Galileo Technology®

GT-64014

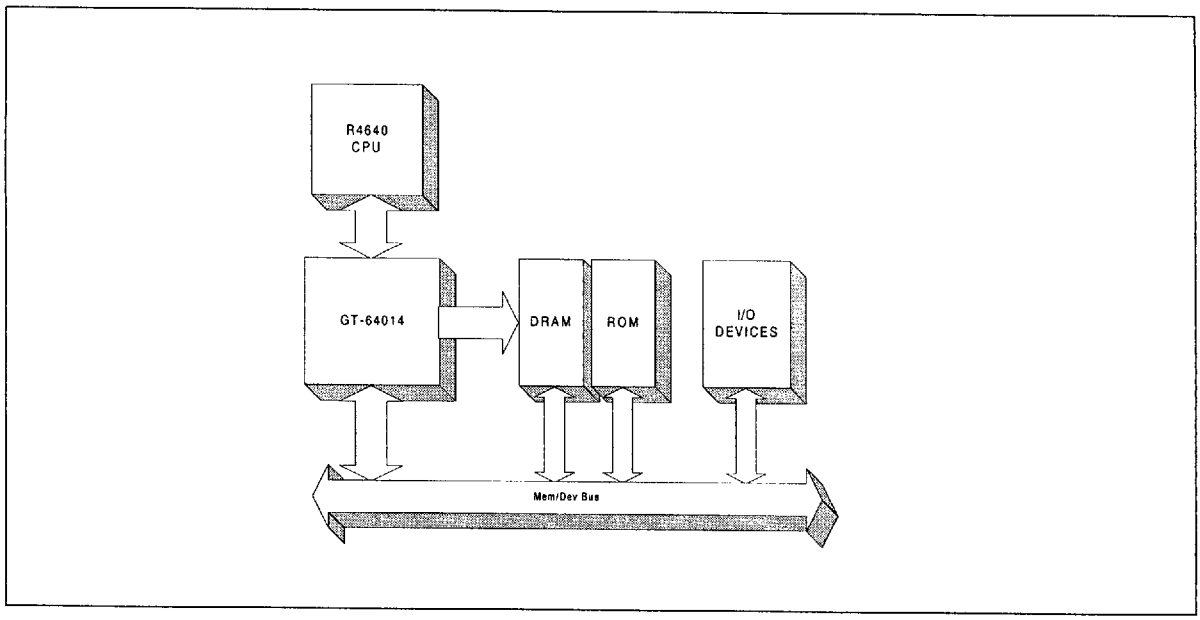
R4640 System Controller

Product Preview
Revision 0.1
3/12/97

FEATURES

Please contact Galileo Technology for possible updates before finalizing a design.

- Low-cost integrated system controller for R4640, R4650 CPUs
- Up to 50MHz CPU bus frequency
- 5V main supply voltage, 3.3V to 5V CPU interface
- 64 byte CPU write buffer
  - 32-bits wide, 16 levels deep
- DRAM controller
  - Page mode and EDO DRAMs
  - 512MB address space
  - 256KB-16MB device depth
  - 4 banks supported directly
  - 32-bit, or 64-bit interleaved data width
  - Different size for each bank
- Three 24-bit and one 32-bit timer/counters
- Boot-ROM chip select
- Device controller
  - 5 chip selects with programmable timing
  - Supports several types of standard memories (ROM / Flash / SRAM) and I/O controllers
  - Up to 160MB address space
  - External wait state support
  - 8-,16- 32, and 64-bit width device support
  - Interleaving supported (64-bit)
- External parity support for user selected banks of DRAM and devices
- DMA controller
  - Four independent channels
  - Chaining via linked lists of records
  - Byte alignment on source and destination
  - Transfers through 32-byte internal FIFO
  - Moves data between memory, and devices
- 208 PQFP



www.galileoT.com info@galileoT.com Tel: (408)451-1400 Fax: (408)451-1404

NOTICE: This is a Product Preview Data Sheet

This document is a PRODUCT PREVIEW data sheet. The specifications in this document are subject to change. DO NOT BASE A FINAL DESIGN on the information in this document.

1. FUNCTIONAL DESCRIPTION

1.1 Overview

The GT-64014 is a highly integrated system controller that supports high-performance embedded control applications with state-of-the-art 32-bit IDT processors, while significantly reducing their cost, complexity, device count, and board space.

The architecture of the GT-64014 supports several system implementations for different applications and cost/performance points. It is possible to design a powerful system with minimal glue logic, or add commodity logic (controlled by the GT-64014) for differentiated system architectures that attain higher performance.

The GT-64014 is software compatible with Galileo's GT-64010A PCI System Controller for R4600/4650/4700/5000 CPUs and Galileo's GT-64011 PCI System Controller for R4640 CPUs.

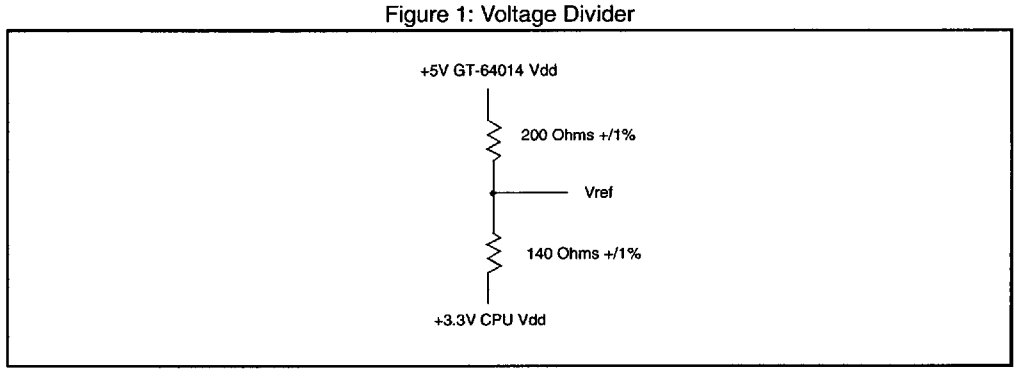
1.2 Processor Interface

The GT-64014 supports without glue logic the low-cost IDT R4640 processor and bus compatible devices. The GT-64014 supports bus frequencies of up to 50MHz, while the processor can operate internally at 80 to 150MHz. The GT-64014 has a 32 byte, 16 level deep write buffer that can absorb up to four CPU write transactions. The processor interface supports the big and little endian options of the CPU.

The GT-64014 can also support bus masters on the local bus (SysAD bus) other than the R4640 processor. Reserved encodings in the SysCmd bus have been used by the GT-64014 to support 1, 2, 4 and 8 word bursts as well as subword accesses. These encodings can be used by co-processors on the SysAD bus that require access to GT-64014 controlled devices.

The GT-64014 processor interface voltage swing is controlled by the Vref pin. The voltage on this pin controls the Voh level for all CPU interface pins. For 3.3V CPUs, Vref must be tied to a voltage divider as shown in Figure 1. This voltage divider fixes the Vref voltage at 0.7V above the 3.3V CPU bus voltage. Galileo recommends this voltage divider design as it minimizes noise sensitivity in the system.

Figure 1: Voltage Divider



1.3 Address Space Decode

The GT-64014 address space decode is programmable. The system resources are divided into eight groups: RAS[1:0], RAS[3:2], CS[2:0], CS[3] & BootCS, and Internal. Each group can have a minimum of 2 Mbytes and a maximum of 256 Mbytes of address space. The individual devices in the device groups (e.g. RAS[0]) are further sub decoded to 1 Mbyte resolution. The sub decoding is performed in the Device Controller unit. The system resources groups can be mapped into a 64 Gbyte address space for CPU accesses and into 4 Gbyte address space for DMA accesses.

When the CPU tries to access an address that is not supported, the GT-64014 will latch the address into the Bus Error registers, and will issue a bus error (over SysCmd[5]) if the access was a read access, and an interrupt if it was a read or write access.

1.4 Parity Support

Memory or device parity generation and checking is supported with external logic. The external logic should generate parity in write accesses to devices and memory and check parity in read accesses. When a parity error is detected by the external logic it needs to drive the ParErr* pin of the GT-64014. The GT-64014 has a programmable parity integrity bit for each bank, which indicates if parity is supported.

In CPU read accesses from a 32-bit device or memory, the GT-64014 will not assert SysCmd[4] even if the bank that was accessed has the parity integrity bit set. If a parity error is detected in this case (indicated by ParErr*), the GT-64014 will return the data with SysCmd[5] asserted and will cause a parity error interrupt.

In DMA read accesses, detection of a parity error from a bank with the parity integrity bit set, will cause an interrupt.

1.5 Memory Control

All memory and I/O devices in a GT-64014 system are connected to the AD bus; the SysAD bus is used primarily as a point-to-point connection between CPU and chipset. The GT-64014 AD bus can support 8-, 16-, and 32-bit devices. 64-bit interleaving is also supported on the AD bus to improve performance.

1.6 DRAM Controller

The GT-64014 has a flexible DRAM controller that supports EDO as well as standard page mode DRAMs. The depth of the DRAM devices can vary for each bank separately from 256K to 16M, and the width of each bank may be 32 bits or 64 bits interleaved. With these options, each DRAM bank size can vary from 1 Mbyte to 128 Mbytes. Furthermore, 0.5K, 1K, 2K, and 4K refresh DRAMs can be used, as well as asymmetric RAS/ CAS addressing.

Some of the DRAM timing parameters are programmable to allow for different system timing optimizations. RAS-to-CAS delay can be programmed for two or three cycles, and CAS can be LOW for one or two cycles.

DRAM performance in CPU read accesses is 7-1-1-1-1-1-1-1 at 50MHz (when interleaved 64-bit memory is implemented), which means 4 wait states to first data and zero wait states for each additional word, and 7-2-2-2-2-2-2-2 with 32-bit DRAMs. DMA burst accesses can be one per clock for a maximum of 8 consecutive 32-bit words.

The DRAM controller supports different depth devices in each bank allowing a base configuration at manufacturing and field upgrades by end users.

Refresh can be programmed to different frequencies of occurrences by the refresh counter. For example, if the refresh counter is programmed to 0x200, then at 50MHz a refresh sequence will occur every 10uS (20nS x 200). Staggered and non-staggered refresh modes are supported. In staggered mode, the four banks of DRAM will be refreshed with one cycle delay between each bank, while in non-staggered mode all four banks will be refreshed together.

Parity generation and checking is supported externally and is optional for each bank of DRAM or any other device on the memory bus.

1.7 Device Controller

The device controller has programmable timing parameters for each device bank to accommodate different device types (e.g. Flash, SRAM, ROM, FIFOs, I/O Controllers). The devices share the local AD bus with the DRAM, but unlike the DRAM, the devices use the AD bus as a multiplexed address and data bus. In the address phase, the device controller puts on the bus 22-bits of address, four general purpose Chip Select signals (CS[3:0]*), one Boot Chip Select (BootCS*), four DMA Acknowledge signals (DMAAck[3:0]*), and an indication as to whether the access is a read or write (DevRW*).

A bus cycle starts by the assertion of ALE and ADS* for one cycle when a CS* signal and/or a DMAAck* signal are active. The CS* and DMAAck* need to be externally latched and qualified with CSTiming*. The CSTiming* signal will be valid for the programmable number of cycles of the specific CS* that is active.

There are eight byte write signals (EWr[3:0]* and OWr[3:0]*). The write signals can be shaped by specifying the following: the number of cycles from the assertion of ADS* to the first assertion of write, the number of cycles the write pulse is active (LOW) - could be extended by Ready*, and the number of cycles the write signal is non-active between consecutive writes. The timing parameters of the write signals determine the length of active CS* (or DMAAck*) signals, as well as the external latch control and burst address change timing.

For read cycles, a device access time to first data (could be extended by Ready*) and to the following data (could be extended by Ready*) in burst accesses defines the cycle parameters. The access time determines the timing that data will be latched, and when the burst address will change.

The device controller supports up to 8 word burst accesses. The burst address is supported by a three bit wide address bus (BAdr[2:0]) that is different from the multiplexed AD bus. The same bus also supports the packing of data into a 64-bit double word, in reads from devices that are 8-bits to 32-bits wide. Devices that are 8-bits or 16-bits wide only are supported by partial reads (up to 64-bits). The controller supports CPU writes of 1 to 8 bytes to 8-bit or 16-bit wide devices. It supports DMA writes of 1 to 4 bytes to 8-bit or 16-bit wide devices. 64-bit interleaved devices are also supported.

Parity generation and checking is supported externally and is optional for each bank of DRAM or any other device on the memory bus.

1.8 Ready* Support

The Ready* pin is used to extend accesses on the AD bus beyond the internal wait-states. The Ready* pin is sampled on two different occasions: on the last rising edge of the WrActive phase during a write cycle and one clock before the data is sampled to the GT-64014 during both AccToFirst and AccToNext phases. During all other phases Ready* is not sampled by the GT64014.

If Ready* is not asserted during these clocks, the WrActive, AccToFirst or AccToNext phases are extended until Ready* is asserted again. See the timing diagrams added for read and write cycles that are controlled by Ready*

1.9 DMA Controller

The GT-64014 incorporates four high performance DMA engines. Each DMA engine has the ability to transfer data between devices residing on the memory bus. All DMA transfers use an internal 32-byte FIFO for moving data. Data is transferred from the source device into the internal FIFO, and from the internal FIFO to the destination device.

The DMA supports increment/decrement/hold on source and destination addresses independently. The length of each transfer of DMA can be limited to 1, 2, 4, 8, 16 or 32 bytes. Accesses can be non-aligned both in the source and the destination. The DMA can be programmed to move up to 64 KBytes of data in each transaction.

The DMA controller supports chained and non-chained modes of operation. In the non-chained mode the CPU programs the DMA channel for each DMA transaction. In chained mode, the DMA controller programs itself via a linked list of records that is loaded by the DMA controller into the channel's working set when a DMA transaction ends.

The DMA controller can be programmed to assert an interrupt in chained mode at the end of every DMA transaction, or when the Next Pointer Register is Null and Byte Count reaches terminal count. In non-chained mode, the DMA will assert an interrupt every time the Byte Count reaches terminal count.

DMA accesses can be initiated by an external request by asserting one of the four DMAReq[3:0]* pins (Demand mode), or by setting an internal bit in a register (Block mode).

Accesses by the four DMA channels can be prioritized via a programmable arbiter. Channels 0 and 1 are in one group and channels 2 and 3 are in another group. Inside each group, the priority can be fixed so a selected channel number can have a higher priority, or both can have the same priority in round-robin fashion. The same scheme applies between the two groups, they can have fixed or round-robin priority.

1.10 Interrupt Controller

The interrupt controller groups all the internal interrupt sources and asserts an interrupt to the CPU when one or more internal interrupts are asserted. There is one Cause register and two Mask registers. The Cause register has one bit for each interrupt source. If the source asserts an interrupt, its respective bit in the Cause register will be set. This bit can be read by the CPU. The interrupt will be acknowledged by the CPU by resetting its bit in the Cause register (writing zero to the specific bit and one to all the other bits.). Each interrupt source has one mask bit in the CPU Mask register. A zero in the CPU Mask register bit will mask the interrupt from asserting an interrupt to the CPU.

1.11 Timer/Counters

The GT-64014 has three 24-bit and one 32 bit timers/counters. When programmed as a counter, the counter will decrement every clock, will set an interrupt and will stop counting. In the timer mode, it will set the interrupt but will reload to the initial value and continue to count down. The initial value for each timer counter is programmable. Write accesses are done to the timer/counter register but read accesses (from the same address) are directly from the counter outputs.

1.12 Reset Configuration

The GT-64014 must acquire some knowledge about the system before it is configured by the software. Special modes of operation are sampled on Reset in order to enable the GT-64014 to function in consistence with the specific system it is used in. Certain pins must be pulled up or down (4.7Kohm recommended) externally to accomplish this. The following configuration pins are continuously sampled from Rst* assertion until 3 TClk cycles after Rst* is deasserted .

Pin	Configuration Function
Interrupt*:	Endianess
0-	Big endian data format
1-	Little endian data format
DAdr[11:10]:	Device Boot Bus Width
00-	8 bits
01-	16 bits
10-	32 bits
11-	64 bits
DAdr[9]:	LEAdrE/DMAReq[2]* Selection
0-	DMAReq[2]*
1-	LEAdrE
DAdr[8]:	OEB Polarity
0-	Active LOW
1-	Active HIGH
DAdr[7]:	External Latches Presence

Pin	Configuration Function
0- 1-	Latches are present System without latches
DAdr[6]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[5]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[4]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[3]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[2]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[1]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DAdr[0]:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DMAReq[3]*:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DMAReq[1]*/ ParErr*:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)
DMAReq[0]*/ Ready*:	Reserved
0- 1-	Test Mode Normal Operation (must pull up)

Notes:

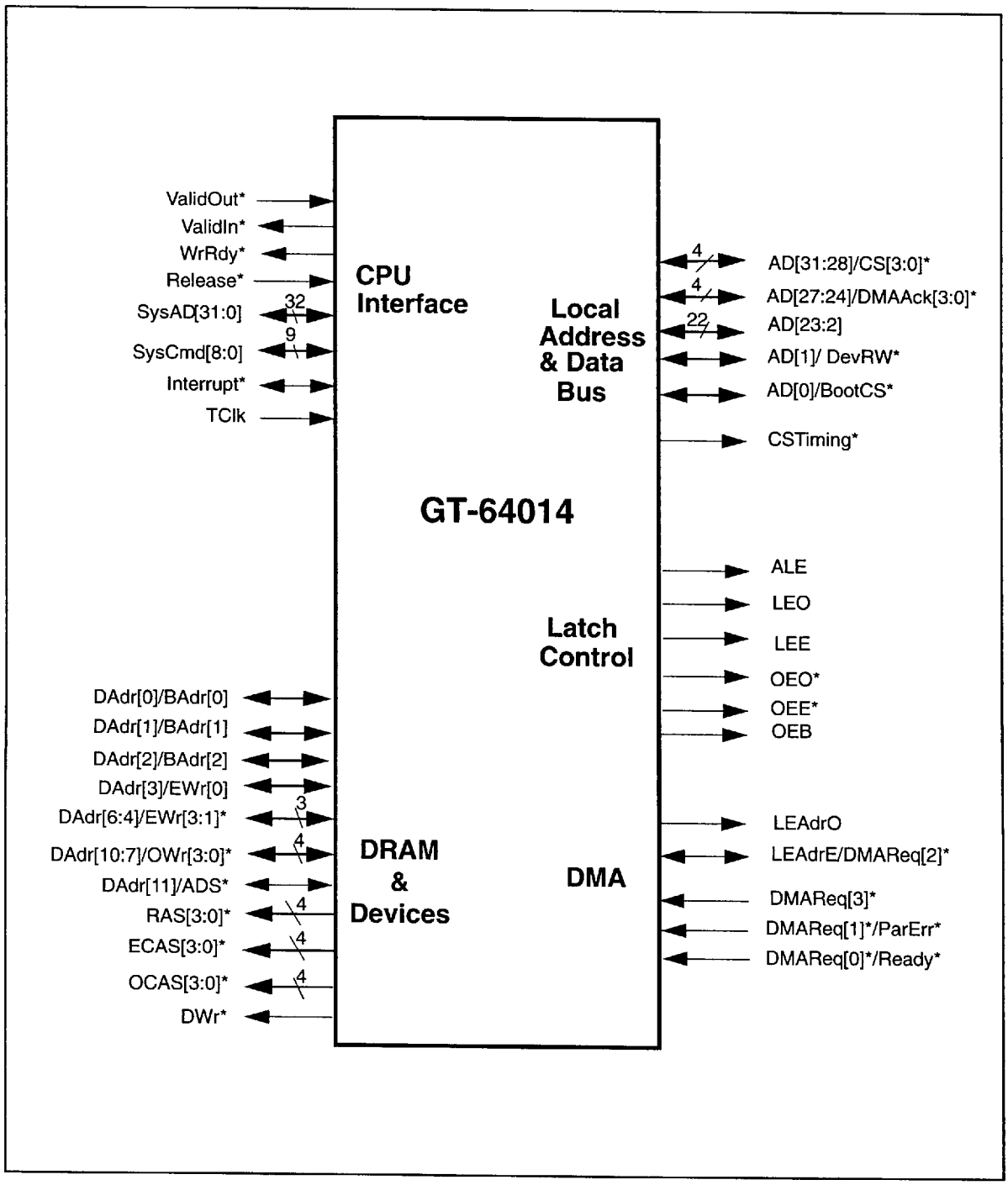
1. LEAdrE/DMAReq[2]* should be selected '0' whenever LEAdrE is not used in the system (e.g., DRAM is not operating in decrement mode).

1.13 Relationship to the GT-64011

The GT-64014 is software and hardware compatible with the GT-64011 PCI System Controller for the R4640 Processor. Software written for the GT-64014 is 100% compatible with the GT-64011. The PCI interface found on the GT-64011 is disabled in early versions of the GT-64014 and will be completely removed in future steppings. Some of the reserved register functions are related to the disabled PCI interface. DO NOT PROGRAM THESE REGISTERS.

2. Pinout and Pin Descriptions

2.1 Pinout



2.2 Pin Descriptions

Pin Name	Type	Description
CPU Interface		
Release*	I	**Release Interface:** Signals to the GT-64014 that the processor is releasing the system interface to slave state.
WrRdy*	O	**Write Ready:** The GT-64014 signals that it can accept a processor write request.
ValidIn*	O	**Valid Input:** The GT-64014 signals that it is driving valid data on the SysAD bus, and a valid data identifier on the SysCmd bus.
ValidOut*	I	**Valid Output:** Signals that the processor is driving valid address or data on the SysAD bus and a valid command or data identifier on the SysCmd bus.
SysAD[31:0]	I/O	**System Address/Data Bus:** A 32-bit address and data bus for communication between the processor and GT-64014.
SysCmd[8:0]	I/O	**System Command/Data Identifier Bus:** A 9-bit bus for command and data identifier transmission between the processor and GT-64014. Some reserved SysCmd encodings are used to support 2, and 4 word bursts for non-R4640 SysAD bus masters.
Interrupt*	I/O	**Interrupt:** An "OR" of all the internal interrupt sources on the GT-64014. This pin is also sampled as an input at reset for configuration purposes.
TClk	I	**Clock:** The input clock to the GT-64014 (up to 50MHz).
Vref	I	**Voltage Reference:** This pin sets the voltage for logical high on the CPU interface pins. For 3.3V CPUs this pin must be tied to a 4V reference voltage (see text.)
DRAM & Devices		
DWr*	O	**DRAM Write:** It is LOW when the GT-64014 writes to the DRAM.
DAdr[0]/BAdr[0]	O	**DRAM Address 0 / Burst Address 0:** This pin has two functions. In an access to a DRAM bank, this pin functions as a DRAM address bit. In write and read accesses from devices that are 8-bit wide, this pin functions as byte address 0 in the packing process of data into 64-bits. In accesses to a word wide (32-bit) device, this bit functions as address 0 in a burst access (equivalent to SysAD[2]). Not used for 16/64 bit devices.
DAdr[1]/BAdr[1]	O	**DRAM Address [1] / Burst Address [1]:** In DRAM accesses, this pin functions as an address bit. In read accesses to devices that are 8-, or 16-bit wide, BAdr[2:1] function as a half word address in the packing process of data into 64 bits. In accesses to a 32-bit bank, BAdr[2:1] function as part of the (two MSB) burst address bits of an address into an eight word line or when packing/unpacking a 64-bit access (equivalent to SysAD[4:3]). In accesses to a 64-bit bank, BAdr[2:1] function as the two burst address bits of a four double word line (equivalent to SysAD[4:3]).

Pin Name	Type	Description
DAdr[2]/BAdr[2]	O	**DRAM Address [2] / Burst Address [2]:** In DRAM accesses, this pin functions as an address bit. In read access to devices that are 8- or 16-bit wide, BAdr[2:1] function as a half word address in the packing process of data into 64 bits. In accesses to a 32-bit bank, BAdr[2:1] function as part of the (two MSB) burst address bits of an address into an eight word line or when packing/unpacking a 64-bit access (equivalent to SysAD[4:3]). In accesses to a 64-bit bank, BAdr[2:1] function as the two burst address bits of a four double word line (equivalent to SysAD[4:3]).
DAdr[3]/EWr[0]*	O	**DRAM Address [3] / Even Bank Byte Write [0]:** In DRAM accesses this pin functions as DRAM address. In device writes it functions as a byte write enable indication to the even bank byte 0.
DAdr[6:4]/ EWr[3:1]*	I/O	**DRAM Address [6:4] / Even Bank Byte Write [3:1]:** In DRAM accesses these pins function as DRAM address. In device writes, they function as byte write enable indications to the even bank bytes [3:1]. These pins are sampled as inputs at reset for configuration purposes.
DAdr[10:7]/ OWr[3:0]*	I/O	**DRAM Address [10:7] / Odd Bank Byte Write [3:0]:** In DRAM accesses these pins function as DRAM address. In device writes, they function as byte write enable indications to the odd bank bytes [3:0]. These pins are sampled as inputs at reset for configuration purposes.
DAdr[11]/ADS*	I/O	**DRAM Address [11] / Address Strobe:** In DRAM accesses this pin functions as a DRAM address. In device accesses it is active for one cycle when the address for the device is on the AD bus. Optionally, this pin is software configurable to only behave as ADS* via bit 17 of the DRAM Configuration register. This pin is sampled as an input at reset for configuration purposes.
RAS[3:0]*	O	**Row Address Select:** Supports four banks of DRAM. The DRAM banks can be 32-(36-) bit or 64-(72-) bit wide.
ECAS[3:0]*	O	**Even Column Address Select:** Supports byte writes/reads to the even bank of the DRAM (when interleaved.) If the bank is not interleaved, ECAS[3:0]* is the same as OCAS[3:0]*.
OCAS[3:0]*	O	**Odd Column Address Select:** Supports byte writes/reads to the odd bank of the DRAM (when interleaved.) If the bank is not interleaved, OCAS[3:0]* is the same as ECAS[3:0]*.
Local AD Bus		
AD[31:28]/CS[3:0]*	I/O	**Data [31:28] / Chip Select [3:0]:** In the data phase, the pins function as data bits [31:28]. In the address phase, Device Chip Selects are valid (and should be latched). The Chip Selects need to be qualified with the CSTiming* signal. Latching is done via ALE.
AD[27:24]/ DMAAck[3:0]*	I/O	**Data [27:24] / DMA Acknowledge[3:0]:** In the data phase, the pins function as data bits [27:24]. In the address phase, DMA Acknowledges are valid (and should be latched). They need to be qualified with the CSTiming* signal. Latching is done via ALE.
AD[23:2]	I/O	**Address/Data[23:2]:** Multiplexed address and data bus to the DRAM (data only) and the devices (address and data).

Pin Name	Type	Description
AD[1]/DevRW*	I/O	**Data [1] / Device Read-Write:** In the data phase it is data bit 1. In the address phase, it indicates if an access to a device is a read ('1') or a write ('0'). Latching is done via ALE.
AD[0]/BootCS*	I/O	**Data [0]/ Boot Chip Select:** In the data phase it is data bit 0. In the address phase, it is the boot device chip select. Latching is done via ALE.
CSTiming*	O	**Chip Select Timing:** Active for the number of cycles that the device that is currently being accessed was programmed to. Used to qualify the CS[3:0]*, BootCS and the DMAAck[3:0]* signals.
Latch Control		
ALE	O	**Address Latch Enable:** Used to latch the Address, BootCS*, CS[3:0]*, DevRW* and DMAAck[3:0]* from the AD bus.
LEO	O	**Latch Enable Odd:** Used to latch data to or from the odd bank devices.
LEE	O	**Latch Enable Even:** Used to latch data to or from the even bank devices.
OEO*	O	**Output Enable Odd:** Output data from the latch of the odd bank to the AD bus.
OEE*	O	**Output Enable Even:** Output data from the latch of the even bank to the AD bus.
OEB	O	**Output Enable Write:** Output data from the latch of the AD bus to the memory bus. This signal is only active during writes to DRAM or devices, and its polarity is programmable at reset.
LEAdrO	O	**Latch Enable Address Odd:** Used to latch the DRAM address and device burst address of the odd bank.
LEAdrE/ DMAReq[2]*	I/O	**Latch Enable Address Even / DMA Request:** Multiplexed signal that can be used to latch the DRAM address and device address of the even bank or, as a DMA request indication by an external device. Its function is designated at reset.
DMA		
DMAReq[3]*	I	**DMA Request[3]:** DMA request indication by an external device. This pin is sampled on Rst* to enable auto-load mode of PCI configuration registers. 0 - Auto-load mode Enabled 1 - Auto-load mode Disabled
DMAReq[1]*/ ParErr*	I	**DMA Request [1] / DMA Parity Error:** DMA request indication by an external device or parity error indication by external logic. The function of this pin is programmable at reset.
DMAReq[0]*/ Ready*	I	**DMA Request [0] / Ready:** This pin has two functions: it serves as a DMA request indication by an external device, or as a cycle extender (when inactive during a device access, an access will extend until Ready* is asserted). The function of this pin is programmable at reset.

3. REGISTER TABLES

3.1 Register Access

The GT-64014's internal registers can be accessed by the CPU via memory-mapping. The registers' address is comprised of the value in the "Internal Space Decode" register and the register Offset. The value in the "Internal Space Decode" register [10:0] is matched against bits [31:21] of the actual address; therefore, this value should be the actual address bits [31:21] shifted right once.

For example, to access "Channel 0 DMA Byte Count" register (offset 0x800) immediately after Reset*, the full address will be the default value in the "Internal Space Decode" register which is 0x0a0 shifted left once, which gives 0x140, two zero's and the offset 0x800, to become a 32-bit address of 0x14000800. The location of the registers in the memory space can be changed by changing the value programmed into the "Internal Space Decode" register. For example after changing the value in the "Internal Space Decode" register by writing to 0x14000068 a value of "0bd", an access to the "Channel 0 DMA Byte Count" register will be with 0x17a00800.

3.2 Register Map

Description	Offset
CPU Interface	
CPU Interface Configuration	0x000
Processor Address Space	
RAS[1:0] Low Decode Address	0x008
RAS[1:0] High Decode Address	0x010
RAS[3:2] Low Decode Address	0x018
RAS[3:2] High Decode Address	0x020
CS[2:0] Low Decode Address	0x028
CS[2:0] High Decode Address	0x030
CS[3] & Boot CS Low Decode Address	0x038
CS[3] & Boot CS High Decode Address	0x040
Reserved 0 Low Decode Address	0x048
Reserved 0 High Decode Address	0x050
Reserved 1 Low Decode Address	0x058
Reserved 1 High Decode Address	0x060
Internal Space Decode	0x068
Bus Error Address Low Processor	0x070
Read Only '0'	0x078
Reserved 2 Low Decode Address	0x080
Reserved 2 High Decode Address	0x088
DRAM and Device Address Space	
RAS[0] Low Decode Address	0x400
RAS[0] High Decode Address	0x404
RAS[1] Low Decode Address	0x408
RAS[1] High Decode Address	0x40c
RAS[2] Low Decode Address	0x410
RAS[2] High Decode Address	0x414
RAS[3] Low Decode Address	0x418
RAS[3] High Decode Address	0x41c
CS[0] Low Decode Address	0x420
CS[0] High Decode Address	0x424
CS[1] Low Decode Address	0x428
CS[1] High Decode Address	0x42c
CS[2] Low Decode Address	0x430
CS[2] High Decode Address	0x434
CS[3] Low Decode Address	0x438
CS[3] High Decode Address	0x43c
Boot CS Low Decode Address	0x440
Boot CS High Decode Address	0x444

Address Decode Error	0x470
DRAM Configuration	
DRAM Configuration	0x448
DRAM Parameters	
DRAM Bank0 Parameters	0x44c
DRAM Bank1 Parameters	0x450
DRAM Bank2 Parameters	0x454
DRAM Bank3 Parameters	0x458
Device Parameters	
Device Bank0 Parameters	0x45c
Device Bank1 Parameters	0x460
Device Bank2 Parameters	0x464
Device Bank3 Parameters	0x468
Device Boot Bank Parameters	0x46c
DMA Record	
Channel 0 DMA Byte Count	0x800
Channel 1 DMA Byte Count	0x804
Channel 2 DMA Byte Count	0x808
Channel 3 DMA Byte Count	0x80c
Channel 0 DMA Source Address	0x810
Channel 1 DMA Source Address	0x814
Channel 2 DMA Source Address	0x818
Channel 3 DMA Source Address	0x81c
Channel 0 DMA Destination Address	0x820
Channel 1 DMA Destination Address	0x824
Channel 2 DMA Destination Address	0x828
Channel 3 DMA Destination Address	0x82c
Channel 0 Next Record Pointer	0x830
Channel 1 Next Record Pointer	0x834
Channel 2 Next Record Pointer	0x838
Channel 3 Next Record Pointer	0x83c
DMA Channel Control	
Channel 0 Control	0x840
Channel 1 Control	0x844
Channel 2 Control	0x848
Channel 3 Control	0x84c
DMA Arbiter	
Arbiter Control	0x860
Timer/Counter	
Timer /Counter 0	0x850

Timer /Counter 1	0x854
Timer /Counter 2	0x858
Timer /Counter 3	0x85c
Timer /Counter Control	0x864
Reserved Internal	
Reserved	0xc00
Reserved	0xc04
Reserved	0xc08
Reserved	0xc0c
Reserved	0xc10
Reserved	0xc14
Reserved	0xc28
Reserved	0xc34
Reserved	0xc3c
Reserved	0xcf8
Reserved	0xcfc
Interrupts	
Interrupt Cause	0xc18
CPU Mask	0xc1c
Reserved	0xc24

3.3 Notes on Reserved Locations

Reserved locations in the memory map are registers used by other GT-6401x family members. These registers may be implemented in some versions of the GT-64014. **Do not attempt program these registers as functionality is not guaranteed.**

Leave all reserved registers at their defualt (unprogrammed) values. The only exception is the Reserved Decode Registers, which may be reprogrammed to suit the target application's memory map (if needed.) Do not program the Reserved Decode Registers to overlapping regions as unpredicatable operation will result.

3.4 CPU Interface Register

The CPU Interface Configuration register determines which of the different MIPS write protocols is supported, as well as CPU endianess. The differences in protocol are minimal and they include support for pipelined write, etc.

CPU Interface Configuration, Offset: 0x000

Bits	Field name	Function	Initial Value
10:0	Reserved	Read Only '0'.	0x0
11	WriteMod	Write mode. 0 - Pipelined writes mode 1 - R4000 mode (2 dead-cycles minimum between consecutive address-phases)	0x0
12	Endianess	Byte Orientation. 0 - Big Endian 1 - Little Endian	Sampled at reset via the Interrupt* pin

3.5 Processor Address Space Registers

The Decode Address registers determine which physical device group will be accessed when the CPU issues an address. The decode to the specific bank (RAS or CAS) in each group is done in the memory control unit. The address decoding is done by comparing bits 31:28 of the address to bits 10:7 of the Low field of all the Low Decode registers to find a match, and by comparing address bits 27:21 to be greater than or equal to bits 6:0 of the Low fields, and less than or equal to the High field. When an address is out of range (of all the Decode Address registers), the CPU will be interrupted during a write and read access and a bus error will be asserted during a read access. The invalid address will be captured in the Bus Error Address Low and High registers. The DMA controller uses the Processor's address decoding.

RAS[1:0] Low Decode Address, Offset: 0x008

Bits	Field Name	Function	Initial Value
10:0	Low	DRAM banks 1 and 0 will be accessed when the decoded addresses are between Low and High.	0x000

RAS[1:0] High Decode Address, Offset: 0x010

Bits	Field Name	Function	Initial Value
6:0	High	DRAM banks 1 and 0 will be accessed when the decoded addresses are between Low and High.	0x07

RAS[3:2] Low Decode Address, Offset: 0x018

Bits	Field Name	Function	Initial Value
10:0	Low	DRAM banks 3 and 2 will be accessed when the decoded addresses are between Low and High.	0x008

RAS[3:2] High Decode Address, Offset: 0x020

Bits	Field Name	Function	Initial Value
6:0	High	DRAM banks 3 and 2 will be accessed when the decoded addresses are between Low and High.	0x0f

CS[2:0] Low Decode Address, Offset: 0x028

Bits	Field Name	Function	Initial Value
10:0	Low	Device banks 2, 1 and 0 will be accessed when the decoded addresses are between Low and High.	0x0e0

CS[2:0] High Decode Address, Offset: 0x030

Bits	Field Name	Function	Initial Value
6:0	High	Device banks 2, 1 and 0 will be accessed when the decoded addresses are between Low and High.	0x70

CS[3] & Boot CS Low Decode Address, Offset: 0x038

Bits	Field Name	Function	Initial Value
10:0	Low	Device bank 3 and the boot bank will be accessed when the decoded addresses are between Low and High.	0x0f8

CS[3] & Boot CS High Decode Address, Offset: 0x040

Bits	Field Name	Function	Initial Value
6:0	High	Device bank 3 and the boot bank will be accessed when the decoded addresses are between Low and High.	0x7f

Internal Space Decode, Offset: 0x068

Bits	Field Name	Function	Initial Value
10:0	IntDecode	Registers inside the GT-64014 will be accessed when SysAD bits 35:21 match the value programmed in bits 14:0.	0x0a0

Bus Error Address Processor, Offset: 0x070

Bits	Field Name	Function	Initial Value
31:0	IlegLoAdd	This register captures bits 31:0 of an illegal 32-bit address.	0x00000000

Reserved, Offset: 0x078

Bits	Field Name	Function	Initial Value
31:0	Reserved	Read Only '0'.	0x0

Reserved 0/1/2 Decode Low, Offset: 0x048, 0x58, 0x80

Bits	Field Name	Function	Initial Value
10:0	Low	Reserved decode regions.	-

Reserved 0/1/2 Decode High, Offset: 0x04C, 0x5C, 0x84

Bits	Field Name	Function	Initial Value
6:0	High	Reserved decode regions.	-

3.6 DRAM and Device Address Space Registers

The Decode Address registers determine which physical device will be accessed when the CPU or DMA issues an address. The address decoding is done by comparing address bits 27:20 to be greater than or equal to the value in the Low fields, and less than or equal to the value in the High fields. In case that no match occurs, an interrupt will be issued and the address causing the error will be latched in the Address Decode Error register. This error can occur when the CPU decoding matches the address while the sub-decoding done in the memory unit doesn't match any of the addresses defined in the address space.

Note: If the value of the "Low Field" is greater than the value of the "High Field", the specified bank is disabled.

RAS[0] Low Decode Address, Offset: 0x400

Bits	Field Name	Function	Initial Value
7:0	Low	DRAM bank 0 will be accessed when the decoded addresses are between Low and High.	0x00

RAS[0] High Decode Address, Offset: 0x404

Bits	Field Name	Function	Initial Value
7:0	High	DRAM bank 0 will be accessed when the decoded addresses are between Low and High.	0x07

RAS[1] Low Decode Address, Offset: 0x408

Bits	Field Name	Function	Initial Value
7:0	Low	DRAM bank 1 will be accessed when the decoded addresses are between Low and High.	0x08

RAS[1] High Decode Address, Offset: 0x40c

Bits	Field Name	Function	Initial Value
7:0	High	DRAM bank 1 will be accessed when the decoded addresses are between Low and High.	0x0f

RAS[2] Low Decode Address, Offset: 0x410

Bits	Field Name	Function	Initial Value
7:0	Low	DRAM bank 2 will be accessed when the decoded addresses are between Low and High.	0x10

RAS[2] High Decode Address, Offset: 0x414

Bits	Field Name	Function	Initial Value
7:0	High	DRAM bank 2 will be accessed when the decoded addresses are between Low and High.	0x17

RAS[3] Low Decode Address, Offset: 0x418

Bits	Field Name	Function	Initial Value
7:0	Low	DRAM bank 3 will be accessed when the decoded addresses are between Low and High.	0x18

RAS[3] High Decode Address, Offset: 0x41c

Bits	Field Name	Function	Initial Value
7:0	High	DRAM bank 3 will be accessed when the decoded addresses are between Low and High.	0x1f

CS[0] Low Decode Address, Offset: 0x420

Bits	Field Name	Function	Initial Value
7:0	Low	Device bank 0 will be accessed when the decoded addresses are between Low and High.	0xc0

CS[0] High Decode Address, Offset: 0x424

Bits	Field Name	Function	Initial Value
7:0	High	Device bank 0 will be accessed when the decoded addresses are between Low and High.	0xc7

CS[1] Low Decode Address, Offset: 0x428

Bits	Field Name	Function	Initial Value
7:0	Low	Device bank 1 will be accessed when the decoded addresses are between Low and High.	0xc8

CS[1] High Decode Address, Offset: 0x42c

Bits	Field Name	Function	Initial Value
7:0	High	Device bank 1 will be accessed when the decoded addresses are between Low and High.	0xcf

CS[2] Low Decode Address, Offset: 0x430

Bits	Field Name	Function	Initial Value
7:0	Low	Device bank 2 will be accessed when the decoded addresses are between Low and High.	0xd0

CS[2] High Decode Address, Offset: 0x434

Bits	Field Name	Function	Initial Value
7:0	High	Device bank 2 will be accessed when the decoded addresses are between Low and High.	0xdf

CS[3] Low Decode Address, Offset: 0x438

Bits	Field Name	Function	Initial Value
7:0	Low	Device bank 3 will be accessed when the decoded addresses are between Low and High.	0xf0

CS[3] High Decode Address, Offset: 0x43c

Bits	Field Name	Function	Initial Value
7:0	High	Device bank 3 will be accessed when the decoded addresses are between Low and High.	0xfb

Boot CS Low Decode Address, Offset: 0x440

Bits	Field Name	Function	Initial Value
7:0	Low	Boot bank will be accessed when the decoded addresses are between Low and High.	0xfc

Boot CS High Decode Address, Offset: 0x444

Bits	Field Name	Function	Initial Value
7:0	High	Boot bank will be accessed when the decoded addresses are between Low and High.	0xff

Address Decode Error, Offset: 0x470

Bits	Field Name	Function	Initial Value
31:0	ErrAddr	The addresses of accesses to invalid address ranges (those not in the range programmed in the DRAM or device decode registers) will be captured in this register.	0xffffffff

3.7 DRAM Configuration Register

The DRAM Configuration register specifies refresh parameters and optional usage of two of the GT-64014 pins related to the DRAM controller. The time between refresh cycles is programmable, with the option to refresh all the banks at the same time or in staggered fashion. The pin functionality of DRAM address bit 11 can be programmed to be ADS* only for systems that do not have deep DRAMs. This pin can also be programmed to be ADS* in device accesses, and to function as DRAM address 11 in DRAM accesses.

DRAM Configuration, Offset: 0x448

Bits	Field name	Function	Initial Value
13:0	RefIntCnt	Refresh interval count value.	0x0200
15:14	Reserved		
16	StagRef	Staggered refresh. 0 - Staggered refresh 1- All banks are refreshed together	0x0

Bits	Field name	Function	Initial Value
17	ADSFunct	Defines the function of the DAdr[11]/ADS* pin. 0 - ADS* in device accesses & DRAM address [11] in DRAM accesses. 1 - ADS* only	0x0
18	DRAMLatch	Sets the latch operation mode. 0 -The latch control signals are active. 1 - The external data latches are transparent in DRAM accesses when CAS is programmed to be one cycle long	0x0

3.8 DRAM Parameter Registers

DRAM timing parameters, bank width, 32-bit wide bank location, parity support, and refresh support for different DRAM sizes, can be set for each DRAM bank independently. The number of cycles CAS* is active (LOW) in read or write accesses can be programmed to one or two cycles. The number of cycles between the cycle RAS* becomes active and the cycle CAS* becomes active in reads or writes is programmable as well.

DRAM Bank0 Parameters, Offset: 0x44c

Bits	Field name	Function	Initial Value
0	CASWr	The number of cycles CAS* is LOW in a write access. 0 - One cycle 1 - Two cycles	0x1
1	RAStoCASWr	The number of cycles between RAS* going active and CAS* going active in a write access. 0 - Two cycles 1 - Three cycles	0x1
2	CASRd	The number of cycles CAS* is LOW in a read access. 0 - One cycle 1 - Two cycles	0x1
3	RAStoCASRd	The number of cycles between RAS* going active and CAS* going active in a read access. 0 - Two cycles 1 - Three cycles	0x1
5:4	Refresh	DRAM type support. 00 - 1/2K Refresh (9 bits row, 9 to 12 bits column) 01 - 1K Refresh (10 bits row, 9 to 12 bits column) 10 - 2K Refresh (11 bits row, 9 to 12 bits column) 11 - 4K Refresh (12 bits row, 9 to 12 bits column)	0x0
6	BankWidth	Width of DRAM bank. 0- 32 (36) bit wide DRAM 1- reserved	0x0
7	BankLoc	Location of a 32-bit wide bank. 0- Even 1- Odd	0x0

Bits	Field name	Function	Initial Value
8	Parity	Parity support for the bank. 0- No parity support 1- Parity supported	0x0
9	Reserved	Must be Programmed '0'.	0x0

DRAM Bank1 Parameters, Offset: 0x450

Bits	Field Name	Function	Initial Value
8:0	Various	Fields function as in DRAM Bank0.	0xf

DRAM Bank2 Parameters, Offset: 0x454

Bits	Field Name	Function	Initial Value
8:0	Various	Fields function as in DRAM Bank0.	0xf

DRAM Bank3 Parameters, Offset: 0x458

Bits	Field Name	Function	Initial Value
8:0	Various	Fields function as in DRAM Bank0.	0xf

3.9 Device Parameter Registers

Device parameters can be different for each bank. The shape of the different control signals that are active in a device access can be programmed. The access time of the device (in number of cycles) during read accesses should be programmed into the AccToFirst field, to set the time that data from the device will be latched into the external latch. AccToNext should be programmed for the time that data from the device can be latched in consecutive accesses during burst accesses. To prevent bus contention, the TurnOff field specifies the number of cycles (from the deassertion of CSTiming*) to the beginning of the next bus transaction. The write signals pulse should be shaped as well. The parameters specify the number of cycles from the beginning of the cycle to the assertion of the write signals (ADSToWr), the number of cycles the write is active (WrActive), and the number of cycles the write signals are inactive (WrHigh) between consecutive writes in a burst access.

Device width can be programmed to 8-, 16-, or 32-bits (default is 32-bits except for the Boot bank). The device controller can pack data during reads from a word that is less than 32-bits wide. For devices that are less then 32-bits, the device can be located in the odd or the even bank. Devices that are 8-bits or 16-bits wide only support partial reads (up to 64-bits).

Performance when reading from a device can be optimized to save a cycle in each access by making the latch transparent for devices that can supply the data to the GT-64014 on time, without the need to be latched.

Device Bank0 Parameters, Offset: 0x45c

Bits	Field name	Function	Initial Value
2:0	TurnOff	The number of cycles between the deassertion of DevOE* (an externally extracted signal which is the logical OR between CSTiming* and inverted DevRW*) to a new AD bus cycle.	0x7
6:3	AccToFirst	The number of cycles in a read access from the assertion of CS* to the cycle that the data will be latched (by the external latches). Can be extended via the Ready* pin.	0xf
10:7	AccToNext	The number of cycles in a read access from the cycle that the first data was latched to the cycle that the next data will be latched (in burst accesses). Can be extended via the Ready* pin.	0xf
13:11	ADStoWr	The number of cycles from ADS* active to the assertion of EWr* or OWr*.	0x7
16:14	WrActive	The number of cycles EWr* or OWr* are active. Can be extended via the Ready* pin.	0x7
19:17	WrHigh	The number of cycles between deassertion and assertion of EWr* or OWr*.	0x7
21:20	DevWidth	Device width. 00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - 64 bits	0x2
22	Reserved	Must be programmed '1'.	0x1
23	DevLoc	32-bit, 16-bit, or 8-bit device location. 0 - Even bank 1 - Odd bank	0x0
24	Reserved	Read only.	0x0
25	LatchFunct	Latch function in read cycles. 0 - Always transparent 1 - Latch enable signals are active.	0x0
27:26	Reserved	Read only.	0x1
29:28	Reserved	Read only.	0x1
30	Parity	Parity support for the bank. 0- No parity support 1- Parity supported	0x0

Device Bank1 Parameters, Offset: 0x460

Bits	Field Name	Function	Initial Value
30:0	Various	Fields function as in Device Bank0.	0x146fffff

Device Bank2 Parameters, Offset: 0x464

Bits	Field Name	Function	Initial Value
30:0	Various	Fields function as in Device Bank0.	0x146fffff

Device Bank3 Parameters, Offset: 0x468

Bits	Field Name	Function	Initial Value
30:0	Various	Fields function as in Device Bank0.	0x146fffff

Device Boot Bank Parameters, Offset: 0x46c

Bits	Field Name	Function	Initial Value
30:0	Various	Fields function as in Device Bank0.	0x14?fffff

In case of the Boot Bank, bits 23:20 are shown as '?' because bits 21:20 are sampled at reset via DAdr[11:10] to define the width of the boot device.

3.10 DMA Record Registers

Each DMA record includes four registers: byte count, source address, destination address, and a pointer to the next record. The record can be written by the CPU or DMA controller in the process of fetching a new record from memory.

Channel 0 DMA Byte Count, Offset: 0x800

Bits	Field Name	Function	Initial Value
15:0	ByteCt	The number of bytes that are left in DMA transfers.	0x0

Channel 1 DMA Byte Count, Offset: 0x804

Bits	Field Name	Function	Initial Value
15:0	ByteCt	The number of bytes that are left in DMA transfers.	0x0

Channel 2 DMA Byte Count, Offset: 0x808

Bits	Field Name	Function	Initial Value
15:0	ByteCt	The number of bytes that are left in DMA transfers.	0x0

Channel 3 DMA Byte Count, Offset: 0x80c

Bits	Field Name	Function	Initial Value
15:0	ByteCt	The number of bytes that are left in DMA transfers.	0x0

Channel 0 DMA Source Address, Offset: 0x810

Bits	Field Name	Function	Initial Value
31:0	SrcAdd	The address that the DMA controller will read the data from.	0x0

Channel 1 DMA Source Address, Offset: 0x814

Bits	Field Name	Function	Initial Value
31:0	SrcAdd	The address that the DMA controller will read the data from.	0x0

Channel 2 DMA Source Address, Offset: 0x818

Bits	Field Name	Function	Initial Value
31:0	SrcAdd	The address that the DMA controller will read the data from.	0x0

Channel 3 DMA Source Address, Offset: 0x81c

Bits	Field Name	Function	Initial Value
31:0	SrcAdd	The address that the DMA controller will read the data from.	0x0

Channel 0 DMA Destination Address, Offset: 0x820

Bits	Field Name	Function	Initial Value
31:0	DestAdd	The address that the DMA controller will write the data to.	0x0

Channel 1 DMA Destination Address, Offset: 0x824

Bits	Field Name	Function	Initial Value
31:0	DestAdd	The address that the DMA controller will write the data to.	0x0

Channel 2 DMA Destination Address, Offset: 0x828

Bits	Field Name	Function	Initial Value
31:0	DestAdd	The address that the DMA controller will write the data to.	0x0

Channel 3 DMA Destination Address, Offset: 0x82c

Bits	Field Name	Function	Initial Value
31:0	DestAdd	The address that the DMA controller will write the data to.	0x0

Channel 0 Next Record Pointer, Offset: 0x830

Bits	Field Name	Function	Initial Value
31:0	NextRecPtr	The address for the next record of DMA. A value of 0 means a NULL pointer (end of the chained list).	0x0

Channel 1 Next Record Pointer, Offset: 0x834

Bits	Field Name	Function	Initial Value
31:0	NextRecPtr	The address for the next record of DMA. A value of 0 means a NULL pointer (end of the chained list).	0x0

Channel 2 Next Record Pointer, Offset: 0x838

Bits	Field Name	Function	Initial Value
31:0	NextRecPtr	The address for the next record of DMA. A value of 0 means a NULL pointer (end of the chained list).	0x0

Channel 3 Next Record Pointer, Offset: 0x83c

Bits	Field Name	Function	Initial Value
31:0	NextRecPtr	The address for the next record of DMA. A value of 0 means a NULL pointer (end of the chained list).	0x0

3.11 DMA Channel Control Registers

Each DMA channel has a control register to set its mode of operation independently of the other three channels. A channel can be programmed to transfer data through the GT-64014. The DMA reads data from the source address (Devices or DRAM) into an internal 32-byte FIFO. From the internal FIFO, the data is written to a destination address (Devices or DRAM) that can be independent from the source address.

Source addresses and destination addresses can be programmed to increment, decrement, or hold the same value throughout the DMA transfer. For devices that can absorb a limited number of bytes at a time, the channel can be programmed to limit the number of bytes transferred in each DMA cycle. DMA accesses can be initiated by an external source (Demand mode) by asserting one of the four DMAReq[3:0]* pins, or by an internal request (Block mode) until the byte count reaches zero.

All four channels have chaining support via linked lists of records. When the chaining mode is enabled, the DMA controller will fetch the information (the record) for a new DMA transfer directly out of memory without involving the CPU. The location of the next record is in the Next Record Pointer register (NextRecPtr) and the DMA controller will fetch records every DMA transfer end until it reaches the NULL pointer (NULL pointer is zero).

There are several mechanisms for status and control of the DMA operations. An status interrupt can be programmed to be asserted every time the DMA byte count reaches zero, or only when byte count reaches zero and the record is the last record in the chain (the NextRecPtr is NULL). In addition, there is a status bit that indicates if a channel is active or not. A channel is active when it is enabled and its byte count is other than zero, or in chained mode when both its byte count is not equal to zero or its NextRecPtr is not equal to NULL, or when it is disabled and its internal FIFO is not empty. A channel can be controlled by disabling it temporarily, and a next record fetch can be forced through FetNexRec in Chained mode even if the current DMA has not ended.

Channel 0 Control, Offset: 0x840

Bits	Field name	Function	Initial Value
0	Reserved	Must be '0'	0x0
1	Reserved	Must be '0'	0x0
3:2	SrcDir	Source Direction. 00 - Increment source address 01 - Decrement source address 10 - Hold in the same value	0x0
5:4	DestDir	Destination Direction. 00 - Increment destination address 01 - Decrement destination address 10 - Hold in the same value	0x0

Bits	Field name	Function	Initial Value
8:6	DatTransLim	Data Transfer Limit in each DMA access. 101 - 1 Byte 110 - 2 Bytes 000 - 4 Bytes 001 - 8 Bytes 011 - 16 Bytes 111 - 32 Bytes	0x0
9	ChainMod	Chained Mode. 0 - Chained mode; when a DMA access is terminated, the parameters of the next DMA access will come from a record in memory that a NextRecPtr register points at. 1 - Non-Chained mode; only the values that are programmed by the CPU directly into the ByteCt, SrcAdd, and DestAdd registers are used.	0x0
10	IntMode	Interrupt Mode. 0 - Interrupt asserted every time the DMA byte count reaches terminal count. 1 - Interrupt every NULL pointer (in Chained mode)	0x0
11	TransMod	Transfer Mode. 0 - Demand 1 - Block	0x0
12	ChanEn	Channel Enable. 0 - Disable 1 - Enable	0x0
13	FetNexRec	Fetch Next Record. 1 - Forces a fetch of the next record (even if the current DMA has not ended). This bit is reset after fetch is completed (meaningful only in Chained mode).	0x0
14	DMAActSt	DMA Activity Status (read only). 0 - Channel is not active 1 - Channel is active	0x0

Channel 1 Control, Offset: 0x844

Bits	Field Name	Function	Initial Value
14:0	Various	Fields function as in Channel 0 Control.	0x0

Channel 2 Control, Offset: 0x848

Bits	Field Name	Function	Initial Value
14:0	Various	Fields function as in Channel 0 Control.	0x0

Channel 3 Control, Offset: 0x84c

Bits	Field Name	Function	Initial Value
14:0	Various	Fields function as in Channel 0 Control.	0x0

3.11.1 DMA Programming Notes

3.11.1.1 Non-Chained mode

In this mode, Source, Destination and Byte Count should be initialized prior to enabling the channel. ChainMod should be set to '1'.

3.11.1.2 Chained mode

All the channel record's parameters for the current transaction (Source, Byte Count, Destination and Next Record Pointer) should be initialized in memory devices. The address of the first record should be initialized by writing it to the NextRecPtr of the channel. The channel should be enabled via ChanEn=1, the FetNexRec should be set to '1' and the ChainMod should be set to '0'.

3.11.1.3 Restarting a disabled channel

In Non-Chained mode, ChanEn should be set to '1'.

In Chained mode, the software should find out if the first fetch took place. If it did, only ChanEn should be set to '1'. If it did not, the FetNexRec should also be set to '1'.

3.11.1.4 Reprogramming an active channel

The channel should first be disabled via ChanEn=0. Then it must be assured that the channel is no longer active (for example by polling the DMAActSt of the channel). New DMA parameters should be programmed prior to re-enabling the channel via ChanEn=1.

3.12 DMA Arbiter Register

The DMA controller has a programmable arbitration scheme between its four channels. The channels are grouped into two groups, one group includes channel 0 and 1, and the other group includes channels 2 and 3. The channels in each group can be programmed to have priority so that a selected channel has the higher priority, or to have the same priority in round robin. The priority between the two groups can be programmed in a similar way so that a selected group has a higher priority, or to have the same priority in round robin. The priority scheme has additional flexibility with the programmable Priority Option. With the Priority Option the DMA bandwidth allocation can be divided in a fairer way. For example, if the PrioOpt bit is set to '0' and the PrioGrps field is set as '10', the requesting devices will get the DMA in the order 0,1,2,0,1,3,0,1,2,0,1,3,..... (assuming that PrioChan1/0 and PrioChan3/2 are set to round robin), while if the PrioOpt bit is set to '1' the requesting devices will get the DMA in the order 0,1,0,1,0,1,2,3,2,3,2,3,..... The DMA arbiter control register can be reprogrammed any time regardless of the channels' status (active or not active).

Some arbitration examples follow to facilitate the understanding of this register:

1. Assuming all 4 channels are requested all the time,
   with Arbiter Control register = 0x40, the order will be: 0,2,1,3,0,2,1,3,.....
   with Arbiter Control register = 0x0, the order will be: 0,1,2,3,0,1,2,3,.....
2. Assuming 3 channels are requested (0,1,2),

with Arbiter Control register = 0x40, the order will be: 0,2,1,2,0,2,1,2,.....
with Arbiter Control register = 0x0, the order will be: 0,1,2,0,1,2,0,1,2,.....

3. Assuming all 4 channels are requested,
with Arbiter Control register = 0x45, the order will be: 1,3,1,3,1,3,.....,0,2,0,2,0,2,.....
with Arbiter Control register = 0x5, the order will be: 1,0,3,2,1,0,3,2,1,0,3,2,.....
4. Assuming 3 channels are requested (0,1,2),
with Arbiter Control register = 0x45, the order will be: 1,2,1,2,1,2,.....,0,0,0,0,0,0,.....
with Arbiter Control register = 0x5, the order will be: 0,1,2,0,1,2,.....
5. Assuming all 4 channels are requested,
with Arbiter Control register = 0x55, the order will be: 3,3,3,3,3,3,2,2,2,2,2,1,1,1,1,0,0,0,.....
with Arbiter Control register = 0x15, the order will be: 3,2,1,3,2,0,3,2,1,3,2,0,.....
6. Assuming 3 channels are requested (0,1,2),
with Arbiter Control register = 0x55, the order will be: 2,2,2,2,1,1,1,1,0,0,0,0,.....
with Arbiter Control register = 0x15, the order will be: 2,1,2,0,2,1,2,0,.....
7. Assuming 3 channels are requested (0,2,3),
with Arbiter Control register = 0x55, the order will be: 3,3,3,...,2,2,2,...,0,0,0,.....
with Arbiter Control register = 0x15, the order will be: 3,2,0,3,2,0,3,2,0,.....

Arbiter Control, Offset: 0x860

Bits	Field name	Function	Initial Value
1:0	PrioChan1/0	Priority between Channel 0 and Channel 1. 00 - Round Robin 01 - Priority to channel 1 over channel 0 10 - Priority to channel 0 over channel 1 11 - Reserved	0x0
3:2	PrioChan3/2	Priority between Channel 2 and Channel 3. 00 - Round Robin 01 - Priority to channel 3 over 2 10 - Priority to channel 2 over 3 11 - Reserved	0x0
5:4	PrioGrps	Priority between the group of channels 0/1 and the group of channels 2/3. 00 - Round Robin 01 - Priority to channels 2/3 over 0/1 10 - Priority to channels 0/1 over 2/3 11 - Reserved	0x0
6	PrioOpt	Defines the arbiter behavior for high priority device. 0 - High priority device will relinquish the bus for a requesting device for one DMA transaction after it was serviced. 1 - High priority device will be granted as long as it requests the bus.	0x0

3.13 Timer / Counter Registers

There are three 24-bit wide and one 32-bit wide timer/counter on the GT-64014. Each one can be selected to operate as a timer or as a counter. In Counter mode, the counter will count down to terminal count, will stop and issue an interrupt. In Timer mode, it will count down, will issue an interrupt on terminal count, and will reload itself to the programmed value and continue to count. Reads from the counter or timer are done from the counter itself, while writes are to its register. For example, note that even though the registers are programmed to an initial value of '0' the counters will

read 0xffffff. In order to reprogram a timer/counter, it should first be disabled, then it should be loaded with a new value and after that it should be enabled as appropriate (counter or timer).

Timer/Counter 0, Offset: 0x850

Bits	Field Name	Function	Initial Value
31:0	TC0Value	The counter or timer initial value.	0x0

Timer/Counter 1, Offset: 0x854

Bits	Field Name	Function	Initial Value
23:0	TC1Value	The counter or timer initial value.	0x0

Timer/Counter 2, Offset: 0x858

Bits	Field Name	Function	Initial Value
23:0	TC2Value	The counter or timer initial value.	0x0

Timer/Counter 3, Offset: 0x85c

Bits	Field Name	Function	Initial Value
23:0	TC3Value	The counter or timer initial value.	0x0

Timer/Counter Control, Offset: 0x864

Bits	Field name	Function	Initial Value
0	EnTC0	The timer/counter will count only when it is enabled. 0 - Disable 1 - Enable	0x0
1	SelTC0	Timer or counter selection. 0 - Counter 1 - Timer	0x0
2	EnTC1	The timer/counter will count only when it is enabled. 0 - Disable 1 - Enable	0x0
3	SelTC1	Timer or counter selection. 0 - Counter 1 - Timer	0x0
4	EnTC2	The timer/counter will count only when it is enabled. 0 - Disable 1 - Enable	0x0

Bits	Field name	Function	Initial Value
5	SelTC2	Timer or counter selection . 0 - Counter 1 - Timer	0x0
6	EnTC3	The timer/counter will count only when it is enabled. 0 - Disable 1 - Enable	0x0
7	SelTC3	Timer or counter selection. 0 - Counter 1 - Timer	0x0

3.14 Interrupt Registers

Bits	Field Name	Function	Initial Value
31:12	BankSize	Specifies the RAS[1:0] address mapping in conjunction with the RAS[1:0] Base Address register. Set to '0' indicates that the corresponding bit in the address and in the base address must be equal in order to have a hit. Set to '1' indicates that the corresponding bit in the address is a don't-care. For example, bit 12 set to '1' indicates that the RAS[1:0] size is 8KBytes (address bits [12:0] are changeable/don't-care). The set bits in the Bank Size must be sequential (e.g. 000...001, 000...011, 000...111 are correct values, whereas 000...010 and 000...100 are not).	0x00fff

IntSum in the Interrupt Cause register is the logical OR of bits[29:1], regardless of the Mask registers' values. This is in order to be notified via polling if any interrupt occurred within the GT-64014. Therefore, bit[0] of the CPU Mask is read-only '0'.

CPUIntSum in the Interrupt Cause register is the logical OR of bits[29:26,20:1], masked by bits[29:26,20:1] of the CPU Mask register. Therefore, bits[25:21] of the CPU Mask register, being non-relevant to interrupts directed to the CPU, are read-only '0'. Also bits[31:30], being summaries, are read-only '0'.

Interrupt Cause, Offset: 0xc18 *(all bits are cleared by writing a value of '0' by the CPU, unless stated otherwise)*

Bits	Field Name	Function	Initial Value
0	IntSum	Interrupt summary. Logical OR of all the interrupt bits, regardless of the Mask registers' values.	0x0 Read only
1	MemOut	Asserts when the CPU accesses an address out of range in the memory decoding or a burst access to 8-/16-bit devices.	0x0
2	DMAOut	Asserts when the DMA accesses an address out of range.	0x0
3	CPUOut	Asserts when the CPU accesses an address out of range.	0x0
4	DMA0Comp	Asserts at completion of DMA Channel 0 transfer.	0x0

Bits	Field Name	Function	Initial Value
5	DMA1Comp	Asserts at completion of DMA Channel 1 transfer.	0x0
6	DMA2Comp	Asserts at completion of DMA Channel 2 transfer.	0x0
7	DMA3Comp	Asserts at completion of DMA Channel 3 transfer.	0x0
8	T0Exp	Asserts when Timer 0 expires.	0x0
9	T1Exp	Asserts when Timer 1 expires.	0x0
10	T2Exp	Asserts when Timer 2 expires.	0x0
11	T3Exp	Asserts when Timer 3 expires.	0x0
12	Reserved	Must write '0'.	0x0
13	Reserved	Must write '0'.	0x0
14	Reserved	Must write '0'.	0x0
15	Reserved	Must write '0'.	0x0
16	AddrErr	Asserts when the GT-64014 detects a parity error on the address lines.	0x0
17	MemErr	Asserts when a memory parity error is detected. Applicable only when an external parity checking device is used.	0x0
18	MasAbort	Asserts upon master abort.	0x0
19	TarAbort	Asserts upon target abort.	0x0
20	RetryCtr	Asserts when the retry counter expires.	0x0
25:21	Reserved	Must write '0'.	0x0
29:26	Reserved	Must write '0'.	0x0
30	Reserved	Must write '0'.	0x0
31	Reserved	Must write '0'.	0x0

CPU Mask, Offset: 0xc1c

Bits	Field Name	Function	Initial Value
31:0	CPUMask	Mask to the CPU interrupt line for the appropriate bits in the Interrupt Cause register. Reserved bits 12:15, and 21:31 must be masked.	0x00000000

4. RESTRICTIONS

4.1 CPU Interface

a) The CPU should not attempt an access before 10 TClk cycles following deassertion of Rst* have expired.

b) The CPU Interface supports only DDDDDDDD and DXDXDXDXDXDXDXDX write patterns.

c) A write of more than 4 bytes to internal space will be ignored. A read of more than 4 bytes to internal space will result in transaction termination with bus-error indication (SysCmd[5] equal '1').

4.2 Memory Interface

a) Unless the boot device is 32-bits wide, the boot will be on the even bank.

b) All Device Parameters (section 3.7) must be greater or equal to 3. i.e., AccToFirst, AccToNext, ADSToWr, WrActive and WrHigh.

c) When working with an 8- or 16-bit bus from CPU, a read/write operation can not exceed 64-bits (8 bytes).

d) When working with an 8- or 16-bit bus from DMA, a read/write operation can't exceed 32-bits (4 bytes).

e) When an erroneous address is issued or a burst operation is performed to an 8- or 16-bit device, the GT-64011 forces an interrupt (unless masked). If a sequence of address misses occurs, there will be no other interrupt prior to resetting the appropriate bit in the cause register and no new address will be registered in the Address Decode Error register (0x470) prior to reading it.

f) When the CPU reads from an address which is decoded in the CPU Interface Unit as being a hit for CS[2:0]* or CS[4:3]* and decoded as a miss in the DRAM/Device Interface Unit, the cycle will complete only if Ready* is asserted (i.e., driven low). Although being a result of improper and inconsistent programming of the address space defining registers, the following 2 workarounds exist:

- Ready* should always be asserted (low) when CSTiming* is inactive (high).
- If the Ready* signal is not needed in the system, the DMAReq[0]/Ready* pin should either be programmed as Ready* and constantly driven active (low) or be programmed as DMAReq[0]*.

4.3 DMA

a) Transfers of less than 4 bytes are not supported.

b) In order to restart a channel after it has been enabled, it should first be checked that the DMAActSt bit is set to NOT ACTIVE (see section 3.9).

c) When Source or Destination address is decremented, both addresses should be word-aligned (that is, A1 and A0 should be both zero), and Byte Count should be a multiple of 4 (this applies for burst limits greater than 4 bytes).

d) When burst limit is less than or equal to 4 bytes, no support in decrement.

e) When using the address hold option in the source direction (SrcDir in section 3.9), the source address should keep the following rules:

- word-aligned if burst limit is greater or equal to 4 bytes.
- bits [1:0] equal to 00, 01 or 10 if burst limit is equal to 2 bytes.
- no restriction for burst limit equal to 1 byte.

f) When using the address hold option in the destination direction (DestDir in section 3.9), the following rules should be kept:

- both Source and Destination addresses should be word-aligned if burst limit is greater or equal to 4 bytes.
- bit [0] of both Source and Destination addresses should be equal to 0 if burst limit is equal to 2 bytes.
- no restriction for burst limit equal to 1 byte.

g) Fly-by is not supported.

h) Records' addresses should be a multiple of 16.

5. DC CHARACTERISTICS

NOTICE: Preliminary specifications

These specifications are *targets* only and are subject to change without notice.

5.1 Absolute Maximum Ratings

Symbol	Parameter	Min.	Max.	Unit
Vdd	Supply Voltage	-0.3	6.5	V
Vi	Input Voltage	-0.3	Vdd+0.3	V
Vo	Output Voltage	-0.3	Vdd+0.3	V
Io	Output Current		24	mA
Iik	Input Protect Diode Current		+-20	mA
Iok	Output Protect Diode Current		+-20	mA
Tc	Operating Case Temperature	0	70	C
Tstg	Storage Temperature	-40	125	C
ESD	ESD Voltage		2000	V

5.2 Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
Vdd	Supply Voltage	4.75		5.25	V
Vi	Input Voltage	0		Vdd	V
Vo	Output Voltage	0		Vdd	V
Tc	Operating Case Temperature	0		70	C
Cin	Input Capacitance		7.2		pF
Cout	Output Capacitance		7.2		pF

5.3 DC Electrical Characteristics Over Operating Range

(Tc=0-70°C; Vdd=+5V, +/-5%)

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
Vih	Input HIGH level	Guaranteed Logic HIGH level	2.0		Vdd + 0.5V	V
Vil	Input LOW level	Guaranteed Logic LOW level	-0.5		0.8	V

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
Voh	Output HIGH Voltage	IoH = 2 mA IoH = 4 mA IoH = 8 mA IoH = 12 mA IoH = 16 mA IoH = 24 mA	2.4			V
Vol	Output LOW Voltage	IoL = 2 mA IoL = 4 mA IoL = 8 mA IoL = 12 mA IoL = 16 mA IoL = 24 mA			0.4	V
Iih	Input HIGH Current				+-1	uA
Iil	Input LOW Current				+-1	uA
Iozh	High Impedance Output Current				+-1	uA
Iozl	High Impedance Output Current				+-1	uA
Vh	Input Hysteresis		TBD	TBD	TBD	mV
Icc	Operating Current				400	mA

NOTE: Pullup/Pulldown resistors are 45KOhm minimum, 65KOhm typical, 80KOhm maximum.

5.4 Thermal Characteristics

Symbol	Max.	Unit
Θjc	TBD	C/W
Θja	TBD, Estimated to be 5 C/W	C/W

6. AC TIMINGS

NOTICE: Preliminary Specifications

These specifications are *targets* only and are subject to change without notice.

6.1 AC Specifications

(TC= 0-70°C; VDD= +5V, +/- 5%)

Symbol	Signals	Description	Min	Max	Unit
t1	TClk	Pulse Width High	8		nS
t2	TClk	Pulse Width Low	8		nS
t3	TClk	Clock Period	20	30	nS
t4	TClk	Rise Time		3	nS
t5	TClk	Fall Time		3	nS
t6	Rst*	Active	10		TClk
t7	DMAReq[3]*, LEAdrE/ DMAReq[2]*, DMAReq[1]*/ParErr*, AD[31:0]	Setup	5		nS
t8	DMAReq[0]*/Ready*,	Setup	11		nS
t9	WrRdy*, ValidIn*, SysCmd[8:0], Interrupt*	Delay	2	12	nS
t10	DAdr[11:0]	Delay (Row Address)	3	18	nS
t11	DAdr[11:0]	Delay (Column Address)	3	11	nS
t12	BAdr[2:0], ADS*, SysAD[31:0]	Delay	2	13	nS
t13	EWr[3:0]*, OWr[3:0]*	Delay From TClk Falling Edge	2	13	nS
t14	DWr*, CSTiming*, ALE,	Delay	2	10	nS
t15	ECAS[3:0]*, OCAS[3:0]*, OEB, OEO*, OEE*, AD[31:0]	Delay	2	9	nS
t16	ALE, LEO, LEE	Delay from TClk Falling Edge	2	10	nS
t17	ValidOut*, Release*, DMAReq[3]*, LEAdrE/ DMAReq[2]*, DMAReq[1]*/ParErr*, SysAD[63:0], SysCmd[8:0], AD[31:0], DMAReq[0]*/Ready*	Hold	1		nS
t18	ValidOut*, Release*,SysAD[31:0], SysCmd[8:0]	Setup	3		nS
t19	LEAdrE/DMAReq[2]*, LEAdrO	Delay	2	10	nS

t20	LEAdrE/DMAReq[2]*, LEAdrO	Delay From TClk Falling Edge	2	10	nS
t21	LEO, LEE	Delay	2	9	nS
t22	RAS[3:0]*	Delay	3	8	nS

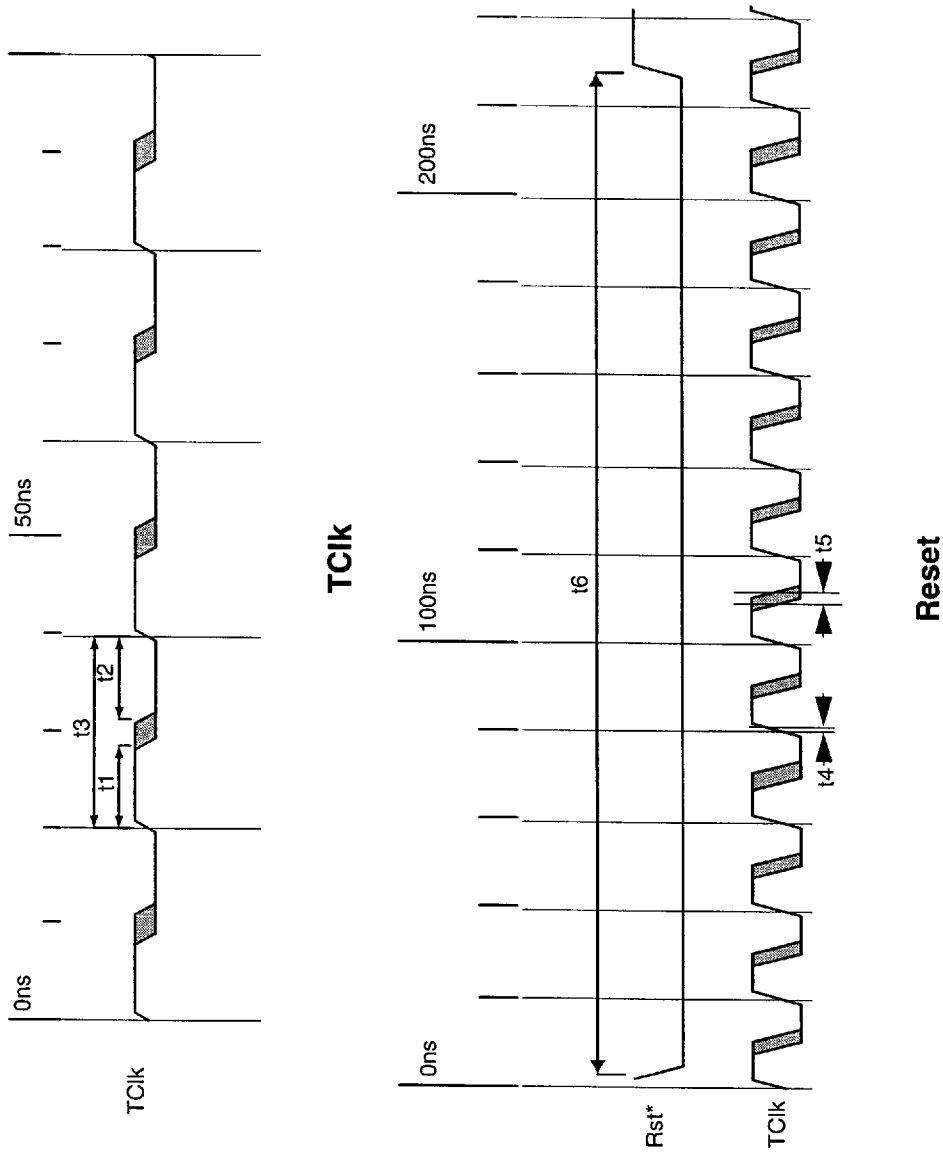
Notes:

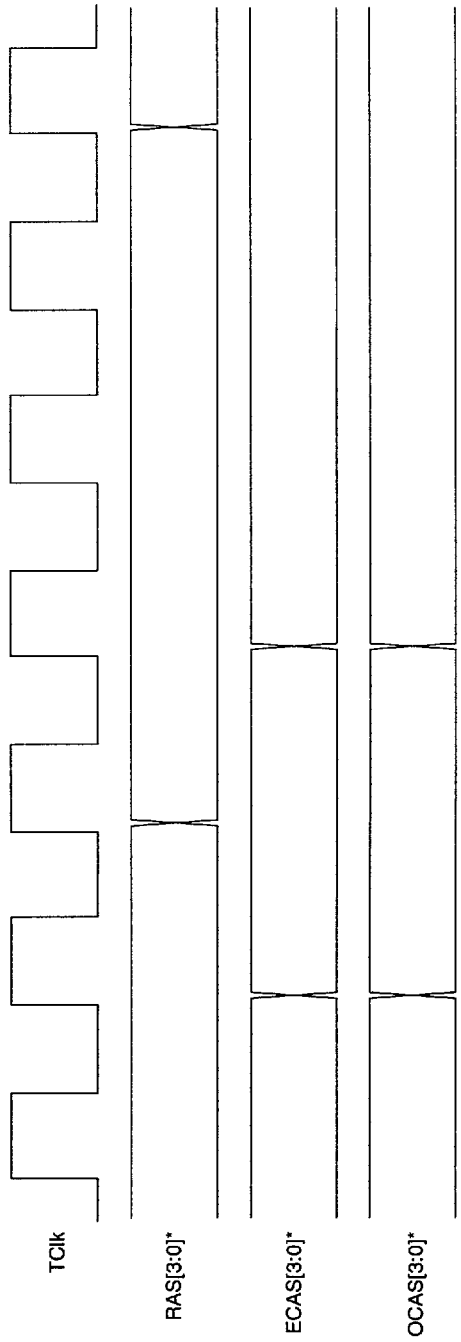
1. All Delays, Setup, and Hold times are referred to TClk rising edge, unless stated otherwise.

2. All outputs are specified for 50pF load except: WrRdy*, ValidIn*, ALE, LEAdrO, LEAdrE/DMAReq[2]* - 30pF, Interrupt* - 20 pF.

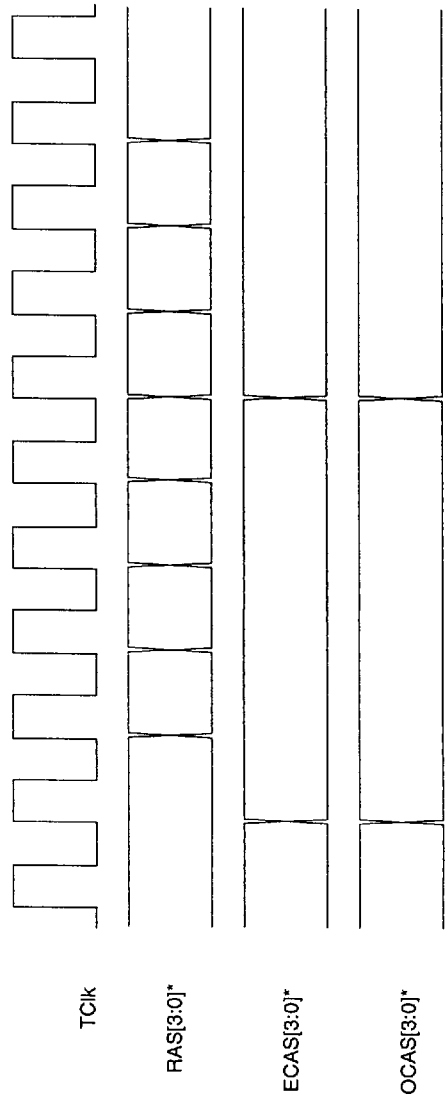
6.2 Waveforms

See the Galileo website at www.galileoT.com for the full GT-64014 waveforms.





Non Staggered Refresh



Staggered Refresh

7. Functional Waveforms

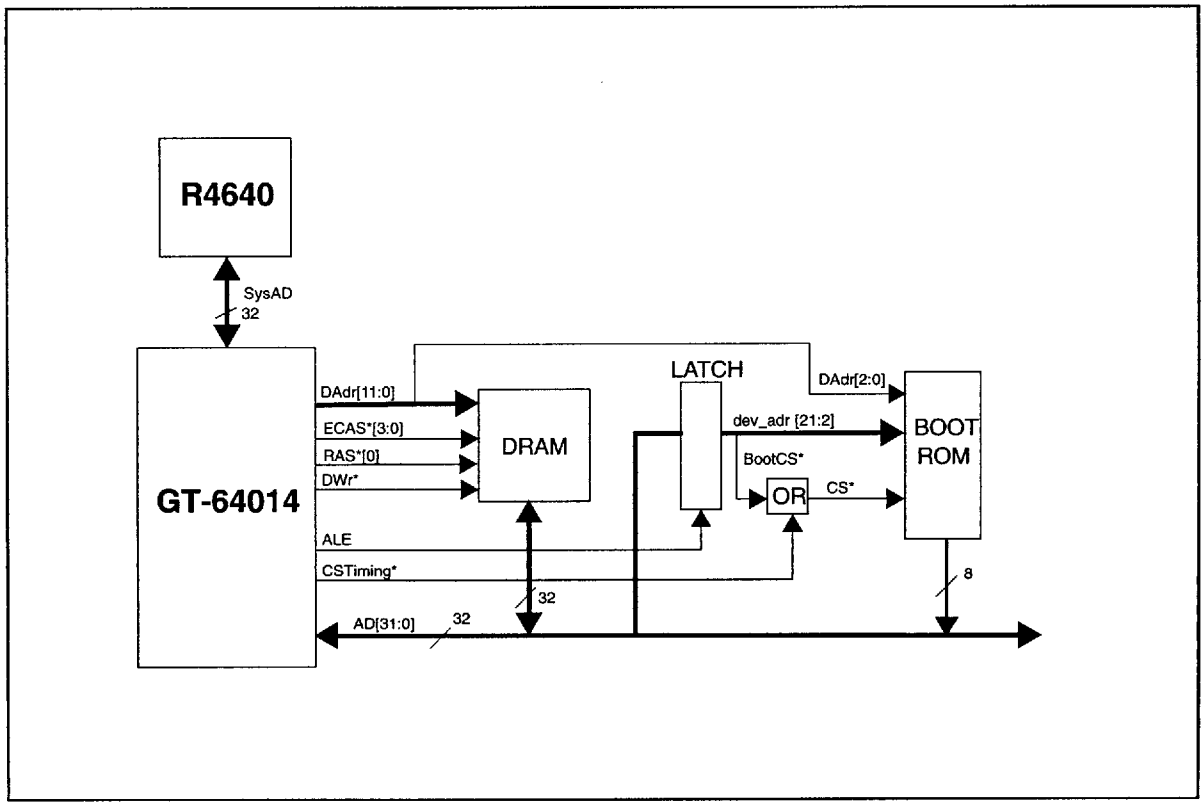
Functional waveforms for the GT-64014 are identical to those of the GT-64011. All functional waveforms can be found on our FTP site at ftp://ftp.galileoT.com.

A subset of the functional waveforms will be added to later revisions of this document.

8. APPLICATIONS: SYSTEM CONFIGURATIONS

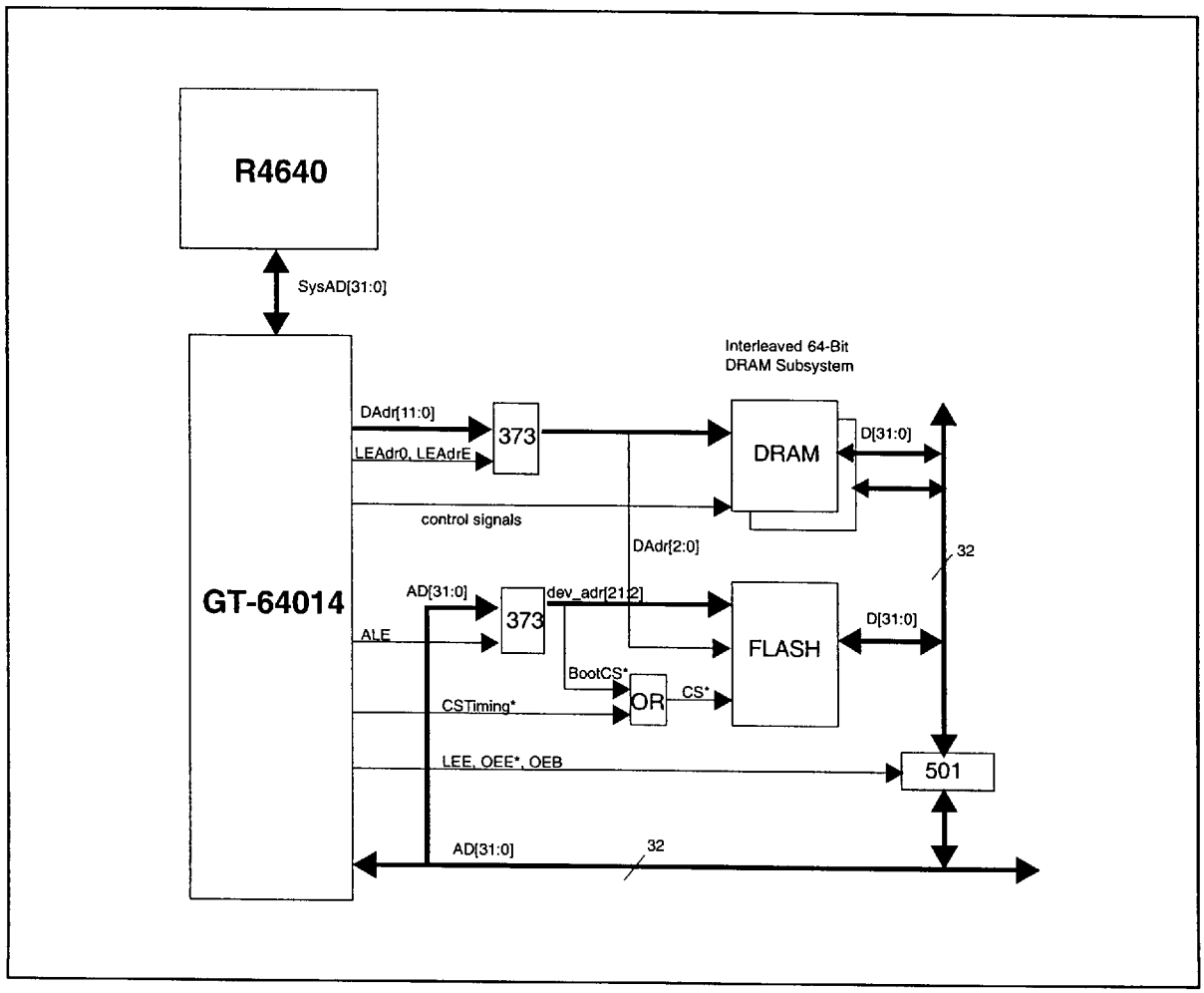
8.1 Minimal System Configuration

A minimal system configuration is shown below. It includes an 8-bit wide boot ROM and a 32-bit wide DRAM. This configuration can be appealing to applications that need high performance and are limited to a minimal board space.



8.2 Typical System

The high performance system shown below includes 32-bit wide memories, Flash for storing code and data, and DRAM as main memory. The CPU can read from the DRAM at peak bandwidth of 200Mbytes per second. CPU writes have peak bandwidth of 400Mbytes per second for all devices through the GT-64014 on-chip write buffer.



8.3 Interface to Asynchronous Devices

In this case, we show the connectivity between the GT-64014 and 32 bit-wide standard memory devices (e.g. SRAM). In this example the latches selected are industry standard FCT16501 for data, FCT16373 for the address, and FCT16373 for the burst address for the odd bank. The data latches that interface to the AD bus are used to interleave the data in read and write access from the GT-64014 and enable data rate of one data per cycle at 50MHz (200 Mbytes per second peak rate). The data latches that interface the SysAD bus are used only in one direction in CPU reads. The SysAD latches enable low latency to first word and up to 400 Mbytes per second bandwidth in CPU burst reads. FCT16373 logic chips are used to latch the address, the CS* and the DevRW*for the Devices. The latch for the odd bank burst address is needed for the address interleaving. This system configuration is for little endian, no decrement.

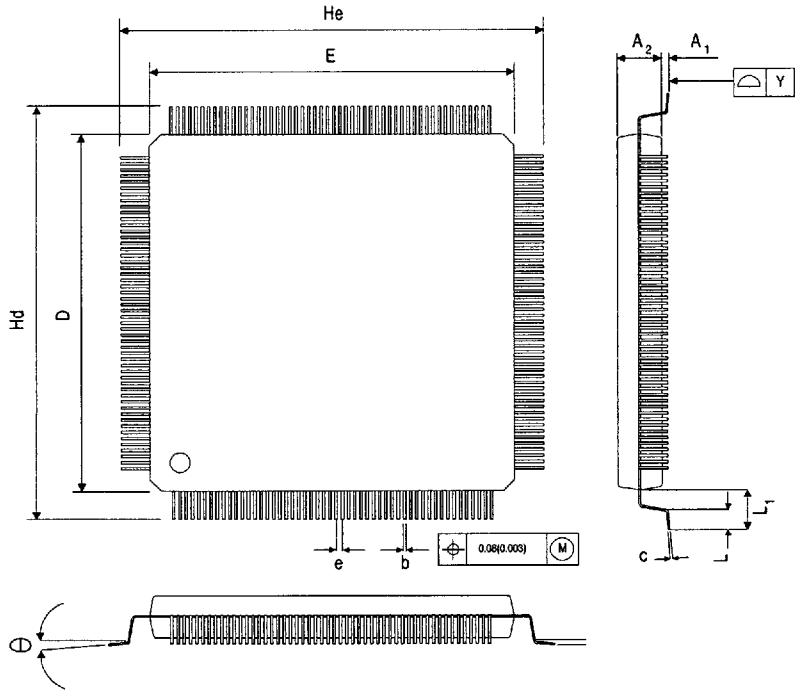
9. Pin List, 208 pin PQFP (sorted by number)

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
1	VDD	36	N/C	71	SysCmd[1]
2	N/C	37	N/C	72	SysCmd[0]
3	N/C	38	VSS	73	SysAD[0]
4	N/C	39	N/C	74	SysAD[1]
5	N/C	40	N/C	75	SysAD[2]
6	N/C	41	N/C	76	SysAD[3]
7	VSS	42	N/C	77	SysAD[4]
8	VSS	43	N/C	78	SysAD[5]
9	VDD	44	VSS	79	VSS
10	N/C	45	VDD	80	VRef
11	N/C	46	N/C	81	VDD
12	N/C	47	N/C	82	SysAD[6]
13	N/C	48	N/C	83	SysAD[7]
14	N/C	49	N/C	84	SysAD[8]
15	VSS	50	N/C	85	SysAD[9]
16	VDD	51	VSS	86	SysAD[10]
17	VSS	52	VDD	87	SysAD[11]
18	N/C	53	N/C	88	SysAD[12]
19	N/C	54	N/C	89	SysAD[13]
20	N/C	55	N/C	90	VSS
21	N/C	56	N/C	91	TClk
22	VDD	57	N/C	92	VDD
23	VSS	58	VSS	93	SysAD[14]
24	VDD	59	Should be tied to VDD	94	SysAD[15]
25	N/C	60	DMAReq[3]*	95	SysAD[16]
26	N/C	61	Interrupt*	96	SysAD[17]
27	N/C	62	SysCmd[8]	97	SysAD[18]
28	N/C	63	SysCmd[7]	98	VSS
29	N/C	64	SysCmd[6]	99	VDD
30	N/C	65	SysCmd[5]	100	SysAD[19]
31	VSS	66	SysCmd[4]	101	SysAD[20]
32	VDD	67	VSS	102	SysAD[21]
33	N/C	68	VDD	103	SysAD[22]
34	N/C	69	SysCmd[3]	104	SysAD[23]
35	N/C	70	SysCmd[2]	105	SysAD[24]

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
106	SysAD[25]	141	AD[23]	176	OCAS[2]*
107	SysAD[26]	142	AD[22]	177	OCAS[1]*
108	SysAD[27]	143	VSS	178	OCAS[0]*
109	SysAD[28]	144	VDD	179	RAS[3]*
110	VSS	145	AD[21]	180	RAS[2]*
111	VDD	146	AD[20]	181	RAS[1]*
112	SysAD[29]	147	AD[19]	182	RAS[0]*
113	SysAD[30]	148	AD[18]	183	DAdr[11]/ADS*
114	SysAD[31]	149	AD[17]	184	DAdr[10]/OWr[3]*
115	ValidOut*	150	AD[16]	185	VSS
116	ValidIn*	151	AD[15]	186	DAdr[9]/OWr[2]*
117	WrRdy*	152	AD[14]	187	DAdr[8]/OWr[1]*
118	Release*	153	AD[13]	188	DAdr[7]/OWr[0]*
119	DMAReq[0]*/Ready*	154	VSS	189	DAdr[6]/EWr[3]*
120	DMAReq[1]*/ParErr*	155	AD[12]	190	DAdr[5]/EWr[2]*
121	LEAdrE/DMAReq[2]*	156	AD[11]	191	DAdr[4]/EWr[1]*
122	LEAdrO	157	AD[10]	192	DAdr[3]/EWr[0]*
123	OEB	158	AD[9]	193	DAdr[2]/BAdr[2]
124	OEE*	159	AD[8]	194	DAdr[1]/BAdr[1]
125	OEO*	160	AD[7]	195	DAdr[0]/BAdr[0]
126	VSS	161	AD[6]	196	N/C
127	LEE	162	AD[5]	197	Rst*
128	LEO	163	AD[4]	198	VDD
129	ALE	164	AD[3]	199	VDD
130	CSTiming*	165	VSS	200	VSS
131	AD[31]/CS[3]*	166	AD[2]	201	VSS
132	AD[30]/CS[2]*	167	AD[1]/DevRW*	202	N/C
133	AD[29]/CS[1]*	168	AD[0]/BootCS*	203	VSS
134	AD[28]/CS[0]*	169	DWr*	204	N/C
135	AD[27]/DMAAck[3]*	170	ECAS[3]*	205	N/C
136	AD[26]/DMAAck[2]*	171	ECAS[2]*	206	N/C
137	AD[25]/DMAAck[1]*	172	ECAS[1]*	207	N/C
138	VSS	173	ECAS[0]*	208	VSS
139	VDD	174	VDD		
140	AD[24]/DMAAck[0]*	175	OCAS[3]*		

10. PACKAGING

The GT-64014 is packaged in a JEDEC standard 208-pin PQFP.



208 LEAD PQFP PACKAGE OUTLINE

SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A_1	0.05	0.25	0.50
A_2	3.17	3.32	3.47
b	0.10	0.20	0.30
c	0.10	0.15	0.20
D	27.90	28.00	28.10
E	27.90	28.00	28.10
e		0.50	
Hd	30.35	30.60	30.85
He	30.35	30.60	30.85
L	0.45	0.60	0.75
L_1		1.30	
Y			0.08
Q	0		7

11. Revision History

Table 1: Revision History

Document Type	Revision Number	Date	Comments
PRODUCT PREVIEW	0.1	2/97	First revision of PRODUCT PREVIEW.