/O
J4	VO_D[0]	3.3	O
J5	VDD_1.8	1.8	–
J8	VSS	GND	–
J9	VSS	GND	–
J10	VSS	GND	–
J11	VSS	GND	–
J12	VSS	GND	–
J13	VSS	GND	–
J16	VDD_2.5	2.5/3.3	–
J17	SDRAM_DQ[21]	2.5/3.3	I/O
J18	SDRAM_DQ[20]	2.5/3.3	I/O
J19	SDRAM_DQS[2] (For DDR parts only)	2.5/3.3	I/O
J20	SDRAM_DQ[19]	2.5/3.3	I/O
K1	BIO_PHY_DATA[5]	3.3	I/O
K2	BIO_PHY_DATA[6]	3.3	I/O
K3	BIO_PHY_DATA[7]	3.3	I/O
K4	BIO_LINK_ON	3.3	I
K5	VDD_3.3	3.3	–
K8	VSS	GROUND	–
K9	VSS	GROUND	–
K10	VSS	GROUND	–
K11	VSS	GROUND	–
K12	VSS	GROUND	–
K13	VSS	GROUND	–

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
K16	VSS_2.5	2.5/3.3	–
K17	SDRAM_DQ[18]	2.5/3.3	I/O
K18	SDRAM_DQ[17]	2.5/3.3	I/O
K19	SDRAM_DQ[16]	2.5/3.3	I/O
K20	SDRAM_DQM[2]	2.5/3.3	O
L1	BIO_PHY_CLK	3.3	I
L2	BIO_PHY_CTL[1]	3.3	I/O
L3	BIO_PHY_DATA[1]	3.3	I/O
L4	BIO_LPS	3.3	O
L5	VDD_3.3	3.3	–
L8	VSS	GROUND	–
L9	VSS	GROUND	–
L10	VSS	GROUND	–
L11	VSS	GROUND	–
L12	VSS	GROUND	–
L13	VSS	GROUND	–
L16	VDD_2.5	2.5/3.3	–
L17	SDRAM_DQM[1]	2.5/3.3	O
L18	SDRAM_DQ[8]	2.5/3.3	I/O
L19	SDRAM_DQ[11]	2.5/3.3	I/O
L20	SDRAM_DQ[13]	2.5/3.3	I/O
M1	BIO_LREQ	3.3	O
M2	BIO_PHY_DATA[3]	3.3	I/O
M3	BIO_PHY_DATA[2]	3.3	I/O
M4	ATAPI_DATA[8] SBP_DATA[0]	3.3/5	I/O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
M5	VDD_1.8	1.8	–
M8	VSS	GROUND	–
M9	VSS	GROUND	–
M10	VSS	GROUND	–
M11	VSS	GROUND	–
M12	VSS	GROUND	–
M13	VSS	GROUND	–
M16	MCONFIG[0]	3.3/5	I
M17	SDRAM_DQ[12]	2.5/3.3	I/O
M18	SDRAM_DQ[14]	2.5/3.3	I/O
M19	SDRAM_DQS[1] (For DDR parts only)	2.5/3.3	I/O
M20	SDRAM_DQ[10]	2.5/3.3	I/O
N1	ATAPI_DATA[7] SD_DATA[7]	3.3/5	I/O
N2	ATAPI_DATA[6] SD_DATA[6]	3.3/5	I/O
N3	ATAPI_DATA[9] SBP_DATA[1]	3.3/5	I/O
N4	ATAPI_DATA[5] SD_DATA[5]	3.3/5	I/O
N8	VSS	GROUND	–
N9	VSS	GROUND	–
N10	VSS	GROUND	–
N11	VSS	GROUND	–
N12	VSS	GROUND	–
N13	VSS	GROUND	–
N17	SDRAM_DQM[0]	2.5/3.3	O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
N18	SDRAM_DQ[15]	2.5/3.3	I/O
N19	SDRAM_DQ[9]	2.5/3.3	I/O
N20	$\overline{\text{SDRAM\_CLK}}$[0] (For DDR parts only)	2.5/3.3	O
P1	ATAPI_DATA[4] SD_DATA[4]	3.3/5	I/O
P2	ATAPI_DATA[10] SBP_DATA[2]	3.3/5	I/O
P3	ATAPI_DATA[3] SD_DATA[3] CD_C2PO	3.3/5	I/O
P4	ATAPI_DATA[2] SD_DATA[2] CD_BCK	3.3/5	I/O
P17	SDRAM_DQ[4]	2.5/3.3	I/O
P18	SDRAM_DQ[6]	2.5/3.3	I/O
P19	SDRAM_DQ[7]	2.5/3.3	I/O
P20	SDRAM_CLK[0]	2.5/3.3	O
R1	ATAPI_DATA[11] SBP_DATA[3]	3.3/5	I/O
R2	ATAPI_DATA[12] SBP_DATA[4]	3.3/5	I/O
R3	ATAPI_DATA[1] SD_DATA[1] CD_LRCK	3.3/5	I/O
R4	ATAPI_DATA[0] SD_DATA[0] CD_DATA	3.3/5	I/O
R17	SDRAM_DQ[0]	2.5/3.3	I/O
R18	SDRAM_DQ[2]	2.5/3.3	I/O
R19	SDRAM_DQS[0] (For DDR parts only)	2.5/3.3	I/O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
R20	SDRAM_DQ[5]	2.5/3.3	I/O
T1	$\overline{\text{ATAPI\_DIOW}}$ SD_$\overline{\text{WRREQ}}$	3.3	O
T2	ATAPI_DATA[13] SBP_DATA[5]	3.3/5	I/O
T3	ATAPI_IORDY	3.3	I
T4	VDD_3.3	3.3	–
T9	VDD_1.8	1.8	–
T10	VDD_1.8	1.8	–
T11	VDD_3.3	3.3	–
T12	VDD_3.3	3.3	–
T17	VSS_DLL	GROUND	–
T18	SIO_SPI_MISO M_A[1]	3.3/5 3.3	I O
T19	SDRAM_DQ[1]	2.5/3.3	I/O
T20	SDRAM_DQ[3]	2.5/3.3	I/O
U1	$\overline{\text{ATAPI\_DMAACK}}$ SD_$\overline{\text{SECSTART}}$	3.3 3.3/5	O I/O
U2	ATAPI_DATA[14] SBP_DATA[6]	3.3/5	I/O
U3	ATAPI_ADDR[3] SBP_$\overline{\text{REQ}}$	3.3	O
U4	VDD_1.8	1.8	–
U5	VDD_5	5	–
U6	H_DATA[23] M_A[24]	3.3/5 3.3	I/O O
U7	$\overline{\text{H\_INT}}$ M_GPIO[0]	3.3 3.3/5	op dr O I/O
U8	H_RD/$\overline{\text{WR}}$ M_RD/$\overline{\text{WR}}$	3.3/5 3.3	I O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
U9	H_DTACK M_DTACK	3.3 3.3	3-st O I
U10	H_DATA[14] M_A[20]/M_D[14]	3.3/5	I/O
U11	H_DATA[8] M_A[14]/M_D[8]	3.3/5	I/O
U12	H_DATA[5] M_A[11]/M_D[5]	3.3/5	I/O
U13	M_ALE	3.3	O
U14	SIO_IRTX1 M_A[2]	3.3	O
U15	VDD_DLL	1.8	–
U16	VDD_DLL	1.8	–
U17	VSS_DLL	GROUND	–
U18	SIO_SPI_MOSI M_A[25]	3.3	O
U19	SIO_UART2_TX	3.3	O
U20	SIO_UART1_TX M_A[4]	3.3	O
V1	ATAPI_ADDR[2] SBP_RD	3.3	O
V2	ATAPI_ADDR[1] SBP_ACK	3.3 3.3/5	O I
V3	ATAPI_DIOR SD_RDREQ	3.3	O
V4	ATAPI_ADDR[4] SBP_CLK	3.3 3.3/5	O I
V5	H_DATA[29] M_CS[3]	3.3/5 3.3	I/O O
V6	H_DATA[26] M_CS[0]	3.3/5 3.3	I/O O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
V7	H_DATA[21] M_A[22]	3.3/5 3.3	I/O O
V8	H_DATA[17] M_A[2]	3.3/5 3.3	I/O O
V9	$\overline{\text{H\_WAIT}}$ $\overline{\text{M\_WAIT}}$	3.3 3.3/5	3-st O I
V10	H_DATA[12] M_A[18]/M_D[12]	3.3/5	I/O
V11	H_DATA[13] M_A[19]/M_D[13]	3.3/5	I/O
V12	H_DATA[10] M_A[16]/M_D[10]	3.3/5	I/O
V13	H_DATA[7] M_A[13]/M_D[7]	3.3/5	I/O
V14	H_DATA[3] M_A[9]/M_D[3]	3.3/5	I/O
V15	$\overline{\text{H\_DMAREQ}}$ $\overline{\text{M\_UWE}}/\overline{\text{UDS}}$	3.3	O
V16	SIO_SPI_CS[2] M_A[22]	3.3	O
V17	SIO_UART1_CTS M_$\overline{\text{RD}}/\overline{\text{WR}}$	3.3	I O
V18	SIO_SPI_CS[1] M_A[23]	3.3	O
V19	SIO_SCL	3.3/5	I/O
V20	SIO_UART1_RTS M_A[3]	3.3	O
W1	ATAPI_DMARQ SD_ERROR	3.3/5	I
W2	ATAPI_INTRQ SD_ACK	3.3/5	I
W3	ATAPI_ADDR[0] SBP_FRAME	3.3 3.3/5	O I/O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
W4	H_DATA[31] M_CS[5]	3.3/5	I/O
W5	H_DATA[28] M_CS[2]	3.3/5 3.3	I/O O
W6	H_DATA[22] M_A[23]	3.3/5 3.3	I/O O
W7	H_DATA[20] M_A[5]	3.3/5 3.3	I/O O
W8	H_DATA[16] M_A[1]	3.3/5 3.3	I/O O
W9	H_ADDR[2] M_GPIO[3]	3.3/5 3.3/5	I I/O
W10	H_ADDR[1] M_GPIO[2]	3.3/5 3.3/5	I I/O
W11	H_DATA[9] M_A[15]/M_D[9]	3.3/5	I/O
W12	H_DATA[6] M_A[12]/M_D[6]	3.3/5	I/O
W13	H_DATA[2] M_A[8]/M_D[2]	3.3/5	I/O
W14	H_DATA[1] M_A[7]/M_D[1]	3.3/5	I/O
W15	H_DATA[0] M_A[6]/M_D[0]	3.3/5	I/O
W16	$\overline{\text{H\_RST}}$ $\overline{\text{M\_RST}}$	3.3/5	I
W17	SIO_IRTX2 $\overline{\text{M\_OE}}$	3.3	O
W18	SIO_SPI_CS[3] M_A[5]	3.3	O
W19	SIO_UART1_RX $\overline{\text{M\_CS}}$[0]	3.3	I O
W20	SIO_UART2_RX	3.3/5	I

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
Y1	ATAPI_RESET	3.3	O
Y2	ATAPI_DATA[15] SBP_DATA[7]	3.3/5	I/O
Y3	H_DATA[30] M_CS[4]	3.3/5	I/O
Y4	H_DATA[27] M_CS[1]	3.3/5 3.3	I/O O
Y5	H_DATA[25] M_A[26]	3.3/5 3.3	I/O O
Y6	H_DATA[24] M_A[25]	3.3/5 3.3	I/O O
Y7	H_DATA[18] M_A[3]	3.3/5 3.3	I/O O
Y8	H_DATA[19] M_A[4]	3.3/5 3.3	I/O O
Y9	H_CS M_GPIO[5]	3.3/5 3.3/5	I I/O
Y10	H_RD M_GPIO[4]	3.3/5 3.3/5	I I/O
Y11	H_DATA[15] M_A[21]/M_D[15]	3.3/5	I/O
Y12	H_DATA[11] M_A[17/M_D[11]	3.3/5	I/O
Y13	H_ADDR[0] M_GPIO[1]	3.3/5 3.3/5	I I/O
Y14	H_DATA[4] M_A[10]/M_D[4]	3.3/5	I/O
Y15	M_OE	3.3	O
Y16	MCONFIG[1]	3.3/5	I
Y17	SIO_SPI_CLK M_A[26]	3.3 3.3	O O

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.34 DMN-8600 Pin List (Cont.)

Number	Pin Name	Voltage[1]	I/O Type
Y18	SIO_SDA	3.3/5	I/O
Y19	SIO_SPI_CS[0] M_A[24]	3.3	O
Y20	SIO_IRRX	3.3/5	I

1. For input pins, “voltage” means tolerance; for output pins, “voltage” means output drive level.
2. DDR refers to Double Data Rate SDRAMs, which have double the bandwidth of standard SDRAMs and are used for high-performance, non-power critical applications.

Note: The DMN-8600 core operates at 1.8 V 5%. Some I/O interface pins can be interfaced with 3.3 V or 5 V devices depending on the voltage applied to the VDD pins associated with them.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.44 308-Pad BGA Pinout (Sheet 1 of 2)

A1 VI_D[9]	B1 NC	C1 VI_VSYNC[0]	D1 VO_D[13]	E1 VO_D[10]	F1 VO_D[7]	G1 VO_D[4]	H1 VO_CLK	J1 BIO_PHY_DATA[4]	K1 BIO_PHY_DATA[5]
A2 VI_D[8]	B2 VI_D[2]	C2 VI_D[4]	D2 NC	E2 VO_D[12]	F2 VO_D[8]	G2 VO_D[5]	H2 VO_D[1]	J2 BIO_PHY_DATA[0]	K2 BIO_PHY_DATA[6]
A3 NC	B3 VI_D[7]	C3 VI_D[3]	D3 VI_D[5]	E3 VO_D[14]	F3 VO_D[11]	G3 VO_D[6]	H3 VO_D[2]	J3 BIO_PHY_CTL[0]	K3 BIO_PHY_DATA[7]
A4 NC	B4 NC	C4 VI_D[6]	D4 VDD_1.8	E4 VDD_3.3	F4 VO_D[15]	G4 VO_D[9]	H4 VO_D[3]	J4 VO_D[0]	K4 BIO_LINK_ON
A5 VI_CLK[0]	B5 NC	C5 NC	D5 VDD_3.3					J5 VDD_1.8	K5 VDD_3.3
A6 NC	B6 TMS	C6 NC	D6 NC						
A7 PLL_BYPASS	B7 TDO	C7 TDI	D7 TRST						
A8 VSS_A	B8 VDD_A	C8 VSS_A	D8 TCK				H8 VSS	J8 VSS	K8 VSS
A9 VSS_A	B9 VDD_A	C9 VSS_A	D9 VDD_A	E9 VDD_3.3			H9 VSS	J9 VSS	K9 VSS
A10 CLKI	B10 CLKX	C10 VSS_X	D10 VDD_A	E10 VDD_1.8			H10 VSS	J10 VSS	K10 VSS
A11 XVDD	B11 VSS_RREF	C11 RREF	D11 VDD_RREF	E11 VDD_3.3			H11 VSS	J11 VSS	K11 VSS
A12 CLKO	B12 AI_SCLK	C12 AI_FSYNC	D12 AI_D[1]	E12 VDD_3.3			H12 VSS	J12 VSS	K12 VSS
A13 AI_MCLKO	B13 AO_MCLKO	C13 AI_D[0]	D13 AO_FSYNC				H13 VSS	J13 VSS	K13 VSS
A14 AO_SCLK	B14 AO_IEC958	C14 AO_D[1]	D14 AO_D[0]						
A15 AO_D[3]	B15 AO_D[2]	C15 SDRAM_CKE	D15 VDD_2.5						
A16 $\overline{\text{SDRAM\_WE}}$	B16 $\overline{\text{SDRAM\_RAS}}$	C16 VDD_2.5	D16 SDRAM_VREF[1]					J16 VDD_2.5	K16 VSS_2.5
A17 SDRAM_A[2]	B17 $\overline{\text{SDRAM\_CAS}}$	C17 SDRAM_A[3]	D17 VDD_2.5	E17 SDRAM_A[12]	F17 SDRAM_A[11]	G17 SDRAM_DQ[27]	H17 SDRAM_DQ[29]	J17 SDRAM_DQ[21]	K17 SDRAM_DQ[18]
A18 SDRAM_A[15]	B18 SDRAM_A[4]	C18 SDRAM_A[6]	D18 SDRAM_A[8]	E18 SDRAM_A[13]	F18 SDRAM_DQ[26]	G18 SDRAM_DQ[28]	H18 SDRAM_DQ[22]	J18 SDRAM_DQ[20]	K18 SDRAM_DQ[17]
A19 SDRAM_A[0]	B19 SDRAM_A[1]	C19 SDRAM_A[10]	D19 SDRAM_A[9]	E19 SDRAM_DQ[25]	F19 SDRAM_DQ[30]	G19 SDRAM_DQM[3]	H19 SDRAM_DQ[23]	J19 SDRAM_DQS[2][1]	K19 SDRAM_DQ[16]
A20 SDRAM_A[5]	B20 SDRAM_A[7]	C20 SDRAM_A[14]	D20 SDRAM_DQ[24]	E20 SDRAM_DQS[3][1]	F20 SDRAM_DQ[31]	G20 SDRAM_CLK[1]	H20 $\overline{\text{SDRAM\_CLK}}$[1][1]	J20 SDRAM_DQ[19]	K20 SDRAM_DQM[2]

1. For DDR parts only. DDR refers to Double Data Rate SDRAMs, which have double the bandwidth of standard SDRAMs and are used for high-performance, nonpower critical applications.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.44 308-Pad BGA Pinout (Sheet 2 of 2)

L1 BIO_PHY_CLK	M1 BIO_LREQ	N1 ATAPI_DATA[7] SD_DATA[7]	P1 ATAPI_DATA[4] SD_DATA[4]	R1 ATAPI_DATA[11] SBP_DATA[3]	T1 ATAPI_DIOW SD_WRREQ	U1 ATAPI_DMAACK SD_SECSTART	V1 ATAPI_ADDR[2] SBP_RD	W1 ATAPI_DMARQ SD_ERROR	Y1 ATAPI_RESET
L2 BIO_PHY_CTL[1]	M2 BIO_PHY_DATA[3]	N2 ATAPI_DATA[6] SD_DATA[6]	P2 ATAPI_DATA[10] SBP_DATA[2]	R2 ATAPI_DATA[12] SBP_DATA[4]	T2 ATAPI_DATA[13] SBP_DATA[5]	U2 ATAPI_DATA[14] SBP_DATA[6]	V2 ATAPI_ADDR[1] SBP_ACK	W2 ATAPI_INTRQ SD_ACK	Y2 ATAPI_DATA[15] SBP_DATA[7]
L3 BIO_PHY_DATA[1]	M3 BIO_PHY_DATA[2]	N3 ATAPI_DATA[9] SBP_DATA[1	P3 ATAPI_DATA[3] SD_DATA[3] CD_C2PO	R3 ATAPI_DATA[1] SD_DATA[1] CD_LRCK	T3 ATAPI_IORDY	U3 ATAPI_ADDR[3] SBP_REQ	V3 ATAPI_DIOR SD_RDREQ	W3 ATAPI_ADDR[0] SBP_FRAME	Y3 H_DATA[30] M_CS[4]
L4 BIO_LPS	M4 ATAPI_DATA[8] SBP_DATA[0]	N4 ATAPI_DATA[5] SD_DATA[5]	P4 ATAPI_DATA[2] SD_DATA[2] CD_BCK	R4 ATAPI_DATA[0] SD_DATA[0] CD_DATA	T4 VDD_3.3	U4 VDD_1.8	V4 ATAPI_ADDR[4] SBP_CLK	W4 H_DATA[31] M_CS[5]	Y4 H_DATA[27] M_CS[1]
L5 VDD_3.3	M5 VDD_1.8					U5 VDD_5	V5 H_DATA[29] M_CS[3]	W5 H_DATA[28] M_CS[2]	Y5 H_DATA[25] M_A[26]
						U6 H_DATA[23] M_A[24]	V6 H_DATA[26] M_CS[0]	W6 H_DATA[22] M_A[23]	Y6 H_DATA[24] M_A[25]
						U7 H_INT M_GPIO[0]	V7 H_DATA[21] M_A[22]	W7 H_DATA[20] M_A[5]	Y7 H_DATA[18] M_A[3]
L8 VSS	M8 VSS	N8 VSS				U8 H_RD/WR M_RD/WR	V8 H_DATA[17] M_A[2]	W8 H_DATA[16] M_A[1]	Y8 H_DATA[19] M_A[4]
K9 VSS	M9 VSS	N9 VSS			T9 VDD_1.8	U9 H_DTACK M_DTACK	V9 H_WAIT M_WAIT	W9 H_DATA[16] M_A[1]	Y9 H_CS M_GPIO[5]
L10 VSS	M10 VSS	N10 VSS			T10 VDD_1.8	U10 H_DATA[14] M_A[20]/M_D[14]	V10 H_DATA[12] M_A[18]/M_D[12]	W10 H_ADDR[1] M_GPIO[2]	Y10 H_RD M_GPIO[4]
L11 VSS	M11 VSS	N11 VSS			T11 VDD_3.3	U11 H_DATA[8] M_A[14]/M_D[8	V11 H_DATA[13] M_A[19]/M_D[13]	W11 H_DATA[9] M_A[15]/M_D[9]	Y11 H_DATA[15] M_A[21]/M_D[15]
L12 VSS	M12 VSS	N12 VSS			T12 VDD_3.3	U12 H_DATA[5] M_A[11]/M_D[5]	V12 H_DATA[10] M_A[16]/M_D[10]	W12 H_DATA[6] M_A[12]/M_D[6]	Y12 H_DATA[11] M_A[17/M_D[11]
L13 VSS	M13 VSS	N13 VSS				U13 M_ALE	V13 H_DATA[7] M_A[13]/M_D[7]	W13 H_DATA[2] M_A[8]/M_D[2]	Y13 H_ADDR[0] M_GPIO[1]
						U14 SIO_IRTX1 M_A[2]	V14 H_DATA[3] M_A[9]/M_D[3]	W14 H_DATA[1] M_A[7]/M_D[1]	Y14 H_DATA[4] M_A[10]/M_D[4]
						U15 VDD_DLL	V15 H_DMAREQ M_UWE/UDS	W15 H_DATA[0] M_A[6]/M_D[0]	Y15 M_OE
L16 VDD_2.5	M16 MCONFIG[0]					U16 VDD_DLL	V16 SIO_SPI_CS[2] M_A[22]	W16 H_RST M_RST	Y16 MCONFIG[1]
L17 SDRAM_DQM[1]	M17 SDRAM_DQ[12]	N17 SDRAM_DQM[0]	P17 SDRAM_DQ[4]	R17 SDRAM_DQ[0]	T17 VSS_DLL	U17 VSS_DLL	V17 SIO_UART1_CTS M_RD/WR	W17 SIO_IRTX2 M_OE	Y17 SIO_SPI_CLK M_A[26]
L18 SDRAM_DQ[8]	M18 SDRAM_DQ[14]	N18 SDRAM_DQ[15]	P18 SDRAM_DQ[6]	R18 SDRAM_DQ[2]	T18 SIO_SPI_MISO M_A[1]	U18 SIO_SPI_MOSI M_A[25]	V18 SIO_SPI_CS[1] M_A[23]	W18 SIO_SPI_CS[3] M_A[5]	Y18 SIO_SDA
L19 SDRAM_DQ[11]	M19 SDRAM_DQS[1][1]	N19 SDRAM_DQ[9]	P19 SDRAM_DQ[7]	R19 SDRAM_DQS[0][1]	T19 SDRAM_DQ[1]	U19 SIO_UART2_TX	V19 SIO_SCL	W19 SIO_UART1_RX M_CS[0]	Y19 SIO_SPI_CS[0] M_A[24]
L20 SDRAM_DQ[13]	M20 SDRAM_DQ[10]	N20 SDRAM_CLK[0] [1]	P20 SDRAM_CLK[0]	R20 SDRAM_DQ[5]	T20 SDRAM_DQ[3]	U20 SIO_UART1_TX M_A[4]	V20 SIO_UART1_RTS M_A[3]	W20 SIO_UART2_RX	Y20 SIO_IRRX

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.35 308 BGA Alphabetical Pin List

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AI_D[0]	C13	ATAPI_DIOR		H_DATA[16]		H_RST M_RST	W16	SDRAM_DQ[28]	G18
AI_D[1]	D12	SD_RDREQ	V3	M_A[1]	W8	H_WAIT M_WAIT	V9	SDRAM_DQ[29]	H17
AI_FSYNC	C12	ATAPI_DIOW		H_DATA[17]		M_ALE	U13	SDRAM_DQ[2]	R18
AI_MCLKO	A13	SD_WRREQ	T1	M_A[2]	V8	M_OE	Y15	SDRAM_DQ[30]	F19
AI_SCLK	B12	ATAPI_DMAACK		H_DATA[18]		MCONFIG[0]	M16	SDRAM_DQ[31]	F20
AO_D[0]	D14	SD_SECSTART	U1	M_A[3]	Y7	MCONFIG[1]	Y16	SDRAM_DQ[3]	T20
AO_D[1]	C14	ATAPI_DMARQ		H_DATA[19]		NC	A3	SDRAM_DQ[4]	P17
AO_D[2]	B15	SD_ERROR	W1	M_A[4]	Y8	NC	A4	SDRAM_DQ[5]	R20
AO_D[3]	A15	ATAPI_INTRQ		H_DATA[1]		NC	A6	SDRAM_DQ[6]	P18
AO_FSYNC	D13	SD_ACK	W2	M_A[7]/M_D[1]	W14	NC	B1	SDRAM_DQ[7]	P19
AO_IEC958	B14	ATAPI_IORDY	T3	H_DATA[20]		NC	B4	SDRAM_DQ[8]	L18
AO_MCLKO	B13	ATAPI_RESET	Y1	M_A[5]	W7	NC	B5	SDRAM_DQ[9]	N19
AO_SCLK	A14	BIO_LINK_ON	K4	H_DATA[21]		NC	C5	SDRAM_	
ATAPI_ADDR[0]		BIO_LPS	L4	M_A[22]	V7	NC	C6	DQM[0]	N17
SBP_FRAME	W3	BIO_LREQ	M1	H_DATA[22]		NC	D2	SDRAM_	
ATAPI_ADDR[1]		BIO_PHY_CLK	L1	M_A[23]	W6	NC	D6	DQM[1]	L17
SBP_ACK	V2	BIO_PHY_		H_DATA[23]		PLL_BYPASS	A7	SDRAM_	
ATAPI_ADDR[2]		CTL[0]	J3	M_A[24]	U6	RREF	C11	DQM[2]	K20
SBP_RD	V1	BIO_PHY_		H_DATA[24]		SDRAM_A[0]	A19	SDRAM_	
ATAPI_ADDR[3]		CTL[1]	L2	M_A[25]	Y6	SDRAM_A[10]	C19	DQM[3]	G19
SBP_REQ	U3	BIO_PHY_		H_DATA[25]		SDRAM_A[11]	F17	SDRAM_DQS[0][1]	R19
ATAPI_ADDR[4]		DATA[0]	J2	M_A[26]	Y5	SDRAM_A[12]	E17	SDRAM_DQS[1][1]	M19
SBP_CLK	V4	BIO_PHY_		H_DATA[26]		SDRAM_A[13]	E18	SDRAM_DQS[2][1]	J19
ATAPI_DATA[0]		DATA[1]	L3	M_CS[0]	V6	SDRAM_A[14]	C20	SDRAM_DQS[3][1]	E20
SD_DATA[0]		BIO_PHY_		H_DATA[27]		SDRAM_A[15]	A18	SDRAM_RAS	B16
CD_DATA	R4	DATA[2]	M3	M_CS[1]	Y4	SDRAM_A[1]	B19	SDRAM_	
ATAPI_DATA[10]		BIO_PHY_		H_DATA[28]		SDRAM_A[2]	A17	VREF[1]	D16
SBP_DATA[2]	P2	DATA[3]	M2	M_CS[2]	W5	SDRAM_A[3]	C17	SDRAM_WE	A16
ATAPI_DATA[11]		BIO_PHY_		H_DATA[29]		SDRAM_A[4]	B18	SIO_IRRX	Y20
SBP_DATA[3]	R1	DATA[4]	J1	M_CS[3]	V5	SDRAM_A[5]	A20	SIO_IRTX1	
ATAPI_DATA[12]		BIO_PHY_		H_DATA[2]		SDRAM_A[6]	C18	M_A[2]	U14
SBP_DATA[4]	R2	DATA[5]	K1	M_A[8]/M_D[2]	W13	SDRAM_A[7]	B20	SIO_IRTX2	
ATAPI_DATA[13]		BIO_PHY_		H_DATA[30]		SDRAM_A[8]	D18	M_OE	W17
SBP_DATA[5]	T2	DATA[6]	K2	M_CS[4]	Y3	SDRAM_A[9]	D19	SIO_SCL	V19
ATAPI_DATA[14]		BIO_PHY_		H_DATA[31]		SDRAM_CAS	B17	SIO_SDA	Y18
SBP_DATA[6]	U2	DATA[7]	K3	M_CS[5]	W4	SDRAM_CKE	C15	SIO_SPI_CLK	
ATAPI_DATA[15]		CLKI	A10	H_DATA[3]		SDRAM_CLK[0]	P20	M_A[26]	Y17
SBP_DATA[7]	Y2	CLKO	A12	M_A[9]/M_D[3]	V14	SDRAM_CLK[0][1]	N20	SIO_SPI_CS[0]	
ATAPI_DATA[1]		CLKX	B10	H_DATA[4]		SDRAM_CLK[1]	G20	M_A[24]	Y19
SD_DATA[1]		H_ADDR[0]		M_A[10]/M_D[4]	Y14	SDRAM_CLK[1][1]	H20	SIO_SPI_CS[1]	
CD_LRCK	R3	M_GPIO[1]	Y13	H_DATA[5]		SDRAM_DQ[0]	R17	M_A[23]	V18
ATAPI_DATA[2]		H_ADDR[1]		M_A[11]/M_D[5]	U12	SDRAM_DQ[10]	M20	SIO_SPI_CS[2]	
SD_DATA[2]		M_GPIO[2]	W10	H_DATA[6]		SDRAM_DQ[11]	L19	M_A[22]	V16
CD_BCK	P4	H_ADDR[2]		M_A[12]/M_D[6]	W12	SDRAM_DQ[12]	M17	SIO_SPI_CS[3]	
ATAPI_DATA[3]		M_GPIO[3]	W9	H_DATA[7]		SDRAM_DQ[13]	L20	M_A[5]	W18
SD_DATA[3]		H_CS M_GPIO[5]	Y9	M_A[13]/M_D[7]	V13	SDRAM_DQ[14]	M18	SIO_SPI_MISO	
CD_C2PO	P3	H_DATA[0]		H_DATA[8]		SDRAM_DQ[15]	N18	M_A[1]	T18
ATAPI_DATA[4]		M_A[6]/M_D[0]	W15	M_A[14]/M_D[8]	U11	SDRAM_DQ[16]	K19	SIO_SPI_MOSI	
SD_DATA[4]	P1	H_DATA[10]		H_DATA[9]		SDRAM_DQ[17]	K18	M_A[25]	U18
ATAPI_DATA[5]		M_A[16]/M_D[10]	V12	M_A[15]/M_D[9]	W11	SDRAM_DQ[18]	K17	SIO_UART1_CTS	
SD_DATA[5]	N4	H_DATA[11]		H_DMAREQ		SDRAM_DQ[19]	J20	M_RD/WR	V17
ATAPI_DATA[6]		M_A[17/M_D[11]	Y12	M_UWE/UDS	V15	SDRAM_DQ[1]	T19	SIO_UART1_RTS	
SD_DATA[6]	N2	H_DATA[12]		H_DTACK M_		SDRAM_DQ[20]	J18	M_A[3]	V20
ATAPI_DATA[7]		M_A[18]/M_D[12]	V10	DTACK	U9	SDRAM_DQ[21]	J17	SIO_UART1_RX	
SD_DATA[7]	N1	H_DATA[13]		H_INT M_		SDRAM_DQ[22]	H18	M_CS[0]	W19
ATAPI_DATA[8]		M_A[19]/M_D[13]	V11	GPIO[0]	U7	SDRAM_DQ[23]	H19	SIO_UART1_TX	
SBP_DATA[0]	M4	H_DATA[14]		H_RD M_		SDRAM_DQ[24]	D20	M_A[4]	U20
ATAPI_DATA[9]		M_A[20]/M_D[14]	U10	GPIO[4]	Y10	SDRAM_DQ[25]	E19	SIO_UART2_RX	W20
SBP_DATA[1]	N3	H_DATA[15]		H_RD/WR M_		SDRAM_DQ[26]	F18	SIO_UART2_TX	U19
		M_A[21]/M_D[15]	Y11	RD/WR	U8	SDRAM_DQ[27]	G17		

NC pins are not connected.

1. For DDR parts only. DDR refers to Double Data Rate SDRAMs, which have double the bandwidth of standard SDRAMs and are used for high-performance, nonpower critical applications.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Table 18.35 308 BGA Alphabetical Pin List (Cont.)

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
TCK	D8	VO_D[7]	F1	VDD_DLL	U15	VSS	M13	VO_D[14]	E3
TDI	C7	VO_D[8]	F2	VDD_DLL	U16	VSS	N8	VO_D[15]	F4
TDO	B7	VO_D[9]	G4	VDD_RREF	D11	VSS	N9	VO_D[1]	H2
TMS	B6	VSS	H8	VI_CLK[0]	A5	VSS	N10	VO_D[2]	H3
TRST	D7	VSS	H9	VI_D[2]	B2	VSS	N11	VO_D[3]	H4
VDD_1.8	D4	VSS	H10	VI_D[3]	C3	VSS	N12	VO_D[4]	G1
VDD_1.8	E10	VSS	H11	VI_D[4]	C2	VSS	N13	VO_D[5]	G2
VDD_1.8	E12	VSS	H12	VI_D[5]	D3	VSS_2.5	K16	VO_D[6]	G3
VDD_1.8	J5	VSS	H13	VI_D[6]	C4	VSS_A	A8	VO_D[7]	F1
VDD_1.8	M5	VSS	J8	VI_D[7]	B3	VSS_A	A9	VO_D[8]	F2
VDD_1.8	T9	VSS	J9	VI_D[8]	A2	VSS_A	C8	VO_D[9]	G4
VDD_1.8	T10	VSS	J10	VI_D[9]	A1	VSS_A	C9	VSS	H8
VDD_1.8	U4	VSS	J11	VI_VSYNC[0]	C1	VSS_DLL	T17	VSS	H9
VDD_2.5	C16	VSS	J12	VO_CLK	H1	VSS_DLL	U17	VSS	H10
VDD_2.5	D15	VSS	J13	VO_D[0]	J4	VSS_RREF	B11	VSS	H11
VDD_2.5	D17	VSS	K8	VO_D[10]	E1	VSS_X	C10	VSS	H12
VDD_2.5	J16	VSS	K9	VO_D[11]	F3	XVDD	A11	VSS	H13
VDD_2.5	L16	VSS	K10	VO_D[12]	E2	VCC_1.8	M5	VSS	J8
VDD_3.3	D5	VSS	K11	VO_D[13]	D1	VDD_1.8	T9	VSS	J9
VDD_3.3	E4	VSS	K12	VO_D[14]	E3	VDD_1.8	T10	VSS	J10
VDD_3.3	E9	VSS	K13	VO_D[15]	F4	VDD_1.8	U4	VSS	J11
VDD_3.3	E11	VSS	L8	VO_D[1]	H2	VDD_2.5	C16	VSS	J12
VDD_3.3	K5	VSS	L9	VO_D[2]	H3	VDD_2.5	D15	VSS	J13
VDD_3.3	L5	VSS	L10	VO_D[3]	H4	VDD_2.5	D17	VSS	K8
VDD_3.3	T4	VSS	L11	VO_D[4]	G1	VDD_2.5	J16	VSS	K9
VDD_3.3	T11	VSS	L12	VO_D[5]	G2	VDD_2.5	L16	VSS	K10
VDD_3.3	T12	VSS	L13	VO_D[6]	G3	VDD_3.3	D5	VSS	K11
VDD_5	U5	VSS	M8	VO_D[7]	F1	VDD_3.3	E4	VSS	K12
VDD_A	B8	VSS	M9	VO_D[8]	F2	VDD_3.3	E9	VSS	K13
VDD_A	B9	VSS	M10	VO_D[9]	G4	VDD_3.3	E11	VSS	L8
VDD_A	D9	VSS	M11	VSS	H8	VDD_3.3	K5	VSS	L9
VDD_A	D10	VSS	M12	VSS	H9	VDD_3.3	L5	VSS	L10
VDD_DLL	U15	VSS	M13	VSS	H10	VDD_3.3	T4	VSS	L11
VDD_DLL	U16	VSS	N8	VSS	H11	VDD_3.3	T11	VSS	L12
VDD_RREF	D11	VSS	N9	VSS	H12	VDD_3.3	T12	VSS	L13
VI_CLK[0]	A5	VSS	N10	VSS	H13	VDD_5	U5	VSS	M8
VI_D[2]	B2	VSS	N11	VSS	J8	VDD_A	B8	VSS	M9
VI_D[3]	C3	VSS	N12	VSS	J9	VDD_A	B9	VSS	M10
VI_D[4]	C2	VSS	N13	VSS	J10	VDD_A	D9	VSS	M11
VI_D[5]	D3	VSS_2.5	K16	VSS	J11	VDD_A	D10	VSS	M12
VI_D[6]	C4	VSS_A	A8	VSS	J12	VDD_DLL	U15	VSS	M13
VI_D[7]	B3	VSS_A	A9	VSS	J13	VDD_DLL	U16	VSS	N8
VI_D[8]	A2	VSS_A	C8	VSS	K8	VDD_RREF	D11	VSS	N9
VI_D[9]	A1	VSS_A	C9	VSS	K9	VI_CLK[0]	A5	VSS	N10
VI_VSYNC[0]	C1	VSS_DLL	T17	VSS	K10	VI_D[2]	B2	VSS	N11
VO_CLK	H1	VSS_DLL	U17	VSS	K11	VI_D[3]	C3	VSS	N12
VO_D[0]	J4	VSS_RREF	B11	VSS	K12	VI_D[4]	C2	VSS	N13
VO_D[10]	E1	VSS_X	C10	VSS	K13	VI_D[5]	D3	VSS_2.5	K16
VO_D[11]	F3	XVDD	A11	VSS	L8	VI_D[6]	C4	VSS_A	A8
VO_D[12]	E2	VDD_3.3	K5	VSS	L9	VI_D[7]	B3	VSS_A	A9
VO_D[13]	D1	VDD_3.3	L5	VSS	L10	VI_D[8]	A2	VSS_A	C8
VO_D[14]	E3	VDD_3.3	T4	VSS	L11	VI_D[9]	A1	VSS_A	C9
VO_D[15]	F4	VDD_3.3	T11	VSS	L12	VI_VSYNC[0]	C1	VSS_DLL	T17
VO_D[1]	H2	VDD_3.3	T12	VSS	L13	VO_CLK	H1	VSS_DLL	U17
VO_D[2]	H3	VDD_5	U5	VSS	M8	VO_D[0]	J4	VSS_RREF	B11
VO_D[3]	H4	VDD_A	B8	VSS	M9	VO_D[10]	E1	VSS_X	C10
VO_D[4]	G1	VDD_A	B9	VSS	M10	VO_D[11]	F3	XVDD	A11
VO_D[5]	G2	VDD_A	D9	VSS	M11	VO_D[12]	E2		
VO_D[6]	G3	VDD_A	D10	VSS	M12	VO_D[13]	D1		

NC pins are not connected.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.4 Package Mechanical Specifications

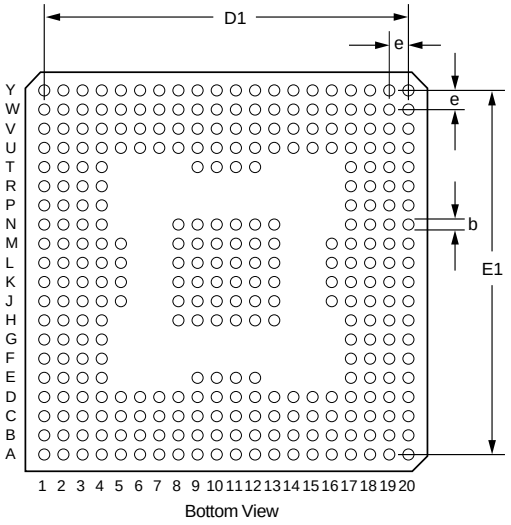
This section provides the dimensions and recommended manufacturing conditions for the DMN-8600 processor.

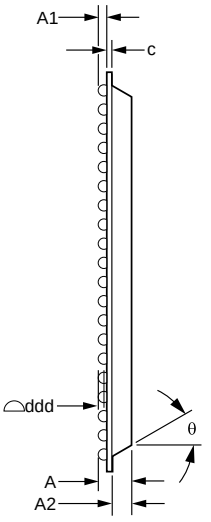
18.4.1 Package Dimensions

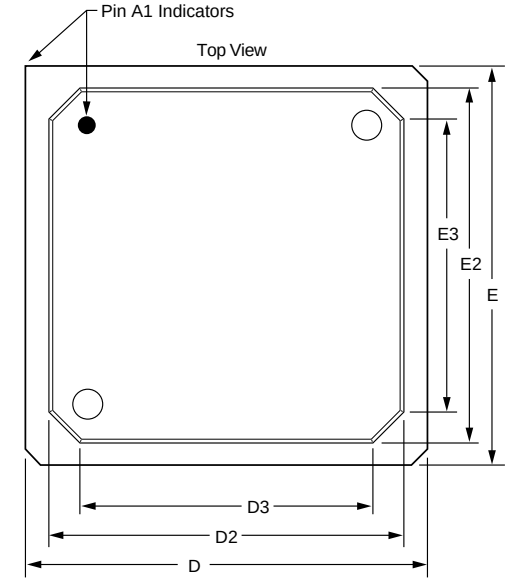
The DMN-8600 processor is available in a 308-pin ball grid array (BGA) package illustrated in Figure 18.45.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Figure 18.45 308-Pin BGA Package Mechanical Dimensions







Symbol	Dimensions in mm			Dimensions in inches		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	2.20	2.33	2.50	0.087	0.092	0.098
A1	–	0.60	–	–	0.024	–
A2	1.12	1.17	1.22	0.044	0.046	0.048
b	–	0.75	–	–	0.030	–
c	0.51	0.56	0.61	0.020	0.022	0.024
D	26.80	27.00	27.20	1.055	1.063	1.071
D1	–	24.13	–	–	0.950	–
D2	23.80	24.00	24.20	0.937	0.945	0.953
D3	17.95	18.00	18.05	0.707	0.709	0.711
E	26.80	27.00	27.20	1.055	1.063	1.071
E1	–	24.13	–	–	0.950	–
E2	23.80	24.00	24.20	0.937	0.945	0.953
E3	17.95	18.00	18.05	0.707	0.709	0.711
e	–	1.27	–	–	0.050	–
ddd	–	–	0.15	–	–	0.006
θ	30˚ Typ			30˚ Typ		

1. The metric (millimeter) values are the controlling dimensions and should be used for PC board design.
2. Dimension b is measured at the maximum solder ball diameter.
3. Minimum clearance of 0.25 mm between the edge of the solder ball and the body edge.

Important: **This drawing may not be the latest version.**

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

18.4.2 Recommended Manufacturing Conditions

LSI Logic recommends the following conditions for surface mounting the DMN-8600 processor:

- Use chlorineless rosin flux.
- Remove flux sufficiently from the product during the washing process.
- Be aware that resonance may occur during ultrasonic washing based on the frequency and PCB shape. This resonance might affect ball strength.

Table 18.36 specifies the recommended hot air or infrared (IR) solder reflow conditions.

Table 18.36 Recommended Hot Air or IR Solder Reflow Conditions

Parameter	Specification
Preheating Temperature	140 to 160 ˚C
Preheating Temperature Hold Time	60 to 120 s
Heating Acceleration	1 to 3 ˚C/s maximum
Package Case Temperature	220˚C
Peak Temperature Hold Time	10 s maximum
High Temperature Region Hold Time	60 s maximum @ 183 ˚C
Cooling Rate	5 ˚C/s maximum

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Chapter A
Register Listing

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.

Reader's Comments

Fax your comments to: LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *DMN-8600 DVD Recorder System Processor.* Place a check mark in the appropriate blank for each category.

	Excellent	Good	Average	Fair	Poor
Completeness of information	____	____	____	____	____
Clarity of information	____	____	____	____	____
Ease of finding information	____	____	____	____	____
Technical content	____	____	____	____	____
Usefulness of examples and illustrations	____	____	____	____	____
Overall manual	____	____	____	____	____

What could we do to improve this document?

_____

_____

_____

_____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

_____

_____

_____

Please complete the information below so that we may contact you directly for clarification or additional information.

Name _____ Date _____

Telephone _____ Fax _____

Title _____

Department _____ Mail Stop _____

Company Name _____

Street _____

City, State, Zip _____

Copyright © 2001, 2002 by LSI Logic Corporation. All rights reserved.