LG Semicon Co.,Ltd.

GM73V1892AH16L
GM73V1892AH17L
2,097,152 WORDS x/9 BIT
Rambus DRAM

Description

The GM73V1892AH16L / GM73V1892AH17L Rambus Dynamic Random Access Memory (RDRAM) is a next generation high-speed CMOS DRAM organized as 2,097,152 x9 bits and capable of bursting up to 256 bytes of data at less than 1.67 nanoseconds per byte. The use of Rambus Signaling Logic (RSL) technology makes transfer rates greater than 600 MB/s achievable while using conventional system and board design methodologies. Lower effective latency is attained by operating the single 2KByte sense amplifier as high speed cache, or by using Random Access mode to facilitate large block transfers. The GM73V1892AH16L / GM73V1892AH17L uses a 32-pin plastic surface horizontal mount package (SHP). The GM73V1892AH16L / GM73V1892AH17L is general purpose   high-performance memory devices suitable for use in a broad range of applications including PC and consumer main memory, graphics, video, and any other application where high-performance at low cost are required.

Features

■ Rambus Interface:

- 600MB/sec peak transfer rate per RDRAM
- Rambus Signaling Logic (RSL) interface
- Synchronous protocol for fast block-oriented transfers
- Direct connection to Rambus ASICs,MPUs, and Peripherals
- Only 15 active signals that require just 31 pins total on the controller interface
- 3.3 $\pm$ 0.15 volt operation
- Additional/multiple Rambus Channels provide an additional 600 MB/second bandwidth each

■ 2KByte sense amplifier may be operated as cache for low latency access

■ Random Access mode enables any burst order at full bandwidth

■ Features for graphics include random-access mode, write-per-bit and mask-per-bit operations

■ Control and refresh logic entirely self-contained

■ On-chip registers for flexible addressing and timing

■ Available in horizontal and vertical surface mount plastic packages

■ Higher performance than VRAM at DRAM cost

System Benefits

■ Superior performance at a lower system cost than alternative multi-DRAM chip solutions

■ Fully engineered solution includes clock chips, memory expansion sockets, simple layout

■ For graphics subsystems addressing display resolutions of 1024x768x16 or above, provides lowest cost frame buffer, greater performance, fewest controller pins, ease of memory expansion

■ For Pentium   processor class main memory, provides faster memory subsystem, fewer components, and 2Mbyte granularity

■ Sufficient performance for unified memory system architectures in cost sensitive consumer products

Part Number

Option	580MHz Low Latency	600MHz Low Latency
18M SHP	GM73V1892AH-17L	GM73V1892AH-16L

Rambus System Overview

A typical Rambus memory system has three main elements : the Rambus Controller, the Channel, and the RDRAMs. The logical representation of this is shown in Figure 1.
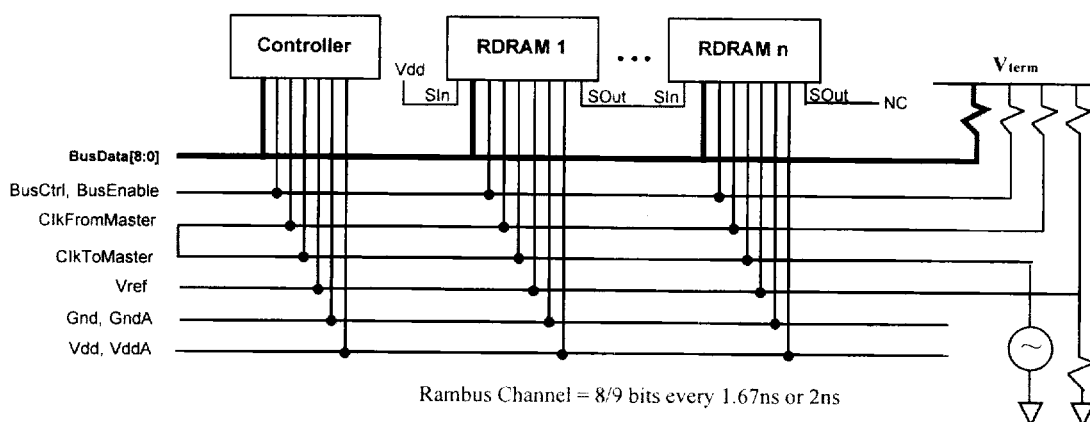

Rambus Channel = 8/9 bits every 1.67ns or 2ns

Figure 1. Controller and RDRAMs Connect to Terminated Transmission Lines

The Rambus Channel is a high-speed, byte-wide, synchronous bus used to connect Rambus devices together. The Channel carries all address, data, and control information to and from devices. Transfer of data on the Rambus Channel is managed through the use of a high level block-oriented protocol.

The Rambus Interface is implemented on both master and slave devices. Rambus master (ASIC devices, memory controllers, graphics engines, peripheral chips, or microprocessors) are the only devices capable of generating transaction requests. RDRAMs are slave devices and can only respond to requests from master devices.

The Rambus Channel has thirteen high speed Rambus Signaling Logic (RSL) I/O signals that are used to transfer information at 1.67 nanosecond intervals. These signals use low voltage swings (logic 0 is about 2.5V and logic 1 is about 1.7V when Vref is 2.1V) to achieve high bus speeds. Two TTL level signals are used for initialization and powerdown operation. Fourteen signals supply power and DC voltage references to the RDRAM, and the remaining pins are No Connects (reserved for future expansion).

Figure 2 shows a typical physical implementation of a Rambus system. It includes a controller ASIC that acts as the Channel master and a base set of RDRAMs soldered directly to the board. A RSocket™ is included on the Channel for memory upgrade using RModule™ expansion cards.
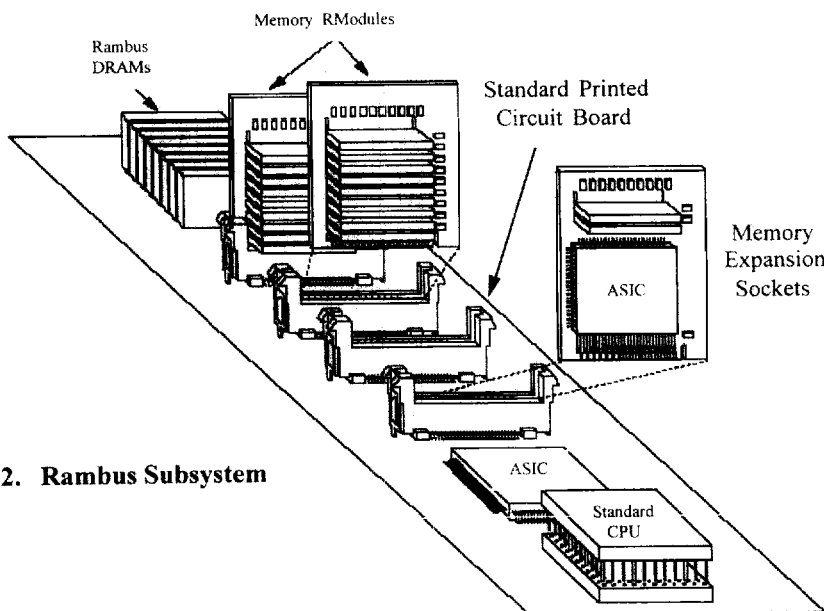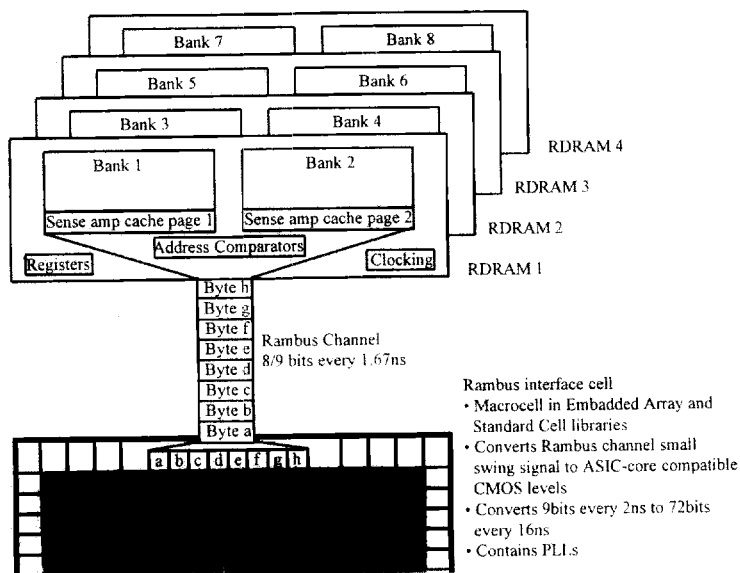

Figure 2. Rambus Subsystem

Data Transfer on the Rambus Channel

Rambus interface cell
• Macrocell in Embadded Array and Standard Cell libraries
• Converts Rambus channel small swing signal to ASIC-core compatible CMOS levels
• Converts 9bits every 2ns to 72bits every 16ns
• Contains PLLs

RDRAM Packages and Pinouts

RDRAMs is available in an EIAJ standard 32-pin TSOP-style horizontal surface mount package (SHP).

This package has 32 signal pins and four mechanical pins that provide support for the device.

Table 1 : Pin Descriptions

Signal	I/O	Description
BusData[8-0]	I/O	Signal lines for request, write data, and read data packets. The request packet contains the address, operation codes. and the count of the bytes to be transferred. These are low-swing, active-low signals referenced to Vref
RxClk	I	Receive clock. Incoming request and write data packets are aligned to this clock. This is a low-swing, active-low signal referenced to Vref.
TxClK	I	Transmit clock. Outgoing acknowledge and read data packets are aligned with this clock. This is a low-swing, active-low signal referenced to Vref.
Vref	I	Logic threshold reference voltage for low swing signals.
BusCtrl	I/O	Control signal to frame packets, transmit part of the oper-ation code, to acknowledge requests, and to interrupt (terminate) pending transactions. This is a low-swing, active-low signal referenced to Vref.
BusEnable	I	Control signal to manage the operating modes of the RDRAMs and to transfer column addresses for random - access (non-sequential) transactions. This is a low-swing, active-low signal referenced to Vref.
Vdd, VddA		+3.3V power supply. VddA is a separate analog supply for clock recovery in the RDRAM
Gnd, GndA		Circuit ground. GndA is a separate analog ground for clock generation in the RDRAM
SIn	I	Initialization daisy chain input. TTL levels. Active high .
SOut	O	Initialization daisy chain output. TTL levels. Active high.

Pin numbering (SHP):

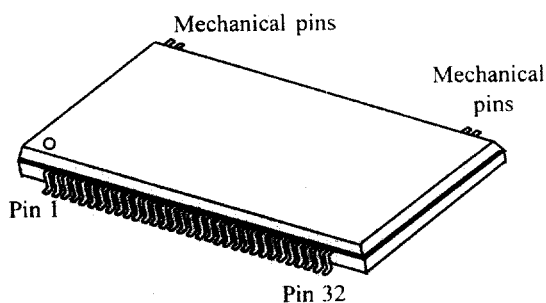| Pin	Signal
1	V_{DD}
2	Gnd
3	BusData8
4	Gnd
5	BusData7
6	(NC)
7	BusEnable
8	V_{DD}
9	BusData6
10	Gnd
11	BusData5
12	V_{DDA}
13	RxClk
14	GndA
15	TxClk
16	V_{DD}
17	BusData4
18	Gnd
19	BusCtrl
20	SIn
21	V_{REF}
22	SOut
23	BusData3
24	Gnd
25	BusData2
26	(NC)
27	BusData1
28	Gnd
29	BusData0
30	(NC)
31	Gnd
32	V_{DD}

Figure 3. SHP Pin Numbering

Figure 4. SHP Packages

Protocol

The transaction protocol used in Rambus systems is built from several types of information packets. These include the request, acknowledge, serial mode, serial address and data packets.

Request Packet : A master device initiates a transaction by generating a six- byte request packet containing address, control, and byte count information as shown in Figure 5.
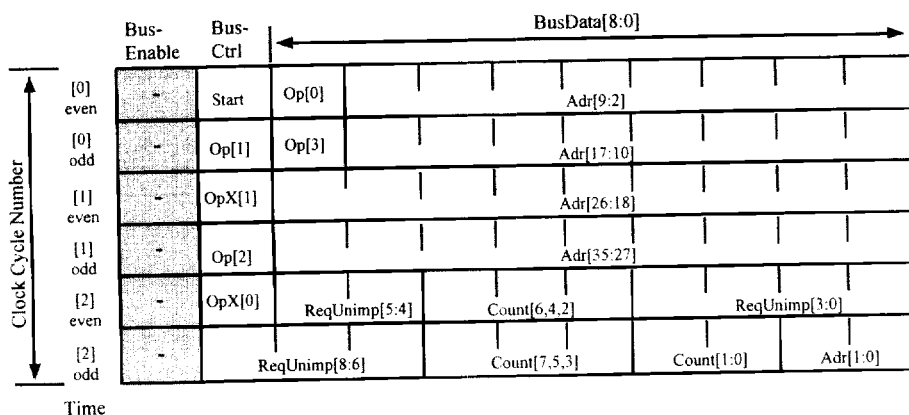

Figure 5. Request Packet[*1]

Acknowledge Packet : Each RDRAM monitors the Channel for a request to access its assigned memory range. The device matching the address range requested then devices an acknowledge packet back to the master.
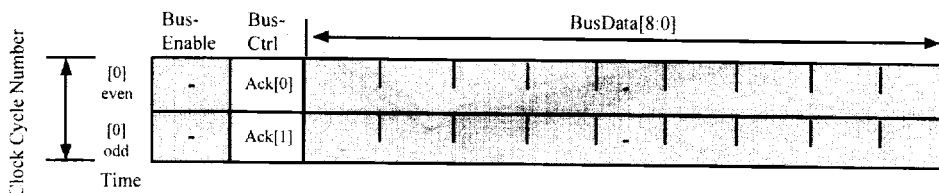

Figure 6. Acknowledge Packet[i]

The Ack[1:0] field in the Acknowledge packet carries the RDRAMs response to the request. If the RDRAM is able to complete the operation as requested, it returns an Okay response. If the RDRAM is unable to complete the operation as requested, it returns a negative acknowledge response(Nack). The encoding of the Ack[1:0] bits is shown in Table 3.

Table 3 : Ack[1:0] Encoding

| Ack [1:0]	Name	Description
00	Non-existent	Indicates passive acceptance of the request(WregB), or indicates that the addressed device did not respond (all other commands).
01	Okay	Indicates that the request was accepted by the addressed (responding) device.
10	Nack	Indicates that the request could not be accepted because the state of the responding device prevented an access at the fixed timing slot.
11	Ack 3	This should not be returned by this responding device. Initiating devices will, when presented with this combination, have an undefined response.

RDRAM Commands

The Op and OpX fields in the Request packet contain a command that is used to instruct the RDRAM which operation is being requested. A summary of these commands is shown in Table 2.

Table 2 : Command Summary

Op[3:0]	OpX[1:0]	Name	Description
0000	00	Rseq	Read sequential data from memory space.
0000	01	Rnsq	Read random-access (non-sequential) data from memory space.
0100	00	WseqNpb	Write sequential data to memory space with no per-bit mask application.
0100	01	WseqDpb	Write sequential data to memory space with data-per-bit masking. Static bit masks are supplied by the MDReg while write data is supplied in the data packet.
0100	10	WseqBpb	Write sequential data to memory space with mask-per-bit masking. Both write data and dynamic bit masks are supplied in the data packet.
0100	11	WseqMpb	Write sequential data to memory space with mask-per-bit masking. Static write data is supplied by the MDReg while dynamic bit masks are supplied in the data packet.
0110	00	Rreg	Read data from register space.
0111	00	Wreg	Write data to register space.
1000	00	WnsqNpb	Write random-access (non-sequential) data to memory space with no per-bit mask application.
1000	01	WnsqDpb	Write random-access(non-sequential) data to memory space with data-per-bit masking. Static bit masks are supplied by the MDReg while write data is supplied in the data packet.
1000	10	WnsqBpb	Write random- access (non-sequential) data to memory space with mask-per-bit masking. Both write data and dynamic bit masks are supplied in the data packet.
1000	11	WnsqMpb	Write random-access(non-sequential) data to memory space with mask-per-bit masking. Static write data is supplied by the MDReg while dynamic bit masks are supplied in the data packet.
1100	00	WbnsNpb	Write random-access (non-sequential) data to memory space with byte masking and no per-bit mask application. Both byte masks and write data are supplied in the data packet.
1100	01	WbnsDpb	Write random-access (non-sequential) data to memory space with byte masking and data-per-bit masking. Static bit masks are supplied by the MDReg while byte masks and write data are supplied in the data packets.
1100	11	WbnsMpb	Write random-access (non-sequential) data to memory space with byte masking and mask-per-bit masking. Static write data is supplied by the MDReg while byte masks and dynamic bit masks are supplied in the data packets.
1111		WregB	Broadcast write to register space of all responding devices with no acknowledge permitted

In response to an acknowledged command, the RDRAM either drives a data packet back to the master in the case of a read, or accepts a data packet from the master in the case of a write, Figure 7 shows example 16 byte read and write transactions. The actual timing from the end of a request packet to data and acknowledge packets is adjustable through RDRAM register settings.

Figure 7. Sample 16-Byte Read and Write Transactions
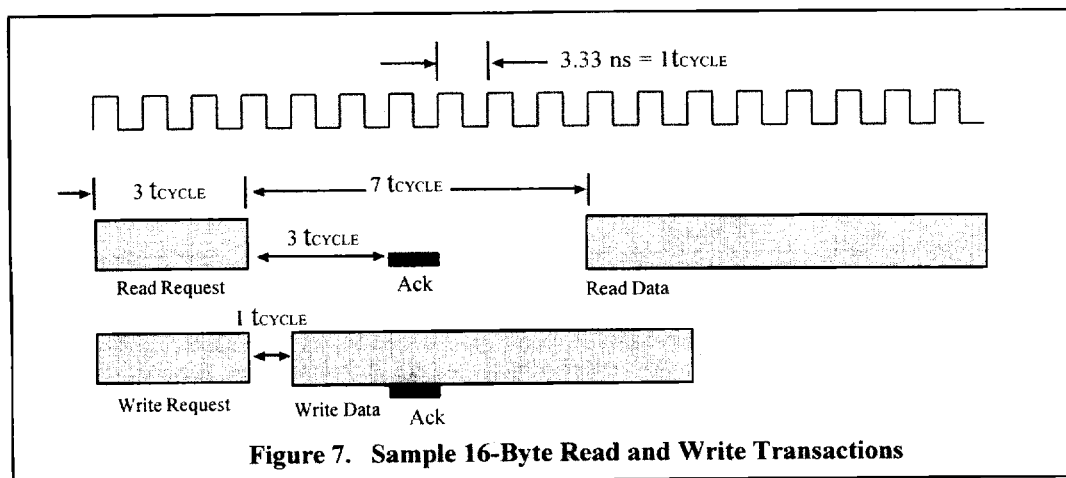
Serial Address Packet : The non-sequential (Random Access) commands specify the eight column address bits needed to access random octbytes within the open page. These address bits are provided using a Serial Address packets.
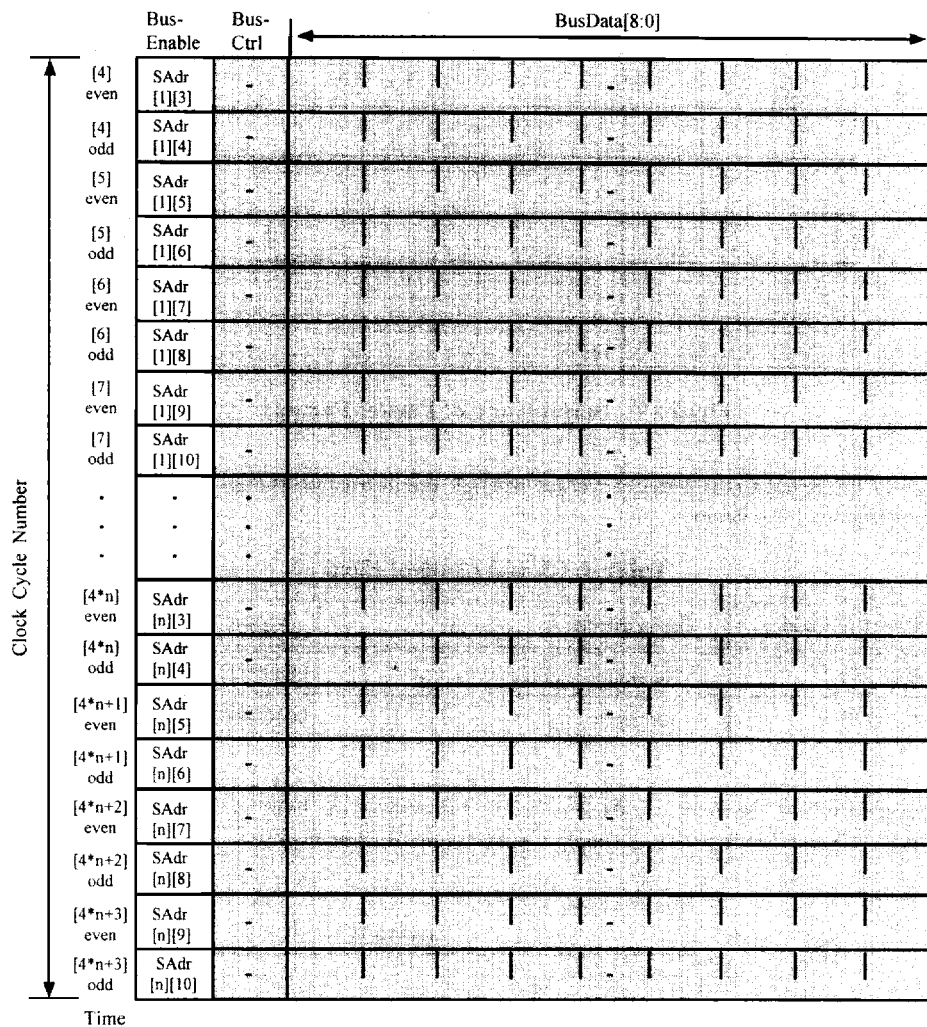

Figure 8.   Serial Address Packet Format[*i]

Serial Control Packet : The protocol also allows the Channel master to issue an early termination instruction for a memory read or write transaction. This is done using a Serial Control packet.
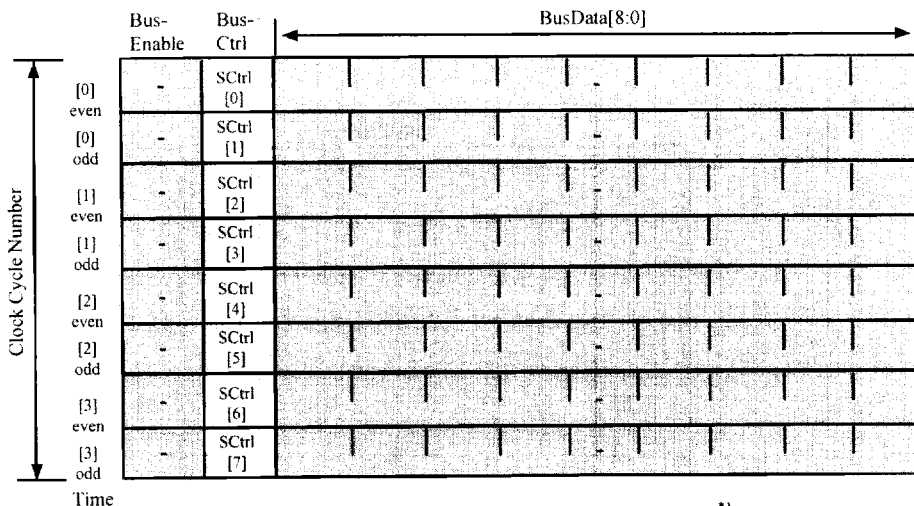

Figure 9. Serial Control Packet Format[*1]

The value of the Serial Control field specifies whether the instruction should continue or terminate. This is shown in the table below.

Table 4. Serial Control Field

Serial Control Field	Description	Value
SCtrl[7:0]	Continue	00000000
SCtrl[7:0]	Terminate	00100000

Serial Mode Packet : Serial Mode Packets are used to instruct the RDRAM to perform a operating mode change. A Serial Mode packet is simply a pulse on the BusEnable line.
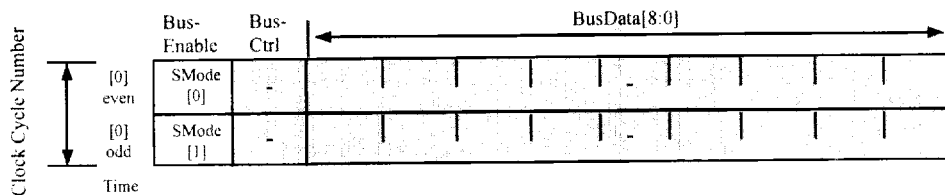

Figure 10. Serial Mode Packet[*1]

RDRAM Overview

The figure 12 is a block diagram of the RDRAM device. The Rambus Channel interface consists of a clock generator, a receiver, and a transmitter. The clock generator uses the external clock signals RxClk and TxClk (tapped off the Channel traces ClockFromMaster and ClockToMaster) and creates the internal signals RClk and TClk. These are used by the receiver and transmitter, respectively, to transfer a bit every 1.67 nanoseconds on each wire between the RDRAM and the master device. The receiver and transmitter blocks also contain multiplexing and storage hardware to permit the internal RDRAM data paths to operate at the slower clock rate (but equivalent bandwidth) of eight bytes transferred every 13.3 nanoseconds (four clock cycles).

The RDRAM also contains control logic and configuration registers. The registers are read and written using special register space commands and control various aspects of RDRAM operation as described on Page 17.

The remaining logic consists of a standard DRAM memory core and row sense amplifier caches.

RDRAM Operation

The RDRAM consists of one bank of memory, this bank storing a full 1 Mbyte of data (see Figure 12). This bank has a 2KByte open page associated with it that is built out of sense amplifier arrays. These sense amplifiers hold the last accessed row of their associated bank in the sense amplifiers. This allows further accesses to the same row of memory to result in page hits. With the row already stored in the sense amplifiers, subsequent data can be accessed with very low latency. Each RDRAM added to a system adds an open page to the memory system, helping to increase hit rates.

A page miss results when a row is accessed that is not currently stored as one of the open pages. When this happens, the requesting master is sent a NACK Acknowledge packet indicating the requested row is not yet available. The RDRAM then loads the requested row into the sense amplifiers and waits for the master to submit a retry of the previous request.

The amount of time that is needed before the retry can be serviced depends on whether the data in the open page is clean or dirty(for the Normal RDRAM). The sense amplifiers act as a "write back" cache in that data written to the open page is not written into the actual DRAM cells until the page is closed. If the data in an open page is clean (not previously written) when a new page is requested, the open page does not need to be written back into the DRAM. If the data in an open page is dirty, then additional time must be added to the miss retry delay to account for the write back operation.

In the case of Low Latency, the sense amplifiers act as a "write through" cache. The data in the open page is always clean because it is automatically written-through to the RDRAM core. The clean and dirty retry times are identical. Figure 11 shows an example of a read miss followed by a read hit for a 32 byte memory read operation to a Low Latency RDRAM.
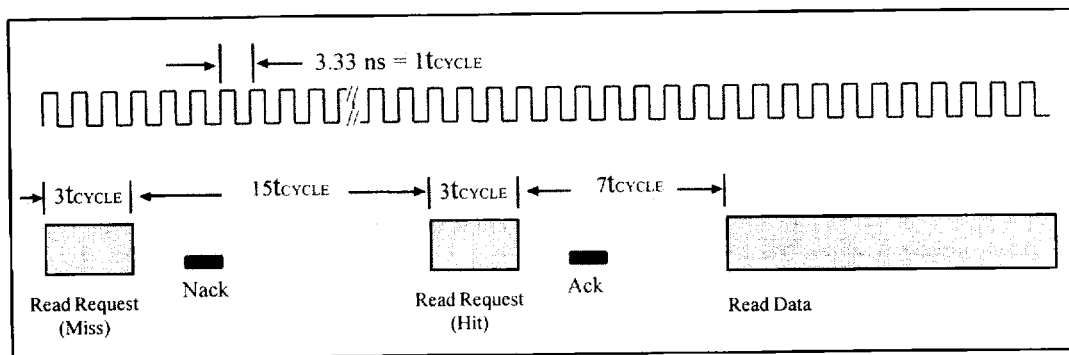
Figure 11. Sample 32-Byte Read Miss and Read Hit Transactions

Address Mapping

Address mapping hardware is provided to increase page hit rates. Adjacent blocks of data (2K or greater) can be separated across several RDRAMs, and therefore across several open pages. This allows a more optimal mapping of the pages as caches and creates higher effective page hit rates. In a typical system containing, for example, eight RDRAMs, hit rates could be expected to be as high as 95%.

Transaction Concurrency

Concurrent transactions can be used to optimize RDRAM utilization in high performance applications by taking advantage of available Channel bandwidth during page miss latency periods. When a miss in one RDRAM takes place, that device will be busy loading a new row into one of its sense amp caches. During this time a transaction to another RDRAM can be scheduled.

Pretouching can be used in system where certain memory accesses are predictable, such as video applications. If the next access a RDRAM is known in advance, a transaction can be first generated that will cause a row miss and prepare the RDRAM for its next access. When the device is next accessed, the required row of data will already be loaded in the open page and a page hit will take place.

Random Access Mode

Non-contiguous blocks of memory can be accessed through the use of the read and write non-sequential (Random Access) operations. With these commands, multiple eight-byte blocks (octbytes) of data within an open page can be accessed in any order. The controller does this using the serial address packet (see Figure 8) to specify the address of each octbyte. Random Access mode can be used to satisfy the burst order of processors like the Intel^{TM} Pentium^{TM} at full RDRAM bandwidth.

Rambus Channel

ClockToMaster

ClockFromMaster

BusEnable, BusCtrl, BusData[8:0]

SIn   SOut   1

RxClk   TxClk

| Receiver | RClk | Clock Generator | TClk | Transmitter |

Request, SMode, SAdr, SCtrl   WriteData[7:0][8:0]   Ack[1:0]   ReadData[7:0][8:0]

37   25   8 Adr[10:3]   64/72   Internal Data Bus   64/72

Op[3:0]
OpX[1:0]   Adr[35:11]
Adr[2:0]
Count[7:0]   Address
Start, Close   Mapping
SMode[1:0]   Logic
SAdr[31:1][10:3]
SCtrl[7:0]

AdrS[35:11]

AddressSelect[3:0][8:0]

Device ID   DeviceId[3:0][8:0]
Compare
DeviceType[3:0][8:0]

Delay[3:0][8:0]

Command and   Mode[3:0][8:0]
Control Logic
DeviceManufacture[3:0][8:0]

RowAddress   Row[3:0][8:0]
Compare
CoreAddress   RefRow[3:0][8:0]
Logic
MinInterval[3:0][8:0]

2   RasInterval[3:0][8:0]
CoreAccess   CoreBankAddr
CoreRestore   1   64/72

CoreColAddr   8   9   MDReg[7:0][8:0]   64/72
CoreRow
Addr   64/72   64/72

64/72   64/72

64/72 x256 RowSenseAmpLatch   64/72 x256 RowSenseAmpLatch

64/72 x256   64/72 x256

64/72 x256x512 DRAM Array - Bank 0   64/72 x256x512 DRAM Array - Bank 1
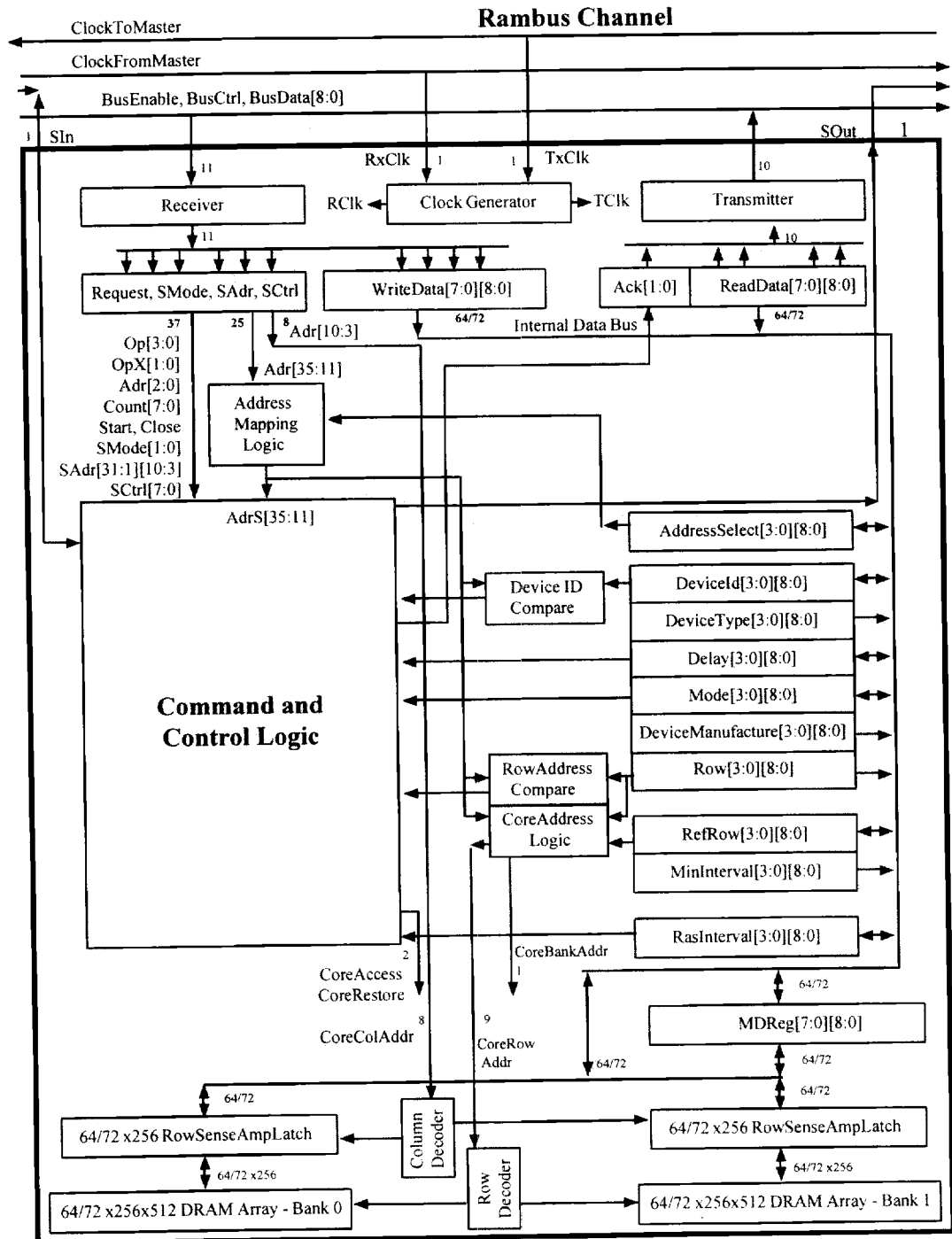
Column Decoder

Row Decoder

Figure 12.   16/18M RDRAM Block Diagram

Bit Masking

Three forms of bit masking are available for memory write operations. These operations are referred to as data-per-bit (Dpb), mask-per-bit (Mpb), and both-per-bit (Bpb) masking. An eight-byte Mask Data register (MDReg) within the RDRAM is used to hold the static value of either mask or data information for these operations. The Mask Data register is an RDRAM internal register that is written by the Bpb commands, and is used by the Mpb and Dpb commands.

With the Dpb operation, the MDReg is used to hold a static mask that is applied to all octbytes of data written to the RDRAM core. With the Mpb operation, the MDReg is used to hold an octbyte of static data that is masked by dynamic bit masks supplied in the data packets before being written to the RDRAM core.

The Bpb operation requires data packets to alternate between mask and data octbytes. The even data packets (starting with data packet 0) carry bit masking information which is placed in the MDReg while the odd data packets carry the data to be masked by the latest contents on the MDReg. This type of operation is also used to set the MDReg for later use in Dpb and Mpb operations.

Table 5 shows the source of the mask and data for each of the write commands. The first eight columns show the Wseq and Wnsq sequential and non-sequential (random-access) write commands. Each has four bit mask sub-commands: Npb,Dpb, Mpb, and Bpb.

A write command consists of writing from one to 32 octbytes of data W[31:0]. Each octbyte of data is masked by an octbyte of bitmask Bit[31: 0]; that is, if the bitmask bit is set, the corresponding bit of write data is written, if the bitmask bit is clear, the bit in memory is left unchanged.

Each of the first eight columns of Table 5 shows the source of the up-to-32 octbytes of write data and bitmask for the eight write commands. The Npb commands use no bitmask at all - effectively the bitmask is all ones. The Dpb command takes a single octbyte of bitmask from the MDReg and applies it to all data octbytes that are written. The Dpb command does the reverse and takes a single octbyte of write data from the MDReg and writes it to each octbyte of memory using a different bitmask from the data packet. The Bpb commands take an octbyte of bitmask and an octbyte of write data alternately, and writes them to a single octbyte of memory.

Byte Masking

Contiguous byte masking is supported by the WseqNpb command. This command uses the Adr[2:0] and Count[2:0] fields of the request packet (Figure 5) to specify the byte masks of the first and last octbytes of a data packet. Refer to the *RDRAM Design Guide* for a more complete description of contiguous byte masking . Table 5 does not show the contiguous byte masking option.

The RDRAM also supports more general form of byte masking called non-contiguous byte masking. This is available with the Wbns write commands.

As can be seen in column nine of Table 5 for the WbnsNpb command, the first data packet, and every ninth thereafter, contain byte masking information that is applied to the eight data packets that follow. This means data packets 0, 9, 18, and 27 (gray boxes in the table) are not written to memory, but are instead used as byte masks for the eight octbytes of data that follow. This means that the data packet Data[35:0] may consist of up to 36 octbytes of information: 4 octbytes of byte mask information and 32 octbytes of data that is actually written to memory.

Each bit of the 64-bit byte mask (one octbyte) controls whether a byte of the following 64 bytes of data (eight octbytes) is written (one) or not written (zero) to memory.

Static bit masking is also available with non-contiguous byte masking. This is shown in the last two columns of Table 5 labelled WbnsDpb and WbnsMpb. The WbnsDpb command applies a static bit held in the MDReg to each octbyte of data W[31:0]. Each bit of the data octbyte is written if the corresponding bits of both the bitmask and bytemask are a one, and is not written if either is a zero. The WbnsMpb command applies a different bitmask Bit[31:0] to a static data octbyte held in the MDReg. Each bit of the data octbyte is written if the corresponding bits of both the bitmask and bytemask are a one, and is not written if either is a zero.

Figure 13 shows how the bits of each Data[i][7:0][8:0] octbyte are flow through the write circuitry as a function of the operation type. The data W[i][j][8:0] is written if the corresponding Bit[i][j][8:0] and Byte[i][j] mask bits are set.
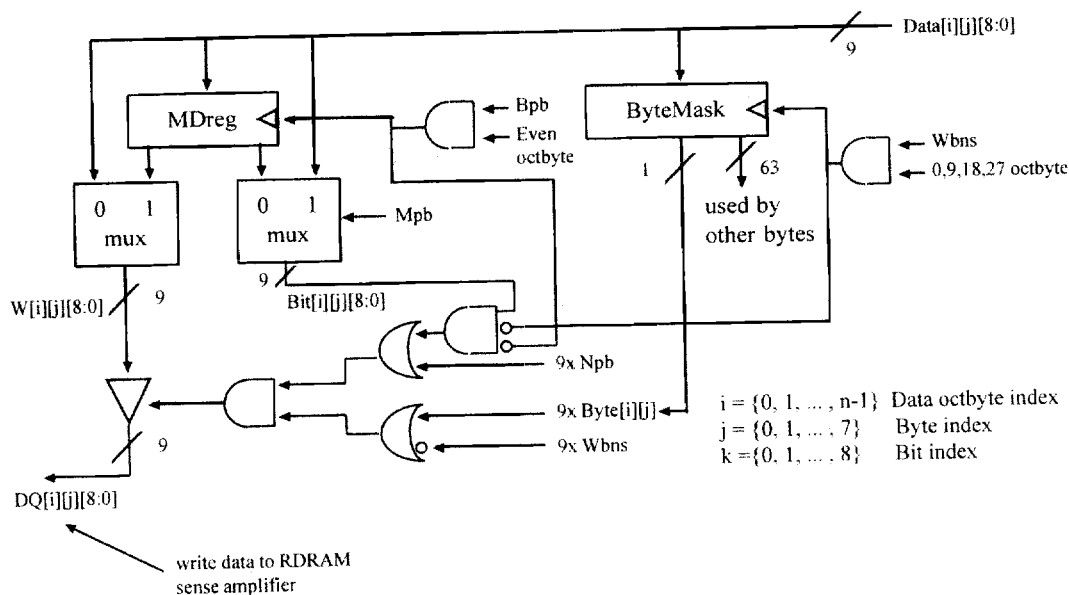

Figure 13. MDReg with BitMask and ByteMask Logic-one byte slice

Table 5: WriteDate, BitMask, and ByteMask Sources [2]

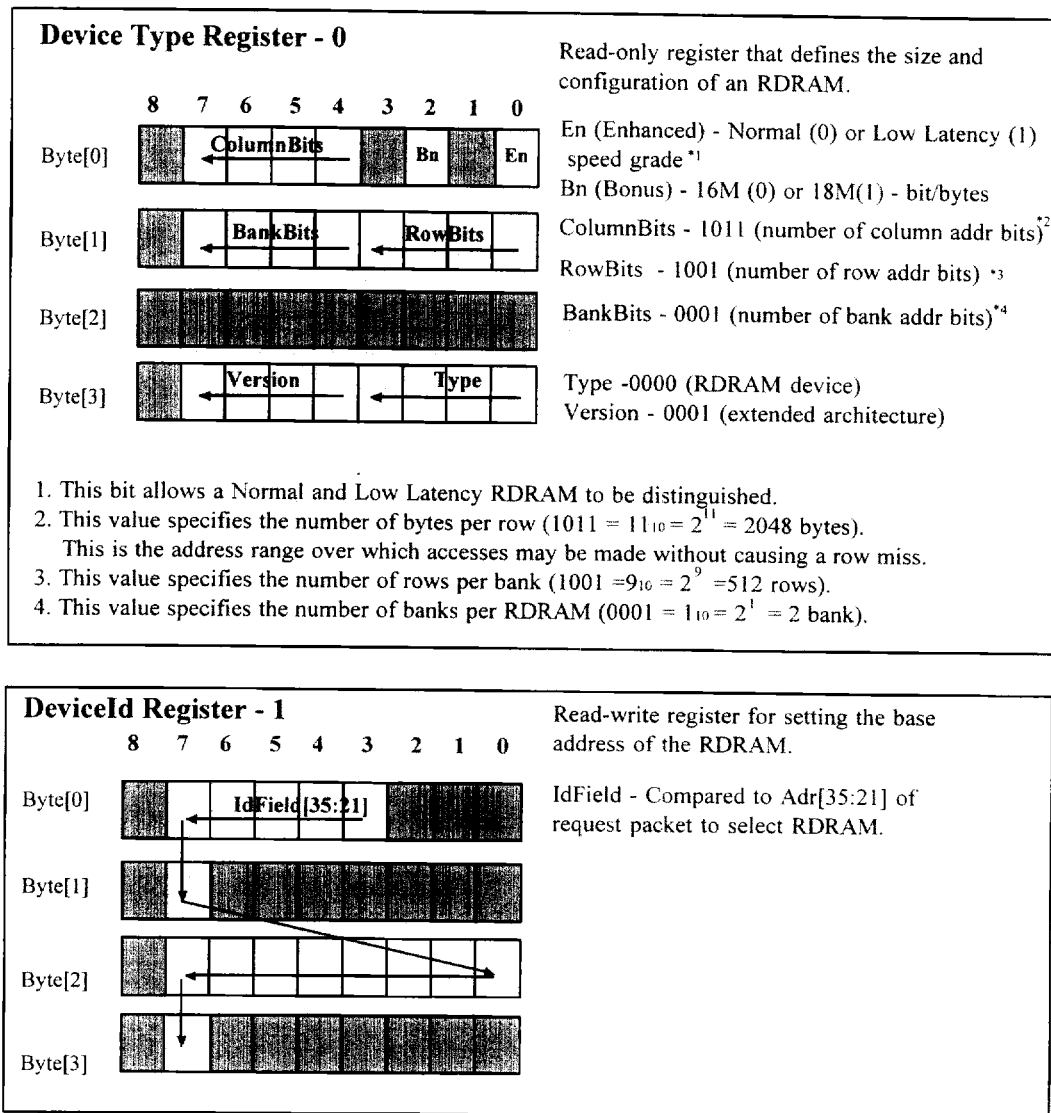| Data Octbyte	Wseq Npb	Wseq Dpb	Wseq Mpb	Wseq Bpb	Wnseq Npb	Wnseq Dpb	Wnseq Mpb	Wnseq Bpb	Wbns Npb	Wbns Dpb	Wbns Mpb
MDReg	-	Bit [35:0]	W [35:0]	-	-	Bit [35:0]	W [35:0]	-	-	Bit [35:0]	W [35:0]
Data[0]	W[0]	W[0]	Bit[0]	Bit[0]	W[0]	W[0]	Bit[0]	Bit[0]	Byte[7:0]	Byte[7:0]	Byte[7:0]
Data[1]	W[1]	W[1]	Bit[1]	W[0]	W[1]	W[1]	Bit[1]	W[0]	W[0]	W[0]	Bit[0]
Data[2]	W[2]	W[2]	Bit[2]	Bit[1]	W[2]	W[2]	Bit[2]	Bit[1]	W[1]	W[1]	Bit[1]
Data[3]	W[3]	W[3]	Bit[3]	W[1]	W[3]	W[3]	Bit[3]	W[1]	W[2]	W[2]	Bit[2]
Data[4]	W[4]	W[4]	Bit[4]	Bit[2]	W[4]	W[4]	Bit[4]	Bit[2]	W[3]	W[3]	Bit[3]
Data[5]	W[5]	W[5]	Bit[5]	W[2]	W[5]	W[5]	Bit[5]	W[2]	W[4]	W[4]	Bit[4]
Data[6]	W[6]	W[6]	Bit[6]	Bit[3]	W[6]	W[6]	Bit[6]	Bit[3]	W[5]	W[5]	Bit[5]
Data[7]	W[7]	W[7]	Bit[7]	W[3]	W[7]	W[7]	Bit[7]	W[3]	W[6]	W[6]	Bit[6]
Data[8]	W[8]	W[8]	Bit[8]	Bit[4]	W[8]	W[8]	Bit[8]	Bit[4]	W[7]	W[7]	Bit[7]
Data[9]	W[9]	W[9]	Bit[9]	W[4]	W[9]	W[9]	Bit[9]	W[4]	Byte[15:8]	Byte[15:8]	Byte[15:8]
Data[10]	W[10]	W[10]	Bit[10]	Bit[5]	W[10]	W[10]	Bit[10]	Bit[5]	W[8]	W[8]	Bit[8]
Data[11]	W[11]	W[11]	Bit[11]	W[5]	W[11]	W[11]	Bit[11]	W[5]	W[9]	W[9]	Bit[9]
Data[12]	W[12]	W[12]	Bit[12]	Bit[6]	W[12]	W[12]	Bit[12]	Bit[6]	W[10]	W[10]	Bit[10]
Data[13]	W[13]	W[13]	Bit[13]	W[6]	W[13]	W[13]	Bit[13]	W[6]	W[11]	W[11]	Bit[11]
Data[14]	W[14]	W[14]	Bit[14]	Bit[7]	W[14]	W[14]	Bit[14]	Bit[7]	W[12]	W[12]	Bit[12]
Data[15]	W[15]	W[15]	Bit[15]	W[7]	W[15]	W[15]	Bit[15]	W[7]	W[13]	W[13]	Bit[13]
Data[16]	W[16]	W[16]	Bit[16]	Bit[8]	W[16]	W[16]	Bit[16]	Bit[8]	W[14]	W[14]	Bit[14]
Data[17]	W[17]	W[17]	Bit[17]	W[8]	W[17]	W[17]	Bit[17]	W[8]	W[15]	W[15]	Bit[15]
Data[18]	W[18]	W[18]	Bit[18]	Bit[9]	W[18]	W[18]	Bit[18]	Bit[9]	Byte[23:16]	Byte[23:16]	Byte[23:16]
Data[19]	W[19]	W[19]	Bit[19]	W[9]	W[19]	W[19]	Bit[19]	W[9]	W[16]	W[16]	Bit[16]
Data[20]	W[20]	W[20]	Bit[20]	Bit[10]	W[20]	W[20]	Bit[20]	Bit[10]	W[17]	W[17]	Bit[17]
Data[21]	W[21]	W[21]	Bit[21]	W[10]	W[21]	W[21]	Bit[21]	W[10]	W[18]	W[18]	Bit[18]
Data[22]	W[22]	W[22]	Bit[22]	Bit[11]	W[22]	W[22]	Bit[22]	Bit[11]	W[19]	W[19]	Bit[19]
Data[23]	W[23]	W[23]	Bit[23]	W[11]	W[23]	W[23]	Bit[23]	W[11]	W[20]	W[20]	Bit[20]
Data[24]	W[24]	W[24]	Bit[24]	Bit[12]	W[24]	W[24]	Bit[24]	Bit[12]	W[21]	W[21]	Bit[21]
Data[25]	W[25]	W[25]	Bit[25]	W[12]	W[25]	W[25]	Bit[25]	W[12]	W[22]	W[22]	Bit[22]
Data[26]	W[26]	W[26]	Bit[26]	Bit[13]	W[26]	W[26]	Bit[26]	Bit[13]	W[23]	W[23]	Bit[23]
Data[27]	W[27]	W[27]	Bit[27]	W[13]	W[27]	W[27]	Bit[27]	W[13]	Byte[31:24]	Byte[31:24]	Byte[31:24]
Data[28]	W[28]	W[28]	Bit[28]	Bit[14]	W[28]	W[28]	Bit[28]	Bit[14]	W[24]	W[24]	Bit[24]
Data[29]	W[29]	W[29]	Bit[29]	W[14]	W[29]	W[29]	Bit[29]	W[14]	W[25]	W[25]	Bit[25]
Data[30]	W[30]	W[30]	Bit[30]	Bit[15]	W[30]	W[30]	Bit[30]	Bit[15]	W[26]	W[26]	Bit[26]
Data[31]	W[31]	W[31]	Bit[31]	W[15]	W[31]	W[31]	Bit[31]	W[15]	W[27]	W[27]	Bit[27]
Data[32]	-	-	-	-	-	-	-	-	W[28]	W[28]	Bit[28]
Data[33]	-	-	-	-	-	-	-	-	W[29]	W[29]	Bit[29]
Data[34]	-	-	-	-	-	-	-	-	W[30]	W[30]	Bit[30]
Data[35]	-	-	-	-	-	-	-	-	W[31]	W[31]	Bit[31]

RDRAM Registers

The 16M/18M RDRAM contains ten registers. These are read and written with the Rreg, Wreg, and WregB commands. They are used to provide configuration information to the RDRAM controller (DeviceType, MinInterval, and DeviceManufacture), to control device, bank, and row addressing(DeviceId, AddressSelect, and Row), to control refresh (RefRow), to control RDRAM timing (Delay and RasInterval), and to control RDRAM operation (Mode). The following table summarizes these functions:

Table 6 : Register Summary

Register Name	Reg. #	Description
DeviceType [3:0][8:0]	0	Read-only register that defines the size and configuration of the RDRAM.
DeviceId [3:0][8:0]	1	Used to specify the base address for the RDRAM.
Delay [3:0][8:0]	2	Used to specify CAS timing parameters.
Mode [3:0][8:0]	3	Used to initialize the RDRAM and set the Iᴏʟ output current.
RefRow [3:0][8:0]	5	Used to specify the next row and bank of the RDRAM to be refreshed.
RasInterval [3:0][8:0]	6	Used to specify RAS timing parameters.
MinInterval [3:0][8:0]	7	Read-only register defining minimum timing parameters for CAS accesses.
AddressSelect [3:0][8:0]	8	Used to specify address bit swapping to maximize RDRAM cache hit rate.
DeviceManufacturer [3:0][8:0]	9	Read-only register containing a manufacturer code.
Row [3:0][8:0]	128	Used to specify the currently sensed row in the back.

The following diagrams show the individual fields of the RDRAM registers. The color of a field denotes its usage: dark-gray is unimplemented, light-gray is read-only, and white is read- write. The arrow within each multi-bit field points from least-significant bit to most-significant bit. Bit and byte numbering use little-endien notation.

Figure 13. Registers

Device Type Register - 0

Read-only register that defines the size and configuration of an RDRAM.

	8	7	6	5	4	3	2	1	0
Byte[0]			ColumnBits				Bn		En
Byte[1]			BankBits			RowBits			
Byte[2]									
Byte[3]			Version			Type			

En (Enhanced) - Normal (0) or Low Latency (1) speed grade [1]

Bn (Bonus) - 16M (0) or 18M(1) - bit/bytes

ColumnBits - 1011 (number of column addr bits) [2]

RowBits - 1001 (number of row addr bits) [3]

BankBits - 0001 (number of bank addr bits) [4]

Type -0000 (RDRAM device)
Version - 0001 (extended architecture)

1. This bit allows a Normal and Low Latency RDRAM to be distinguished.
2. This value specifies the number of bytes per row ($1011 = 11_{10} = 2^{11} = 2048$ bytes).
   This is the address range over which accesses may be made without causing a row miss.
3. This value specifies the number of rows per bank ($1001 = 9_{10} = 2^{9} = 512$ rows).
4. This value specifies the number of banks per RDRAM ($0001 = 1_{10} = 2^{1} = 2$ bank).

DeviceId Register - 1

Read-write register for setting the base address of the RDRAM.

	8	7	6	5	4	3	2	1	0
Byte[0]			IdField[35:21]						
Byte[1]									
Byte[2]									
Byte[3]									

IdField - Compared to Adr[35:21] of request packet to select RDRAM.

Delay Register - 2

Read-write and read-only register to program the CAS access delays.

	8	7	6	5	4	3	2	1	0
Byte[0]					AckWinDelay		AckWinBits		
Byte[1]					ReadDelay		ReadBits		
Byte[2]					AckDelay		AckBits		
Byte[3]					WriteDelay		WriteBits		

AckWinBits- 011 (Number of AckWinDelay bits)
AckWinDelay - Normally set to 101 (5 busclocks)

ReadBits - 011 (Number of ReadDelay bits)
ReadDelay -Normally set to 111 (7 busclocks)

AckBits- 010 (Number of AckDelay bits)
AckDelay - Normally set to 011 (3 busclocks)

WriteBits- 011 (Number of WriteDelay bits)
WriteDelay - Normally set to 001 (1 busclocks)

AckWinDelay adjusts the size of the acknowledge window. ReadDelay, WriteDelay, and AckDelay adjust the time from the end of the request packet to the start of read data, write data, and the acknowledge packets, respectively.

Mode Register - 3
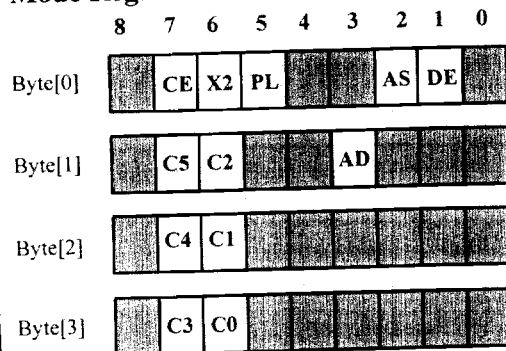
Read-write register for initializing the RDRAM and for controlling operating modes.

	8	7	6	5	4	3	2	1	0
Byte[0]		CE	X2	PL			AS	DE	
Byte[1]		C5	C2				AD		
Byte[2]		C4	C1						
Byte[3]		C3	C0						

DE(DevEn)-Selects RDRAM at initialization
AS - Set to one

PL(PwrLng) - Selects powerdown wake-up time
AD (AckDis) - Acknowledge disable [1]

X2(CCMult) - Set to one. Specifies I_{OL} current
CE (CCEnable) - Specifies manual(0)/auto(1)
                Auto control is normally used.

C[5:0] (CCValue) - Specifies I_{OL} - 63=minimum
                                    0=maximum

1. This bit permits the Low Latency RDRAM to suppress the Acknowledge response when set to one (Not available in Normal RDRAM)

RefRow Register - 5

```
    8   7   6   5   4   3   2   1   0
```

Byte[0] | RowField |

Byte[1] | BankField |

Byte[2]

Byte[3]
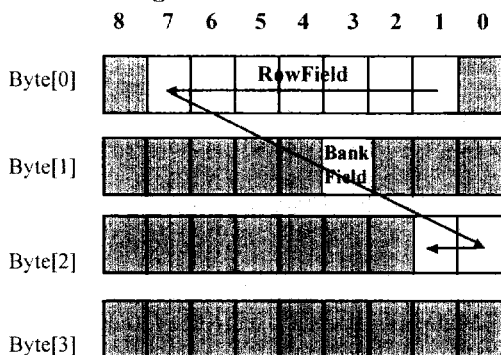
Read-write register for setting the next row refreshed by SetRR burst refresh.

RowField - Next row to be refreshed by SetRR

BankField - Next bank to be refreshed by SetRR

RasInterval Register - 6

```
    8   7   6   5   4   3   2   1   0
```

Byte[0] | RowPrecharge |

Byte[1] | RowSense |

Byte[2] | RowImpRestore |

Byte[3] | RowExpRestore |
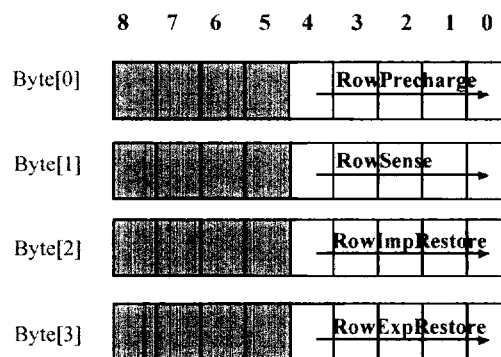
Read-write register to program the RAS access intervals. Note that fields are in bit-reversed order.

RowPrecharge - Set the precharge interval
(Program to 01000 / 10000 / 010000)[1]

RowSense - Set the sense interval
(Program to 01100 / 11100 / 11000)[1]

RowRestore - Set the implicit restore interval
(Program to 10010 / 01010 / 00000)[1]

RowExpRestore - Set the explicit restore interval
(Program to 00100 / 00100 / 00000)[1]

1. Normal 500MHz / Normal 533MHz / Low Latency RDRAM

MinInterval Register - 7

```
    8   7   6   5   4   3   2   1   0
```

Byte[0]

Byte[1] MinAckDly MinReadDly MinWriteDly

Byte[2]

Byte[3] | SpecFunc |
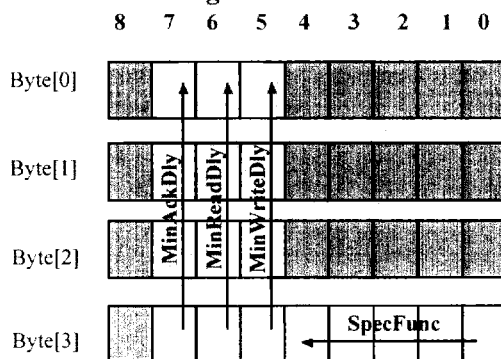
Read-only register (configuration info) and write-only register (special control).

MinWriteDelay - 0001
(minimum WriteDelay of RDRAM)

MinReadDelay - 0111
(minimum ReadDelay of RDRAM)

MinAckDelay - 0011
(minimum AckDelay of RDRAM)

SpecFunc - Performs SetRR burst refresh and SetPD powerdown entry

MinWriteDelay, MinReadDelay, and MinAckDelay specify the minimum number of cycles allowed between a request packet and a write data, read data, and acknowledge packet, respectively. SpecFunc is a write-only field that is used to initiate a SetRR burst refresh or to place the RDRAM into powerdown mode.

AddressSelect Register - 8

```
    8   7   6   5   4   3   2   1   0
Byte[0]            SwapField

Byte[1]

Byte[2]

Byte[3]
```
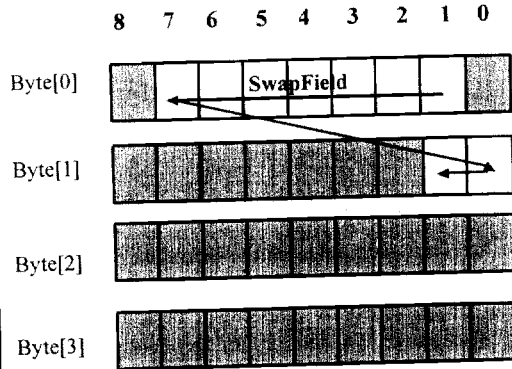
Read-write register for swapping subfields of the Adr field of request packet. This maximizes the rowhit rate for many applications.

SwapField - Each bit swaps a pair of bits from Adr[28:20] and Adr[19:11].

DeviceManufacture Register - 9

```
    8   7   6   5   4   3   2   1   0
Byte[0]         ManufactureCode

Byte[1]         ManufactureCode

Byte[2]         Manufacturer

Byte[3]         Manufacturer
```
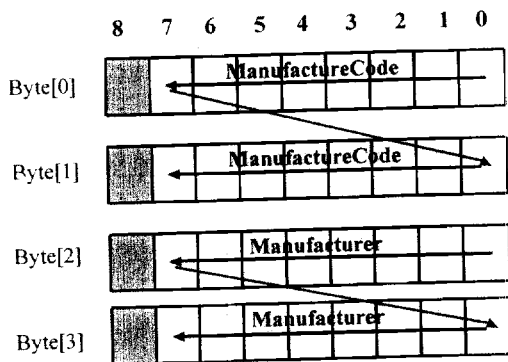
Read-only register with configuration information for RDRAM.

ManufactureCode - Manufacturer specific information.

Manufacture - Contains ID number specifying the manufacturing company.

Row Register - 128

```
    8   7   6   5   4   3   2   1   0
Byte[0]         SensedRow[0]

Byte[1]

Byte[2]         SensedRow[1]

Byte[3]
```
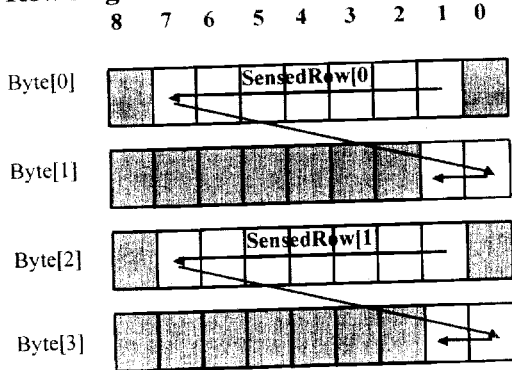
Read-write register with address of currently-sensed row in each bank of RDRAM.

SensedRow[0] - Address of currently sensed row of bank 0.

SensedRow[1] - Address of currently sensed row of bank 1.

Refresh

The RDRAM is a dynamic device, and the memory array must be refreshed every 17 ms (t_{REF}). To support this need, the RDRAM includes all of the logic necessary to support three refresh modes built in. These refresh modes are:

■ Manual Refresh: The Rambus Channel master uses a register write transaction (SetRR) to initiate a single burst refresh of four rows.
■ Touching : A single row is refreshed each time that a read or write request is made to that row in the RDRAM.
■ Powerdown Mode Refresh : A single row is refreshed with each pulse on the SIn/SOut pins.

When the RDRAM is in PowerDown mode, it can be refreshed by passing a periodic pulse at a frequency of 60.2 KHz or greater through the SIn/SOut pins. This minimum frequency is equal to $1024/t_{REF}$.

Operating Modes

The RDRAM has three operating modes; Active, Standby and Power Down. The three modes are distinguished by two factors, their power consumption, and the time that it will take the RDRAM to service a request from that mode.

The control logic within the RDRAM includes a counter that counts Serial Mode packets. It takes a specific number of packets to cause the RDRAM to transition from a low-power mode to the Active state. This counter is active in all three operating modes.

In Active mode, the RDRAM is active and ready to immediately service a request packet. Power consumption is also highest in Active mode.

An RDRAM automatically transitions to Standby mode at the end of a transaction. While in this low power state, each RDRAM monitors the BusEnable signal for a serial mode packet while ignoring other activity on the remaining Channel signals. The Channel master sends a serial mode packet bring all RDRAMs temporarily out of Standby and into Active mode so they can respond to a request packet. Once the request packet is acknowledged, all of the RDRAM return to Standby mode with the exception of the one responding to the request. That device returns to Standby mode once the read or write operation is complete.

Unlike conventional DRAM memory systems where each device in an entire bank of memory must be kept active and consuming power through an entire access, Rambus memory systems use only one active device while all others remain in a lower power state.

Power consumption may be greatly reduced by using the PowerDown mode. This mode is entered manually by setting the Special Function bit *SetPD* in the MinInterval register. Entering this mode causes the device to write back and precharge its cache line, disable the internal clock generator, and disable most DC current sources. The BusEnable receiver is kept active to detect serial mode packets used to exit powerdown mode. The only significant power consumption in powerdown mode is due to refresh.

Since the RDRAM's internal clocks are disabled while in powerdown mode, refresh must be maintained manually by the master device. This is done by supplying a low frequency square wave on the SIn TTL signal. This propagates through each RDRAM and is used to initiate asynchronous refresh operations in each device.

Each RDRAM may be placed in either low or high threshold powerdown mode. Threshold refers to the number of serial mode packets required to wake up the RDRAM. A low threshold requires relatively few serial mode packets while a high threshold requires a larger number. The actual power dissipation is identical in both modes.

An example of where these modes are used is in a portable computer application. In this example, sleep mode is implemented by placing a majority of the RDRAMs in high threshold powerdown while the RDRAM that contains the frame buffer is placed in low threshold powerdown. This permits screen refresh to take place without powering up the entire memory system.
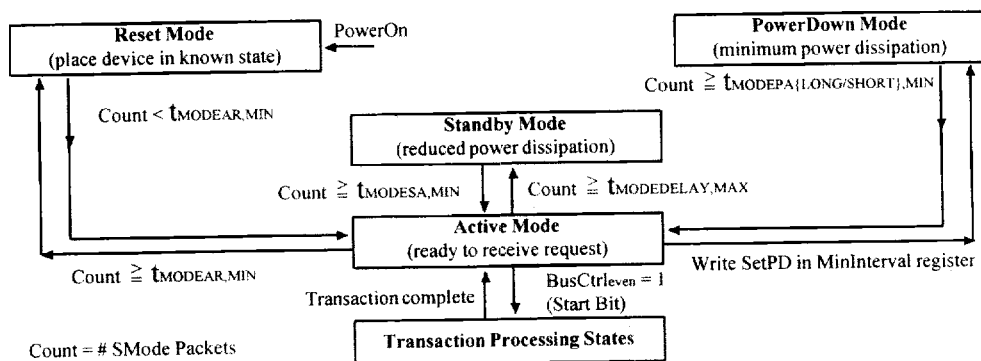
FIGURE 14. RDRAM Operating Modes

Initialization

The Channel master resets the RDRAM devices on the Channel by asserting the BusEnable signal for $t_{MODEAR,MIN}$ clock cycles. After the RDRAM has been reset, the base address and register space registers revert to their default values. Because the default address of all devices is zero until initialization is complete, individual devices cannot be addressed from the Channel, although the devices can and will respond to broadcast commands.

In a typical system application, the SIn pin of the first RDRAM is tied to VDD (refer to Figure 1). The SIn pin of the next RDRAM is connected to the SOut pin of the first RDRAM, and so on. SOut of the last device is then connected to SIn of the Channel master to complete the daisy chain. A RDRAM will not respond to a write command (other than a broadcast write) until the SIn pin is set to 1. Note that if PowerDown mode is to be used by the application, it must be possible to connect a 60.2KHz pulse source to SIn of the first RDRAM to provide refresh.

To start the initialization sequence, the RAC cell in the controller is reset, its DLL is allowed to lock, and its current control register is loaded. Next, the RDRAMs are put into Reset state by asserting the BusEnable wire for $t_{MODEAR,MIN}$ cycle. The $t_{LOCK,RESET}$ interval is observed to allow the RDRAM DLLs to lock.

To start the initialization sequence, the RAC cell in the controller is reset, its DLL is allowed to lock, and its current control register is loaded. Next, the RDRAMs are put into Reset state by asserting the BusEnable wire for $t_{MODEAR,MIN}$ cycle. The $t_{LOCK,RESET}$ interval is observed to allow the RDRAM DLLs to lock.

After the RDRAMs have been placed in Reset state, broadcast writes are made to all control registers needing values different from their Reset values. Next, SIn is asserted high on the first RDRAM in the chain. This enables it. The Channel master then writes the desired device address to the DeviceID register and sets the *DevEn* bit. This asserts the SOut pin (and the SIn pin on the next device in the chain) to 1. Before proceeding to the next RDRAM, these additional steps are taken on the current RDRAM:

1. Current control calibration. The value written into the CCValue field of the Mode register is fine tuned to maximize signal margin. This calibration process must take place before the controller performs any register or memory reads or any acknowledge responses.

2. Check read-only fields of control registers. This confirms which type of Base RDRAMs are present. This also provides an indication of when the end of the Channel is reached.

3. Set the RasInterval register fields. The values used depend upon the type of RDRAM.

4. Touch the RDRAM with eight successive memory read transactions. This settles timing circuitry.

This process continues until all of the RDRAMs have been initialized. When these steps have been completed for every device in the chain, all of the RDRAM devices will have unique, contiguous DeviceID values, and will have their *DevEn* bits set.

Absolute Maximum Ratings

The following table represents stress ratings only, and functional operation at the maximums is not guaranteed. Extended exposure to the maximum ratings may affect device reliability. Furthermore, although devices contain protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

Symbol	Parameter	Min	Max	Unit
$V_{I,ABS}$	Voltage applied to any RSL pin with respect to Gnd	- 0.5	$V_{DD,MAX}+0.5$	V
$V_{I,TTL,ABS}$	Voltage applied to any TTL pin with respect to Gnd	- 0.5	$V_{DD}+0.5$	V
$V_{DD,ABS}$	Voltage on VDD with respect to Gnd	- 0.5	$V_{DD,MAX}+1.0$	V
$T_{J,ABS}$	Junction temperature under bias	- 55	125	℃
T_{STORE}	Storage temperature	- 55	125	℃

Thermal Parameters

Symbol	Parameter and Conditions	Min	Max	Unit
T_j	Junction operating temperature	0	100	℃
Θ_{JC}	Junction-to-Case thermal resistance		5	℃/Watt

Capacitance

Symbol	Parameter and Conditions	Min	Max	Unit
C_I	Low-swing input parasitic capacitance		2	pF
$C_{I,TTL}$	TTL input parasitic capacitance		8	pF

Power Consumption

Mode	Description	Min	Max	Unit
Standby	Device inactive		70 [7]	mA
Active	Device evaluating request packet		230 [7]	mA
Read	Data being transferred from device		360 [7]	mA
Write	Data being transferred to device		330 [7]	mA

Recommended Electrical Conditions

Symbol	Parameter and Conditions	Min	Max	Unit
$V_{DD,VDDA}$	Supply voltage(3.3V version)	3.15	3.45	V
V_{REF}	Reference voltage	1.9	V_{DD} - 0. 8	V
V_{IL}	Input low voltage	V_{REF} - 0. 8	V_{REF} - 0.35	V
V_{IH}	Input high voltage	V_{REF} + 0.35	V_{REF} + 0. 8	V
$V_{IL,TTL}$	TTL input low voltage	- 0.5	0.8	V
$V_{IH,TTL}$	TTL input high voltage	1.8	V_{DD} + 0. 5	V

Electrical Characteristics

Symbol	Parameter and Conditions	Min	Max	Unit
I_{REF}	V_{REF} current @ V_{REF}, MAX	- 10	10	μA
I_{OH}	Output high current @ $(0 < V_{OUT} \leq V_{DD})$	- 10	10	μA
I_0	I_{OL} Output current @ V_{OUT} = 1.6V @ C[5:0] = 111111 (63₁₀)[*8]	0.0	4.0	mA
I_{40}	I_{OL} Output current @ V_{OUT} = 1.6V @ C[5:0] = 011111 (31₁₀)[*8]	36.0	44.0	mA
I_{NONE} (manual)	I_{OL} Output current @ V_{OUT} = 1.6V @ C[5:0] = 111111 (63₁₀)[*9]	0.0	0.0	mA
I_{ALL}(manual)	I_{OL} Output current @ V_{OUT} = 1.6V @ C[5:0] = 000000 (0₁₀)[*9]	40.0	80.0	mA
$I_{I,TTL}$	TTL input leakage current @ $(0 \leq V_{I,TTL} \leq V_{DD})$	- 10.0	10.0	μA
$V_{OL, TTL}$	TTL output voltage @ $I_{OL,TTL}$ = 1.0mA	0.0	0.4	V
$V_{OH, TTL}$	TTL output high voltage @ $I_{OH,TTL}$ = -0.25mA	2.0	V_{DD}	V

Recommended Timing Conditions

Symbol	Parameter	Min	Max	Unit
t_{CR}, t_{CF}	TxClk and RxClk input rise and fall times	0.3	0.8	ns
t_{CYCLE}	TxClk and RxClk cycle times (for 500 /533/ 600 RDRAM)	4.0 / 3.75 / 3.33	4.5	ns
t_{TICK}	Transport time per bit per pin (this timing interval is synthesized by the RDRAM's internal clock generator)	0.5(1.67ns @ t_{CYCLE} = 3.3ns)	0.5(2.25ns @ t_{CYCLE} = 4.5ns)	t_{CYCLE}
t_{CH}, t_{CL}	TxClk and RxClk high and low times	45%	55%	t_{CYCLE}
t_{TR}	TxClk-RxClk differential	0	0.7	t_{CYCLE}
t_{DR}, t_{DF}	Data/Control input rise and fall times	0.3	0.6	ns

Symbol	Parameter	Min	Max	Unit
t_S	Data/Control-to-RxClk setup time	0.35		ns
t_H	RxClk-to-Data/Control hold time	0.35		ns
t_{REF}	Refresh interval		17	ms
$t_{LOCK,RESET}$	RDRAM internal clock generator lock time from Reset mode		$750(3\mu s @ t_{CYCLE} = 4ns)$	t_{CYCLE}
$t_{LOCK, POWERUP}$	RDRAM internal clock generator lock time from PowerUp mode		$750(3\mu s @ t_{CYCLE} = 4ns)$	t_{CYCLE}

Timing Characteristics

Symbol	Parameter	Min	Max	Unit
t_{PIO}	SIn-to-SOut propagation delay @ $C_{LOAD,TTL} = 10 \sim 40pF$	1	25	ns
t_Q	TClk-to-Data/Control output time	$t_{CYCLE}/4 - 0.4ns$	$t_{CYCLE}/4 + 0.4ns$	$t_{CYCLE}/4$ and ns
t_{QR}, t_{QF}	Data/Control output rise and fall times	0.3	0.5	ns

Rambus Channel Timing

The next table shows important timing on the Rambus Channel for common operations.
All timings are from the point of view of the Channel master, and thus have the bus overhead delay of 4ns per bus transversal included where appropriate.

Symbol	Parameter	Min	Max	Unit
$t_{RESPONSE}$	Start of request packet to start of acknowledge packet	6 [10,11]	9 [10]	t_{CYCLE}
$t_{READHIT}$	Start of request packet to start of read data packet for row hit(Okay)	10 [10]	17 [10]	t_{CYCLE}
$t_{WRITEHIT}$	Start of request packet to start of write data packet for row hit(Okay)	4 [10]	11 [10]	t_{CYCLE}
$t_{RETRYSENSED}$ CLEAN (no restore)	Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row had not been written	18/22 [12,13]		t_{CYCLE}
$t_{RETRYSENSED}$ DIRTY (restore)	Start of request packet for row miss (Nack) to start of request packet for row hit (Okay). The previous row had been written	18/30 [12,13]		t_{CYCLE}
$t_{RETRYREFRESH}$ CLEAN (no restore)	Start of request that performs a burst refresh (SetRR) until the start of a request that will not have a Nack acknowledge due to the pending refresh. The previously sensed row had not been written	148/213 [12,13,15]		t_{CYCLE}
$t_{RETRYREFRESH}$ DIRTY (restore)	Start of request that performs a burst refresh (SetRR) until the start of a request that will not have a Nack acknowledge due to the pending refresh. The previously sensed row had been written	148/221 [12,13,15]		t_{CYCLE}

Symbol	Parameter	Min	Max	Unit
$t_{RASAGAIN-CLEAN}$ (no restore)	Start of request packet for row miss (Nack) to start of request packet for row miss (Nack). The previous row had not been written	26/39 *12,13		t_{CYCLE}
$t_{RASAGAIN-DIRTY}$ (restore)	Start of request packet for row miss (Nack) to start of request packet for row miss (Nack). The previous row had been written	26/47 *12,13		t_{CYCLE}
$t_{READBURST32}$	Start of request packet to end of 32 byte read data packet for row hit	26 *14		t_{CYCLE}
$t_{WRITEBURST32}$	Start of request packet to end of 32 byte write data packet for row hit	20 *15		t_{CYCLE}
$t_{READDELAY}$	End of request packet to beginning of read data packet	7 *10	14	t_{CYCLE}
$t_{WRITEDELAY}$	End of request packet to beginning of write data packet	1 *10	8	t_{CYCLE}
$t_{ACKDELAY}$	End of request packet to beginning of acknowledge packet	3 *10,11	6	t_{CYCLE}
$t_{ACKWINDELAY}$	Window in which an acknowledge packet will be sent	5 *10,11	12	t_{CYCLE}
$t_{SERIALREAD-OFFSET}$	Delay from the beginning of a serial address packet or serial control packet to the beginning of the corresponding read data subpacket	13	13	t_{CYCLE}
$t_{SERIALWRITE-OFFSET}$	Delay from the beginning of a serial address packet or serial control packet to the beginning of the corresponding write data subpacket	5	5	t_{CYCLE}
$t_{POSTMEM-WRITEDELAY}$	Delay from the end of the current memory space transaction to the beginning of the next memory space transaction	2 *16		t_{CYCLE}
$t_{POSTREG-WRITEDELAY}$	Delay from the end of the current register space transaction to the beginning of the next register space transaction	4 *16		t_{CYCLE}
$t_{MODEOFFSET}$	Offset from the beginning of SMode packet to request packet for standby to active transaction	4	7/4 *12	t_{CYCLE}
t_{MODESA}	Number of SMode packets to cause a transition from Standby Mode to ActiveMode	1	4/1 *17	t_{CYCLE}
t_{MODEAR}	Number of SMode packets necessary to cause a transition from ActiveState to ResetState	254/288 *18		t_{CYCLE}
$t_{MODEPASHORT}$	Number of SMode packets to cause a transition from PowerDown-Mode[0] to ActiveMode	10/20 *18		t_{CYCLE}
$t_{MODEPALONG}$	Number of SMode packets to cause a transition from PowerDown-Mode[1] to ActiveMode	158/176 *18		t_{CYCLE}
$t_{INTERREQUEST}$	Offset from the beginning of the request packet of the current transaction to the beginning of the request packet of the next transaction	6/8 *12		t_{CYCLE}
$t_{RAS.MAX}$	Time that a row may remained sensed within a bank		125	μs

Notes:

1. A [pin symbol] in this diagram signifies that this pin is not used by this packet. If it used by another packet, it is pulled to a logic zero value.

2. The shade data packet contains byte masking information that is applied to the eight data packets that follow.

3. $t_{MODEARMAX}$ is the maximum number of SMode packets necessary to cause a transition from ActiveState to ResetState.

4. $t_{MODEDELAYMAX}$ is maximum delay (in clock cycles) after a transaction is complete for the RDRAM to enter the StandbyState.

5. $t_{MODESAMIN}$ in the minimum number of SMode packets necessary to cause a transition from StandbyState to AciveState.

6. $t_{MODEPAMIN}$ is the minimum number of SMode packets necessary to cause a transition from PowerDownState to ActiveState.

7. The numbers shown represent "typical" maximum power levels.

8. In auto-calibration mode (CCEnable=1) this is the value written into the C[5:0] field of the Mode register to produce the indicated I_{OL} value. Values of I_{OL} in between the I_0 and I_{40} values are produced by interpolating C[5:0] to intermediate values. For example, C[5:0] = 101000 (40_{10}) produces an I_{OL} in the range of 23.0 to 29.6mA.

9. In manual-calibration mode (CCEnable=0) this is the value written into the C[5:0] field of the Mode register to produce the indicated I_{OL} value. Values of I_{OL} in between the I_{NONE} and I_{ALL} are produced by interpolating C[5:0] to intermediate values. For example, C[5:0] = 101000 (40_{10}) produces an I_{OL} in the range of 14.6 to 29.2mA.

10. Programmable- All RDRAMs will operate across the full progamming range.

11. The acknowledge response may be suppressed in the Low Latency RDRAM with the AD bit in the mode register.

12. The two values shown apply to the Low Latency / Normal speed grades of RDRAM (the AD bit of Mode register is set for $t_{INTERREQUEST} = 6\ t_{CYCLE}$)

13. Minimum at $t_{CYCLE, MIN}$. The delay is programmable to give equivalent timings at longer t_{CYCLE} .

14. Calculated with $t_{READHIT,MIN}$

15. Calculated with $t_{WRITEHIT,MIN}$

16. This is observed after all write transaction when ManufactureCode=0. When ManufactureCode>0, it only applies when a write transaction is followed by a transaction to the same RDRAM. This field is in the DeviceManufacture register.

17. The two values shown apply to RDRAMs with ManufactureCode>0 / ManufactureCode=0. This field is in the DeviceManufacture register.

18. A particular RDRAM will perform the transition at a threshold value within the indicated range; a controller uses the largest value.

Timing Waveforms

Where : $V_{OH,MIN} = V_{TERM,MIN}$
$V_{OL,MAX} = V_{TERM,MAX} - Z_O * (I_{OL,MIN})$
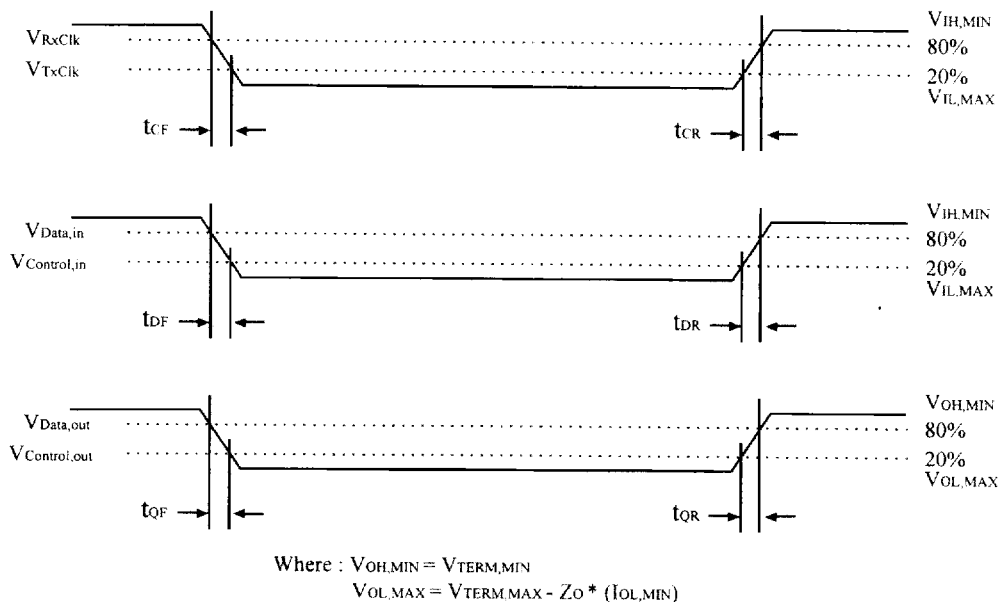
FIGURE 15. Rise/Fall Timing
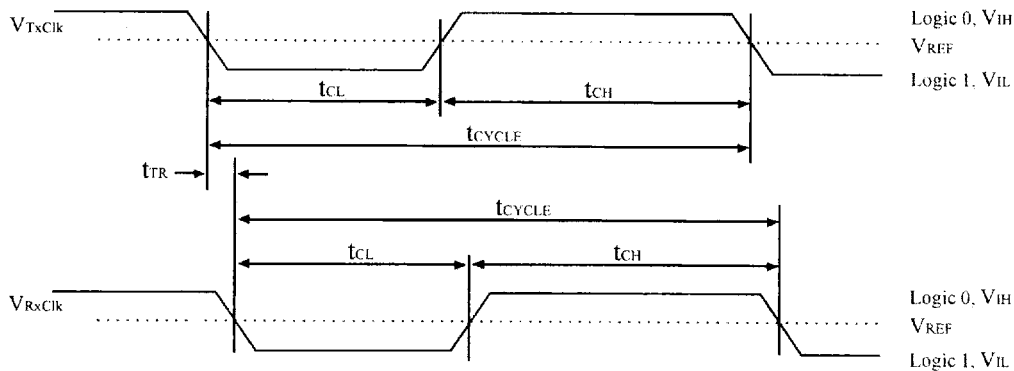
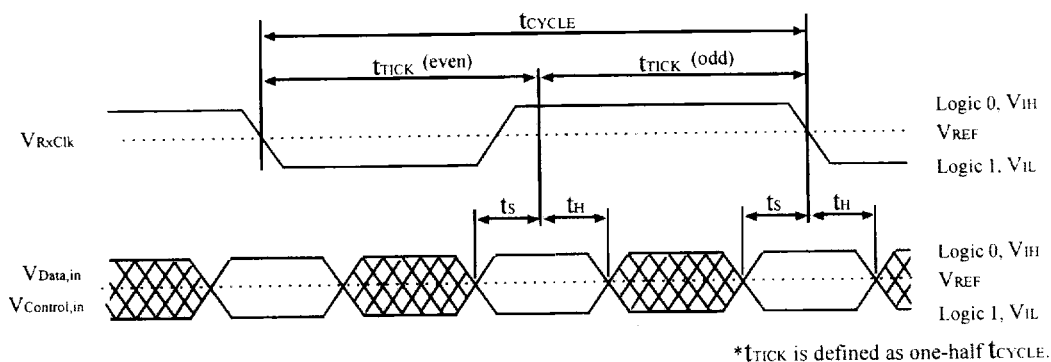

FIGURE 16. Clock Timing
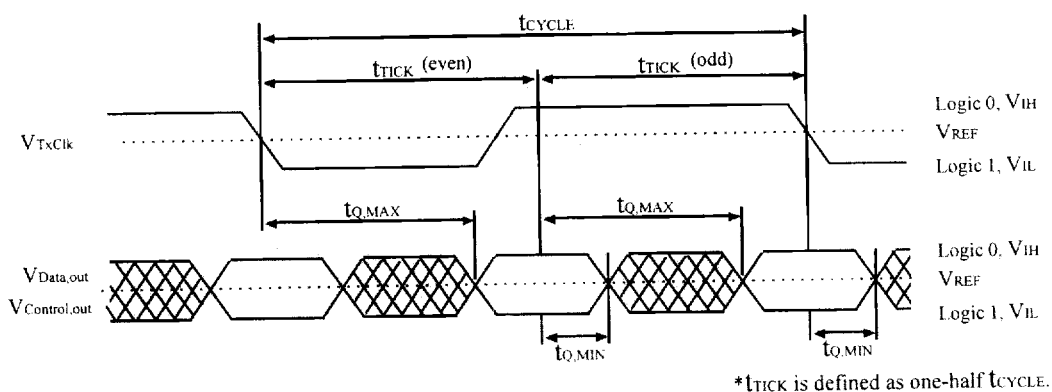
FIGURE 17.   Receive Data Timing
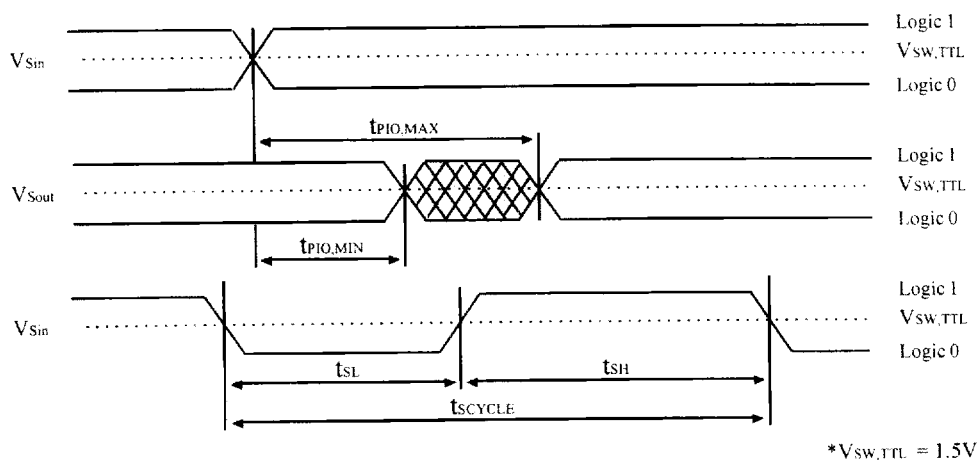

FIGURE 18.   Transmit Data Timing

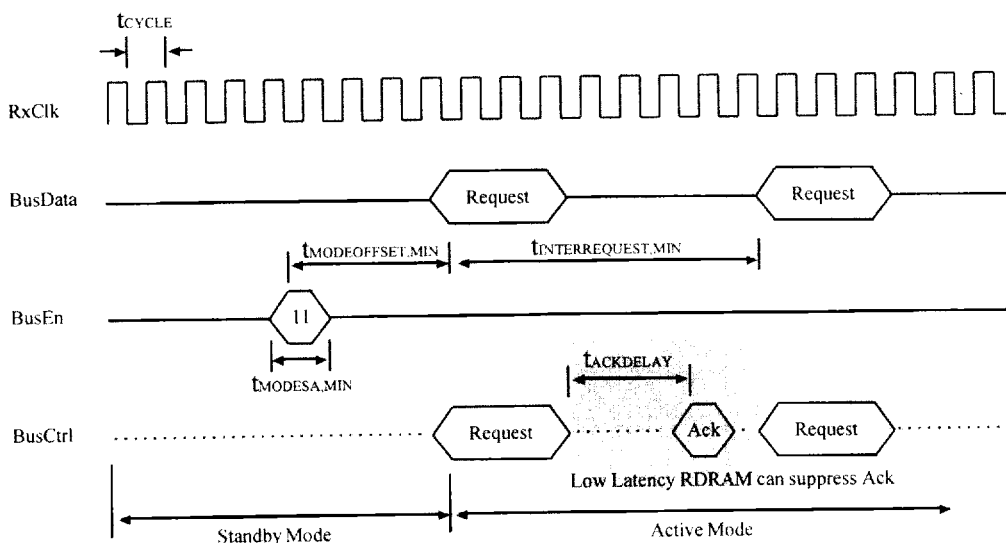**FIGURE 19. Serial Configuration Pin Timing**

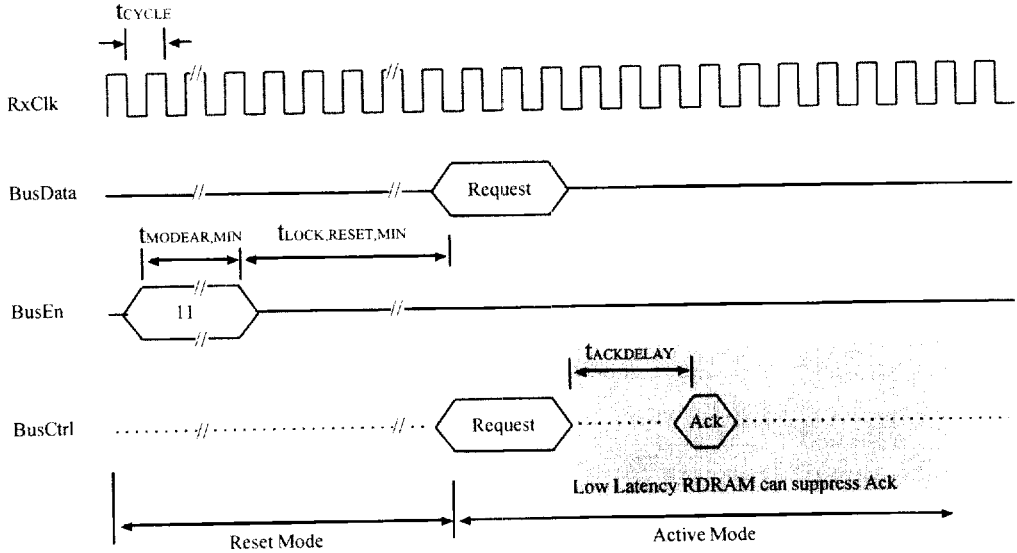**FIGURE 20. Standby Mode to Active Mode Timing and Inter-Request Timing**
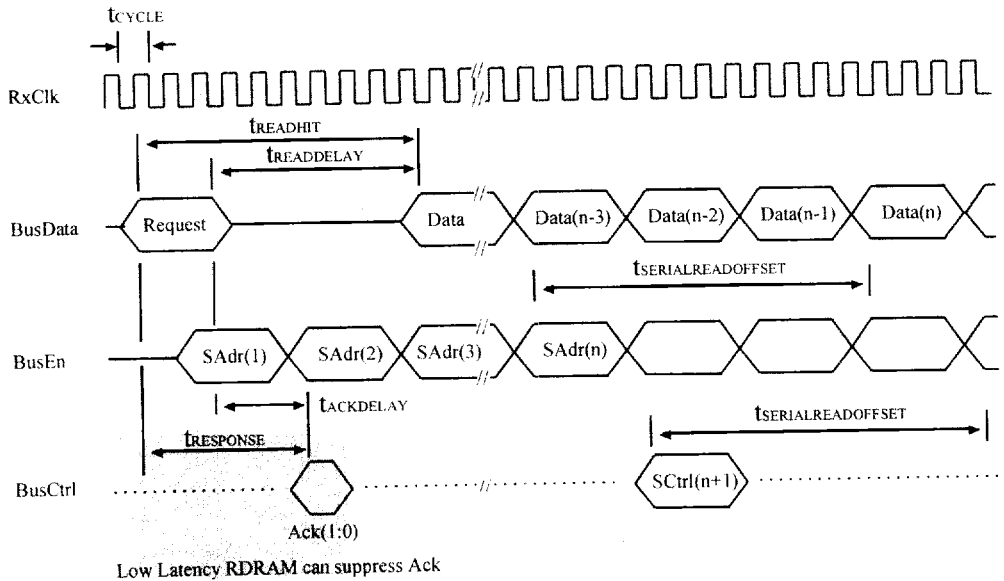
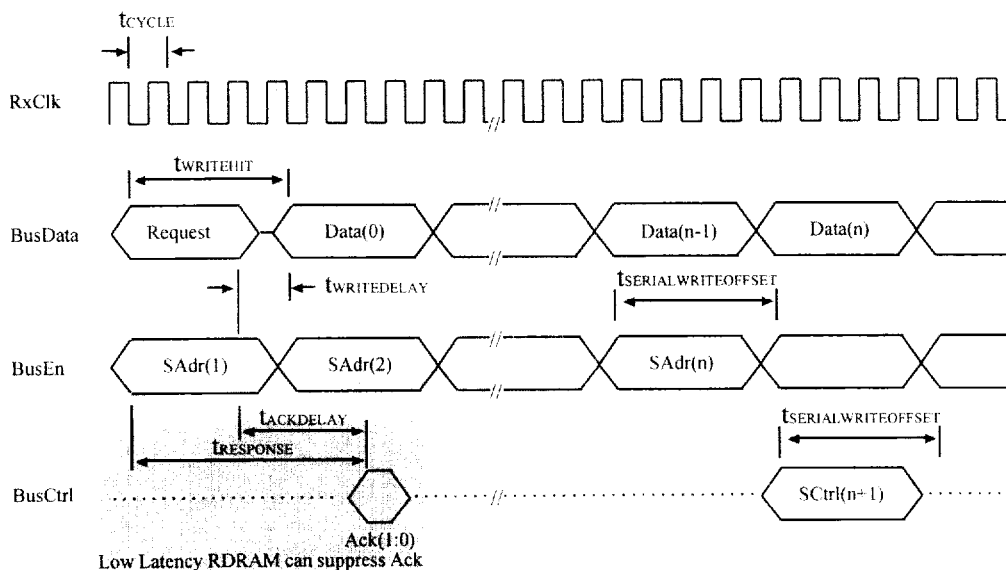**FIGURE 21. Reset Timing**

FIGURE 22. Read Hit Timing Diagram

FIGURE 23.   Write Hit Timing Diagram
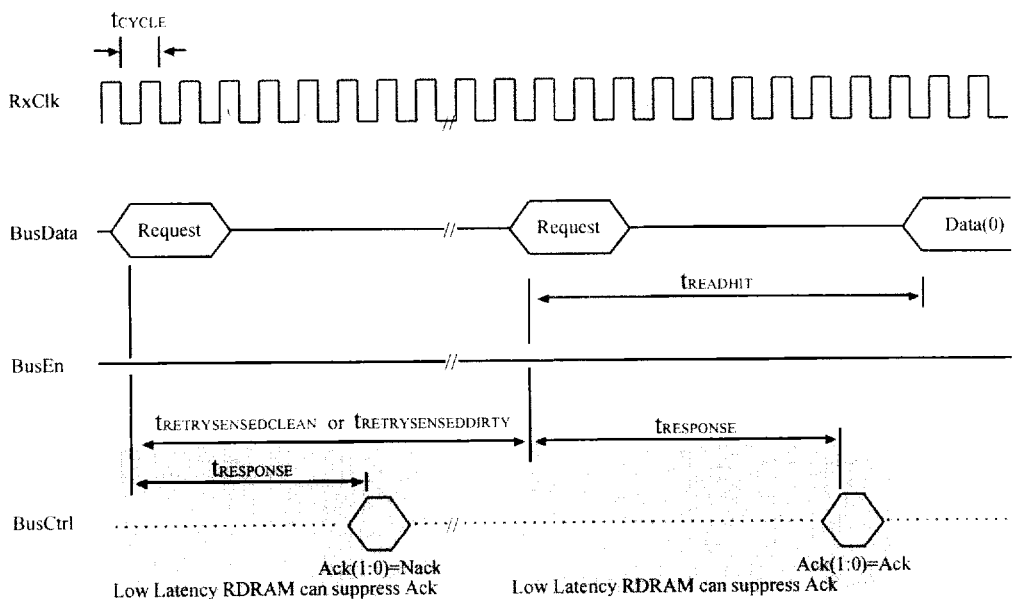
FIGURE 24.   Read Miss Timing Diagram
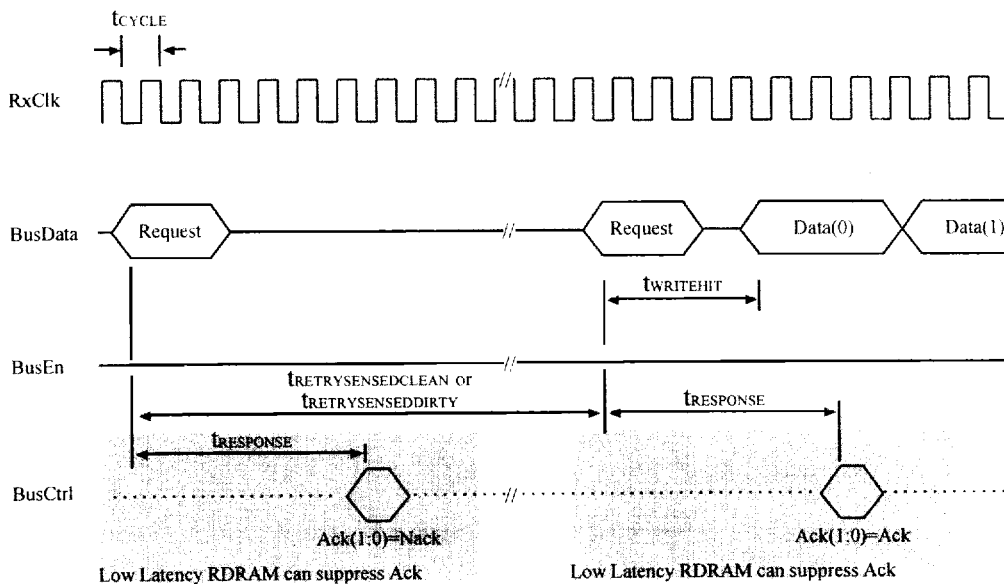
FIGURE 25. Write Miss Timing Diagram
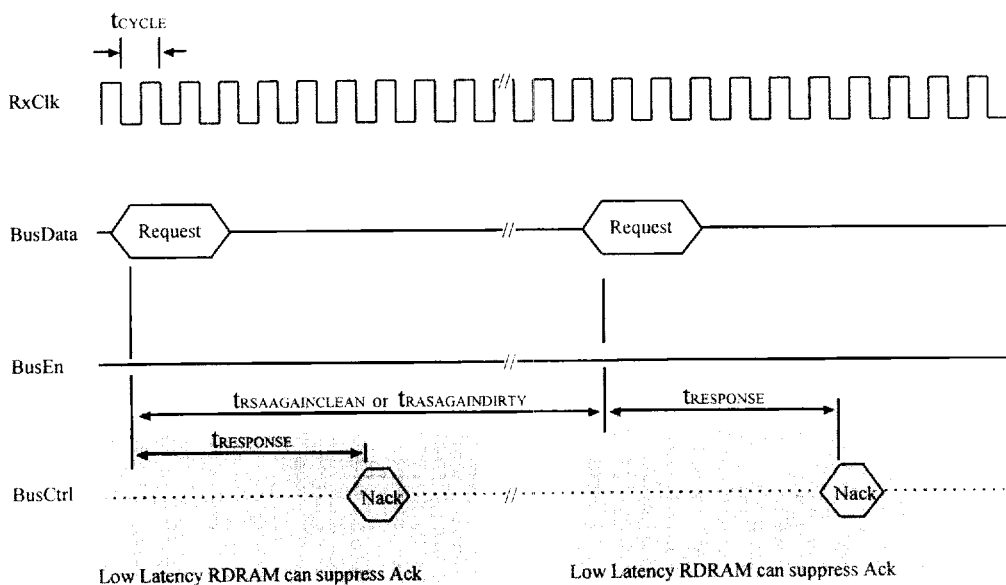
FIGURE 26. RASAgain Timing Diagram
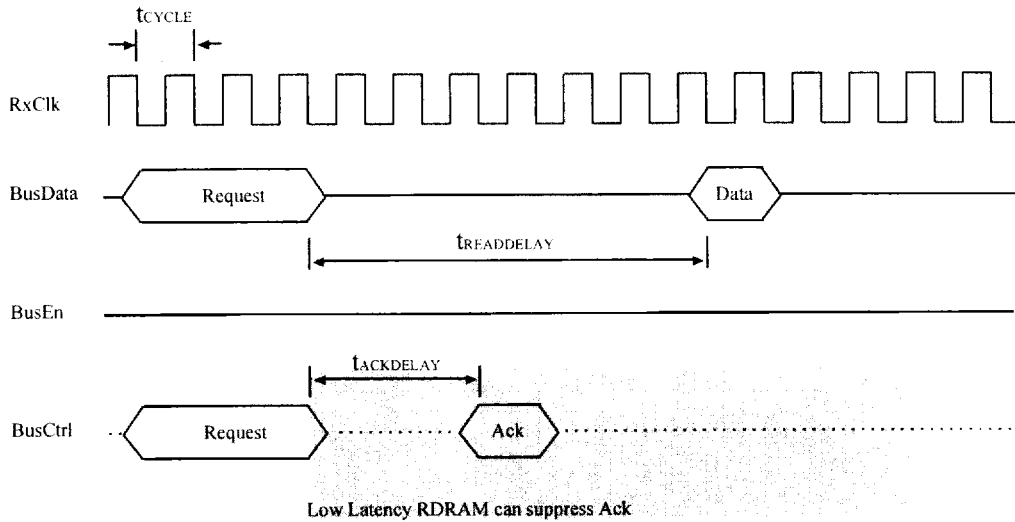
FIGURE 27.   Register Read Timing Diagram
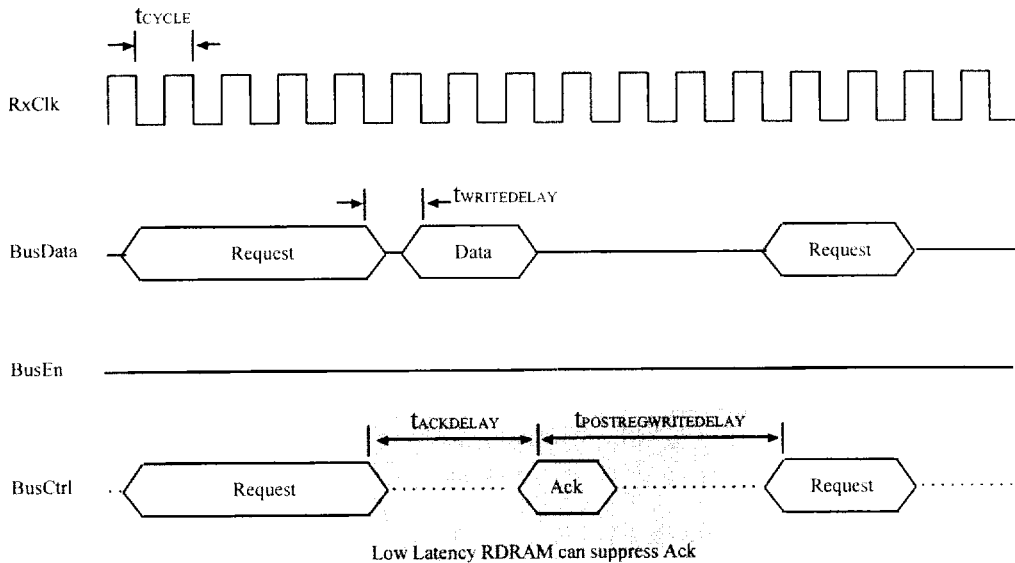
FIGURE 28.   Register Write Timing Diagram

RxClk

t_{CYCLE}

$t_{WRITEDELAY}$

BusData ... Request ... Data ... Request

$t_{RETRYREFRESH}$

BusEn

$t_{ACKDELAY}$

BusCtrl ... Ack

Low Latency RDRAM can suppress Ack
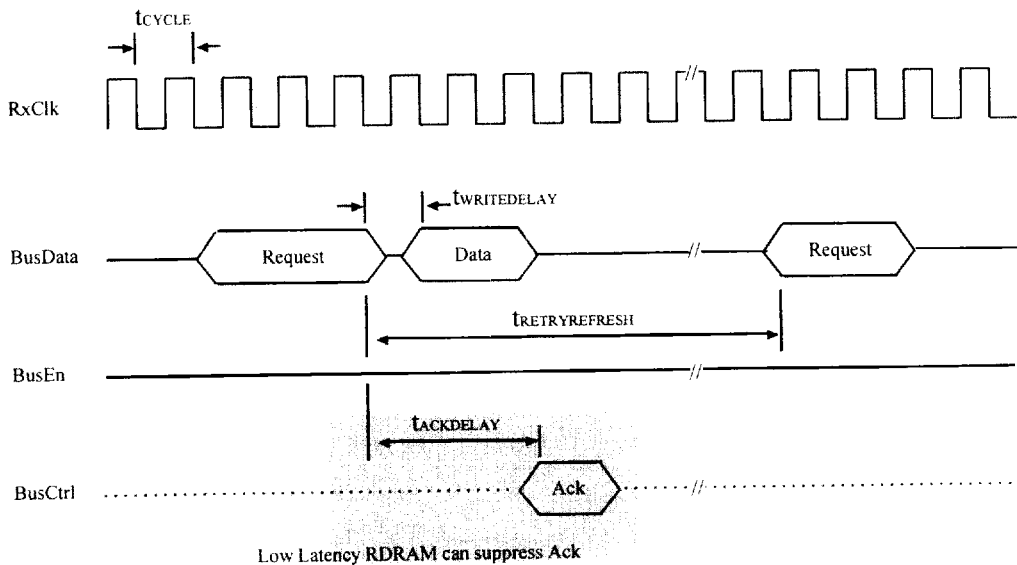
FIGURE 29. Manual Refresh Using SetRR

$t_{REF}/1024$

SIn
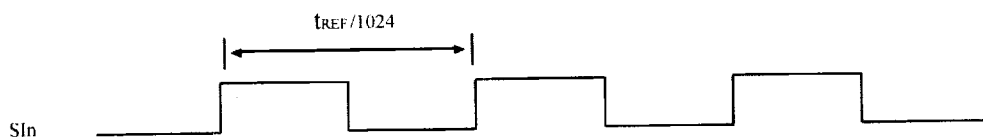
FIGURE 30. Refresh Timing for Powerdown Mode Using SIn, SOut
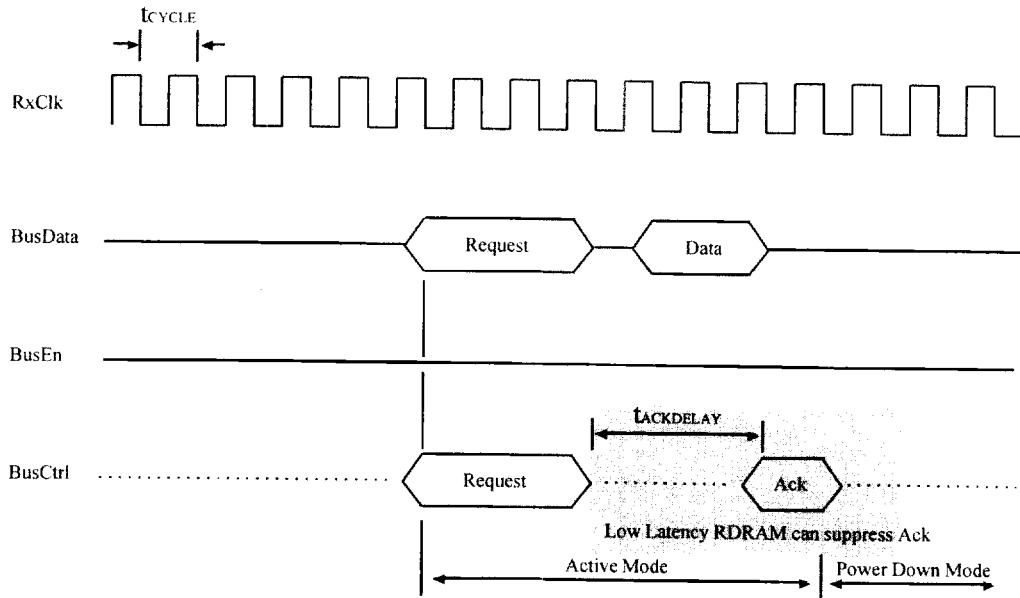
FIGURE 31. Power Down Timing (Register Write)
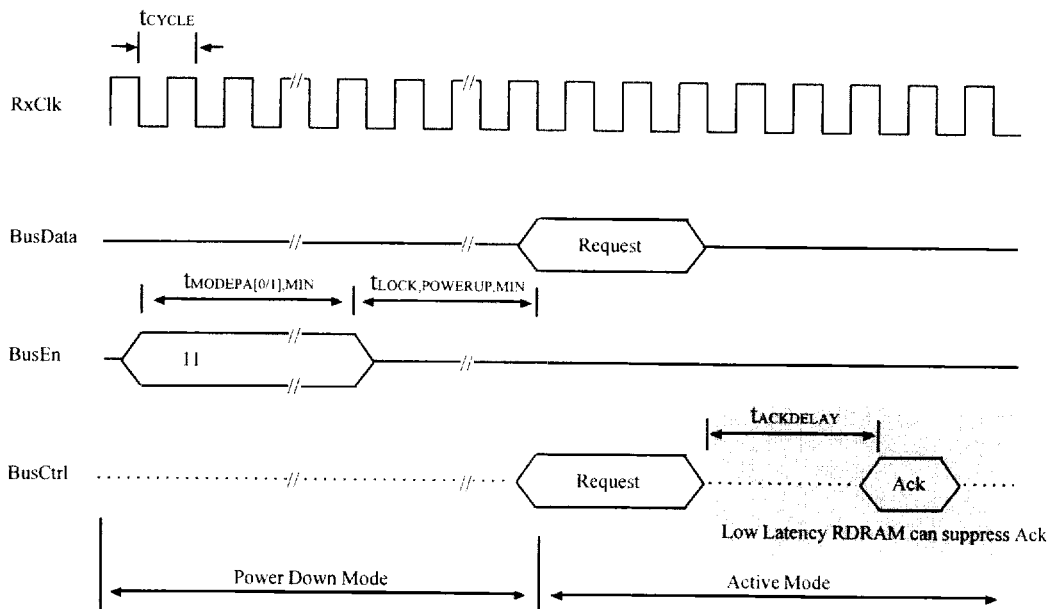
FIGURE 32. Power Up Timing

Mechanical Drawings

The RDRAM is available in horizontal surface mount plastic packages. Dimensions for the Horizontal surface mount plastic package is shown below.
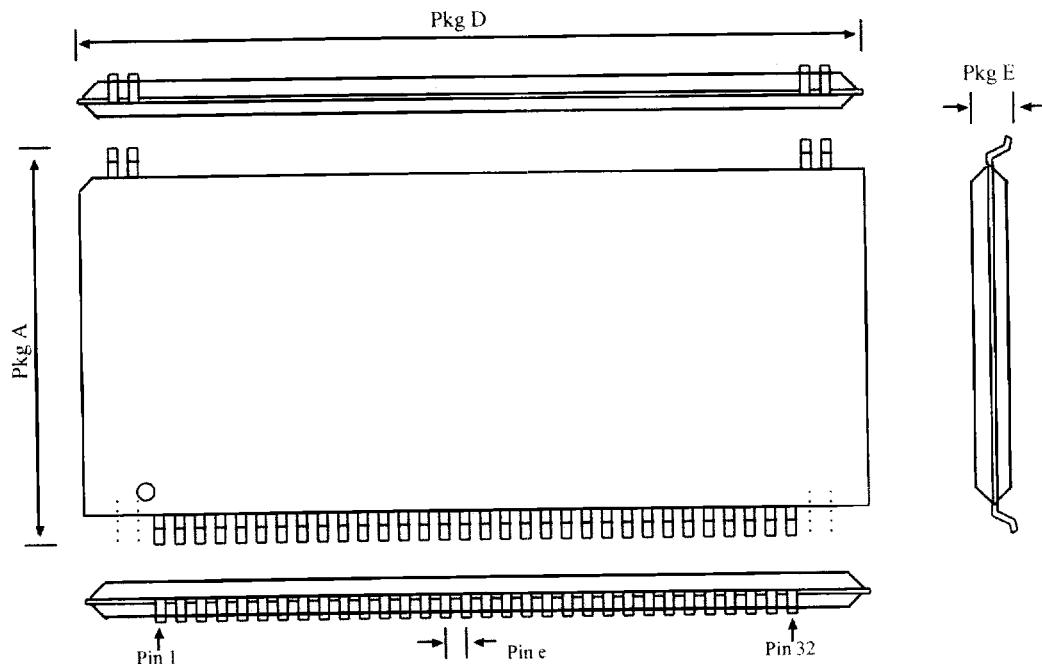
FIGURE 33. SHP-32 Package

The next figure shows the footprint of the SHP-32 package. Plan R-R is the electrical reference plane of the device on the center line of the SMT pads.
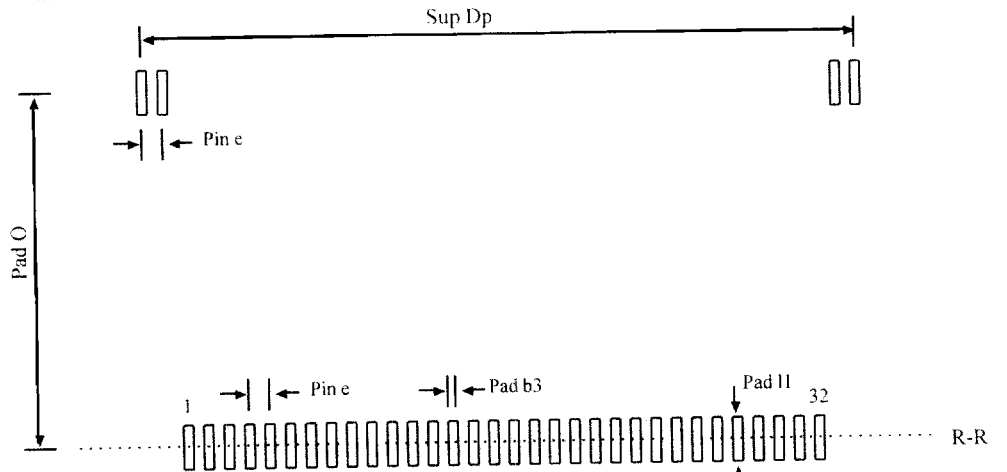
FIGURE 34. SHP-32 Footprint

All pins on 0.65 horizontal grid

This table summarizes the values of the package and footprint dimensions.

Table 8 : SHP-32 Package Dimensions

Symbol	Parameter	Min	Max	Unit
Pin e	Pin Pitch	0.65	0.65	mm
Pkg D [1]	Package Body Length	24.9	25.1	mm
Pkg A	Lead tip to Lead tip Distance	12.9	13.1	mm
Pkg E	Package height	-	1.7	mm
Pad b3	SMT pad width	0.19	0.29	mm
Pad 11	SMT pad length	1.2	1.4	mm
Sup Dp	Support pad outer pitch	22.75	22.75	mm
Pad O	SMT pad offset	12.5	12.5	mm