Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

IDT™ Interprise™ Integrated Communications Processor

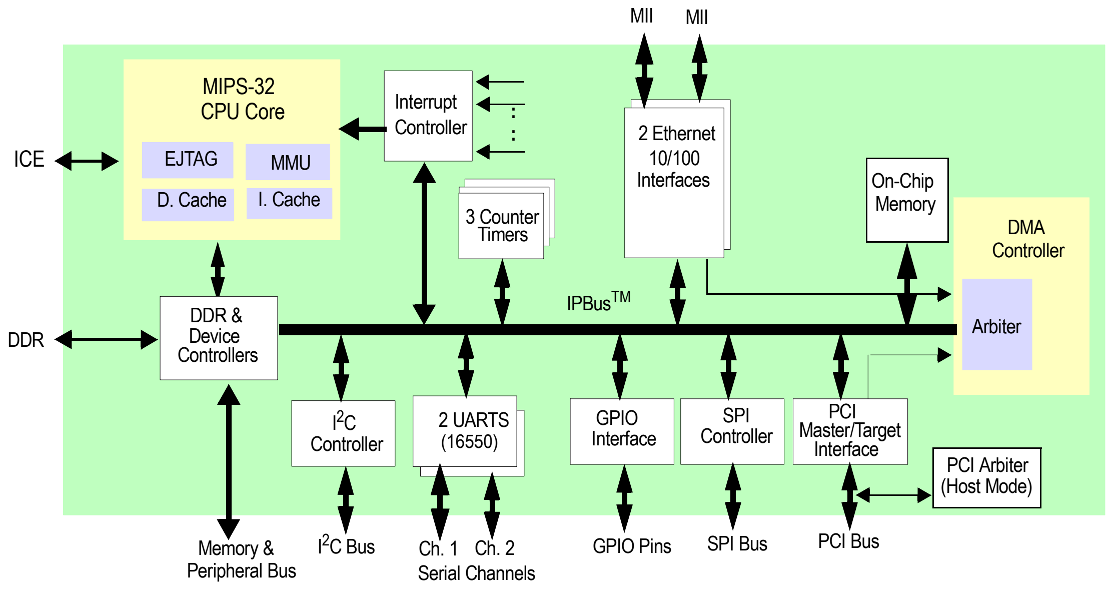
79RC32438

Features

- **32-bit CPU Core**
  - *MIPS32 instruction set*
  - *Cache Sizes: 16KB instruction and data caches, 4-Way set associative, cache line locking, non-blocking prefetches*
  - *16 dual-entry JTLB with variable page sizes*
  - *3-entry instruction TLB*
  - *3-entry data TLB*
  - *Max issue rate of one 32x16 multiply per clock*
  - *Max issue rate of one 32x32 multiply every other clock*
  - *CPU control with start, stop and single stepping*
  - *Software breakpoints support*
  - *Hardware breakpoints on virtual addresses*
  - *Enhanced JTAG and ICE Interface that is compatible with v2.5 of the EJTAG Specification*
- **DDR Memory Controller**
  - *Supports up to 2GB of DDR SDRAM*
  - *2 chip selects (each chip select supports 4 internal DDR banks)*
  - *Supports 16-bit or 32-bit data bus width using 8, 16, or 32-bit devices*
  - *Supports 64Mb, 128Mb, 256Mb, 512Mb, and 1Gb DDR SDRAM devices*
  - *Data bus multiplexing support allows interfacing to standard DDR DIMMs and SODIMMs*
  - *Automatic refresh generation*
- **Memory and Peripheral Device Controller**
  - *Provides "glueless" interface to standard SRAM, Flash, ROM, dual-port memory, and peripheral devices*
  - *Demultiplexed address and data buses: 16-bit data bus, 26-bit address bus, 6 chip selects, supports alternate bus masters, control for external data bus buffers*
  - *Supports 8-bit and 16-bit width devices*

    Automatic byte gathering and scattering
  - *Flexible protocol configuration parameters: programmable number of wait states (0 to 63), programmable postread/post-write delay (0 to 31), supports external wait state generation, supports Intel and Motorola style peripherals*
  - *Write protect capability per chip select*
  - *Programmable bus transaction timer generates warm reset when counter expires*
  - *Supports up to 64 MB of memory per chip select*
- **Counter/Timers**
  - *Three general purpose 32-bit counter timers*
- **PCI Interface**
  - *32-bit PCI revision 2.2 compliant (3.3V only)*
  - *Supports host or satellite operation in both master and target modes*
  - *Support for synchronous and asynchronous operation*
  - *PCI clock supports frequencies from 16 MHz to 66 MHz*
  - *PCI arbiter in Host mode: supports 6 external masters, fixed priority or round robin arbitration*
  - *I_2O "like" PCI Messaging Unit*

Block Diagram



IDT and the IDT logo are trademarks of Integrated Device Technology, Inc.

- **DMA Controller**
  - *10 DMA channels: two channels for PCI (PCI to Memory and Memory to PCI), two for each Ethernet interface, two channels for memory to memory operations, two channels for external operations*
  - *Provides flexible descriptor based operation*
  - *Supports unaligned transfers (i.e., source or destination address may be on any byte boundary) with arbitrary byte length.*
- **Two Ethernet Interfaces**
  - *10 and 100 Mb/s ISO/IEC 8802-3:1996 compliant*
  - *Two IEEE 802.3u compatible Media Independent Interfaces (MII) with serial management interface*
  - *MII supports IEEE 802.3u auto-negotiation speed selection*
  - *Supports 64 entry hash table based multicast address filtering*
  - *512 byte transmit and receive FIFOs*
  - *Supports flow control functions outlined in IEEE Std. 802.3x-1997*
- **Universal Asynchronous Receiver Transmitter (UART)**
  - *Compatible with the 16550 and 16450 UARTs*
  - *Two completely separate serial channels*
  - *Modem control functions (CTS, RTS, DSR, DTR, RI, DCD)*
  - *16-byte transmit and receive buffers*
  - *Programmable baud rate generator derived from the system clock*
  - *Fully programmable serial characteristics:*
    - *5, 6, 7, or 8 bit characters*
    - *Even, odd or no parity bit generation and detection*
    - *1, 1-1/2 or 2 stop bit generation*
  - *Line break generation and detection*
  - *False start bit detection*
  - *Internal loopback mode*
- **I^{2}C-Bus**
  - *Supports standard 100 Kbps mode as well as 400 Kbps fast mode*
  - *Supports 7-bit and 10-bit addressing*
  - *Supports four modes: master transmitter, master receiver, slave transmitter, slave receiver*
- **Additional General Purpose Peripherals**
  - *Two 16550-compatible serial ports*
  - *Interrupt controller*
  - *System integrity functions*
  - *General purpose I/O controller*
  - *Serial peripheral interface (SPI)*
- **On-chip Memory**
  - *4KB of high speed SRAM organized as 1K x 32 bits*
  - *Supports burst and non-burst byte, halfword, triple-byte, and word CPU, PCI, and DMA accesses*
- **Debug Support**
  - *Rev. 2.6 compliant EJTAG Interface*

Device Overview

The RC32438 is a member of the IDT™ Interprise™ family of PCI integrated communications processors. It incorporates a high performance CPU core and a number of on-chip peripherals. The integrated processor is designed to transfer information from I/O modules to main memory with minimal CPU intervention using a highly sophisticated direct memory access (DMA) engine. All data transfers through the RC32438 are achieved by writing data from an on-chip I/O peripheral to main memory and then out to another I/O module.

CPU Execution Core

The 32-bit CPU core is 100% compatible with the MIPS32 instruction set architecture (ISA).

Specifically, this device features the 4Kc CPU core developed by MIPS Technologies Inc. (www.mips.com). This core issues a single instruction per cycle, includes a five stage pipeline, and is optimized for applications that require integer arithmetic. The CPU core includes 16 KB instruction and 16 KB data caches. Both caches are 4-way set associative and can be locked on a per line basis, which allows the programmer control over this precious on-chip memory resource. The core also features a memory management unit (MMU). The CPU core also incorporates an enhanced joint test access group (EJTAG) interface that is used to interface to in-circuit emulator tools, providing access to internal registers and enabling the part to be controlled externally, simplifying the system debug process. The use of this core allows IDT's customers to leverage the broad range of software and development tools available for the MIPS architecture, including operating systems, compilers, and in-circuit emulators.

Double Data Rate Memory Controller

The RC32438 incorporates a high performance double data rate (DDR) memory controller which supports both x16 and x32 memory configurations up to 2GB. This module provides all of the signals required to interface to both memory modules and discrete devices, including two chip selects, differential clocking outputs and data strobes.

Memory and I/O Controller

The RC32438 uses a dedicated local memory/IO controller including a de-multiplexed 16-bit data and 26-bit address bus. It includes all of the signals required to interface directly to as many as six Intel or Motorola-style external peripherals, and the interface can be configured to support both 8-bit and 16-bit peripherals.

DMA Controller

The DMA controller consists of 10 independent DMA channels, all of which operate in exactly the same manner. The DMA controller off-loads the CPU core from moving data among the on-chip interfaces, external peripherals, and memory. The controller supports scatter/gather DMA with no alignment restrictions, appropriate for communications and graphics systems.

PCI Interface

The PCI interface on the RC32438 is compatible with version 2.2 of the PCI specification. An on-chip arbiter supports up to six external bus masters, supporting both fixed priority and rotating priority arbitration schemes. The part can support both satellite and host PCI configurations, enabling the RC32438 to act as a slave controller for a PCI add-in card application, or as the primary PCI controller in the system. The PCI interface can be operated synchronously or asynchronously to the other I/O interfaces on the RC32438 device.

Ethernet Interface

The RC32438 has two Ethernet Channels supporting 10Mbps and 100Mbps speeds to provide a standard media independent interface (MII) off-chip, allowing a wide range of external devices to be connected efficiently.

UART Interface

The RC32438 contains two completely separate serial channels (UARTs) that are compatible with the industry standard 16550 UART.

System Integrity Functions

The RC32438 contains a programmable watchdog timer that generates NMI when the counter expires and an address space monitor that reports errors in response to accesses to undecoded address regions.

General Purpose I/O Controller

The RC32438 contains 32 general purpose input/output pins. Each pin may be used as an active high or active low level interrupt or nonmaskable interrupt input, and each signal may be used as a bit input or output port.

I^{2}C Interface

The standard I2C interface allows the RC32438 to connect to a number of standard external peripherals for a more complete system solution. The RC32438 supports both master and slave operations.

Debug Support

The RC32438 supports the industry standard Rev. 2.6 EJTAG interface.

Thermal Considerations

The RC32438 consumes less than 2.7 W peak power. It is guaranteed in a ambient temperature range of 0° to +70° C for commercial temperature devices and - 40° to +85° for industrial temperature devices.

Revision History

November 7, 2002: Initial publication. Preliminary Information.

November 15, 2002: Added footnotes to Tables 5, 9, and 10.

December 12, 2002: Added Clock Speed parameter to PLL and Core supply in Table 16.

December 19, 2002: Release version.

January 13, 2003: Changed Thermal Considerations to read less than 2.7W instead of 2.5W, added values to CLK parameter in Table 5, and revised EJTAG description.

February 4, 2003: Revised description for EJTAG/JTAG pins in Table 1. Changed DDRDM[7:0] from input/output to output only in Tables 1 and 2 and Logic Diagram. Added new section, Voltage Sense Signal Timing, as part of EJTAG description.

March 4, 2003: In Table 2, removed “pull-up” from PCI pin category and from GPIO [24] and GPIO[30-26]. In Table 20, changed max. values for VccSI/O, VccCore, and VccPLL.

July 9, 2003: In Table 7: changed values for DDRDATA, DDRDM, and DDRADDR—WEN signals, and deleted old footnote #3 and changed values in new footnote #3. In Table 8, changed Tdo values. Changed Figure 7. Changed values in Table 18, Power Consumption. Removed IPBus Monitor feature which included changes to Tables 1, 2, 21, 24, and 25. Deleted Table 13 which resulted in a re-ordering of subsequent tables.

March 8, 2004: Added 300MHz speed grade.

May 25, 2004: In Table 9, signals MIIxRXCLK and MIIxTXCLK, the Min and Max values for Thigh/Tlow_9c were changed to 140 and 260 respectively and the Min and Max values for Thigh/Tlow_9d were changed to 14.0 and 26.0 respectively.

Pin Description Table

The following table lists the functions of the pins provided on the RC32438. Some of the functions listed may be multiplexed onto the same pin.

The active polarity of a signal is defined using a suffix. Signals ending with an "N" are defined as being active, or asserted, when at a logic zero (low) level. All other signals (including clocks, buses and select lines) will be interpreted as being active, or asserted when at a logic one (high) level.

Signal	Type	Name/Description
System		
CLK	I	**Master Clock.** This is the master clock input. The processor frequency is a multiple of this clock frequency. This clock is used as the system clock for all memory and peripheral bus operations.
EXTCLK	O	**External Clock.** This clock is used for all memory and peripheral bus operations.
COLDRSTN	I	**Cold Reset.** The assertion of this signal initiates a cold reset. This causes the processor state to be initialized, boot configuration to be loaded, and the internal PLL to lock onto the master clock (CLK).
RSTN	I/O	**Reset.** The assertion of this bidirectional signal initiates a warm reset. This signal is asserted by the RC32438 during a warm reset.
Memory and Peripheral Bus		
BDIRN	O	**External Buffer Direction.** Memory and peripheral bus external data bus buffer direction control. If the RC32438 memory and peripheral bus is connected to the A side of a transceiver, such as an IDT74FCT245, then this pin may be directly connected to the direction control (e.g., BDIR) pin of the transceiver.
BGN	O	**Bus Grant.** This signal is asserted by the RC32438 to indicate that the RC32438 has relinquished ownership of the memory and peripheral bus.
BOEN	O	**External Buffer Enable.** This signal provides an output enable control for an external buffer on the memory and peripheral data bus.
BRN	I	**Bus Request.** This signal is asserted by an external device to request ownership of the memory and peripheral bus.
BWEN[1:0]	O	**Byte Write Enables.** These signals are memory and peripheral bus byte write enable signals. BWEN[0] corresponds to byte lane MDATA[7:0] BWEN[1] corresponds to byte lane MDATA[15:8]
CSN[5:0]	O	**Chip Selects.** These signals are used to select an external device on the memory and peripheral bus.
MADDR[21:0]	O	**Address Bus.** 22-bit memory and peripheral bus address bus. MADDR[25:22] are available as GPIO alternate functions
MDATA[15:0]	I/O	**Data Bus.** 16-bit memory and peripheral data bus. During a cold reset, these pins function as inputs that are used to load the boot configuration vector.
OEN	O	**Output Enable.** This signal is asserted when data should be driven on by an external device on the memory and peripheral bus.
RWN	O	**Read Write.** This signal indicates if the transaction on the memory and peripheral bus is a read transaction or a write transaction. A high level indicates a read from an external device. A low level indicates a write to an external device.

Table 1 Pin Description (Part 1 of 9)

Signal	Type	Name/Description
WAITACKN	I	**Wait or Transfer Acknowledge.** When configured as wait, this signal is asserted during a memory and peripheral bus transaction to extend the bus cycle. When configured as a transfer acknowledge, this signal is asserted during a transaction to signal the completion of the transaction.
DDR Bus		
DDRADDR[13:0]	O	**DDR Address Bus.** 14-bit multiplexed DDR bus address bus. This bus is used to transfer the addresses to the DDR devices.
DDRBA[1:0]	O	**DDR Bank Address.** These signals are used to transfer the bank address to the DDRs.
DDRCASN	O	**DDR Column Address Strobe.** This signal is asserted during DDR transactions.
DDRCKE	O	**DDR Clock Enable.** The DDR clock enable is asserted during normal DDR operation. This signal is negated during following a cold reset or during a power down operation.
DDRCKN[1:0]	O	**DDR Negative DDR clock.** These signals are the negative clock of the differential DDR clock pair. Two copies of this output are provided to reduce signal loading.
DDRCKP[1:0]	O	**DDR Positive DDR clock.** These signals are the positive clock of the differential DDR clock pair. Two copies of this output are provided to reduce signal loading.
DDRCSN[1:0]	O	**DDR Chip Selects.** These active low signals are used to select DDR device(s) on the DDR bus.
DDRDATA[31:0]	I/O	**DDR Data Bus.** 32-bit DDR data bus used to transfer data between the RC32438 and the DDR devices. Data is transferred on both edges of the clock.
DDRDM[7:0]	O	**DDR Data Write Enables.** Byte data write enables used to enable specific byte lanes during DDR writes. DDRDM[0] corresponds to DDRDATA[7:0] DDRDM[1] corresponds to DDRDATA[15:8] DDRDM[2] corresponds to DDRDATA[23:16] DDRDM[3] corresponds to DDRDATA[31:24] DDRDM[4] corresponds to DDRDATA[39:32] DDRDM[5] corresponds to DDRDATA[47:40] DDRDM[6] corresponds to DDRDATA[55:48] DDRDM[7] corresponds to DDRDATA[54:56] (Refer to the DDR Data Bus Multiplexing section in Chapter 7 of the RC32438 User Reference Manual.)
DDRDQS[3:0]	I/O	**DDR Data Strobes.** DDR byte data strobes used to clock data between DDR devices and the RC32438. These strobes are inputs during DDR reads and outputs during DDR writes. DDRDQS[0] corresponds to DDRDATA[7:0]. DDRDQS[1] corresponds to DDRDATA[15:8]. DDRDQS[2] corresponds to DDRDATA[23:16]. DDRDQS[3] corresponds to DDRDATA[31:24].
DDROEN[3:0]	O	**DDR Bus Switch Output Enables.** These pins are used to enable external data bus switches in systems that support data bus multiplexing.
DDRRASN	O	**DDR Row Address Strobe.** The DDR row address strobe is asserted during DDR transactions.

Table 1 Pin Description (Part 2 of 9)

Signal	Type	Name/Description
DDRVREF	I	**DDR Voltage Reference.** SSTL_2 DDR voltage reference generated by an external source.
DDRWEN	O	**DDR Write Enable.** DDR write enable is asserted during DDR write transactions.
PCI Bus		
PCIAD[31:0]	I/O	**PCI Multiplexed Address/Data Bus**. Address is driven by a bus master during initial PCIFRAMEN assertion. Data is then driven by the bus master during writes or by the bus target during reads.
PCICBEN[3:0]	I/O	**PCI Multiplexed Command/Byte Enable Bus**. PCI command is driven by the bus master during the initial PCIFRAMEN assertion. Byte enable signals are driven by the bus master during subsequent data phase(s).
PCICLK	I	**PCI Clock**. Clock used for all PCI bus transactions.
PCIDEVSELN	I/O	**PCI Device Select**. This signal is driven by a bus target to indicate that the target has decoded the address as one of its own address spaces.
PCIFRAMEN	I/O	**PCI Frame**. Driven by a bus master. Assertion indicates the beginning of a bus transaction. Negation indicates the last data.
PCIGNTN[3:0]	I/O	**PCI Bus Grant**. **In PCI host mode with internal arbiter:** The assertion of these signals indicates to the agent that the internal RC32438 arbiter has granted the agent access to the PCI bus. **In PCI host mode with external arbiter:** PCIGNTN[0]: asserted by an external arbiter to indicate to the RC32438 that access to the PCI bus has been granted. PCIGNTN[3:1]: unused and driven high. **In PCI satellite mode:** PCIGNTN[0]: This signal is asserted by an external arbiter to indicate to the RC32438 that access to the PCI bus has been granted. PCIGNTN[1]: this signal takes on the alternate function of PCIEECS and is used as a PCI Serial EEPROM chip select PCIGNTN[3:2]: unused and driven high. **Note**: When the GPIO register is programmed in the alternate function mode for bits GPIO [26] and [28], these bits become PCIGNTN [4] and [5] respectively.
PCIIRDYN	I/O	**PCI Initiator Ready**. Driven by the bus master to indicate that the current datum can complete.
PCILOCKN	I/O	**PCI Lock**. This signal is asserted by an external bus master to indicate that an exclusive operation is occurring.
PCIPAR	I/O	**PCI Parity**. Even parity of the PCIAD[31:0] bus. Driven by the bus master during address and write Data phases. Driven by the bus target during the read data phase.
PCIPERRN	I/O	**PCI Parity Error**. If a parity error is detected, this signal is asserted by the receiving bus agent 2 clocks after the data is received.

Table 1 Pin Description (Part 3 of 9)

Signal	Type	Name/Description
PCIREQN[3:0]	I/O	**PCI Bus Request.** **In PCI host mode with internal arbiter:** These signals are inputs whose assertion indicates to the internal RC32438 arbiter that an agent desires ownership of the PCI bus. **In PCI host mode with external arbiter:** PCIREQN[0]: asserted by the RC32438 to request ownership of the PCI bus. PCIREQN[3:1]: unused and driven high. **In PCI satellite mode:** PCIREQN[0]: this signal is asserted by the RC32438 to request use of the PCI bus. PCIREQN[1]: function changes to PCIIDSEL and is used as a chip select during configuration read and write transactions. PCIREQN[3:2]: unused and driven high. **Note**: When the GPIO register is programmed in the alternate function mode for bits GPIO [24] and [27], these bits become PCIREQN [4] and [5] respectively.
PCIRSTN	I/O	**PCI Reset**. In host mode, this signal is asserted by the RC32438 to generate a PCI reset. In satellite mode, assertion of this signal initiates a warm reset.
PCISERRN	I/O	**PCI System Error**. This signal is driven by an agent to indicate an address parity error, data parity error during a special cycle command, or any other system error. Requires an external pull-up.
PCISTOPN	I/O	**PCI Stop**. Driven by the bus target to terminate the current bus transaction. For example, to indicate a retry.
PCITRDYN	I/O	**PCI Target Ready**. Driven by the bus target to indicate that the current data can complete.
General Purpose Input/Output		
GPIO[0]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0SOUT Alternate function: UART channel 0 serial output.
GPIO[1]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0SINP Alternate function: UART channel 0 serial input.
GPIO[2]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0RIN Alternate function: UART channel 0 ring indicator input.
GPIO[3]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0DCDN Alternate function: UART channel 0 data carrier detect input.
GPIO[4]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0DTRN Alternate function: UART channel 0 data terminal ready input.
GPIO[5]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0DSRN Alternate function: UART channel 0 data set ready input.

Table 1 Pin Description (Part 4 of 9)

Signal	Type	Name/Description
GPIO[6]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0RTSN Alternate function: UART channel 0 request to send output.
GPIO[7]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U0CTSN Alternate function: UART channel 0 clear to send input.
GPIO[8]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1SOUT Alternate function: UART channel 1 serial output.
GPIO[9]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1SINP Alternate function: UART channel 1 serial input.
GPIO[10]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1DTRN Alternate function: UART channel 1 data terminal ready output.
GPIO[11]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1DSRN Alternate function: UART channel 1 data set ready input.
GPIO[12]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1RTSN Alternate function: UART channel 1 request to send output.
GPIO[13]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: U1CTSN Alternate function: UART channel 1 clear to send input.
GPIO[14]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMAREQN0 Alternate function: External DMA channel 0 request input.
GPIO[15]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMAREQN1 Alternate function: External DMA channel 1 request input.
GPIO[16]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMADONEN0 Alternate function: External DMA channel 0 done input.
GPIO[17]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMADONEN1 Alternate function: External DMA channel 1 done input.

Table 1 Pin Description (Part 5 of 9)

Signal	Type	Name/Description
GPIO[18]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMAFINN0 Alternate function: External DMA channel 0 finished output.
GPIO[19]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: DMAFINN1 Alternate function: External DMA channel 1 finished output.
GPIO[20]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin Alternate function pin name: MADDR[22] Alternate function: Memory and peripheral bus address output.
GPIO[21]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: MADDR[23] Alternate function: Memory and peripheral bus address output.
GPIO[22]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: MADDR[24] Alternate function: Memory and peripheral bus address output.
GPIO[23]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: MADDR[25] Alternate function: Memory and peripheral bus address output.
GPIO[24]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: PCIREQN[4] Alternate function: PCI Request 4 input or output.
GPIO[25]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: AFSPARE1 Alternate function: *reserved.*
GPIO[26]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: PCIGNTN[4] Alternate function: PCI Grant 4 output.
GPIO[27]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: PCIREQN[5] Alternate function: PCI Request 5 input or output.
GPIO[28]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: PCIGNTN[5] Alternate function: PCI Grant 5 output.
GPIO[29]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: Reserved Alternate function: Reserved.

Table 1 Pin Description (Part 6 of 9)

Signal	Type	Name/Description
GPIO[30]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin. Alternate function pin name: PCIMUINTN Alternate function: PCI Messaging unit interrupt output.
GPIO[31]	I/O	**General Purpose I/O.** This pin can be configured as a general purpose I/O pin.
SPI Interface		
SCK	I/O	**Serial Clock**. This signal is used as the serial clock output in SPI mode and in PCI satellite mode with suspended CPU execution during PCI serial EEPROM loading. This pin may be configured as a GPIO pin.
SDI	I/O	**Serial Data Input**. This signal is used to shift in serial data in SPI mode and in PCI satellite mode with suspended CPU execution during PCI serial EEPROM loading. This pin may be configured as a GPIO pin.
SDO	I/O	**Serial Data Output**. This signal is used shift out serial data in SPI mode and in PCI satellite mode with suspended CPU execution during PCI serial EEPROM loading. This pin may be configured as a GPIO pin.
I^{2}C Bus Interface		
SCL	I/O	**I^{2}C Clock.** I^{2}C-bus clock.
SDA	I/O	**I^{2}C Data Bus.** I^{2}C-bus data bus.
Ethernet Interfaces		
MII0CL	I	**Ethernet 0 MII Collision Detected.** This signal is asserted by the ethernet PHY when a collision is detected.
MII0CRS	I	**Ethernet 0 MII Carrier Sense.** This signal is asserted by the ethernet PHY when either the transmit or receive medium is not idle.
MII0RXCLK	I	**Ethernet 0 MII Receive Clock.** This clock is a continuous clock that provides a timing reference for the reception of data.
MII0RXD[3:0]	I	**Ethernet 0 MII Receive Data.** This nibble wide data bus contains the data received by the ethernet PHY.
MII0RXDV	I	**Ethernet 0 MII Receive Data Valid.** The assertion of this signal indicates that valid receive data is in the MII receive data bus.
MII0RXER	I	**Ethernet 0 MII Receive Error.** The assertion of this signal indicates that an error was detected somewhere in the ethernet frame currently being sent in the MII receive data bus.
MII0TXCLK	I	**Ethernet 0 MII Transmit Clock.** This clock is a continuous clock that provides a timing reference for the transfer of transmit data.
MII0TXD[3:0]	O	**Ethernet 0 MII Transmit Data.** This nibble wide data bus contains the data to be transmitted.
MII0TXENP	O	**Ethernet 0 MII Transmit Enable.** The assertion of this signal indicates that data is present on the MII for transmission.
MII0TXER	O	**Ethernet 0 MII Transmit Coding Error.** When this signal is asserted together with MIITXENP, the ethernet PHY will transmit symbols which are not valid data or delimiters.
MII1CL	I	**Ethernet 1 MII Collision Detected.** This signal is asserted by the ethernet PHY when a collision is detected.

Table 1 Pin Description (Part 7 of 9)

Signal	Type	Name/Description
MII1CRS	I	**Ethernet 1 MII Carrier Sense.** This signal is asserted by the ethernet PHY when either the transmit or receive medium is not idle.
MII1RXCLK	I	**Ethernet 1 MII Receive Clock.** This clock is a continuous clock that provides a timing reference for the reception of data.
MII1RXD[3:0]	I	**Ethernet 1 MII Receive Data.** This nibble wide data bus contains the data received by the ethernet PHY.
MII1RXDV	I	**Ethernet 1 MII Receive Data Valid.** The assertion of this signal indicates that valid receive data is in the MII receive data bus.
MII1RXER	I	**Ethernet 1 MII Receive Error.** The assertion of this signal indicates that an error was detected somewhere in the ethernet frame currently being sent in the MII receive data bus.
MII1TXCLK	I	**Ethernet 1 MII Transmit Clock.** This clock is a continuous clock that provides a timing reference for the transfer of transmit data.
MII1TXD[3:0]	O	**Ethernet 1 MII Transmit Data.** This nibble wide data bus contains the data to be transmitted.
MII1TXENP	O	**Ethernet 1 MII Transmit Enable.** The assertion of this signal indicates that data is present on the MII for transmission.
MII1TXER	O	**Ethernet 1 MII Transmit Coding Error.** When this signal is asserted together with MIITXENP, the ethernet PHY will transmit symbols which are not valid data or delimiters.
MIIMDC	O	**MII Management Data Clock.** This signal is used as a timing reference for transmission of data on the management interface.
MIIMDIO	I/O	**MII Management Data.** This bidirectional signal is used to transfer data between the station management entity and the ethernet PHY.
JTAG / EJTAG		
EJTAG_TMS	I	**EJTAG Mode**. The value on this signal controls the test mode select of the EJTAG Controller. When using the JTAG boundary scan, this pin should be left disconnected (since there is an internal pull-up) or driven high.
JTAG_TCK	I	**JTAG Clock**. This is an input test clock used to clock the shifting of data into or out of the boundary scan logic, JTAG Controller, or the EJTAG Controller. JTAG_TCK is independent of the system and the processor clock with a nominal 50% duty cycle.
JTAG_TDI	I	**JTAG Data Input**. This is the serial data input to the boundary scan logic, JTAG Controller, or the EJTAG Controller.
JTAG_TDO	O	**JTAG Data Output**. This is the serial data shifted out from the boundary scan logic, JTAG Controller, or the EJTAG Controller. When no data is being shifted out, this signal is tri-stated.
JTAG_TMS	I	**JTAG Mode**. The value on this signal controls the test mode select of the boundary scan logic or JTAG Controller. When using the EJTAG debug interface, this pin should be left disconnected (since there is an internal pull-up) or driven high.

Table 1 Pin Description (Part 8 of 9)

Signal	Type	Name/Description
JTAG_TRST_N	I	**JTAG Reset**. This active low signal asynchronously resets the boundary scan logic, JTAG TAP Controller, and the EJTAG Debug TAP Controller. An external pull-up on the board is recommended to meet the JTAG specification in cases where the tester can access this signal. However, for systems running in functional mode, one of the following should occur: 1) actively drive this signal low with control logic 2) statically drive this signal low with an external pull-down on the board 3) clock JTAG_TCK while holding EJTAG_TMS and/or JTAG_TMS high.
Debug		
CPU	O	**CPU Transaction.** This signal is asserted during all CPU instruction fetches and data transfers to/from the DDR and devices on the memory and peripheral bus. The signal is negated during PCI and DMA transactions to/from the DDR and devices on the memory and peripheral bus.
INST	O	**Instruction or Data.** This signal is driven high during CPU instruction fetches on the memory and peripheral bus memory or DDR bus.

Table 1 Pin Description (Part 9 of 9)

Pin Characteristics

Note: Some input pads of the RC32438 do not contain internal pull-ups or pull-downs. Unused inputs should be tied off to appropriate levels. This is especially critical for unused control signal inputs (such as BRN) which, if left floating, could adversely affect the RC32438's operation. Also, any input pin left floating can cause a slight increase in power consumption.

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes[1]
Memory and Peripheral Bus	BDIRN	O	LVTTL	High Drive		
	BGN	O	LVTTL	Low Drive		
	BOEN	O	LVTTL	High Drive		
	BRN	I	LVTTL	STI[2]	pull-up	
	BWEN[1:0]	O	LVTTL	High Drive		
	CSN[5:0]	O	LVTTL	High Drive		
	MADDR[21:0]	O	LVTTL	High Drive		
	MDATA[15:0]	I/O	LVTTL	High Drive		
	OEN	O	LVTTL	High Drive		
	RWN	O	LVTTL	High Drive		
	WAITACKN	I	LVTTL	STI	pull-up	

Table 2 Pin Characteristics (Part 1 of 4)

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes[1]
DDR Bus	DDRADDR[13:0]	O	SSTL_2	SSTL_2		
	DDRBA[1:0]	O	SSTL_2	SSTL_2		
	DDRCASN	O	SSTL_2	SSTL_2		
	DDRCKE	O	SSTL_2 / LVCMOS	SSTL_2		
	DDRCKN[1:0]	O	SSTL_2	SSTL_2		
	DDRCKP[1:0]	O	SSTL_2	SSTL_2		
	DDRCSN[1:0]	O	SSTL_2	SSTL_2		
	DDRDATA[31:0]	I/O	SSTL_2	SSTL_2		
	DDRDM[7:0]	O	SSTL_2	SSTL_2		
	DDRDQS[3:0]	I/O	SSTL_2	SSTL_2		
	DDROEN[3:0]	O	SSTL_2	SSTL_2		
	DDRRASN	O	SSTL_2	SSTL_2		
	DDRVREF	I	Analog	SSTL_2		
	DDRWEN	O	SSTL_2	SSTL_2		
PCI Bus Interface[3]	PCIAD[31:0]	I/O	PCI	PCI		
	PCICBEN[3:0]	I/O	PCI	PCI		
	PCICLK	I	PCI	PCI		
	PCIDEVSELN	I/O	PCI	PCI		pull-up on board
	PCIFRAMEN	I/O	PCI	PCI		pull-up on board
	PCIGNTN[3:0]	I/O	PCI	PCI		pull-up on board
	PCIIRDYN	I/O	PCI	PCI		pull-up on board
	PCILOCKN	I/O	PCI	PCI		
	PCIPAR	I/O	PCI	PCI		
	PCIPERRN	I/O	PCI	PCI		
	PCIREQN[3:0]	I/O	PCI	PCI		pull-up on board
	PCIRSTN	I/O	PCI	PCI		pull-down on board
	PCISERRN	I/O	PCI	Open Collector; PCI		pull-up on board
	PCISTOPN	I/O	PCI	PCI		pull-up on board
	PCITRDYN	I/O	PCI	PCI		pull-up on board
General Purpose I/O	GPIO[23:0]	I/O	LVTTL	Low Drive	pull-up	
	GPIO[24]	I/O	PCI			pull-up on board
	GPIO[25]	I/O	LVTTL	Low Drive	pull-up	
	GPIO[30:26][4]	I/O	PCI			pull-up on board
	GPIO[31]	I/O	LVTTL	Low Drive	pull-up	

Table 2 Pin Characteristics (Part 2 of 4)

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes[1]
Serial Interface	SCK	I/O	LVTTL	Low Drive	pull-up	pull-up on board
	SDI	I/O	LVTTL	Low Drive	pull-up	pull-up on board
	SDO	I/O	LVTTL	Low Drive	pull-up	pull-up on board
I^{2}C-Bus Interface	SCL	I/O	LVTTL	Low Drive/STI		pull-up on board[5]
	SDA	I/O	LVTTL	Low Drive/STI		pull-up on board[5]
Ethernet Interfaces	MII0CL	I	LVTTL	STI	pull-down	
	MII0CRS	I	LVTTL	STI	pull-down	
	MII0RXCLK	I	LVTTL	STI	pull-up	
	MII0RXD[3:0]	I	LVTTL	STI	pull-up	
	MII0RXDV	I	LVTTL	STI	pull-down	
	MII0RXER	I	LVTTL	STI	pull-down	
	MII0TXCLK	I	LVTTL	STI	pull-up	
	MII0TXD[3:0]	O	LVTTL	Low Drive		
	MII0TXENP	O	LVTTL	Low Drive		
	MII0TXER	O	LVTTL	Low Drive		
	MII1CL	I	LVTTL	STI	pull-down	
	MII1CRS	I	LVTTL	STI	pull-down	
	MII1RXCLK	I	LVTTL	STI	pull-up	
	MII1RXD[3:0]	I	LVTTL	STI	pull-up	
	MII1RXDV	I	LVTTL	STI	pull-down	
	MII1RXER	I	LVTTL	STI	pull-down	
	MII1TXCLK	I	LVTTL	STI	pull-up	
	MII1TXD[3:0]	O	LVTTL	Low Drive		
	MII1TXENP	O	LVTTL	Low Drive		
	MII1TXER	O	LVTTL	Low Drive		
	MIIMDC	O	LVTTL	Low Drive		
	MIIMDIO	I/O	LVTTL	Low Drive	pull-up	
EJTAG / ICE	JTAG_TRST_N	I	LVTTL	STI	pull-up	
	JTAG_TCK	I	LVTTL	STI	pull-up	
	JTAG_TDI	I	LVTTL	STI	pull-up	
	JTAG_TDO	O	LVTTL	Low Drive		
	JTAG_TMS	I	LVTTL	STI	pull-up	
	EJTAG_TMS	I	LVTTL	STI	pull-up	
Debug	CPU	O	LVTTL	Low Drive		
	INST	O	LVTTL	Low Drive		

Table 2 Pin Characteristics (Part 3 of 4)

Function	Pin Name	Type	Buffer	I/O Type	Internal Resistor	Notes[1]
Miscellaneous	CLK	I	LVTTL	STI		
	EXTCLK	O	LVTTL	High Drive		
	COLDRSTN	I	LVTTL	STI		
	RSTN	I/O	LVTTL	Low Drive / STI	pull-up	pull-up on board

Table 2 Pin Characteristics (Part 4 of 4)

[1] External pull-up required in most system applications. Some applications may require additional pull-ups not identified in this table.

[2] Schmidt Trigger Input (STI).

[3] The PCI pins have internal pull-ups but they are too weak to guarantee system validity. Therefore, board pull-ups are mandatory where indicated. GPIO alternate function pins for PCI must also have board pull-ups.

[4] PCIMUINTN is an alternate function of GPIO[30]. When configured as an alternate function, this pin is tri-stated when not asserted (i.e., it acts as an open collector output).

[5] Use a 2.2K pull-up resistor for I2C pins.

Boot Configuration Vector

The boot configuration vector is read by the RC32438 during a cold reset. The vector defines essential RC32438 parameters that are required once the cold reset completes.

The encoding of the boot configuration vector is described in Table 3, and the vector input is illustrated in Figure 4. The value of the boot configuration vector read in by the RC32438 during a cold reset may be determined by reading the Boot Configuration Vector (BCV) Register.

Signal	Name/Description
MDATA[3:0]	**CPU Pipeline Clock Multiplier**. This field specifies the value by which the PLL multiplies the master clock input (CLK) to obtain the processor clock frequency (PCLK). For master clock input frequency constraints, refer to Table 3.1 in the RC32438 User Manual. 0x0 - PLL Bypass 0x1 - Multiply by 3 0x2 - Multiply by 4 0x3 - Multiply by 6 0x4 - Multiply by 8 0x5 - reserved 0x6 - reserved 0x7 - reserved 0x8 - reserved 0xD - reserved 0xE - reserved 0xF - reserved
MDATA[5:4]	**External Clock Divider**. This field specifies the value by which the IPBus clock (ICLK), which is always 1/2 PCLK, is divided in order to generate the external clock output on the EXTCLK pin. 0x0 - Divide by 1 0x1 - Divide by 2 0x2 - Divide by 4 0x3 - reserved
MDATA[6]	**Endian.** This bit specifies the endianness. 0x0 - little endian 0x1 - big endian

Table 3 Boot Configuration Encoding (Part 1 of 2)

Signal	Name/Description
MDATA[7]	**Boot Device Width**. This field specifies the width of the boot device (i.e., Device 0). 0x0 - 8-bit boot device width 0x1 - 16-bit boot device width
MDATA[8]	**Reset Mode**. This bit specifies the length of time the RSTN signal is driven. 0x0 - Normal reset: RSTN driven for minimum of 4096 clock cycles 0x1 - reserved
MDATA[11:9]	**PCI Mode**. This bit controls the operating mode of the PCI bus interface. The initial value of the EN bit in the PCIC register is determined by the PCI mode. 0x0 - Disabled (EN initial value is zero) 0x1 - PCI satellite mode with PCI target not ready (EN initial value is one) 0x2 - PCI satellite mode with suspended CPU execution (EN initial value is one) 0x3 - PCI host mode with external arbiter (EN initial value is zero) 0x4 - PCI host mode with internal arbiter using fixed priority arbitration algorithm (EN initial value is zero) 0x5 - PCI host mode with internal arbiter using round robin arbitration algorithm (EN initial value is zero) 0x6 - *reserved* 0x7 - *reserved*
MDATA[12]	**Disable Watchdog Timer**. When this bit is set, the watchdog timer is disabled following a cold reset. 0x0 - Watchdog timer enabled 0x1 - Watchdog timer disabled
MDATA[15:13]	**Reserved**. These pins must be driven low during boot configuration.

Table 3 Boot Configuration Encoding (Part 2 of 2)

Logic Diagram — RC32438

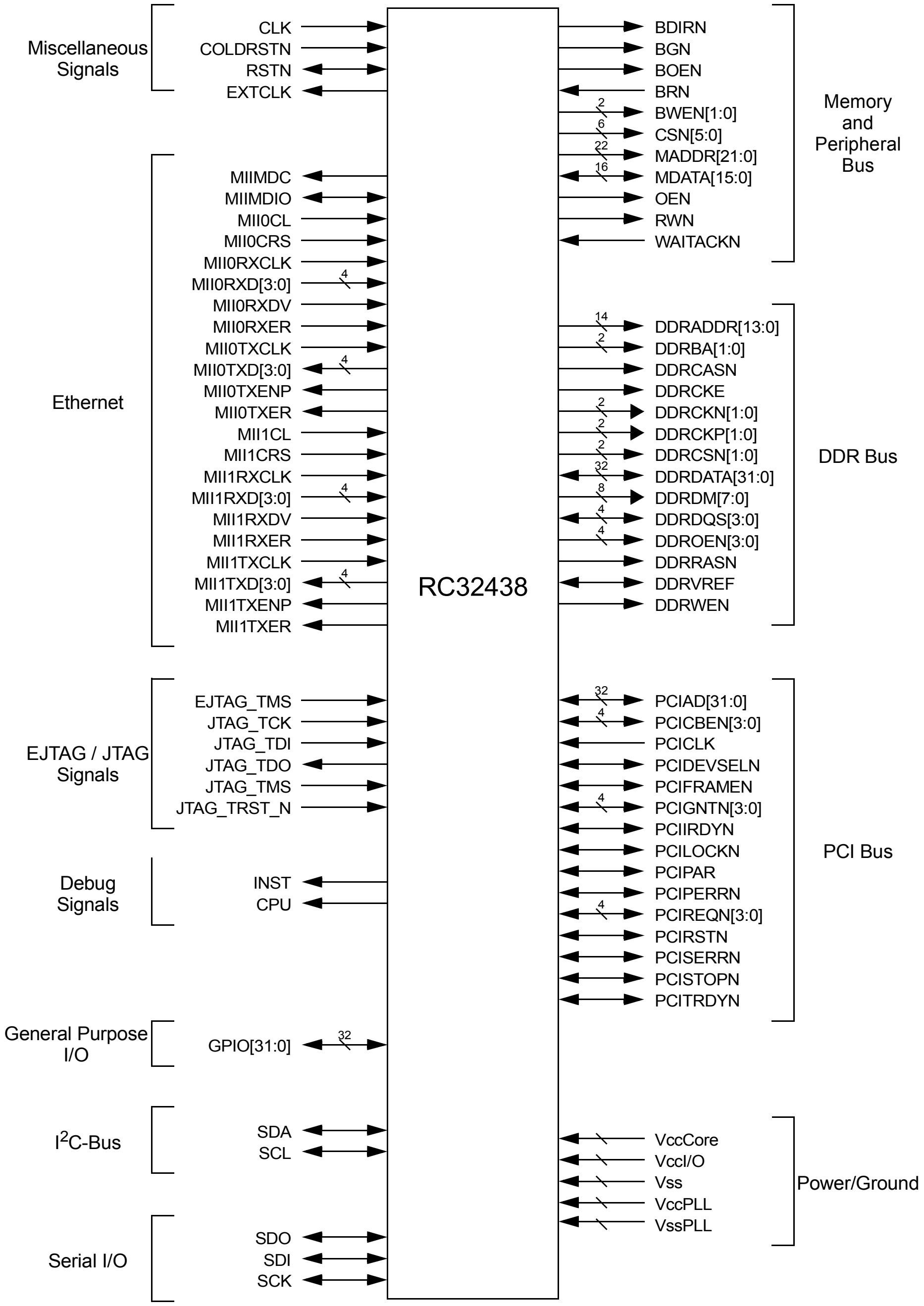


Figure 1 Logic Diagram

AC Timing Definitions

Below are examples of the AC timing characteristics used throughout this document.

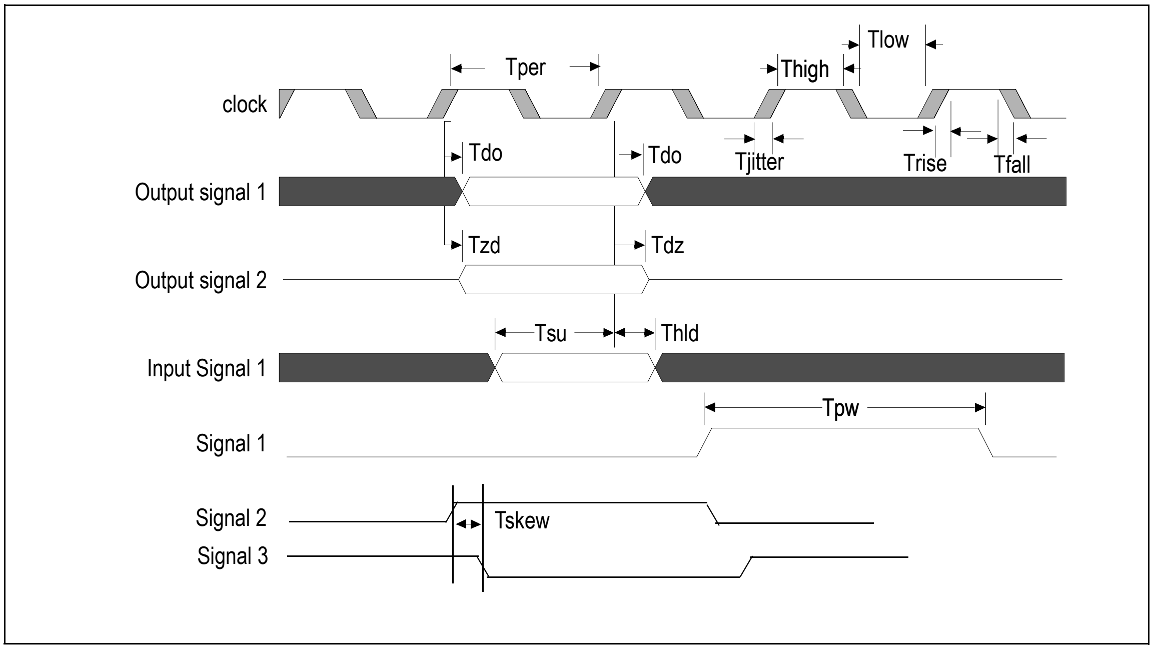


Figure 2 AC Timing Definitions Waveform

Symbol	Definition
Tper	Clock period.
Tlow	Clock low. Amount of time the clock is low in one clock period.
Thigh	Clock high. Amount of time the clock is high in one clock period.
Trise	Rise time. Low to high transition time.
Tfall	Fall time. High to low transition time.
Tjitter	Jitter. Amount of time the reference clock (or signal) edge can vary on either the rising or falling edges.
Tdo	Data out. Amount of time after the reference clock edge that the output will become valid. The minimum time represents the data output hold. The maximum time represents the earliest time the designer can use the data.
Tzd	Z state to data valid. Amount of time after the reference clock edge that the tri-stated output takes to become valid.
Tdz	Data valid to Z state. Amount of time after the reference clock edge that the valid output takes to become tri-stated.
Tsu	Input set-up. Amount of time before the reference clock edge that the input must be valid.
Thld	Input hold. Amount of time after the reference clock edge that the input must remain valid.
Tpw	Pulse width. Amount of time the input or output is active for asynchronous signals.
Tslew	Slew rate. The rise or fall rate for a signal to go from a high to low, or low to high.
X(clock)	Timing value. This notation represents a value of 'X' multiplied by the clock time period of the specified clock. Using 5(CLK) as an example: X = 5 and the oscillator clock (CLK) = 25MHz, then the timing value is 200.
Tskew	Skew. The amount of time two signal edges deviate from one another.

Table 4 AC Timing Definitions

System Clock Parameters

Values based on systems running at recommended supply voltages and operating temperatures, as shown in Tables 15 and 16.

Parameter	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Units	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max		
PCLK[1]	Frequency	none	200	200	200	233	200	266	200	300	MHz	See Figure 3.
	Tper		5.0	5.0	4.2	5.0	3.8	5.0	3.3	5.0	ns	
ICLK[2,3,4]	Frequency	none	100	100	100	116.5	100	133	100	150	MHz	
	Tper		10.0	10.0	10.0	8.5	10.0	7.5	6.7	10.0	ns	
CLK[5]	Frequency	none	25	66.6	25	77.6	25	88.6	25	100	MHz	
	Tper_5a		15.0	40.0	12.9	40.0	11.2	40.0	10	40	ns	
	Thigh_5a, Tlow_5a		40	60	40	60	40	60	40	60	% of Tper_5a	
	Trise_5a, Tfall_5a		—	3.0	—	3.0	—	3.0	—	3.0	ns	
	Tjitter_5a		—	0.1	—	0.1	—	0.1	—	0.1	ns	

Table 5 Clock Parameters

[1] The CPU pipeline clock (PCLK) speed is selected during cold reset by the boot configuration vector (see Table 3).

[2] ICLK is the internal IPBus clock. It is always equal to PCLK divided by 2. This clock cannot be sampled externally.

[3] The ethernet clock (MIIxRXCLK and MIIxTXCLK) frequency must be equal to or less than 1/2 ICLK (MIIxRXCLK and MIIxTXCLK <= 1/2(ICLK)).

[4] PCICLK must be equal to or less than two times ICLK (PCICLK <= 2(ICLK)) with a maximum PCICLK of 66MHz.

[5] The input clock (CLK) is input from the external oscillator to the internal PLL.

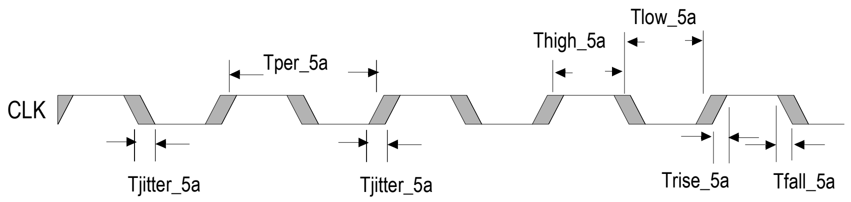


Figure 3 Clock Parameters Waveform

AC Timing Characteristics

Values given below are based on systems running at recommended operating temperatures and supply voltages, shown in Tables 15 and 16.

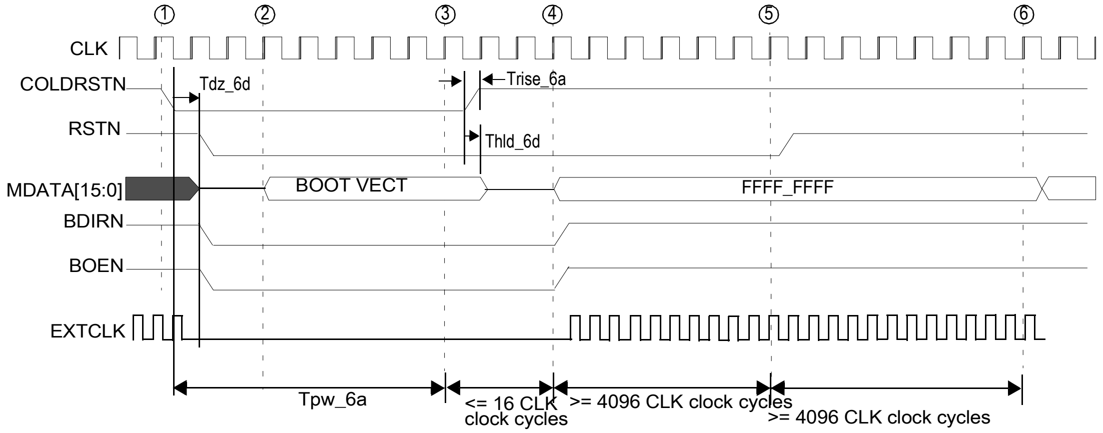
Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
Reset													
COLDRSTN[1]	Tpw_6a[2]	none	OSC + 0.5	—	OSC + 0.5	—	OSC + 0.5	—	OSC + 0.5	—	ms	Cold reset	See Figures 4 and 5.
	Trise_6a	none	—	5.0	—	5.0	—	5.0	—	5.0	ns	Cold reset	
RSTN[3] (input)	Tpw_6b[2]	none	2(CLK)	—	2(CLK)	—	2(CLK)	—	2(CLK)	—	ns	Warm reset	
RSTN[3] (output)	Tdo_6c	COLDRSTN falling	—	15.0	—	15.0	—	15.0	—	15.0	ns	Cold reset	
MDATA[15:0] (boot vector)	Thld_6d	COLDRSTN rising	3.0	—	3.0	—	3.0	—	3.0	—	ns	Cold reset	
	Tdz_6d[2]	COLDRSTN falling	—	30.0	—	30.0	—	30.0	—	30.0	ns	Cold reset	
	Tdz_6d[2]	RSTN falling	—	5(CLK)	—	5(CLK)	—	5(CLK)	—	5(CLK)	ns	Warm reset	
	Tzd_6d[2]	RSTN rising	2(CLK)	—	2(CLK)	—	2(CLK)	—	2(CLK)	—	ns	Warm reset	

Table 6 Reset and System AC Timing Characteristics

[1] The COLDRSTN minimum pulse width is the oscillator stabilization time (OSC) plus 0.5 ms with V_{cc} stable.

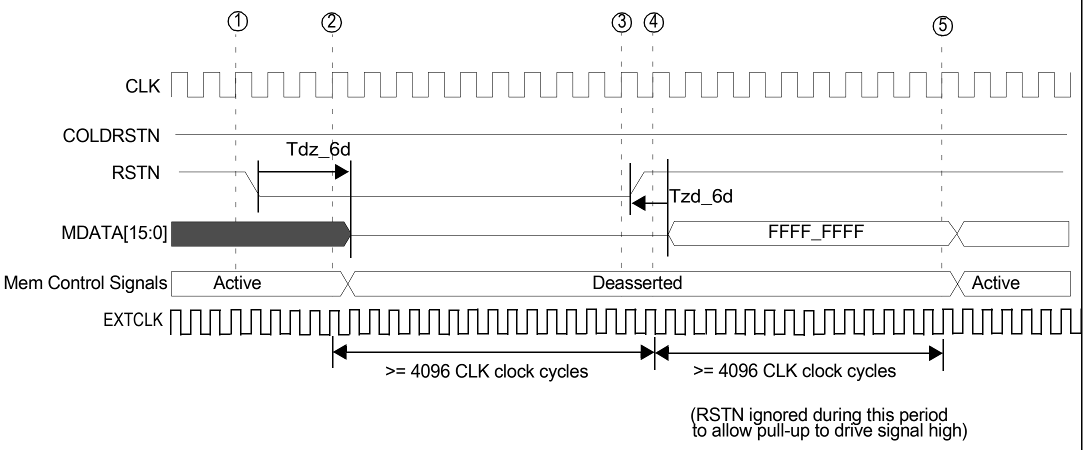
[2] The values for this symbol were determined by calculation, not by testing.

[3] RSTN is a bidirectional signal. It is treated as an asynchronous input.



1. COLDRSTN asserted by external logic. The RC32438 asserts RSTN, asserts BOEN low, drives BDIRN low, disables EXTCLK, and tri-states the data bus and all output pins in response.
2. External logic begins driving valid boot configuration vector on the data bus, and the RC32438 starts sampling it.
3. External logic negates COLDRSTN and tri-states the boot configuration vector on MDATA[15:0]. The boot configuration vector must not be tri-stated before COLDRSTN is negated. The RC32438 stops sampling the boot configuration vector.
4. The RC32438 starts driving the data bus, MDATA[15:0], negates BOEN, drives BDIRN high, and starts driving EXTCLK.
5. RSTN negated by the RC32438.
6. CPU begins executing by taking MIPS reset exception, and the RC32438 starts sampling RSTN as a warm reset input.

Figure 4 Cold Reset AC Timing Waveform



1. Warm reset caused by any of the conditions listed in the Warm Reset section of Chapter 3, Clocking and Initialization, in the RC32438 User Reference Manual.
2. The RC32438 tri-states the data bus, MDATA[15:0], and negates all memory control signals.
3. The RC32438 negates RSTN.
4. The RC32438 starts driving the data bus, MDATA[15:0], again, but does not sample the RSTN input.
5. CPU begins executing by taking a MIPS soft reset exception and also starts sampling the RSTN input again.

Figure 5 Warm Reset AC Timing Waveform

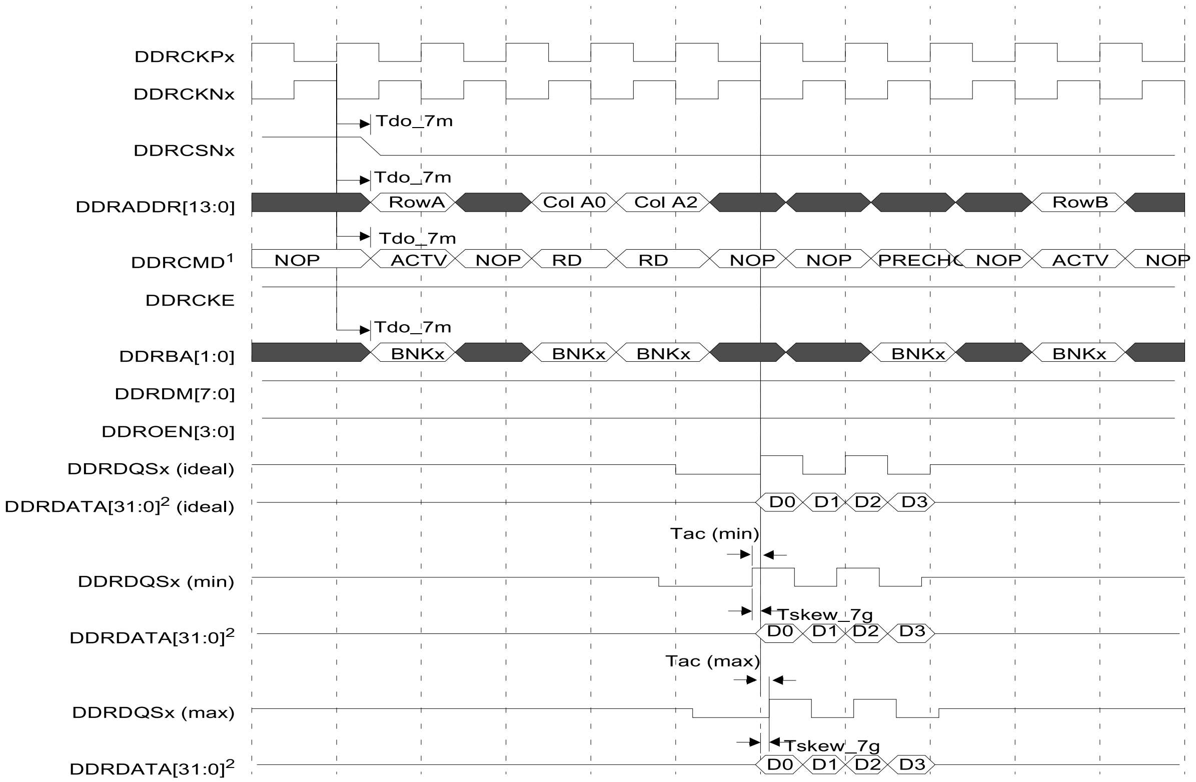
Signal	Symbol[1]	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
Memory Bus - DDR Access													
DDRDATA[31:0]	Tskew_7g[2]	DDRDQSx	0.0	0.9	0.0	0.9	0.0	0.9	0.0	0.8	ns		See Figures 6 and 7.
	Tdo_7k[3]		1.5	3.3	1.1	2.9	0.9	2.7	0.7	2.4	ns		
DDRDM[7:0]	Tdo_7l	DDRDQSx	1.5	3.3	1.1	2.9	0.9	2.7	0.7	2.4	ns		
DDRDQS[3:0]	Tac	DDRCKPx	-0.75	0.75	-0.75	0.75	-0.75	0.75	-0.75	0.75	ns		
DDRADDR[13:0], DDRBA[1:0], DDRCASN, DDRCKE, DDRCSN[1:0], DDROEN[3:0], DDRRASN, DDRWEN	Tdo_7m[4]	DDRCKPx	1.1	4.5	1.1	4.5	1.1	4.5	1.1	4.5	ns		

Table 7 DDR SDRAM Timing Characteristics

[1] In the DDR data sheet: Tskew_7g = $t_{DQSQ;}$ Tdo_7k = t_{DH}, $t_{DS;}$ Tdo_7l = t_{DH}, $t_{DS;}$ Tac = $t_{AC;}$ Tdo_7m = t_{IH}, $t_{IS.}$

[2] Meets DDR timing requirements for DDR 266 SDRAMs with 400 ps remaining margin to compensate for PCB propagation mismatches, which is adequate to guarantee functional timing, provided the RC32438 DDR layout guidelines are followed.

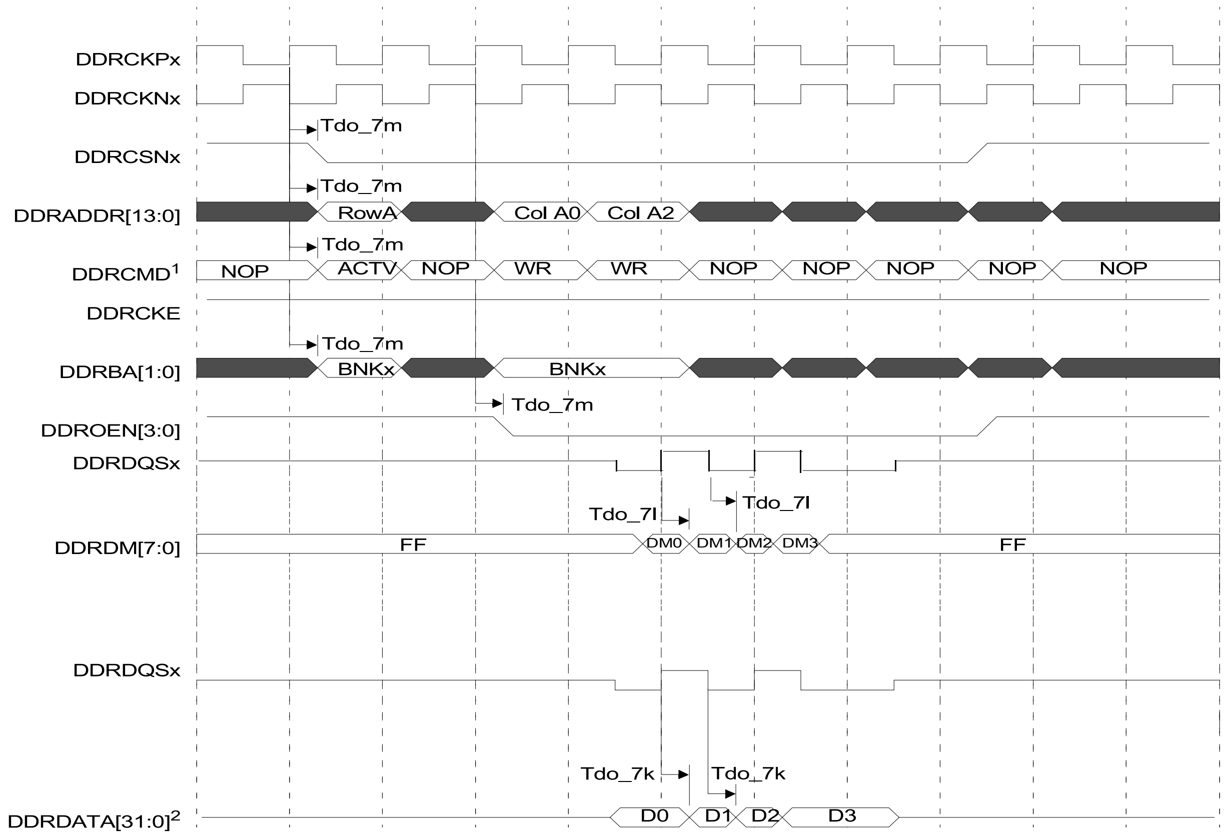
[3] Setup times are calculated as applicable clock period - Tdo max. For example, if the DDR is running at 266MHz, it uses a 133MHz input clock. The period for a 133MHz clock is 7.5ns. If the Tdo max value is 4.5ns, the T_{IS} parameter is 7.5ns minus 4.5ns = 3ns. The DDR spec for this parameter is 1ns, so there is 2ns of slack left over for board propagation. Calculations for T_{DS} are similar, but since this parameter is taken relative to the DDRDQS signals, which are referenced on both edges, the effective period with a 133MHz input clock is only 3.75ns. So, if the max Tdo is 2.7ns, we have 3.75ns minus 2.7ns = 1.05ns for T_{DS}. The DDR data sheet specs a value of 0.5ns for 266MHz, so this leaves 0.55ns slack for board propagation delays.



[1] DDRCMD contains DDRRASN, DDRCASN and DDRWEN.

[2] DDRDATA is either 32-bits or 16-bits wide depending on the DBW control bit in DDRC Register (see Chapter 7, DDR Controller, in the RC32438 User Reference Manual).

Figure 6 DDR SDRAM AC Timing Waveform - SDRAM Read Access



[1] DDRCMD contains DDRRASN, DDRCASN and DDRWEN.

[2] DDRDATA is either 32-bits or 16-bits wide depending on the DBW control bit in DDRC Register (see Chapter 7, DDR Controller, in the RC32438 User Reference Manual).

Figure 7 DDR SDRAM Timing Waveform — Write Access

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
Memory and Peripheral Bus[1]													See Figures 8 and 9.
MADDR[21:0]	Tdo_8a	EXTCLK rising	0.0	5.0	0.0	5.0	0.0	5.0	0.0	5.0	ns		
	Tdz_8a[2]		0.0	0.1	0.0	0.1	0.0	0.1	0.0	0.1	ns		
	Tzd_8a[2]		0.5	2.3	0.5	2.3	0.5	2.3	0.5	2.3	ns		
MADDR[25:22]	Tdo_8b	EXTCLK rising	0.0	6.5	0.0	6.5	0.0	6.5	0.0	6.5	ns		
	Tdz_8b[2]		0.7	1.5	0.7	1.5	0.7	1.5	0.7	1.5	ns		
	Tzd_8b[2]		1.2	3.3	1.2	3.3	1.2	3.3	1.2	3.3	ns		

Table 8 Memory and Peripheral Bus AC Timing Characteristics (Part 1 of 3)

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
MDATA[15:0]	Tsu_8c	EXTCLK rising	7.0	—	7.0	—	7.0	—	7.0	—	ns		See Figures 8 and 9 (cont.)
	Thld_8c		0.0	—	0.0	—	0.0	—	0.0	—	ns		
	Tdo_8c		0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	ns		
	Tdz_8c[2]		0.0	0.1	0.0	0.1	0.0	0.1	0.0	0.1	ns		
	Tzd_8c[2]		0.5	2.2	0.5	2.2	0.5	2.2	0.5	2.2	ns		
EXTCLK[3]	Tper_8d	none	10.0	—	8.33	—	7.5	—	6.66	—	ns		
BDIRN	Tdo_8e	EXTCLK rising	1.0	4.0	1.0	4.0	1.0	4.0	1.0	4.0	ns		
	Tdz_8e[2]		-1.0	-0.1	-1.0	-0.1	-1.0	-0.1	-1.0	-0.1	ns		
	Tzd_8e[2]		0.4	1.0	0.4	1.0	0.4	1.0	0.4	1.0	ns		
BOEN	Tdo_8f	EXTCLK rising	1.0	4.0	1.0	4.0	1.0	4.0	1.0	4.0	ns		
	Tdz_8f[2]		0.1	0.4	0.1	0.4	0.1	0.4	0.1	0.4	ns		
	Tzd_8f[2]		1.1	2.0	1.1	2.0	1.1	2.0	1.1	2.0	ns		
BRN	Tsu_8g	EXTCLK rising	5.5	—	5.5	—	5.5	—	5.5	—	ns		
	Thld_8g		0.0	—	0.0	—	0.0	—	0.0	—	ns		
BGN	Tdo_8h	EXTCLK rising	1.0	4.0	1.0	4.0	1.0	4.0	1.0	4.0	ns		
WAITACKN[4]	Tsu_8h	EXTCLK rising	5.8	—	5.8	—	5.8	—	5.8	—	ns		
	Thld_8h		0.0	—	0.0	—	0.0	—	0.0	—	ns		
	Tpw_8h[2]	none	2(EXTCLK)	—	2(EXTCLK)	—	2(EXTCLK)	—	2(EXTCLK)	—	ns		
CSN[5:0]	Tdo_8i	EXTCLK rising	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	ns		
	Tdz_8i[2]		0.1	0.4	0.1	0.4	0.1	0.4	0.1	0.4	ns		
	Tzd_8i[2]		0.6	2.2	0.6	2.2	0.6	2.2	0.6	2.2	ns		
RWN	Tdo_8j	EXTCLK rising	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	ns		
	Tdz_8j[2]		-0.7	0.1	-0.7	0.1	-0.7	0.1	-0.7	0.1	ns		
	Tzd_8j[2]		0.6	1.1	0.6	1.1	0.6	1.1	0.6	1.1	ns		
OEN	Tdo_8k	EXTCLK rising	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	ns		
	Tdz_8k[2]		-0.4	0.2	-0.4	0.2	-0.4	0.2	-0.4	0.2	ns		
	Tzd_8k[2]		0.8	1.5	0.8	1.5	0.8	1.5	0.8	1.5	ns		
BWEN[1:0]	Tdo_8l	EXTCLK rising	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	ns		
	Tdz_8l[2]		0	0.2	0	0.2	0	0.2	0	0.2	ns		
	Tzd_8l[2]		0.8	1.7	0.8	1.7	0.8	1.7	0.8	1.7	ns		

Table 8 Memory and Peripheral Bus AC Timing Characteristics (Part 2 of 3)

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
DMAREQN[1:0]	Tpw_8n[2]	None	2(ICLK)	—	2(ICLK)		2(ICLK)	—	2(ICLK)	—	ns		See Figures 10 and 11.
DMADONEN[1:0]	Tsu_8o	EXTCLK rising	6.0	—	6.0	—	6.0	—	6.0	—	ns		
	Thld_8o		1.0	—	1.0	—	1.0	—	1.0	—	ns		
DMAFINN[1:0]	Tdo_8p	EXTCLK rising	1.5	6.0	1.5	6.0	1.5	6.0	1.5	6.0	ns		
CPU, INST	Tdo_8m	EXTCLK rising	2.0	10.0	2.0	10.0	2.0	10.0	2.0	10.0	ns		See Figures 8 and 9.

Table 8 Memory and Peripheral Bus AC Timing Characteristics (Part 3 of 3)

[1] The RC32438 provides bus turnaround cycles to prevent bus contention when going from a read to write, write to read, and during external bus ownership. For example, there are no cycles where an external device and the RC32438 are both driving. See Chapter 6, Device Controller, in the RC32438 User Reference Manual.

[2] The values for this symbol were determined by calculation, not by testing.

[3] The frequency of EXTCLK is programmable. See the External Clock Divider description in Table 3 of this data sheet.

[4] WAITACKN must meet the setup and hold times if it is synchronous or the minimum pulse width if it is asynchronous.

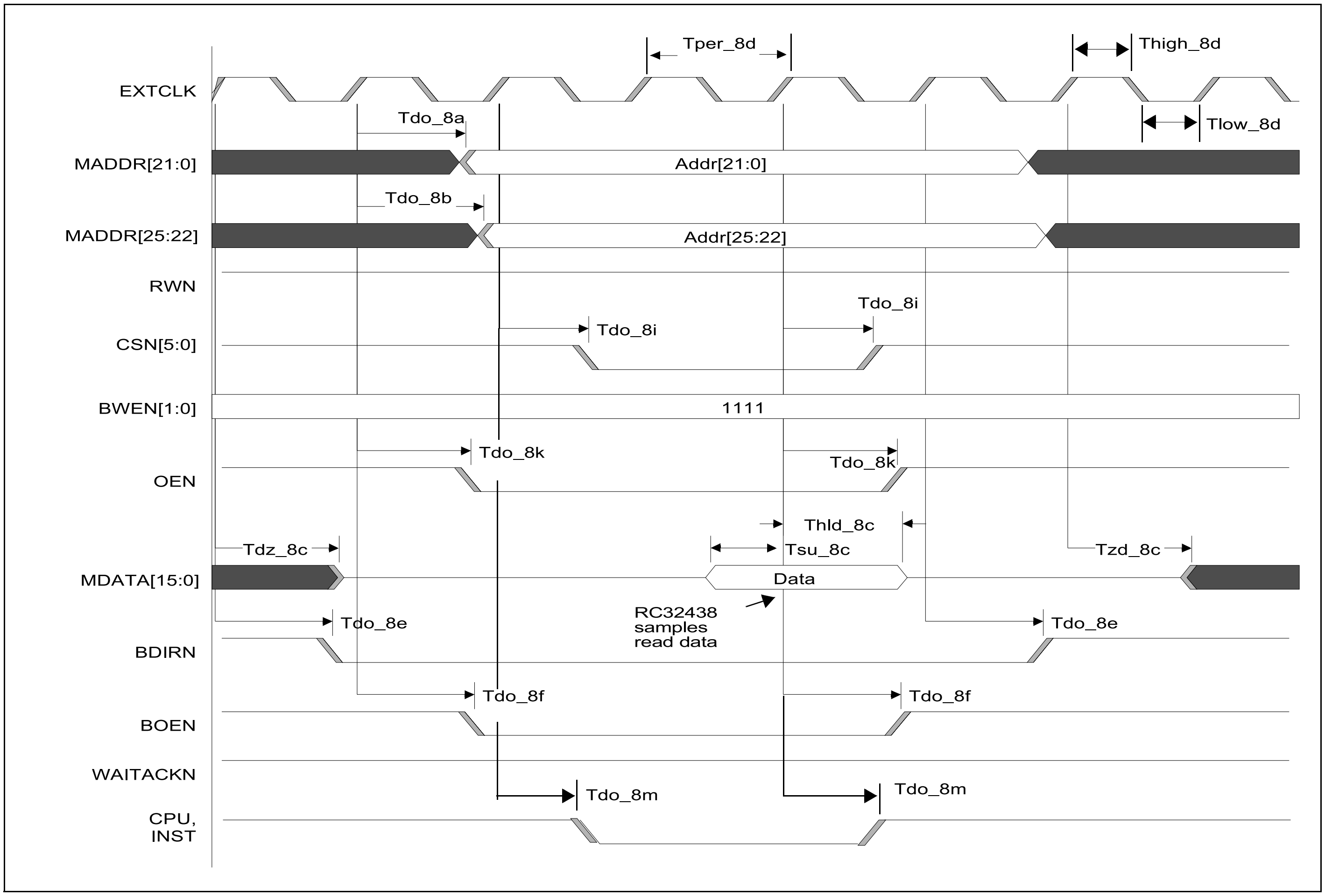


Figure 8 Memory and Peripheral Bus AC Timing Waveform — Read Access