



Features

- ♦ **RC32300 32-bit Microprocessor**
 - Up to 150 MHz operation
 - Enhanced MIPS-II Instruction Set Architecture (ISA)
 - Cache prefetch instruction
 - Conditional move instruction
 - DSP instructions
 - Supports big or little endian operation
 - MMU with 32 page TLB
 - 8kB Instruction Cache, 2-way set associative
 - 2kB Data Cache, 2-way set associative
 - Cache locking per line
 - Programmable on a page basis to implement a write-through no write allocate, write-through write allocate, or write-back algorithms for cache management
 - Compatible with a wide variety of operating systems
- ♦ **Local Bus Interface**
 - Up to 75 MHz operation
 - 26-bit address bus
 - 32-bit data bus
 - Direct control of local memory and peripherals
 - Programmable system watch-dog timers
 - Big or little endian support
- ♦ **Interrupt Controller simplifies exception management**
- ♦ **Four general purpose 32-bit timer/counters**

- ♦ **Programmable I/O (PIO)**
 - Input/Output/Interrupt source
 - Individually programmable
- ♦ **SDRAM Controller (32-bit memory only)**
 - 4 banks, non-interleaved
 - Up to 256MB total SDRAM memory supported
 - Implements full, direct control of discrete, SODIMM, or DIMM memories
 - Supports 16Mb through 256Mb SDRAM device depths
 - Automatic refresh generation
- ♦ **Serial Peripheral Interface (SPI) master mode interface**
- ♦ **UART Interface**
 - Two 16550 compatible UARTs
 - Baud rate support up to 1.5M
 - Modem control signals available on one channel
- ♦ **Memory & Peripheral Controller**
 - 6 banks, up to 64MB per bank
 - Supports 8-, 16-, and 32-bit interfaces
 - Supports Flash ROM, SRAM, dual-port memory, and peripheral devices
 - Supports external wait-state generation
 - 8-bit boot PROM support
 - Flexible I/O timing protocols

Block Diagram

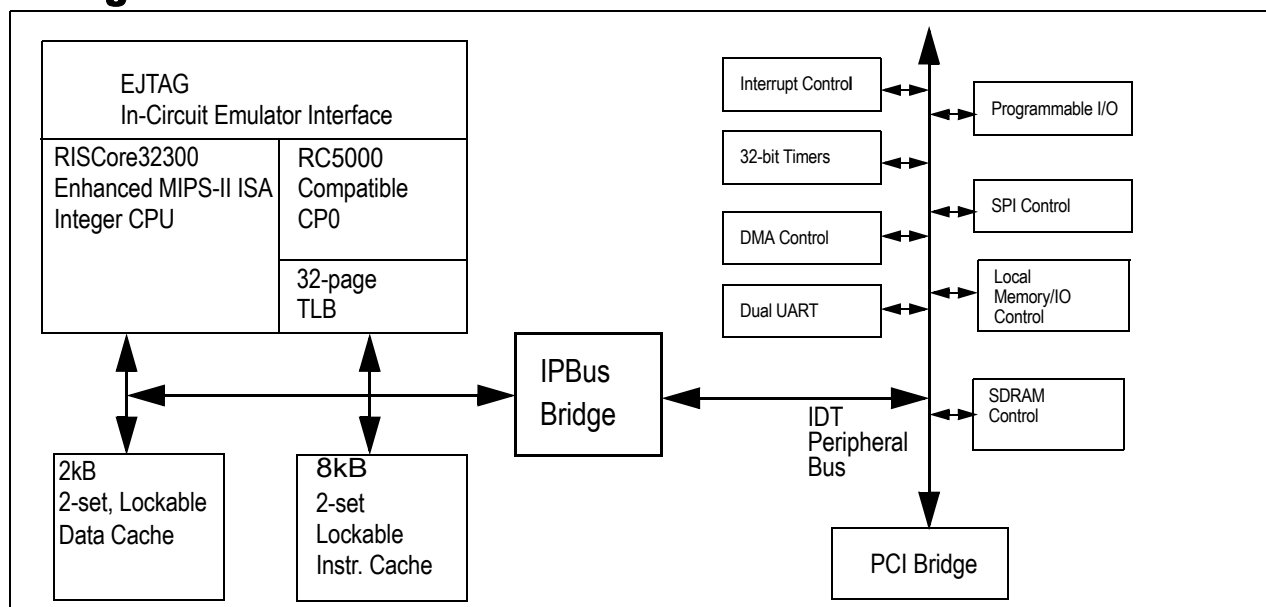


Figure 1 RC32334 Block Diagram

◆ 4 DMA Channels

- 4 general purpose DMA, each with endianness swappers and byte lane data alignment
- Supports scatter/gather, chaining via linked lists of records
- Supports memory-to-memory, memory-to-I/O, memory-to-PCI, PCI-to-PCI, and I/O-to-I/O transfers
- Supports unaligned transfers
- Supports burst transfers
- Programmable DMA bus transactions burst size (up to 16 bytes)

◆ PCI Bus Interface

- 32-bit PCI, up to 66 MHz
- Revision 2.1 compatible
- Target or master
- Host or satellite
- Three slot PCI arbiter
- Serial EEPROM support, for loading configuration registers

◆ Off-the-shelf development tools

◆ JTAG Interface (IEEE Std. 1149.1 compatible)

◆ 256-ball BGA (1.0mm spacing)

◆ 3.3V operation with 5V compatible I/O

◆ EJTAG in-circuit emulator interface

Device Overview

The IDT RC32334 device is an integrated processor based on the RC32300 CPU core. This product incorporates a high-performance, low-cost 32-bit CPU core with functionality common to a large number of embedded applications. The RC32334 integrates these functions to enable the use of low-cost PC commodity market memory and I/O devices, allowing the aggressive price/performance characteristics of the CPU to be realized quickly into low-cost systems.

CPU Execution Core

The RC32334 integrates the RISCore32300, the same CPU core found in the award-winning RC32364 microprocessor.

The RISCore32300 implements the Enhanced MIPS-II ISA. Thus, it is upwardly compatible with applications written for a wide variety of MIPS architecture processors, and it is kernel compatible with the modern operating systems that support IDT's 64-bit RISController product family.

The RISCore32300 was explicitly defined and designed for integrated processor products such as the RC32334. Key attributes of the execution core found within this product include:

- ◆ High-speed, 5-stage scalar pipeline executes to 150MHz. This high performance enables the RC32334 to perform a variety of performance intensive tasks, such as routing, DSP algorithms, etc.
- ◆ 32-bit architecture with enhancements of key capabilities. Thus, the RC32334 can execute existing 32-bit programs, while enabling designers to take advantage of recent advances in CPU architecture.
- ◆ Count leading-zeroes/ones. These instructions are common to a wide variety of tasks, including modem emulation, voice over IP compression and decompression, etc.
- ◆ Cache PREFetch instruction support, including a specialized form intended to help memory coherency. System programmers can allocate and stage the use of memory bandwidth to achieve maximum performance.
- ◆ 8kB of 2-way set associative instruction cache

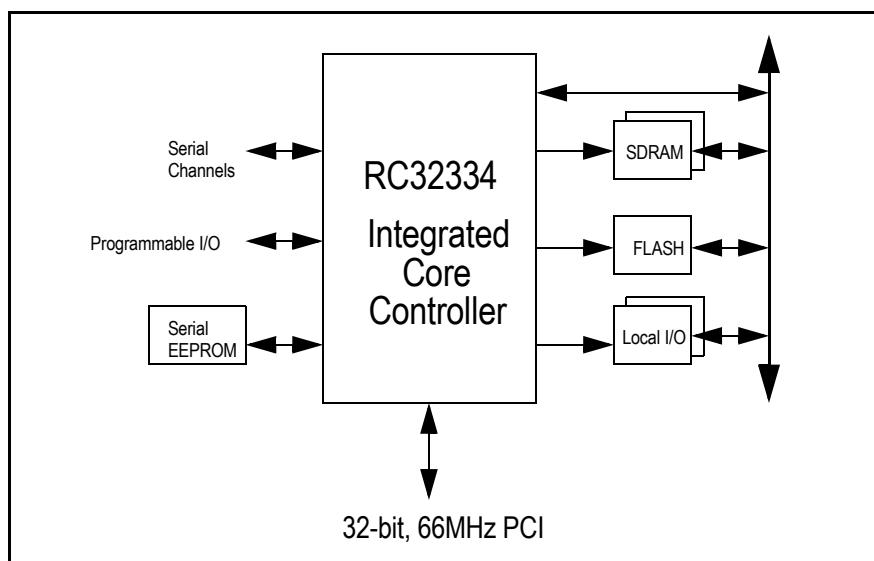


Figure 2 RC32334 Based System Diagram

- ◆ 2KB of 2-way set associative data cache, capable of write-back and write-through operation.
- ◆ Cache locking per line to speed real-time systems and critical system functions
- ◆ On-chip TLB to enable multi-tasking in modern operating systems
- ◆ EJTAG interface to enable sophisticated low-cost in-circuit emulation.

Synchronous-DRAM Interface

The RC32334 integrates a SDRAM controller which provides direct control of system SyncDRAM running at speeds to 75MHz.

Key capabilities of the SDRAM controller include:

- ◆ Direct control of 4 banks of SDRAM (up to 2 64-bit wide DIMMs)
- ◆ On-chip page comparators optimize access latency.
- ◆ Speeds to 75MHz
- ◆ Programmable address map.
- ◆ Supports 16, 64, 128 or 256Mb SDRAM devices
- ◆ Automatic refresh generation driven by on-chip timer
- ◆ Support for discrete devices, SODIMM, or DIMM modules.

Thus, systems can take advantage of the full range of commodity memory that is available, enabling system optimization for cost, real-estate, or other attributes.

Local Memory and I/O Controller

The local memory and I/O controller implements direct control of external memory devices, including the boot ROM as well as other memory areas, and also implements direct control of external peripherals.

The local memory controller is highly flexible, allowing a wide range of devices to be directly controlled by the RC32334 processor. For example, a system can be built using an 8-bit boot ROM, 16-bit FLASH cards (possibly on PCMCIA), a 32-bit SRAM or dual-port memory, and a variety of low-cost peripherals.

Key capabilities include:

- ◆ Direct control of EPROM, FLASH, RAM, and dual-port memories
- ◆ 6 chip-select outputs, supporting up to 64MB per memory space
- ◆ Supports mixture of 8-, 16-, and 32-bit wide memory regions
- ◆ Flexible timing protocols allow direct control of a wide variety of devices
- ◆ Programmable address map for 2 chip selects
- ◆ Automatic wait state generation.

PCI Bus Bridge

In order to leverage the wide availability of low-cost peripherals for the PC market as well as to simplify the design of add-in functions, the RC32334 integrates a full 32-bit PCI bus bridge. Key attributes of this bridge include:

- ◆ 66 MHz operation
- ◆ PCI revision 2.1 compliant
- ◆ Programmable address mappings between CPU/Local memory and PCI memory and I/O
- ◆ On-chip PCI arbiter
- ◆ Extensive buffering allows PCI to operate concurrently with local memory transfers
- ◆ Selectable byte-ordering swapper
- ◆ 5V tolerant I/O.

On-Chip DMA Controller

To minimize CPU exception handling and maximize the efficiency of system bandwidth, the RC32334 integrates a very sophisticated 4-channel DMA controller on chip.

The RC32334 DMA controller is capable of:

- ◆ Chaining and scatter/gather support through the use of a flexible, linked list of DMA transaction descriptors
- ◆ Capable of memory<->memory, memory<->I/O, and PCI<->memory DMA
- ◆ Unaligned transfer support
- ◆ Byte, halfword, word, quadword DMA support.

On-Chip Peripherals

The RC32334 also integrates peripherals that are common to a wide variety of embedded systems.

- ◆ Dual channel 16550 compatible UARTs, with modem control interface on one channel.
- ◆ SPI master mode interface for direct interface to EEPROM, A/D, etc.
- ◆ Interrupt Controller to speed interrupt decode and management
- ◆ Four 32-bit on-chip Timer/Counters
- ◆ Programmable I/O module

Debug Support

To facilitate rapid time to market, the RC32334 provides extensive support for system debug.

First and foremost, this product integrates an EJTAG in-circuit emulation module, allowing a low-cost emulator to interoperate with programs executing on the controller. By using an augmented JTAG interface, the RC32334 is able to reuse the same low-cost emulators developed around the RC32364 CPU.

Secondly, the RC32334 implements additional reporting signals intended to simplify the task of system debugging when using a logic analyzer. This product allows the logic analyzer to differentiate transactions initiated by DMA from those initiated by the CPU and further allows CPU transactions to be sorted into instruction fetches vs. data fetches.

Finally, the RC32334 implements a full boundary scan capability, allowing board manufacturing diagnostics and debug.

Packaging

The RC32334 is packaged using a 256-lead PBGA package, with 1.0mm ball spacing.

Thermal Considerations

The RC32334 consumes less than 2.1 W peak power. The device is guaranteed in an ambient temperature range of 0° to +85° C for commercial temperature devices; -40° to +85° for industrial temperature devices.

Revision History

May 16, 2000: Initial version.

June 8, 2000: In CPU Core Specific Signals section of Table 1, changed `cpu_dr_r_n` pin from Input to Output. Updated document from Advance to Preliminary Information.

June 15, 2000: In Table 1, switched assertion and de-assertion for `debug_cpu_dma_n` signal. In the AC Timing Characteristics table, added SPI section and adjusted parameters in the Reset section.

July 12, 2000: Removed "Preliminary Information" statement. Added information regarding external pull-ups and pull-downs to the Pin Description Table. Made minor revisions in other parts of the data sheet.

August 3, 2000: Added Pin Layout diagram showing power and ground pins. Revised Power Curves section to reflect support of only 2x, 3x, and 4x.

August 30, 2000: Added Standby mode and values to Power Consumption table. Extended Power Curve figure to 75 MHz.

September 25, 2000: Changed MIPS32 ISA to Enhanced MIPS-II. In Local System Interface section of Table 6, changed `Thld2` values for `mem_data[31:0]` from 1.8 to 1.5 ns and changed `Tdoh3` values for `mem_addr[25:2]`, etc. from 1.8 to 1.5 ns.

December 12, 2000: Changed Max values for `cpu_masterclock` period in Table 5 and added footnote. In Table 1, added 2nd alternate function for `spi_mosi`, `spi_miso`, `spi_sck`. In Table 10, removed the "1" from Alt column for `cpu_masterclk` and added "2" in Alt column for pins G3, G4, H2. In RC32334 Alternate Signal Functions table: added pin T2; added pin names in Alt #2 column for pins G3, G4, H2; added `PIO[11]` to Alt #2 column for pin R3.

January 4, 2001: In Table 6 under Interrupt Handling, moved the values for `Tsu9` from the Max to the Min columns.

March 13, 2001: Changed upper ambient temperature for industrial and commercial uses from +70° C to +85° C.

June 7, 2001: In the Clock Parameters table, added footnote 3 to `output_clk` category and added NA to Min and Max columns. In Figure 3 (Reset Specification), enhanced signal line for `cpu_masterclk`. In Local System Interface section of AC Timing Characteristics table, changed values in Min column for last category of signals (`Tdoh3`) from 1.5 to 2.5 for all speeds. In SDRAM Controller section of same table, changed values in Min column for last category of signals (9 signals) from 1 to 2.5 for all speeds.

September 14, 2001: In the Reset category of Table 6: switched `mem_addr[19:17]` from `Tsu22` and `Thld22` to `Tsu10` and `Thld10`; switched `mem_addr[22:20]` from `Tsu10` and `Thld10` to `Tsu22` and `Thld22`; moved `ejtag_pcs[2:0]` from Reset to Debug Interface category under `Tsu20` and `Thld20`.

November 1, 2001: Added Input Voltage Undershoot parameter and 2 footnotes to Table 10.

March 20, 2002: In Local System Interface section of AC Timing Characteristics table, changed values in Min column for last category of signals (`Tdoh3`) from 2.5 to 1.5 for all speeds. In Table 8, PCI Drive Output Pads, the Conditions for parameters V_{OL} , V_{OH} , V_{IL} , and V_{IH} were changed to read Per PCI 2.1.

Pin Description Table

The following table lists the pins provided on the RC32334. Note that those pin names followed by "_n" are active-low signals. All external pull-ups and pull-downs require 10 k Ω resistor.

Name	Type	Drive Strength Capability	Description																														
Local System Interface																																	
mem_data[31:0]	I/O	High	Local system data bus Primary data bus for memory. I/O and SDRAM.																														
mem_addr[25:2]	I/O	[25:16] Low [15:2] High	Memory Address Bus These signals provide the Memory or DRAM address, during a Memory or DRAM bus transaction. During each word data, the address increments either in linear or sub-block ordering, depending on the transaction type. The table below indicates how the memory write enable signals are used to address discreet memory port width types. <table><tr><th>Port Width</th><th colspan="4">Pin Signals</th></tr><tr><td></td><th>mem_we_n[3]</th><th>mem_we_n[2]</th><th>mem_we_n[1]</th><th>mem_we_n[0]</th></tr><tr><td>DMA (32-bit)</td><td>mem_we_n[3]</td><td>mem_we_n[2]</td><td>mem_we_n[1]</td><td>mem_we_n[0]</td></tr><tr><td>32-bit</td><td>mem_we_n[3]</td><td>mem_we_n[2]</td><td>mem_we_n[1]</td><td>mem_we_n[0]</td></tr><tr><td>16-bit</td><td>Byte High Write Enable</td><td>mem_addr[1]</td><td>Not Used (Driven Low)</td><td>Byte Low Write Enable</td></tr><tr><td>8-bit</td><td>Not Used (Driven High)</td><td>mem_addr[1]</td><td>mem_addr[0]</td><td>Byte Write Enable</td></tr></table> mem_addr[22] Alternate function: reset_boot_mode[1]. mem_addr[21] Alternate function: reset_boot_mode[0]. mem_addr[20] Alternate function: reset_pci_host_mode. mem_addr[19] Alternate function: modebit [9]. mem_addr[18] Alternate function: modebit [8]. mem_addr[17] Alternate function: modebit [7]. mem_addr[15] Alternate function: sdram_addr[15]. mem_addr[14] Alternate function: sdram_addr[14]. mem_addr[13] Alternate function: sdram_addr[13]. mem_addr[11] Alternate function: sdram_addr[11]. mem_addr[10] Alternate function: sdram_addr[10]. mem_addr[9] Alternate function: sdram_addr[9]. mem_addr[8] Alternate function: sdram_addr[8]. mem_addr[7] Alternate function: sdram_addr[7]. mem_addr[6] Alternate function: sdram_addr[6]. mem_addr[5] Alternate function: sdram_addr[5]. mem_addr[4] Alternate function: sdram_addr[4]. mem_addr[3] Alternate function: sdram_addr[3]. mem_addr[2] Alternate function: sdram_addr[2].	Port Width	Pin Signals					mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]	DMA (32-bit)	mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]	32-bit	mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]	16-bit	Byte High Write Enable	mem_addr[1]	Not Used (Driven Low)	Byte Low Write Enable	8-bit	Not Used (Driven High)	mem_addr[1]	mem_addr[0]	Byte Write Enable
Port Width	Pin Signals																																
	mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]																													
DMA (32-bit)	mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]																													
32-bit	mem_we_n[3]	mem_we_n[2]	mem_we_n[1]	mem_we_n[0]																													
16-bit	Byte High Write Enable	mem_addr[1]	Not Used (Driven Low)	Byte Low Write Enable																													
8-bit	Not Used (Driven High)	mem_addr[1]	mem_addr[0]	Byte Write Enable																													
mem_cs_n[5:0]	Output	Low	Memory Chip Select Negated Recommend an external pull-up. Signals that a Memory Bank is actively selected.																														
mem_oe_n	Output	High	Memory Output Enable Negated Recommend an external pull-up. Signals that a Memory Bank can output its data lines onto the cpu_ad bus.																														
mem_we_n[3:0]	Output	High	Memory Write Enable Negated Bus Signals which bytes are to be written during a memory transaction. Bits act as Byte Enable and mem_addr[1:0] signals for 8-bit or 16-bit wide addressing.																														

Table 1 Pin Descriptions (Part 1 of 7)

Name	Type	Drive Strength Capability	Description
mem_wait_n	Input	—	Memory Wait Negated Requires an external pullup. SRAM/I/O/IOM modes: Allows external wait-states to be injected during last cycle before data is sampled. DPM (dual-port) mode: Allows dual-port busy signal to restart memory transaction. Alternate function: sdram_wait_n.
mem_245_oe_n	Output	Low	Memory FCT245 Output Enable Negated Controls output enable to optional FCT245 transceiver bank by asserting during both reads and writes to a memory or I/O bank.
mem_245_dt_r_n	Output	High	Memory FCT245 Direction Xmit/Rcv Negated Recommend an external pull-up. Alternate function: cpu_dt_r_n. See CPU Core Specific Signals below.
output_clk	Output	High	Output Clock Optional clock output.

PCI Interface

pci_ad[31:0]	I/O	PCI	PCI Multiplexed Address/Data Bus Address driven by Bus Master during initial frame_n assertion, and then the Data is driven by the Bus Master during writes; or the Data is driven by the Bus Slave during reads.
pci_cbe_n[3:0]	I/O	PCI	PCI Multiplexed Command/Byte Enable Bus Command (not negated) Bus driven by the Bus Master during the initial frame_n assertion. Byte Enable Negated Bus driven by the Bus Master during the data phase(s).
pci_par	I/O	PCI	PCI Parity Even parity of the pci_ad[31:0] bus. Driven by Bus Master during Address and Write Data phases. Driven by the Bus Slave during the Read Data phase.
pci_frame_n	I/O	PCI	PCI Frame Negated Driven by the Bus Master. Assertion indicates the beginning of a bus transaction. De-assertion indicates the last datum.
pci_trdy_n	I/O	PCI	PCI Target Ready Negated Driven by the Bus Slave to indicate the current datum can complete.
pci_irdy_n	I/O	PCI	PCI Initiator Ready Negated Driven by the Bus Master to indicate that the current datum can complete.
pci_stop_n	I/O	PCI	PCI Stop Negated Driven by the Bus Slave to terminate the current bus transaction.
pci_idsel_n	Input	—	PCI Initialization Device Select Uses pci_req_n[2] pin. See the PCI subsection.
pci_perr_n	I/O	PCI	PCI Parity Error Negated Driven by the receiving Bus Agent 2 clocks after the data is received, if a parity error occurs.
pci_serr_n	I/O Open-collector	PCI	System Error Requires an external pull-up. Driven by any agent to indicate an address parity error, data parity during a Special Cycle command, or any other system error.
pci_clk	Input	—	PCI Clock Clock for PCI Bus transactions. Uses the rising edge for all timing references.
pci_rst_n	Input	—	PCI Reset Negated Host mode: Resets all PCI related logic. Satellite mode: with boot from PCI mode: Resets all PCI related logic and also warm resets the 32334.
pci_devsel_n	I/O	PCI	PCI Device Select Negated Driven by the target to indicate that the target has decoded the present address as a target address.

Table 1 Pin Descriptions (Part 2 of 7)

Name	Type	Drive Strength Capability	Description
pci_req_n[2]	Input	—	PCI Bus Request #2 Negated Requires an external pull-up. Host mode: pci_req_n[2] is an input indicating a request from an external device. Satellite mode: used as pci_idsel pin which selects this device during a configuration read or write. Alternate function: pci_idsel (satellite).
pci_req_n[1]	Input	—	PCI Bus Request #1 Negated Requires an external pull-up. Host mode: pci_req_n[2] is an input indicating a request from an external device. Alternate function: Unused (satellite).
pci_req_n[0]	I/O	High	PCI Bus Request #0 Negated Requires an external pull-up for burst mode. Host mode: pci_req_n[0] is an input indicating a request from an external device. Satellite mode: pci_req_n[0] is an output indicating a request from this device.
pci_gnt_n[2]	Output	High	PCI Bus Grant #2 Negated Recommend an external pull-up. Host mode: pci_gnt_n[2] is an output indicating a grant to an external device. Satellite mode: pci_gnt_n[2] is used as the pci_inta_n output pin. Alternate function: pci_inta_n (satellite).
pci_gnt_n[1] / pci_eeprom_cs	I/O	High	PCI Bus Grant #1 Negated Recommend an external pull-up. Host mode: pci_gnt_n[2:1] are outputs indicating grants to external devices. Satellite mode: Used as pci_eprom_cs output pin for Serial Chip Select for loading PCI Configuration Registers in the RC32334 Reset Initialization Vector PCI boot mode. Defaults to the output direction at reset time. 1st Alternate function: pci_eeprom_cs (satellite). 2nd Alternate function: PIO[11].
pci_gnt_n[0]	I/O	High	PCI Bus Grant #0 Negated Host mode: pci_gnt_n[0] is an output indicating a grant to an external device. Recommend external pullup. Satellite mode: pci_gnt_n[0] is an input indicating a grant to this device. Require external pullup.
pci_inta_n	Output Open- collector	PCI	PCI Interrupt #A Negated Uses pci_gnt_n[2]. See the PCI subsection.
pci_lock_n	Input	—	PCI Lock Negated Driven by the Bus Master to indicate that an exclusive operation is occurring.

SDRAM Control Interface

sdram_addr_12	Output	High	SDRAM Address Bit 12 and Precharge All SDRAM mode: Provides SDRAM address bit 12 (10 on the SDRAM chip) during row address and "pre-charge all" signal during refresh, read and write command.
sdram_ras_n	Output	High	SDRAM RAS Negated SDRAM mode: Provides SDRAM RAS control signal to all SDRAM banks.
sdram_cas_n	Output	High	SDRAM CAS Negated SDRAM mode: Provides SDRAM CAS control signal to all SDRAM banks.
sdram_we_n	Output	High	SDRAM WE Negated SDRAM mode: Provides SDRAM WE control signal to all SDRAM banks.
sdram_cke	Output	High	SDRAM Clock Enable SDRAM mode: Provides clock enable to all SDRAM banks.
sdram_cs_n[3:0]	Output	High	SDRAM Chip Select Negated Bus Recommend an external pull-up. SDRAM mode: Provides chip select to each SDRAM bank. SODIMM mode: Provides upper select byte enables [7:4].
sdram_s_n[1:0]	Output	High	SDRAM SODIMM Select Negated Bus SDRAM mode: Not used. SDRAM SODIMM mode: Upper and lower chip selects.

Table 1 Pin Descriptions (Part 3 of 7)

Name	Type	Drive Strength Capability	Description
sdram_bemask_n[3:0]	Output	High	SDRAM Byte Enable Mask Negated Bus (DQM) SDRAM mode: Provides byte enables for each byte lane of all DRAM banks. SODIMM mode: Provides lower select byte enables [3:0].
sdram_245_oe_n	Output	Low	SDRAM FCT245 Output Enable Negated Recommend an external pull-up. SDRAM mode: Controls output enable to optional FCT245 transceiver bank by asserting during both reads and writes to any DRAM bank.
sdram_245_dt_r_n	Output	High	SDRAM FCT245 Direction Transmit/Receive Recommend an external pull-up. Uses cpu_dt_r_n. See CPU Core Specific Signals below.

On-Chip Peripherals

dma_ready_n[1:0] / dma_done_n[1:0]	I/O	Low	DMA Ready Negated Bus Requires an external pull-up. Ready mode: Input pin for each general purpose DMA channel that can initiate the next datum in the current DMA descriptor frame. Done mode: Input pin for each general purpose DMA channel that can terminate the current DMA descriptor frame. dma_ready_n[0] 1st Alternate function PIO[1]; 2nd Alternate function: dma_done_n[0]. dma_ready_n[1] 1st Alternate function PIO[0]; 2nd Alternate function: dma_done_n[1].
pio[15:0]	I/O	Low	Programmable Input/Output General purpose pins that can each be configured as a general purpose input or general purpose output. These pins are multiplexed with other pin functions: uart_cts_n[0], uart_dsr_n[0], uart_dtr_n[0], uart_rts_n[0], pci_gnt_n[1], spi_mosi, spi_sck, spi_ss, spi_miso, uart_rx[0], uart_tx[0], uart_rx[1], uart_tx[1], timer_tc_n[0], dma_ready_n[0], dma_ready_n[1]. Note that spi_mosi, spi_miso, spi_sck, and spi_ss default to outputs at reset time. The others default to inputs.
timer_tc_n[0] / timer_gate_n[0]	I/O	Low	Timer Terminal Count Overflow Negated Terminal count mode (timer_tc_n): Output indicating that the timer has reached its count compare value and has overflowed back to 0. Gate mode (timer_gate_n): input indicating that the timer may count one tick on the next clock edge. 1st Alternate function: PIO[2]. 2nd Alternate function: timer_gate_n[0].
uart_rx[1:0]	I/O	Low	UART Receive Data Bus UART mode: Each UART channel receives data on their respective input pin. uart_rx[0] Alternate function: PIO[6]. uart_rx[1] Alternate function: PIO[4].
uart_tx[1:0]	I/O	Low	UART Transmit Data Bus Recommend an external pull-up. UART mode: Each UART channel sends data on their respective output pin. Note that these pins default to inputs at reset time and must be programmed via the PIO interface before being used as UART outputs. uart_tx[0] Alternate function: PIO[5]. uart_tx[1] Alternate function: PIO[3].
uart_cts_n[0] uart_dsr_n[0] uart_dtr_n[0] uart_rts_n[0]	I/O	Low	UART Transmit Data Bus UART mode: Data bus modem control signal pins for UART channel 0. uart_cts_n[0] Alternate function: PIO[15]. uart_dsr_n[0] Alternate function: PIO[14]. uart_dtr_n[0] Alternate function: PIO[13]. uart_rts_n[0] Alternate function: PIO[12].

Table 1 Pin Descriptions (Part 4 of 7)

Name	Type	Drive Strength Capability	Description
spi_mosi	I/O	Low	SPI Data Output Serial mode: Output pin from RC32334 as an Input to a Serial Chip for the Serial data input stream. In PCI satellite mode, acts as an Output pin from RC32334 that connects as an Input to a Serial Chip for the Serial data input stream for loading PCI Configuration Registers in the RC32334 Reset Initialization Vector PCI boot mode. Defaults to the output direction at reset time. 1st Alternate function: PIO[10]. 2nd Alternate function: pci_eeeprom_mdo.
spi_miso	I/O	Low	SPI Data Input Serial mode: Input pin to RC32334 from the Output of a Serial Chip for the Serial data output stream. In PCI satellite mode, acts as an Input pin from RC32334 that connects as an output to a Serial Chip for the Serial data output stream for loading PCI Configuration Registers in the RC32334 Reset Initialization Vector PCI boot mode. Defaults to input direction at reset time. 1st Alternate function: PIO[7]. 2nd Alternate function: pci_eeeprom_mdi.
spi_sck	I/O	Low	SPI Clock Serial mode: Output pin for Serial Clock. In PCI satellite mode, acts as an Output pin for Serial Clock for loading PCI Configuration Registers in the RC32334 Reset Initialization Vector PCI boot mode. Defaults to the output direction at reset time. 1st Alternate function: PIO[9]. 2nd Alternate function: pci_eeeprom_sk.
spi_ss_n	I/O	Low	SPI Chip Select Output pin selecting the serial protocol device as opposed to the PCI satellite mode EEPROM device. Alternate function: PIO[8]. Defaults to the output direction at reset time.

CPU Core Specific Signals

cpu_nmi_n	Input	—	CPU Non-Maskable Interrupt Requires an external pullup. This interrupt input is active low to the CPU.
cpu_masterclk	Input	—	CPU Master System Clock Provides the basic system clock.
cpu_int_n[5:4], [2:0]	Input	—	CPU Interrupt Requires an external pullup. These interrupt inputs are active low to the CPU.
cpu_coldreset_n	Input	—	CPU Cold Reset This active-low signal is asserted to the RC32334 after V_{cc} becomes valid on the initial power-up. The Reset initialization vectors for the RC32334 are latched by cold reset.
cpu_dt_r_n	Output	—	CPU Direction Transmit/Receive This active-low signal controls the DT/R pin of an optional FCT245 transceiver bank. It is asserted during read operations. 1st Alternate function: mem_245_dt_r_n. 2nd Alternate function: sdram_245_dt_r_n.

Table 1 Pin Descriptions (Part 5 of 7)

Name	Type	Drive Strength Capability	Description
JTAG Interface Signals			
jtag_tck	Input	—	JTAG Test Clock Requires an external pull-down. An input test clock used to shift into or out of the Boundary-Scan register cells. jtag_tck is independent of the system and the processor clock with nominal 50% duty cycle.
jtag_tdi, ejtag_dint_n	Input	—	JTAG Test Data In Requires an external pull-up. On the rising edge of jtag_tck, serial input data are shifted into either the Instruction or Data register, depending on the TAP controller state. During Real Mode, this input is used as an interrupt line to stop the debug unit from Real Time mode and return the debug unit back to Run Time Mode (standard JTAG). This pin is also used as the ejtag_dint_n signal in the EJTAG mode.
jtag_tdo, ejtag_tpc	Output	High	JTAG Test Data Out The jtag_tdo is serial data shifted out from instruction or data register on the falling edge of jtag_tck. When no data is shifted out, the jtag_tdo is tri-stated. During Real Time Mode, this signal provides a non-sequential program counter at the processor clock or at a division of processor clock. This pin is also used as the ejtag_tpc signal in the EJTAG mode.
jtag_tms	Input	—	JTAG Test Mode Select Requires an external pull-up. The logic signal received at the jtag_tms input is decoded by the TAP controller to control test operation. jtag_tms is sampled on the rising edge of the jtag_tck.
jtag_trst_n	Input	—	JTAG Test Reset The jtag_trst_n pin is an active-low signal for asynchronous reset of the debug unit, independent of the processor logic. An external pull-up on the board is recommended to meet the JTAG specification in cases where the tester can not access this signal, however, specific systems ordinarily should either 1) drive low this signal 2) use an external pulldown on the board 3) clock jtag_tclk
ejtag_dclk	Output	—	EJTAG Test Clock Processor Clock. During Real Time Mode, this signal is used to capture address and data from the ejtag_tpc signal at the processor clock speed or any division of the internal pipeline.
ejtag_pcst[2:0]	I/O	Low	EJTAG PC Trace Status Information 111 (STL) Pipe line Stall 110 (JMP) Branch/Jump forms with PC output 101 (BRT) Branch/Jump forms with no PC output 100 (EXP) Exception generated with an exception vector code output 011 (SEQ) Sequential performance 010 (TST) Trace is outputted at pipeline stall time 001 (TSQ) Trace trigger output at performance time 000 (DBM) Run Debug Mode Alternate function: modebit[2:0].
ejtag_debugboot	Input	—	EJTAG DebugBoot Requires an external pull-down. The ejtag_debugboot input is used during reset and forces the CPU core to take a debug exception at the end of the reset sequence instead of a reset exception. This enables the CPU to boot from the ICE probe without having the external memory working. This input signal is level sensitive and is not latched internally. This signal will also set the JtagBrk bit in the JTAG_Control_Register[12].
ejtag_tms	Input	—	EJTAG Test Mode Select Requires an external pull-up. The ejtag_tms is sampled on the rising edge of jtag_tck.

Table 1 Pin Descriptions (Part 6 of 7)

Name	Type	Drive Strength Capability	Description
Debug Signals			
debug_cpu_dma_n	I/O	Low	Debug CPU versus DMA Negated De-assertion high during debug_cpu_ads_n assertion or debug_cpu_ack_n assertion indicates transaction was generated from the CPU. Assertion low during debug_cpu_ads_n assertion or debug_cpu_ack_n assertion indicates transaction was generated from DMA. Alternate function: modebit[6].
debug_cpu_ack_n	I/O	Low	Debug CPU Acknowledge Negated Indicates either a data acknowledge to the CPU or DMA. Alternate function: modebit[4].
debug_cpu_ads_n	I/O	Low	Debug CPU Address/Data Strobe Negated Assertion indicates that either a CPU or a DMA transaction is beginning and that the mem_data[31:4] bus has the current block address. Alternate function: modebit[5].
debug_cpu_i_d_n	I/O	Low	Debug CPU Instruction versus Data Negated Assertion during debug_cpu_ads_n assertion or debug_cpu_ack_n assertion indicates transaction is a CPU or DMA data transaction. De-assertion during debug_cpu_ads_n assertion or debug_cpu_ack_n assertion indicates transaction is a CPU instruction transaction. Alternate function: modebit[3].

Table 1 Pin Descriptions (Part 7 of 7)

Mode Bit Settings to Configure Controller on Reset

The following table lists the mode bit settings to configure the controller on reset.

Pin	Mode Bit	Description	Value	Mode Setting
ejtag_pcst[2:0]	2:0 MSB (2)	Clock Multiplier MasterClock is multiplied internally to generate PClock	0	Multiply by 2
			1	Multiply by 3
			2	Multiply by 4
			3	Reserved
			4	Reserved
			5	Reserved
			6	Reserved
			7	Reserved
debug_cpu_i_d_n	3	EndBit	0	Little-endian ordering
			1	Big-endian ordering
debug_cpu_ack_n	4	Reserved	0	
debug_cpu_ads_n	5	Reserved	0	
debug_cpu_dma_n	6	TmrIntEn Enables/Disables the timer interrupt on Int*[5]	0	Enables timer interrupt
			1	Disables timer interrupt
mem_addr[17]	7	Reserved for future use	1	

Table 2 Boot-Mode Configuration Settings

Pin	Mode Bit	Description	Value	Mode Setting
mem_addr[19:18]	9:8 MSB (9)	Boot-Prom Width specifies the memory port width of the memory space which contains the boot prom.	00	8 bits
			01	16 bits
			10	32 bits
			11	Reserved

Table 2 Boot-Mode Configuration Settings

reset_boot_mode Settings

By using the non-boot mode reset initialization mode the user can change the internal register addresses from base 1800_0000 to base 1900_0000, as required. The RC32334 reset-boot mode initialization setting values and mode descriptions are listed below.

Pin	Reset Boot Mode	Description	Value	Mode Settings
mem_addr[22:21]	1:0 MSB (1)	Reserved	11	
		Reserved	10	
		PCI-boot mode (pci_host_mode must be in satellite mode) RC32334 will reset either from a cold reset or from a PCI reset. Boot code is provided via PCI.	01	PCI_boot_mode
		Standard-boot mode Boot from the RC32334's memory controller (typical system).	00	standard_boot_mode

Table 3 RC32334 reset_boot_mode Initialization Settings

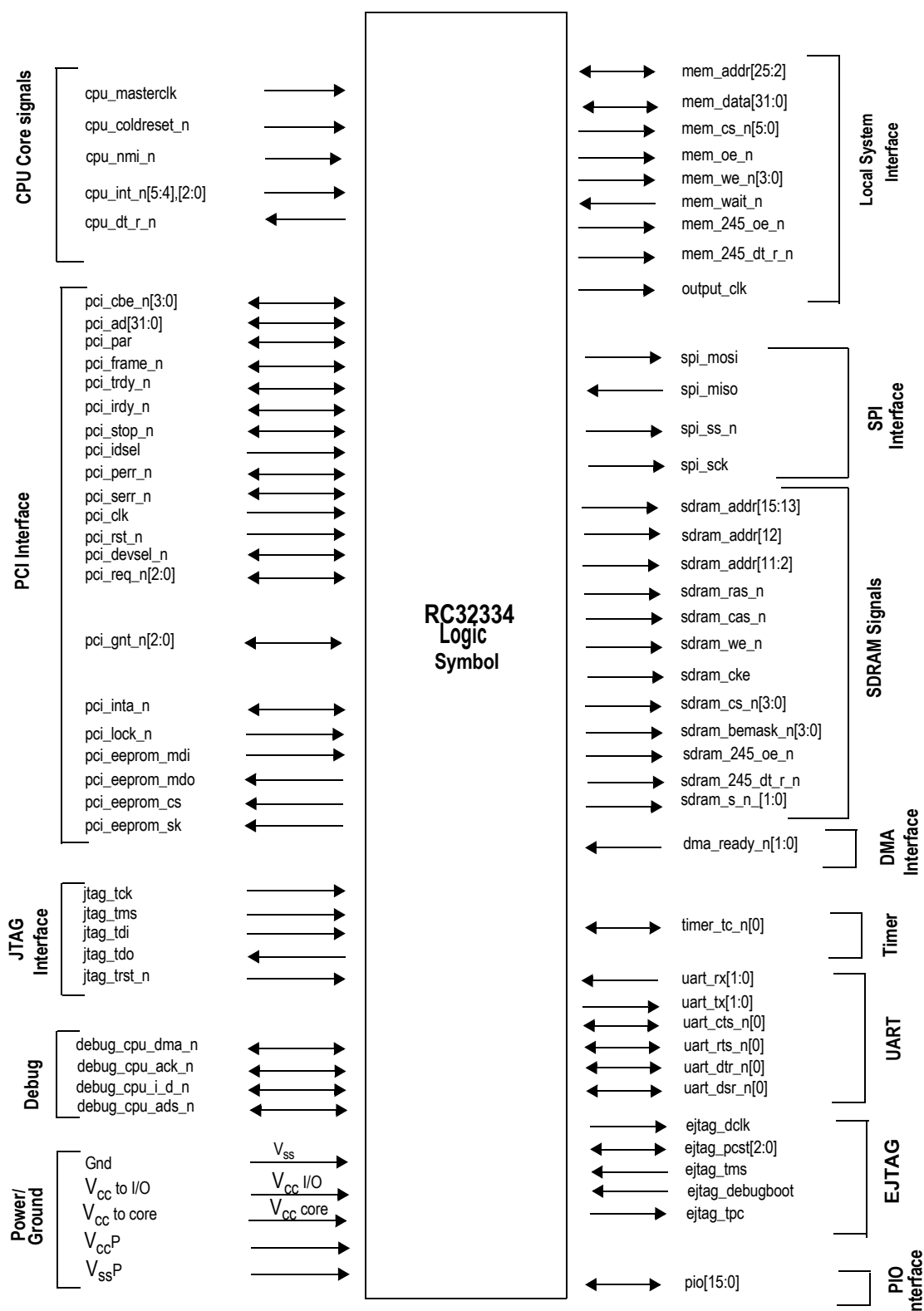
pci_host_mode Settings

During reset initialization, the RC32334's PCI interface can be set to the Satellite or Host mode settings. When set to the Host mode, the CPU must configure the RC32334's PCI configuration registers, including the read-only registers. If the RC32334's PCI is in the PCI-boot mode Satellite mode, read-only configuration registers are loaded by the serial EEPROM.

Pin	Reset Boot Mode	Description	Value	Mode Settings
mem_addr[20]	PCI host mode	PCI is in satellite mode	1	PCI_satellite
		PCI is in host mode (typical system)	0	PCI_host

Table 4 RC32334 pci_host_mode Initialization Settings

Logic Diagram — RC32334



Clock Parameters — RC32334

(Ta = 0°C to +85°C Commercial, Ta = -40°C to +85°C Industrial, V_{cc} I/O = +3.3V±5%, V_{cc} Core = +3.3V±5%)

Parameter	Symbol	Test Conditions	RC32334 100MHz		RC32334 133MHz		RC32334 150MHz		Units
			Min	Max	Min	Max	Min	Max	
cpu_masterclock HIGH	t _{MCHIGH}	Transition ≤ 2ns	8	—	6.75	—	6	—	ns
cpu_masterclock LOW	t _{MCLOW}	Transition ≤ 2ns	8	—	6.75	—	6	—	ns
cpu_masterclock period ¹	t _{MCP}	—	20	66.6	15	66.6	13.33	66.6	ns
cpu_masterclock Rise & Fall Time ²	t _{MCRise} , t _{MCFall}	—	—	3	—	3	—	3	ns
cpu_masterclock Jitter	t _{JITTER}	—	—	± 250	—	± 250	—	± 200	ps
pci_clk Rise & Fall Time	t _{PCRise} , t _{PCFall}	PCI 2.1	—	1.6	—	1.6	—	1.6	ns
pci_clk Period ¹	t _{PCP}	—	15	—	15	—	15	—	ns
jtag_tck Rise & Fall Time	t _{JCRise} , t _{JCFall}	—	—	5	—	5	—	5	ns
ejtag_dclk period	t _{DCK} , t _{t1}	—	10	—	10	—	10	—	ns
jtag_tck clock period	t _{TCK} , t ₃	—	100	—	100	—	100	—	ns
ejtag_dclk High, Low Time	t _{DCK High} , t _g t _{DCK Low} , t ₁₀	—	4	—	4	—	4	—	ns
ejtag_dclk Rise, Fall Time	t _{DCK Rise} , t _g t _{DCK Fall} , t ₁₀	—	—	1	—	1	—	1	ns
output_clk ³	Tdo21	—	N/A	N/A	N/A	N/A	N/A	N/A	—
cpu_coldreset_n Asserted during power-up		power-on sequence	120	—	120	—	120	—	ms

Table 5 Clock Parameters - RC32334

¹ cpu_masterclock should never be below pci_clk if PCI interface is used.

² Rise and fall times are measured between 10% and 90%

³ Output_clk should not be used in a system. Only the cpu_masterclock or its derivative must be used to drive all the subsystems with designs based on the RC32334/RC32332. Refer to the RC32334/RC32332 Device Errata for more information.

Reset Specification

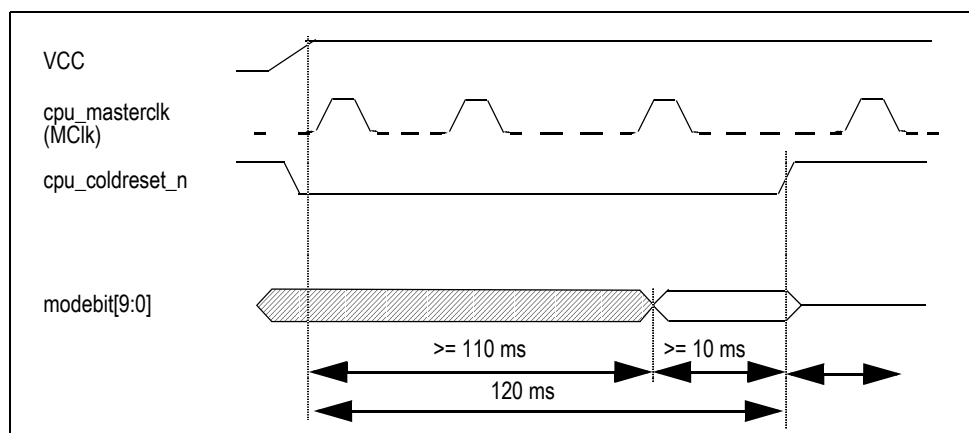


Figure 3 Mode Configuration Interface Reset Sequence

Power Ramp-up

There is no special requirement for how fast Vcc and VccP ramp up to 3.3V. However, all timing references are based on Vcc and VccP stabilized at 3.3V -5%.

AC Timing Characteristics — RC32334

(Ta = 0°C to +85°C Commercial, Ta = -40°C to +85°C Industrial, V_{cc} I/O = +3.3V±5%, V_{cc} Core = +3.3V±5%)

Signal	Symbol	Reference Edge	RC32334 ¹ 100MHz		RC32334 ¹ 133MHz		RC32334 ¹ 150MHz		Unit	User Manual Timing Diagram Reference
			Min	Max	Min	Max	Min	Max		

Local System Interface

mem_data[31:0] (data phase)	Tsu2	cpu_masterclk rising	6	—	5	—	4.8	—	ns	Chapter 9, Figures 9.2 and 9.3
mem_data[31:0] (data phase)	Thld2	cpu_masterclk rising	1.5	—	1.5	—	1.5	—	ns	
cpu_dt_r_n	Tdo3	cpu_masterclk rising	—	15	—	12	—	10	ns	
mem_data[31:0]	Tdo4	cpu_masterclk rising	—	12	—	10	—	9.3	ns	
mem_data[31:0] output hold time	Tdoh1	cpu_masterclk rising	1	—	1	—	1	—	ns	
mem_data[31:0] (tristate disable time)	Tdz	cpu_masterclk rising	—	12 ²	—	10 ²	—	9.3 ²	ns	
mem_data[31:0] (tristate to data time)	Tzd	cpu_masterclk rising	—	12 ²	—	10 ²	—	9.3 ²	ns	
mem_wait_n	Tsu6	cpu_masterclk rising	9	—	7	—	6	—	ns	Chapter 10, Figures 10.6 through 10.8
mem_wait_n	Thld8	cpu_masterclk rising	1	—	1	—	1	—	ns	
mem_addr[25:2]	Tdo5	cpu_masterclk rising	—	12	—	9	—	8	ns	
mem_cs_n[5:0]	Tdo6	cpu_masterclk rising	—	12	—	9	—	8	ns	
mem_oe_n, mem_245_oe_n	Tdo7	cpu_masterclk rising	—	12	—	9	—	8	ns	
mem_we_n[3:0]	Tdo7a	cpu_masterclk rising	—	15	—	12	—	10	ns	
mem_245_dt_r_n	Tdo8	cpu_masterclk rising	—	15	—	12	—	10	ns	
mem_addr[25:2] mem_cs_n[5:0] mem_oe_n, mem_we_n[3:0], mem_245_dt_r_n, mem_245_oe_n	Tdoh3	cpu_masterclk rising	1.5	—	1.5	—	1.5	—	ns	

PCI

pci_ad[31:0], pci_cbe_n[3:0], pci_par, pci_frame_n, pci_trdy_n, pci_irdy_n, pci_stop_n, pci_perr_n, pci_serr_n, pci_devsel_n, pci_lock_n ³	Tsu	pci_clk rising	3	—	3	—	3	—	ns	Per PCI 2.1
pci_idsel, pci_req_n[2], pci_req_n[1], pci_req_n[0], pci_gnt_n[0], pci_inta_n	Tsu	pci_clk rising	5	—	5	—	5	—	ns	
pci_gnt_n[0]	Tsu	pci_clk rising	5	—	5	—	5	—	ns	
pci_ad[31:0], pci_cbe_n[3:0], pci_par, pci_frame_n, pci_trdy_n, pci_irdy_n, pci_stop_n, pci_perr_n, pci_serr_n, pci_rst_n, pci_devsel_n, pci_lock_n ³	Thld	pci_clk rising	1	—	1	—	1	—	ns	
pci_idsel, pci_req_n[2], pci_req_n[1], pci_req_n[0], pci_gnt_n[0], pci_inta_n	Thld	pci_clk rising	1	—	1	—	1	—	ns	
pci_eeprom_mdi	Tsu	pci_clk rising, pci_eeprom_sk falling	15	—	12	—	10	—	ns	

Table 6 AC Timing Characteristics - RC32334 (Part 1 of 4)

Signal	Symbol	Reference Edge	RC32334 ¹ 100MHz		RC32334 ¹ 133MHz		RC32334 ¹ 150MHz		Unit	User Manual Timing Diagram Reference
			Min	Max	Min	Max	Min	Max		
pci_eeeprom_mdi	Thld	pci_clk rising, pci_eeeprom_sk falling	15	—	12	—	10	—	ns	
pci_eeeprom_mdo, pci_eeeprom_cs	Tdo	pci_clk rising, pci_eeeprom_sk falling	—	15	—	12	—	10	ns	
pci_eeeprom_sk	Tdo	pci_clk rising	—	15	—	12	—	10	ns	
pci_ad[31:0], pci_cbe_n[3:0], pci_par, pci_frame_n, pci_trdy_n, pci_irdy_n, pci_stop_n, pci_perr_n, pci_serr_n, pci_devsel_n	Tdo	pci_clk rising	2	6	2	6	2	6	ns	Per PCI 2.1
pci_req_n[0], pci_gnt_[2], pci_gnt_n[1], pci_gnt_n[0], pci_inta_n	Tdo	pci_clk rising	2	6	2	6	2	6	ns	

SDRAM Controller

sdrām_245_dt_r_n	Tdo8	cpu_masterclk rising	—	15	—	12	—	10	ns	Chapter 11, Figures 11.4 and 11.5
sdrām_ras_n, sdrām_cas_n, sdrām_we_n, sdrām_cs_n[3:0], sdrām_s_n[1:0], sdrām_bemask_n[3:0], sdrām_cke	Tdo9	cpu_masterclk rising	—	12	—	9	—	8	ns	
sdrām_addr_12	Tdo10	cpu_masterclk rising	—	12	—	9	—	8	ns	
sdrām_245_oe_n	Tdo11	cpu_masterclk rising	—	12	—	9	—	8	ns	
sdrām_245_dt_r_n	Tdoh4	cpu_masterclk rising	1	—	1	—	1	—	ns	
sdrām_ras_n, sdrām_cas_n, sdrām_we_n, sdrām_cs_n[3:0], sdrām_s_n[1:0], sdrām_bemask_n[3:0] sdrām_cke, sdrām_addr_12, sdrām_245_oe_n	Tdoh4	cpu_masterclk rising	2.5	—	2.5	—	2.5	—	ns	

DMA

dma_ready_n[1:0], dma_done_n[1:0]	Tsu7	cpu_masterclk rising	9	—	7	—	6	—	ns	Chapter 13, Figure 13.4
dma_ready_n[1:0], dma_done_n[1:0]	Thld9	cpu_masterclk rising	2	—	2	—	2	—	ns	

Interrupt Handling

cpu_int_n[5:4], cpu_int_n[2:0], cpu_nmi_n	Tsu9	cpu_masterclk rising	9	—	9	—	6	—	ns	Chapter 14, Figure 14.12
cpu_int_n[5:4], cpu_int_n[2:0], cpu_nmi_n	Thld13	cpu_masterclk rising	1	—	1	—	1	—	ns	

PIO

PIO[15:0]	Tsu7	cpu_masterclk rising	9	—	7	—	6	—	ns	Chapter 15, Figures 15.9 and 15.10
PIO[15:0]	Thld9	cpu_masterclk rising	2	—	2	—	2	—	ns	
PIO[15:10], PIO[8:0]	Tdo16	cpu_masterclk rising	—	15	—	12	—	10	ns	
PIO[9]	Tdo19	cpu_masterclk rising	—	15	—	12	—	10	ns	
PIO[15:10], PIO[8:0]	Tdoh7	cpu_masterclk rising	1	—	1	—	1	—	ns	
PIO[9]	Tdoh7	cpu_masterclk rising	1	—	1	—	1	—	ns	

Timer

timer_tc_n[0], timer_gate_n[0]	Tsu8	cpu_masterclk rising	9	—	7	—	6	—	ns	Chapter 16, Figures 16.6 and 16.7
timer_tc_n[0], timer_gate_n[0]	Thld10	cpu_masterclk rising	2	—	2	—	2	—	ns	
timer_tc_n[0], timer_gate_n[0]	Tdo15	cpu_masterclk rising	—	15	—	12	—	10	ns	
timer_tc_n[0], timer_gate_n[0]	Tdoh6	cpu_masterclk rising	1	—	1	—	1	—	ns	

Table 6 AC Timing Characteristics - RC32334 (Part 2 of 4)

Signal	Symbol	Reference Edge	RC32334 ¹ 100MHz		RC32334 ¹ 133MHz		RC32334 ¹ 150MHz		Unit	User Manual Timing Diagram Reference
			Min	Max	Min	Max	Min	Max		
UARTs										
uart_rx[1:0], uart_tx[1:0], uart_cts_n[0], uart_dsr_n[0], uart_dtr_n[0], uart_rts_n[0]	Tsu7	cpu_masterclk rising	15	—	12	—	10	—	ns	Chapter 17, Figure 17.16
uart_rx[1:0], uart_tx[1:0], uart_cts_n[0], uart_dsr_n[0], uart_dtr_n[0], uart_rts_n[0]	Thld9	cpu_masterclk rising	15	—	12	—	10	—	ns	
uart_rx[1:0], uart_tx[1:0], uart_cts_n[0], uart_dsr_n[0], uart_dtr_n[0], uart_rts_n[0]	Tdo16	cpu_masterclk rising	—	15	—	12	—	10	ns	
uart_rx[1:0], uart_tx[1:0], uart_cts_n[0], uart_dsr_n[0], uart_dtr_n[0], uart_rts_n[0]	Tdoh8	cpu_masterclk rising	1	—	1	—	1	—	ns	
SPI Interface										
spi_clk, spi_mosi, spi_miso	Tsu7	cpu_masterclk rising	15	—	12	—	10	—	ns	Chapter 18, Figures 18.8 and 18.9
spi_clk, spi_mosi, spi_miso	Thld9	cpu_masterclk rising	15	—	12	—	10	—	ns	
spi_clk, spi_mosi, spi_miso	Tdo16	cpu_masterclk rising	—	15	—	12	—	10	ns	
spi_clk, spi_mosi, spi_miso	Tdoh8	cpu_masterclk rising	1	—	1	—	1	—	ns	
Reset										
cpu_coldreset_n	Tsu21	cpu_masterclk rising	9	—	7	—	6	—	ns	Chapter 19 Figures 19.8 and 19.9
cpu_coldreset_n	Thld21	cpu_masterclk rising	1	—	1	—	1	—	ns	
mem_addr[19:17]	Tsu10	cpu_coldreset_n rising	10	—	10	—	10	—	ms	
mem_addr[19:17]	Thld10	cpu_coldreset_n rising	1	—	1	—	1	—	ns	
mem_addr[22:20],	Tsu22	cpu_masterclk rising	9	—	7	—	6	—	ns	
mem_addr[22:20]	Thld22	cpu_masterclk rising	1	—	1	—	1	—	ns	
Debug Interface										
debug_cpu_dma_n, debug_cpu_ack_n, debug_cpu_ads_n, debug_cpu_i_d_n, ejtag_pcst[2:0]	Tsu20	cpu_coldreset_n rising	10	—	10	—	10	—	ms	Chapter 19, Figure 19.9 and Chapter 9, Figure 9.2
debug_cpu_dma_n, debug_cpu_ack_n, debug_cpu_ads_n, debug_cpu_i_d_n, ejtag_pcst[2:0]	Thld20	cpu_coldreset_n rising	1	—	1	—	1	—	ns	
debug_cpu_dma_n, debug_cpu_ack_n, debug_cpu_ads_n, debug_cpu_i_d_n	Tdo20	cpu_masterclk rising	—	15	—	12	—	10	ns	
debug_cpu_dma_n, debug_cpu_ack_n, debug_cpu_ads_n, debug_cpu_i_d_n	Tdoh20	cpu_masterclk rising	1	—	1	—	1	—	ns	
JTAG Interface										
jtag_tms, jtag_tdi, jtag_trst_n	t ₅	jtag_tck rising	10	—	10	—	10	—	ns	See Figure 4 below.
jtag_tms, jtag_tdi, jtag_trst_n	t ₆	jtag_tck rising	10	—	10	—	10	—	ns	
jtag_tdo	t ₄	jtag_tck falling	—	10	—	10	—	10	ns	
EJTAG Interface										
ejtag_tms, ejtag_debugboot	t ₅	jtag_tclk rising	4	—	4	—	4	—	ns	
ejtag_tms, ejtag_debugboot	t ₆	jtag_clk rising	2	—	2	—	2	—	ns	

Table 6 AC Timing Characteristics - RC32334 (Part 3 of 4)

Signal	Symbol	Reference Edge	RC32334 ¹ 100MHz		RC32334 ¹ 133MHz		RC32334 ¹ 150MHz		Unit	User Manual Timing Diagram Reference
			Min	Max	Min	Max	Min	Max		
jtag_tdo Output Delay Time	t _{TDOD0} , t ₄	jtag_tck falling	—	6	—	6	—	6	ns	See Figure 4 below.
jtag_tdi Input Setup Time	t _{TDIS} , t ₅	jtag_tck rising	4	—	4	—	4	—	ns	
jtag_tdi Input Hold Time	t _{TDIH} , t ₆	jtag_tck rising	2	—	2	—	2	—	ns	
jtag_trst_n Low Time	t _{TRSTLow} , t ₁₂	—	100	—	100	—	100	—	ns	
jtag_trst_n Removal Time	t _{TRSTR} , t ₁₃	jtag_tck rising	3	—	3	—	3	—	ns	
ejtag_tpc Output Delay Time	t _{TPCDO} , t ₈	ejtag_dclk rising	-1	3	-1	3	-1	3	ns	
ejtag_pcst Output Delay Time	t _{PCSTDO} , t ₇	ejtag_dclk rising	-1	3	-1	3	-1	3	ns	

Table 6 AC Timing Characteristics - RC32334 (Part 4 of 4)

¹. At all pipeline frequencies.

². Guaranteed by design.

³. pci_rst_n is tested per PCI 2.1 as an asynchronous signal.

Standard EJTAG Timing — RC32334

Figure 4 represents the timing diagram for the EJTAG interface signals.

The standard JTAG connector is a 10-pin connector providing 5 signals and 5 ground pins. For Standard EJTAG, a 24-pin connector has been chosen providing 12 signals and 12 ground pins. This guarantees elimination of noise problems by incorporating signal-ground type arrangement. Refer to the RC32334 User Reference Manual for connector pinout and mechanical specifications.

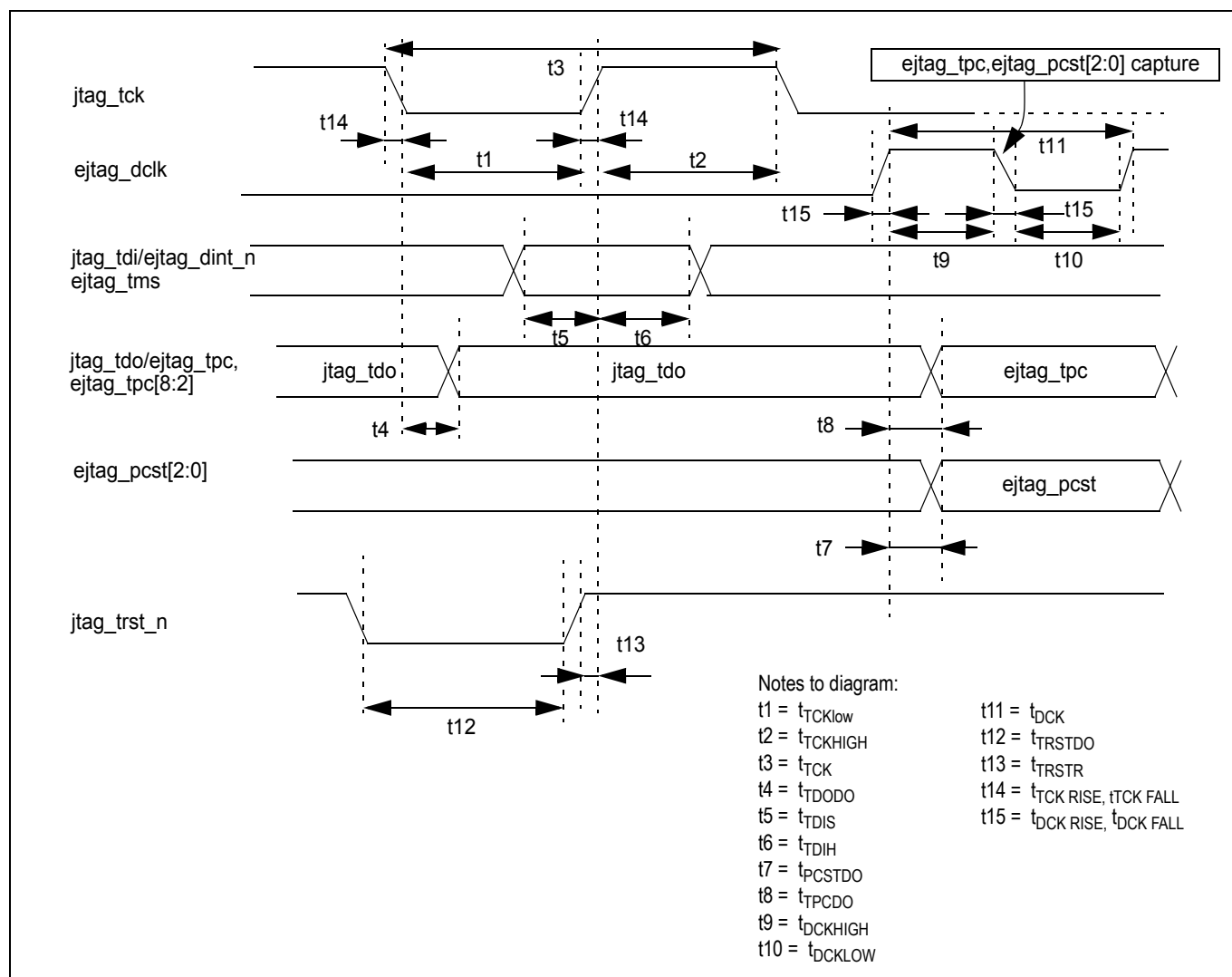
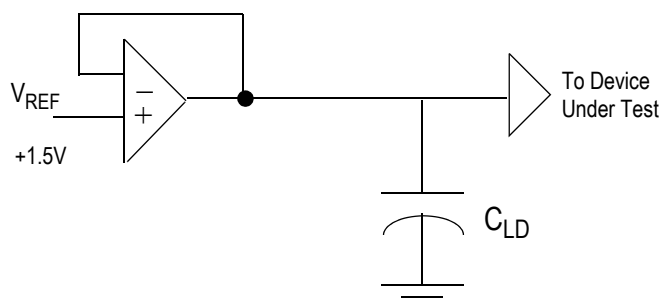


Figure 4 Standard EJTAG Timing

Output Loading for AC Testing



Signal	C _{ld}
All High Drive Signals	50 pF
All Low Drive Signals	25 pF

Figure 5 Output Loading for AC Testing

Note: PCI pins have been correlated to PCI 2.1.

Recommended Operation Temperature and Supply Voltage

Grade	Temperature	Gnd	V _{cc} IO	V _{cc} Core	V _{cc} P
Commercial	0°C to +85°C (Ambient)	0V	3.3V±5%	3.3V±5%	3.3V±5%
Industrial	-40°C to +85°C (Ambient)	0V	3.3V±5%	3.3V±5%	3.3V±5%

Table 7 Temperature and Voltage

DC Electrical Characteristics — RC32334

Commercial Temperature Range—RC32334

(T_a = 0°C to +85°C Commercial, T_a = -40°C to +85°C Industrial, V_{cc} I/O = +3.3V±5%, V_{cc} Core = +3.3V±5%)

	Parameter	RC32334		Pin Numbers	Conditions
		Minimum	Maximum		
LOW Drive Output-Pads	V _{OL}	—	0.4V	A1, A12, A15, A16, B1, B2, B11, B12, B15, C1-C3, C12, C13, C14, D12, D13, E1-E4, F1, F2, G1-G4, H1, H2, J1, J2, K2-K4, L1, L3, L4, P3, P14, R2, R15, R16, T16	I _{OUT} = 6mA
	V _{OH}	V _{cc} - 0.4V	—		I _{OUT} = 8mA
	V _{IL}	—	0.8V		—
	V _{IH}	2.0V	—		—
HIGH Drive Output-Pads	V _{OL}	—	0.4V	A2-A4, A6-A11, A13, A14, B3, B4, B6-B10, B13, B16, C4, C6-C8, C10, C11, C15, C16, D1-D4, D6, D7, D10, D11, D14-D16, E14, E15, F3, F13-F16, G13-G16, H15, H16, J13, J14, K5, K13, K14, K16, L13-L16, M2, M13, M16, P2, P4, R1, R3, R4	I _{OUT} = 7mA
	V _{OH}	V _{cc} - 0.4V	—		I _{OUT} = 16mA
	V _{IL}	—	0.8V		—
	V _{IH}	2.0V	—		—
PCI Drive Output-Pads	V _{OL}	—	—	M15, N4-N7, N10-N16, P5-P13, P15, P16, R5-R9, R11-R14, T4-T15	Per PCI 2.1
	V _{OH}	—	—		
	V _{IL}	—	—		
	V _{IH}	—	—		

Table 8 DC Electrical Characteristics - RC32334 (Part 1 of 2)

	Parameter	RC32334		Pin Numbers	Conditions
		Minimum	Maximum		
	C_{IN}	—	10pF	All except R3, T3	—
	C_{IN}	5pF	12pF	T3	Per PCI 2.1
	C_{IN}		8pF	R3	Per PCI 2.1
	C_{OUT}	—	10pF	All output pads	—
	I/O_{LEAK}	—	10 μ A	All non-internal pull-up pins	Input/Output Leakage
	I/O_{LEAK}	—	50 μ A	All internal pull-up pins	Input/Output Leakage

Table 8 DC Electrical Characteristics - RC32334 (Part 2 of 2)

Capacitive Load Deration — RC32334

Refer to the IDT document “RC32334 IBIS Model” under sub-category RC32334 Integrated Processor on the company’s web page for Processors (<http://www.idt.com/products/pages/Processors.html>).

Power Consumption — RC32334

Note: This table is based on a 2:1 pipeline-to-bus clock ratio.

Parameter		100MHz RC32334		133MHz RC32334		150MHz RC32334		Conditions
		Typical	Max.	Typical	Max.	Typical	Max.	
I_{CC}	(mA) Normal mode	360	480	480	630	550	700	C_L = (See Figure 5, Output Loading for AC Testing) T_a = 25°C V_{CC} core = 3.46V (for max. values) V_{CC} I/O = 3.46V (for max. values) V_{CC} core = 3.3V (for typical values) V_{CC} I/O = 3.3V (for typical values)
	(mA) Standby mode ¹	250	370	330	480	390	540	
P	Power dissipation (w) Normal mode	1.2	1.7	1.5	2.2	1.7	2.4	
	Power dissipation (w) Standby mode ¹	.87	1.3	1.1	1.7	1.3	1.9	

Table 9 Power Consumption

¹. RISCore 32300 CPU core enters Standby mode by executing WAIT instructions. On-chip logic outside the CPU core continues to function.

Power Curves

The following two graphs contain the simulated power curves that show power consumption at various bus frequencies.

Note: Only pipeline frequencies that are integer multiples (2x, 3x, 4x) of bus frequencies are supported.

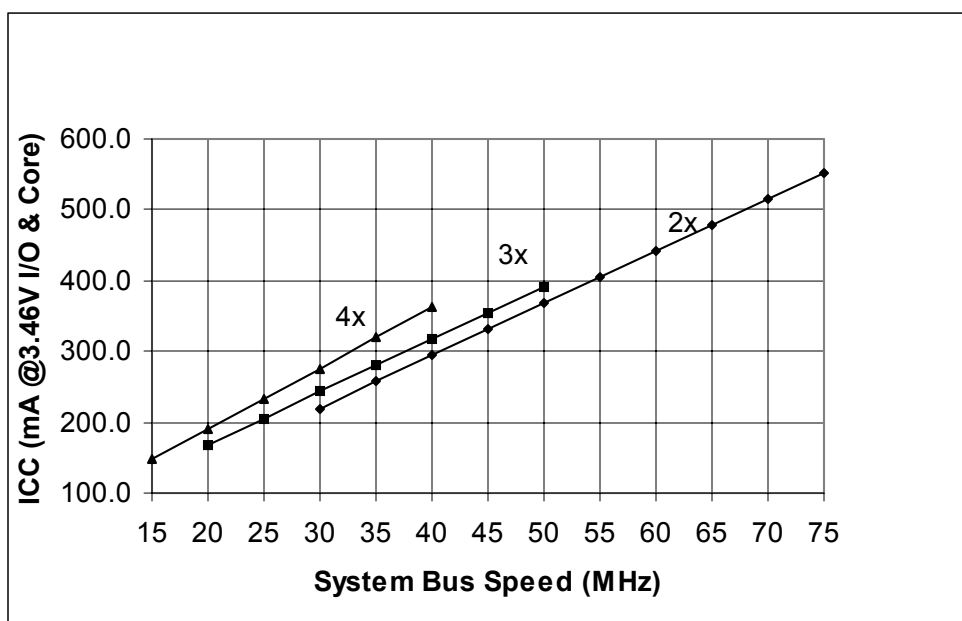


Figure 6 Typical Power Usage - RC32334

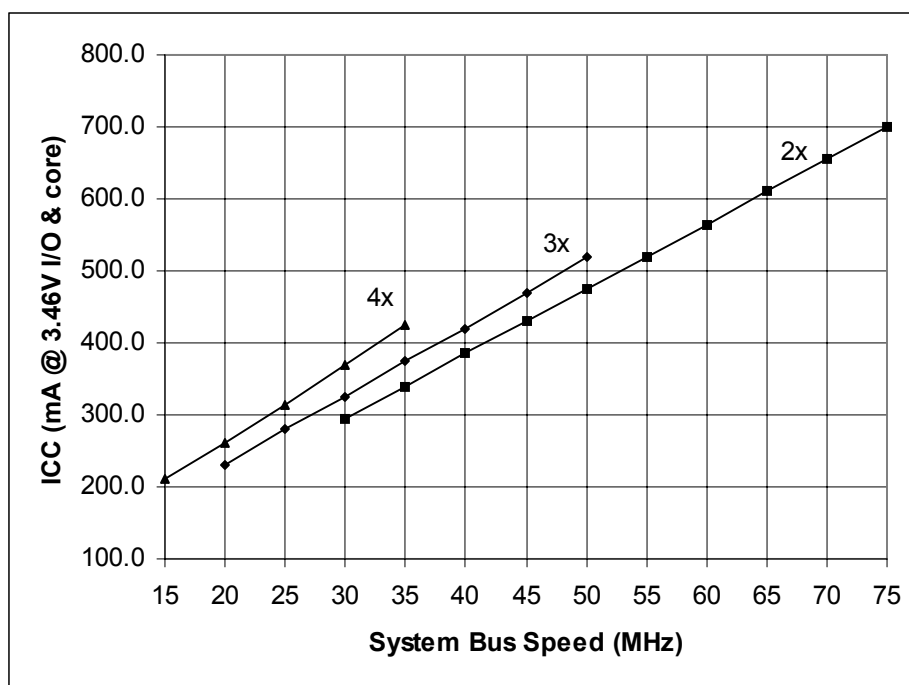


Figure 7 Maximum Power Usage - RC32334

Absolute Maximum Ratings

Symbol	Parameter	Min ¹	Max ¹	Unit
V _{CC}	Supply Voltage	-0.3	3.465	V
V _i	Input Voltage	-0.3	5.5	V
V _{imin}	Input Voltage - undershoot ²	-0.6	—	V
T _a	Ambient Operating Temperature	0	70	degrees C
T _{stg}	Storage Temperature	-40	125	degrees C

Table 10 Absolute Maximum Ratings

¹ Functional and tested operating conditions are given in Table 7. Absolute maximum ratings are stress ratings only, and functional operation at the maximums is not guaranteed. Stresses beyond those listed may affect device reliability or cause permanent damage to the device.

² All PCI pads are fully compatible with PCI Specification version 2.1.

Package Pin-out — 256-PBGA Pinout for RC32334

The following table lists the pin numbers and signal names for the RC32334. Signal names ending with an “_n” are active when low.

Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt
A1	uart_cts_n[0]	1	E1	mem_cs_n[4]		J1	debug_cpu_dma_n	1	N1	cpu_int_n[1]	
A2	sdram_245_oe_n		E2	mem_cs_n[5]		J2	debug_cpu_ack_n	1	N2	cpu_int_n[0]	
A3	sdram_cas_n		E3	mem_cs_n[3]		J3	V _{CC} IO		N3	jtag_tdi	
A4	sdram_bemask_n[1]		E4	mem_cs_n[2]		J4	V _{SS}		N4	pci_ad[30]	
A5	sdram_ras_n		E5	V _{CC} IO		J5	V _{CC} IO		N5	pci_ad[26]	
A6	mem_addr[3]	1	E6	V _{CC} IO		J6	V _{SS}		N6	pci_ad[23]	
A7	mem_addr[7]	1	E7	V _{CC} IO		J7	V _{SS}		N7	pci_ad[19]	
A8	mem_addr[11]	1	E8	V _{CC} IO		J8	V _{SS}		N8	V _{CC} core	
A9	sdram_cke		E9	V _{CC} IO		J9	V _{SS}		N9	V _{SS}	
A10	sdram_bemask_n[2]		E10	V _{CC} IO		J10	V _{SS}		N10	pci_trdy_n	
A11	mem_addr[15]	1	E11	V _{CC} IO		J11	V _{SS}		N11	pci_perr_n	
A12	mem_addr[19]	1	E12	V _{CC} IO		J12	V _{CC} IO		N12	pci_ad[15]	
A13	mem_data[10]		E13	cpu_masterclk		J13	mem_data[26]		N13	pci_ad[1]	
A14	mem_data[20]		E14	mem_data[15]		J14	mem_data[5]		N14	pci_ad[3]	
A15	mem_addr[23]		E15	mem_data[16]		J15	V _{CC} core		N15	pci_ad[4]	
A16	timer_tc_n[0]	2	E16	V _{CC} core		J16	V _{SS}		N16	pci_ad[2]	
B1	uart_rts_n[0]	1	F1	mem_cs_n[0]		K1	ejtag_debugboot		P1	pci_rst_n	
B2	uart_dsr_n[0]	1	F2	mem_cs_n[1]		K2	ejtag_dclk		P2	pci_gnt_n[2]	1
B3	sdram_we_n		F3	mem_oe_n		K3	debug_cpu_i_d_n	1	P3	dma_ready_n[1]	2
B4	sdram_bemask_n[0]		F4	mem_wait_n	1	K4	debug_cpu_ads_n	1	P4	pci_req_n[0]	
B5	sdram_cs_n[1]		F5	V _{CC} IO		K5	V _{CC} IO		P5	pci_ad[27]	
B6	mem_addr[2]	1	F6	V _{SS}		K6	V _{SS}		P6	pci_cbe_n[3]	

Table 11 RC32334 256-pin PBGA Package Pin-Out (Part 1 of 3)

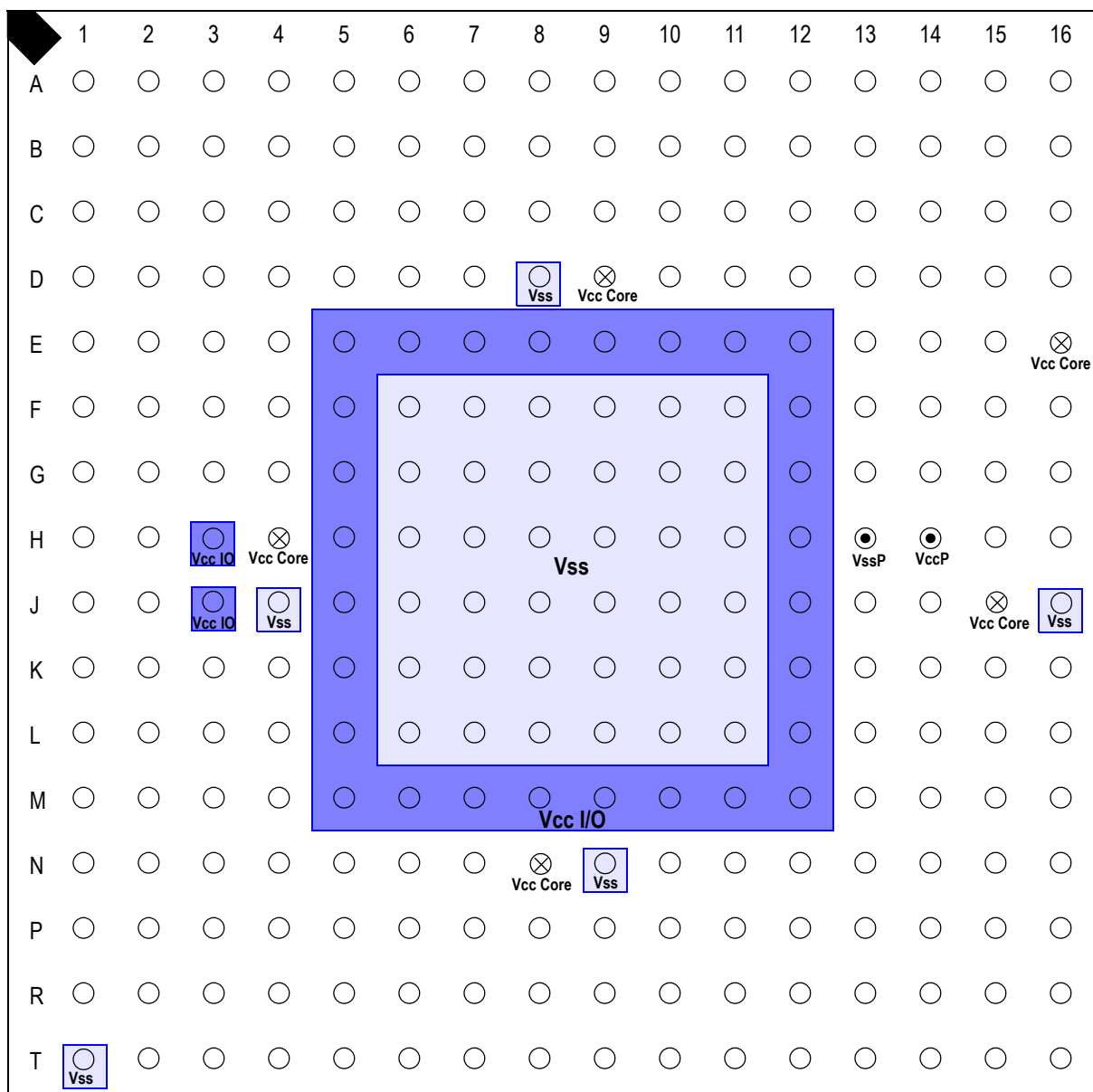
Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt
B7	mem_addr[6]	1	F7	V _{ss}		K7	V _{ss}		P7	pci_ad[20]	
B8	mem_addr[10]	1	F8	V _{ss}		K8	V _{ss}		P8	pci_ad[16]	
B9	sdram_addr_12		F9	V _{ss}		K9	V _{ss}		P9	pci_cbe_n[2]	
B10	sdram_bemask_n[3]		F10	V _{ss}		K10	V _{ss}		P10	pci_devsel_n	
B11	mem_addr[16]		F11	V _{ss}		K11	V _{ss}		P11	pci_serr_n	
B12	mem_addr[20]	1	F12	V _{cc} IO		K12	V _{cc} IO		P12	pci_ad[14]	
B13	mem_data[11]		F13	mem_data[1]		K13	cpu_dt_r_n	2	P13	pci_ad[11]	
B14	cpu_coldreset_n		F14	mem_data[30]		K14	mem_data[6]		P14	cpu_int_n[5]	
B15	mem_addr[25]		F15	mem_data[31]		K15	mem_data[24]		P15	pci_ad[6]	
B16	mem_data[12]		F16	mem_data[0]		K16	mem_data[25]		P16	pci_ad[5]	
C1	uart_rx[0]	1	G1	dma_ready_n[0]	2	L1	ejtag_pcst[0]		R1	pci_req_n[2]	1
C2	uart_tx[0]	1	G2	mem_245_oe_n		L2	jtag_trst_n		R2	cpu_int_n[2]	
C3	uart_dtr_n[0]	1	G3	spi_mosi	2	L3	ejtag_pcst[1]	1	R3	pci_gnt_n[1]	2
C4	sdram_cs_n[0]		G4	spi_miso	2	L4	ejtag_pcst[2]	1	R4	pci_gnt_n[0]	
C5	sdram_s_n[0]		G5	V _{cc} IO		L5	V _{cc} IO		R5	pci_ad[29]	
C6	mem_addr[4]	1	G6	V _{ss}		L6	V _{ss}		R6	pci_ad[25]	
C7	mem_addr[9]	1	G7	V _{ss}		L7	V _{ss}		R7	pci_ad[22]	
C8	output_clk		G8	V _{ss}		L8	V _{ss}		R8	pci_ad[18]	
C9	mem_addr[12]		G9	V _{ss}		L9	V _{ss}		R9	pci_irdy_n	
C10	sdram_cs_n[3]		G10	V _{ss}		L10	V _{ss}		R10	pci_lock_n	
C11	mem_addr[14]	1	G11	V _{ss}		L11	V _{ss}		R11	pci_cbe_n[1]	
C12	mem_addr[18]	1	G12	V _{cc} IO		L12	V _{cc} IO		R12	pci_ad[12]	
C13	mem_addr[22]	1	G13	mem_data[3]		L13	mem_data[7]		R13	pci_ad[10]	
C14	mem_addr[24]		G14	mem_data[28]		L14	mem_data[8]		R14	pci_cbe_n[0]	
C15	mem_data[19]		G15	mem_data[29]		L15	mem_data[22]		R15	uart_tx[1]	1
C16	mem_data[13]		G16	mem_data[2]		L16	mem_data[23]		R16	cpu_int_n[4]	
D1	mem_we_n[1]		H1	spi_ss_n	1	M1	jtag_tms		T1	V _{ss}	
D2	mem_we_n[3]		H2	spi_sck	2	M2	jtag_tdo		T2	pci_req_n[1]	1
D3	mem_we_n[2]		H3	V _{cc} IO		M3	ejtag_tms		T3	pci_clk	
D4	mem_we_n[0]		H4	V _{cc} core		M4	jtag_tck		T4	pci_ad[31]	
D5	sdram_s_n[1]		H5	V _{cc} IO		M5	V _{cc} IO		T5	pci_ad[28]	
D6	mem_addr[5]	1	H6	V _{ss}		M6	V _{cc} IO		T6	pci_ad[24]	
D7	mem_addr[8]	1	H7	V _{ss}		M7	V _{cc} IO		T7	pci_ad[21]	
D8	V _{ss}		H8	V _{ss}		M8	V _{cc} IO		T8	pci_ad[17]	
D9	V _{cc} core		H9	V _{ss}		M9	V _{cc} IO		T9	pci_frame_n	
D10	sdram_cs_n[2]		H10	V _{ss}		M10	V _{cc} IO		T10	pci_stop_n	
D11	mem_addr[13]	1	H11	V _{ss}		M11	V _{cc} IO		T11	pci_par	

Table 11 RC32334 256-pin PBGA Package Pin-Out (Part 2 of 3)

Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt
D12	mem_addr[17]	1	H12	V _{cc} IO		M12	V _{cc} IO		T12	pci_ad[13]	
D13	mem_addr[21]	1	H13	V _{ss} P		M13	mem_data[9]		T13	pci_ad[9]	
D14	mem_data[17]		H14	V _{cc} P		M14	cpu_nmi_n		T14	pci_ad[8]	
D15	mem_data[14]		H15	mem_data[27]		M15	pci_ad[0]		T15	pci_ad[7]	
D16	mem_data[18]		H16	mem_data[4]		M16	mem_data[21]		T16	uart_rx[1]	1

Table 11 RC32334 256-pin PBGA Package Pin-Out (Part 3 of 3)

Pin Layout



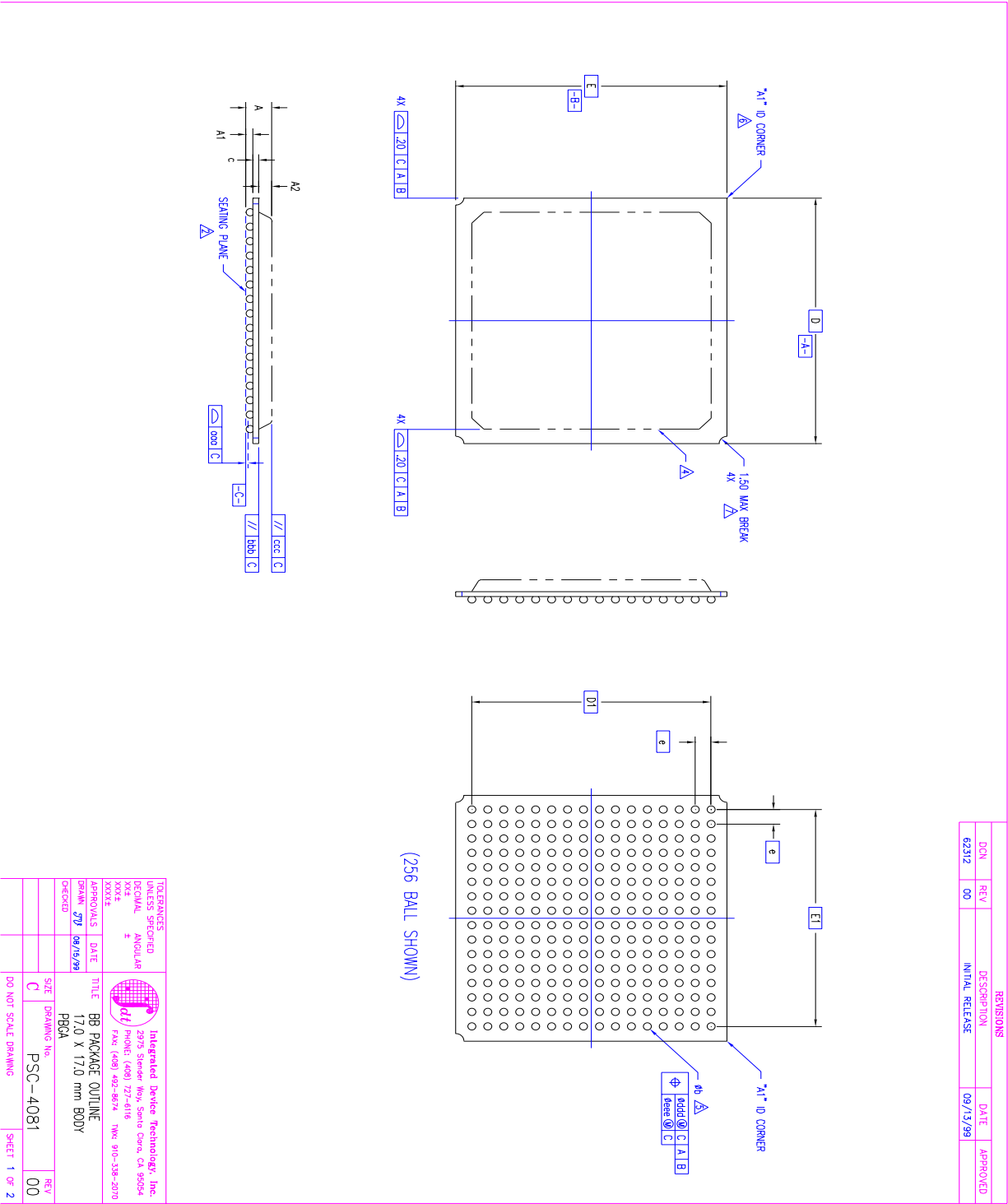
-  The lighter shaded area shows the ground pins (Vss)
-  The darker shaded area shows the supply voltage pins (Vcc I/O)
-  Vcc Core
-  VccP, VssP

Figure 8 RC32334 Chip — Top View

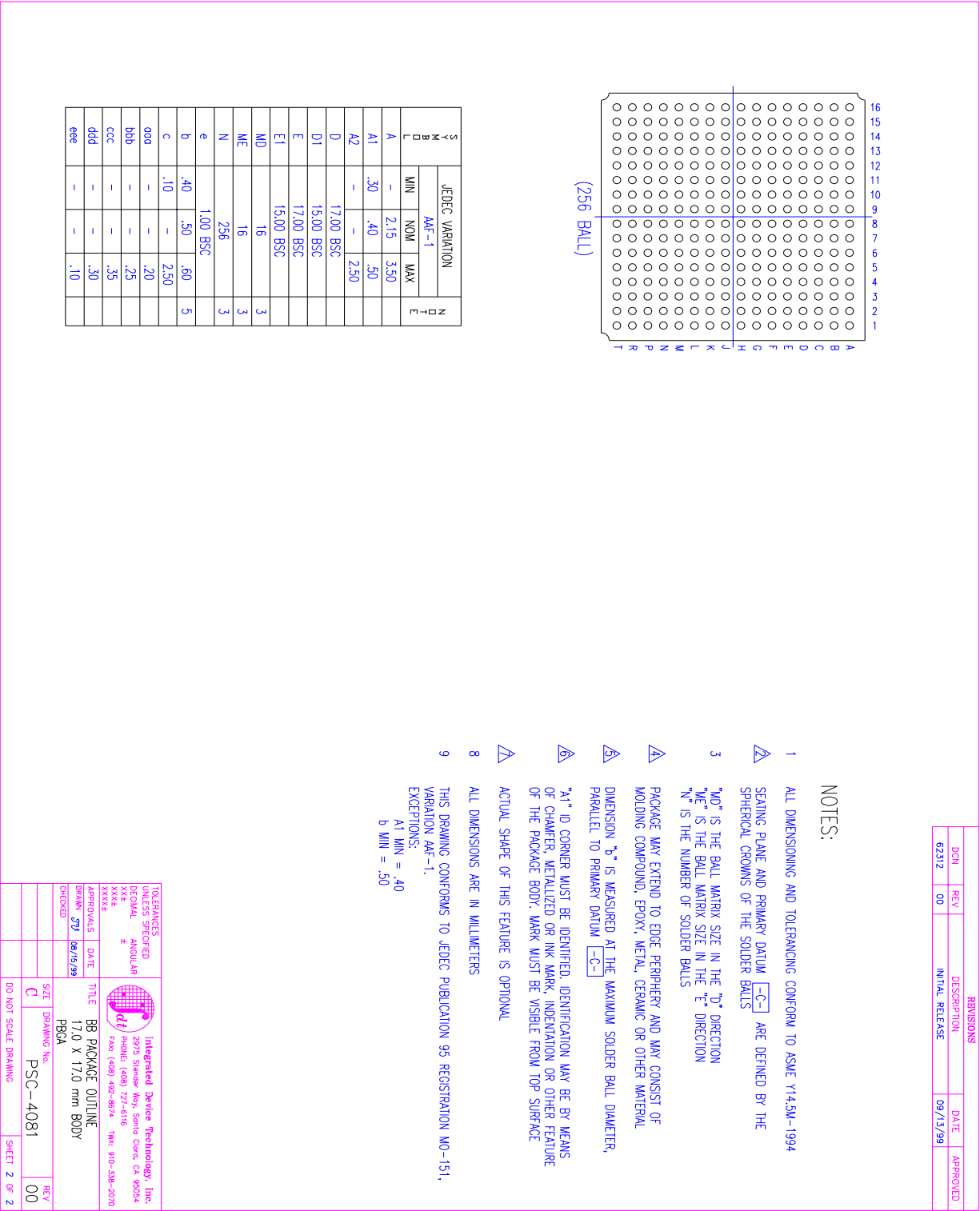
RC32334 Alternate Signal Functions

Pin	Alt #1	Alt #2	Pin	Alt #1	Alt #2	Pin	Alt #1	Alt #2
A1	PIO[15]		C6	sdram_addr[4]		J1	modebit[6]	
A6	sdram_addr[3]		C7	sdram_addr[9]		J2	modebit[4]	
A7	sdram_addr[7]		C11	sdram_addr[14]		K3	modebit[3]	
A8	sdram_addr[11]		C12	modebit[8]		K4	modebit[5]	
A11	sdram_addr[15]		C13	reset_boot_mode[1]		K13	mem_245_dt_r_n	sdram_245_dt_r_n
A12	modebit[9]		D6	sdram_addr[5]		L1	modebit[0]	
A16	PIO[2]	timer_gate_n[0]	D7	sdram_addr[8]		L3	modebit[1]	
B1	PIO[12]		D11	sdram_addr[13]		L4	modebit[2]	
B2	PIO[14]		D12	modebit[7]		P2	pci_inta_n (satellite)	
B6	sdram_addr[2]		D13	reset_boot_mode[0]		P3	PIO[0]	dma_done_n[1]
B7	sdram_addr[6]		F4	sdram_wait_n		R1	pci_idsel (satellite)	
B8	sdram_addr[10]		G1	PIO[1]	dma_done_n[0]	R3	pci_eeeprom_cs (satellite)	PIO[11]
B12	reset_pci_host_mode		G3	PIO[10]	pci_eeeprom_mdo	R15	PIO[3]	
C1	PIO[6]		G4	PIO[7]	pci_eeeprom_mdi	T2	Unused (satellite)	
C2	PIO[5]		H1	PIO[8]		T16	PIO[4]	
C3	PIO[13]	C3	H2	PIO[9]	pci_eeeprom_sk			

RC32334 Package Drawing — 256-pin PBGA



RC32334 Package Drawing — Page Two



Valid Combinations

Commercial

Industrial



for Tech Support:
email: rischelp@idt.com
phone: 408-492-8208