Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	MIPS32
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	233MHz
Co-Processors/DSP	-
RAM Controllers	DDR
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10/100Mbps (2)
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	416-BGA
Supplier Device Package	416-PBGA (27x27)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/79rc32k438-233bbgi

- **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^2C-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 non-maskable interrupt input, and each signal may be used as a bit input or output port.

I^2C 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.

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
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]
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.

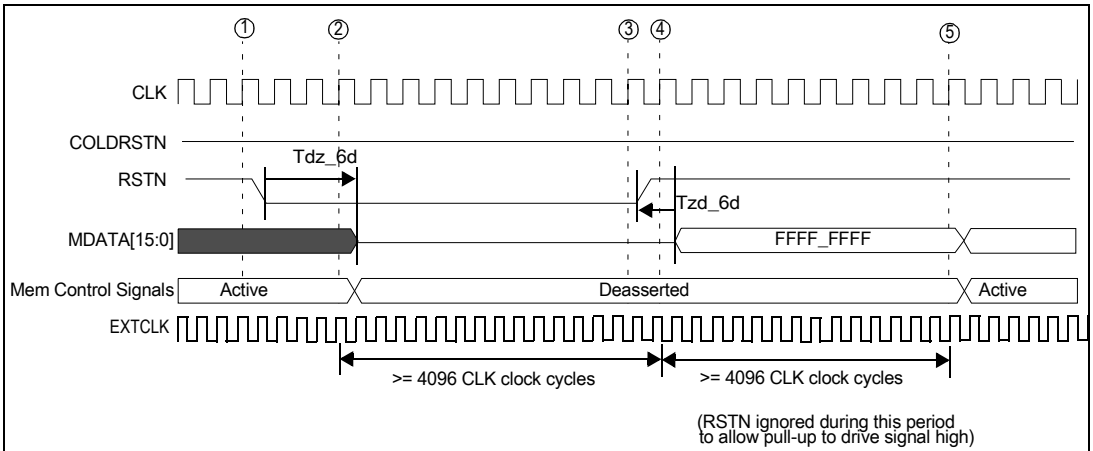
Boot Configuration Vector

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)



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.

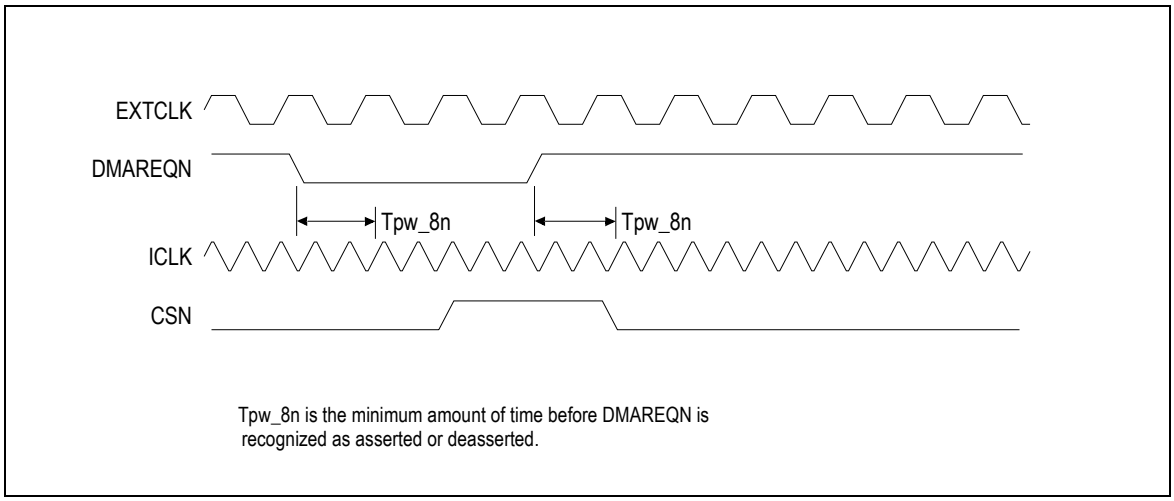


Figure 11 DMAREQN AC Timing Waveform

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
Ethernet[1]													
MIIMDC	Tper_9a	None	40.0	—	33.3	—	30.0	—	30.0	—	ns		See Figure 12.
	Thigh_9a, Tlow_9a		16.0	—	13.0	—	12.0	—	12.0	—	ns		
MIIMDIO	Tsu_9b	MIIMDC rising	10.0	—	10.0	—	10.0	—	10.0	—	ns		
	Thld_9b		0.0	—	0.0	—	0.0	—	0.0	—	ns		
	Tdo_9b[2]		10	300	10	300	10	300	10	300	ns		
MIIxRXCLK, MIIxTXCLK[3]	Tper_9c	None	399.96	400.4	399.96	400.4	399.96	400.4	399.96	400.4	ns	10 Mbps	
	Thigh_9c, Tlow_9c		140	260	140	260	140	260	140	260	ns		
	Trise_9c, Tfall_9c		—	3.0	—	3.0	—	3.0	—	3.0	ns		
MIIxRXCLK, MIIxTXCLK[3]	Tper_9d	None	39.9	40.0	39.9	40.0	39.9	40.0	39.9	40.0	ns	100 Mbps	
	Thigh_9d, Tlow_9d		14.0	26.0	14.0	26.0	14.0	26.0	14.0	26.0	ns		
	Trise_9d, Tfall_9d		—	2.0	—	2.0	—	2.0	—	2.0	ns		
MIIxRXD[3:0], MIIxRXDV, MIIxRXER	Tsu_9e	MIIxRXCLK rising	10.0	—	10.0	—	10.0	—	10.0	—	ns		
	Thld_9e		10.0	—	10.0	—	10.0	—	10.0	—	ns		
MIIxTXD[3:0], MIIxTXENP, MIIxTXER	Tdo_9f	MIIxTXCLK rising	0.0	25.0	0.0	25.0	0.0	25.0	0.0	25.0	ns		

Table 9 Ethernet AC Timing Characteristics

1. There are two MII interfaces and the timing is the same for each. "X" represents interface 0 or 1.

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

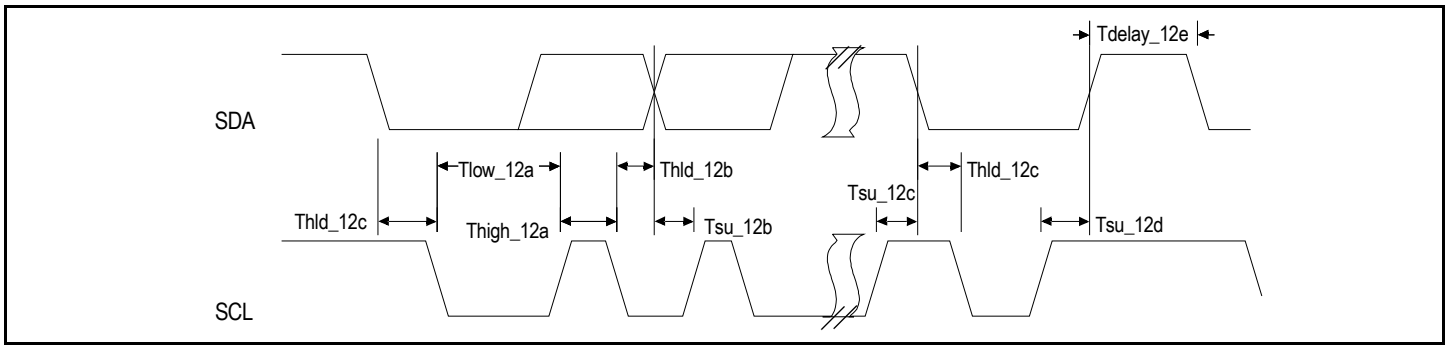


Figure 16 I²C AC Timing Waveform

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
GPIO													
GPIO[31:0][1]	Tpw_13b[2]	None	2(ICLK)	—	2(ICLK)	—	2(ICLK)	—	2(ICLK)	—	ns		See Figure 17.

Table 12 GPIO AC Timing Characteristics

1. GPIO signals must meet the setup and hold times if they are synchronous or the minimum pulse width if they are asynchronous.

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

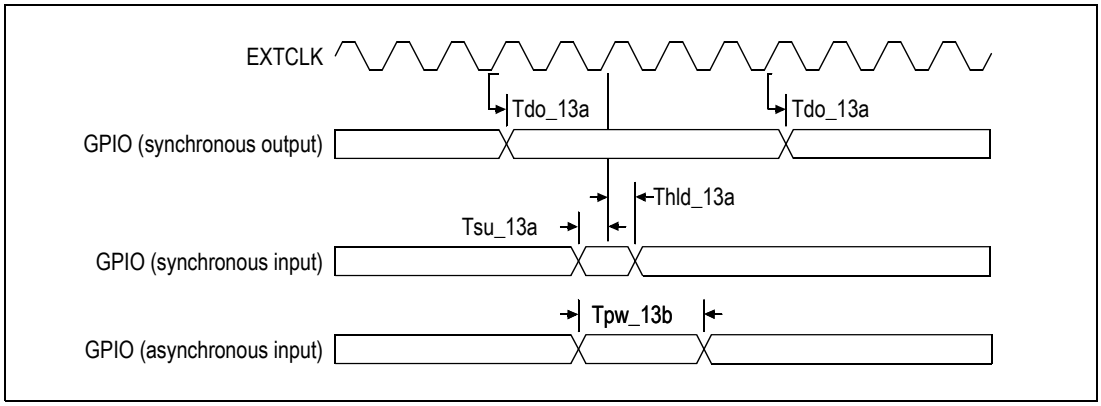


Figure 17 GPIO AC Timing Waveform

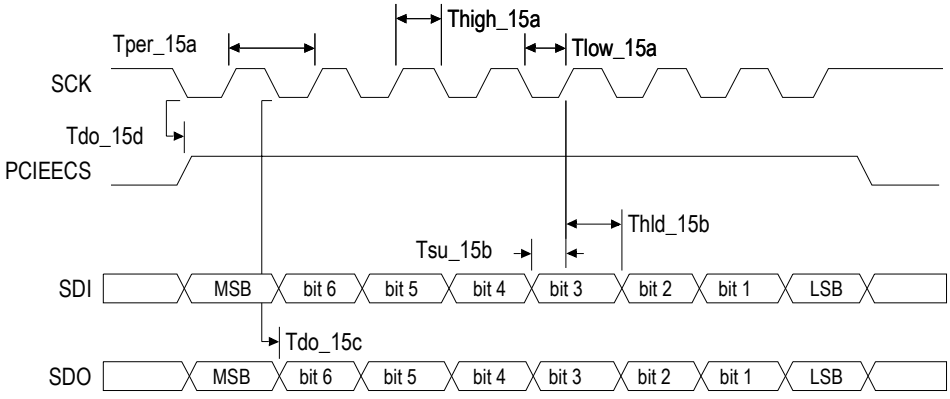
Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
SPI[1]													
SCK	Tper_15a	None	—	1920	—	1920	—	1920	—	1920	ns	33 MHz PCI	See Figures 18, 19, 20 and 21.
	Tper_15a		—	960	—	960	—	960	—	960	ns	66 MHz PCI	
	Tper_15a		100	166667	100	166667	100	166667	100	166667	ns	SPI	
	Thigh_15a, Tlow_15a		930	990	930	990	930	990	930	990	ns	33 MHz PCI	
	Thigh_15a, Tlow_15a		465	495	465	495	465	495	465	495	ns	66 MHz PCI	
	Thigh_15a, Tlow_15a		40	83353	40	83353	40	83353	40	83353	ns	SPI	
SDI	Tsu_15b	SCK rising or falling	60	—	60	—	60	—	60	—	ns	SPI or PCI	
	Thld_15b		60	—	60	—	60	—	60	—	ns		
SDO	Tdo_15c	SCK rising or falling	0	60	0	60	0	60	0	60	ns	SPI or PCI	
PCIEECS[2]	Tdo_15d	SCK rising or falling	0	60	0	60	0	60	0	60	ns	PCI	
SCK, SDI, SDO[3]	Tpw_15e	None	2(ICLK)	—	2(ICLK)	—	2(ICLK)	—	2(ICLK)	—	ns	Bit I/O	

Table 13 SPI AC Timing Characteristics

1. In SPI mode, the SCK period and sampling edge are programmable. In PCI mode, the SCK period is fixed and the sampling edge is rising.

2. PCIEECS is the PCI serial EEPROM chip select. It is an alternate function of PCIGNTN[1].

3. In Bit I/O mode, SCK, SDI, and SDO must meet the setup and hold times if they are synchronous or the minimum pulse width if they are asynchronous.



Loading PCI configuration registers through SPI from an EEPROM.

Figure 18 SPI AC Timing Waveform — PCI Configurations Load

Signal	Symbol	Reference Edge	200MHz		233MHz		266MHz		300MHz		Unit	Conditions	Timing Diagram Reference
			Min	Max	Min	Max	Min	Max	Min	Max			
EJTAG and JTAG													
JTAG_TCK	Tper_16a	none	25.0	50.0	25.0	50.0	25.0	50.0	25.0	50.0	ns		See Figure 22.
	Thigh_16a, Tlow_16a		10.0	25.0	10.0	25.0	10.0	25.0	10.0	25.0	ns		
JTAG_TMS[1], JTAG_TDI	Tsu_16b	JTAG_TCK rising	2.4	—	2.4	—	2.4	—	2.4	—	ns		
	Thld_16b		1.0	—	1.0	—	1.0	—	1.0	—	ns		
JTAG_TDO	Tdo_16c	JTAG_TCK falling	—	11.3	—	11.3	—	11.3	—	11.3	ns		
	Tdz_16c[2]		—	11.3	—	11.3	—	11.3	—	11.3	ns		
JTAG_TRST_N	Tpw_16d[2]	none	25.0	—	25.0	—	25.0	—	25.0	—	ns		
EJTAG_TMS[1]	Tsu_16e	JTAG_TCK rising	2.0	—	2.0	—	2.0	—	2.0	—	ns		
	Thld_6e		1.0	—	1.0	—	1.0	—	1.0	—	ns		
VSENSE	Trise_16f	none	—	2	—	2	—	2	—	2	sec	Measured from 0.5V (T_{active})	See Figure 24.

Table 14 JTAG AC Timing Characteristics

1. The JTAG specification, IEEE 1149.1, recommends that both JTAG_TMS and EJTAG_TMS should be held at 1 while the signal applied at JTAG_TRST_N changes from 0 to 1. Otherwise, a race may occur if JTAG_TRST_N is deasserted (going from low to high) on a rising edge of JTAG_TCK when either JTAG_TMS or EJTAG_TMS is low, because the TAP controller might go to either the Run-Test/Idle state or stay in the Test-Logic-Reset state.

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

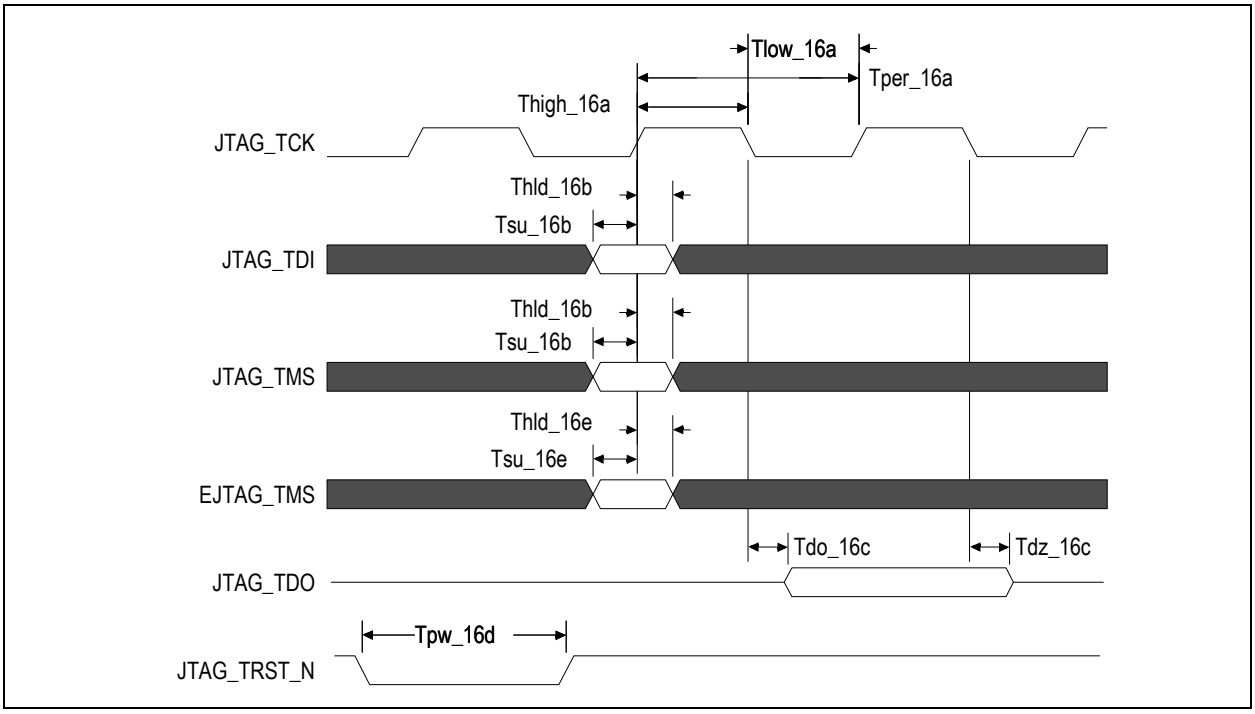


Figure 22 JTAG AC Timing Waveform

The IEEE 1149.1 specification requires that the JTAG and EJTAG TAP controllers be reset at power-up whether or not the interfaces are used for a boundary scan or a probe. Reset can occur through a pull-down resistor on JTAG_TRST_N if the probe is not connected. However, on-chip pull-up resistors are implemented on the RC32438 due to an IEEE 1149.1 requirement. Having on-chip pull-up and external pull-down resistors for the JTAG_TRST_N signal requires special care in the design to ensure that a valid logical level is provided to JTAG_TRST_N, such as using a small external pull-down resistor to ensure this level overrides the on-chip pull-up. An alternative is to use an active power-up reset circuit for JTAG_TRST_N, which drives JTAG_TRST_N low only at power-up and then holds JTAG_TRST_N high afterwards with a pull-up resistor.

Figure 23 shows the electrical connection of the EJTAG probe target system connector.

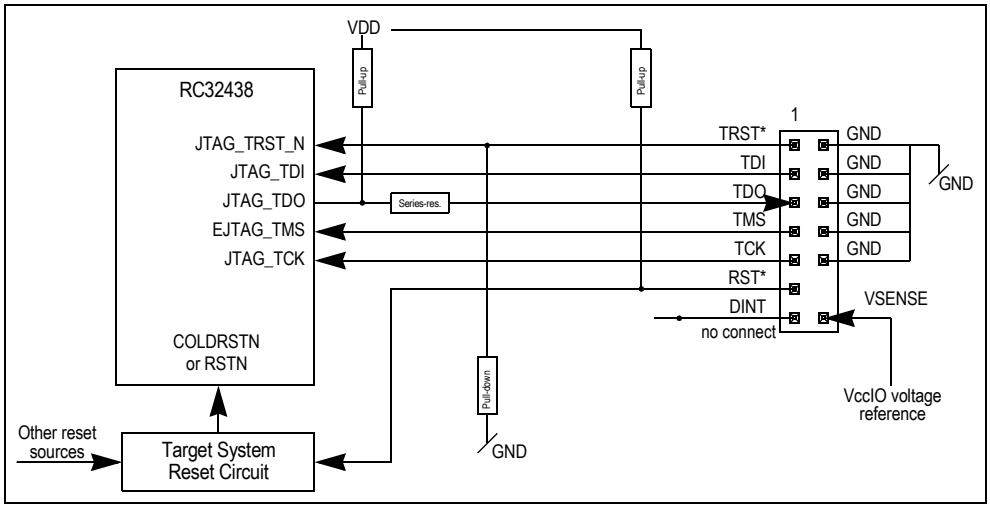


Figure 23 Target System Electrical EJTAG Connection

Package Pin-out — 416-PBGA Signal Pinout for RC32438

The following table lists the pin numbers, signal names, and number of alternate functions for the RC32438 device. Signal names ending with an "_N" or "N" are active when low.

Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt
A1	MII0CL		D11	V_{ss}		P1	GPIO[00]	1	AC17	V_{ss}	
A2	GPIO[25]	1	D12	V_{ss}		P2	MIIMDIO		AC18	V_{ss}	
A3	GPIO[31]		D13	V_{cc} Core		P3	GPIO[02]	1	AC19	V_{ss}	
A4	CSN[05]		D14	V_{cc} Core		P4	V_{cc} I/O		AC20	V_{cc} I/O	
A5	CSN[02]		D15	V_{cc} Core		P23	V_{cc} CORE		AC21	V_{cc} I/O	
A6	BWEN[01]		D16	V_{ss}		P24	DDRDM[03]		AC22	V_{cc} I/O	
A7	BOEN		D17	V_{ss}		P25	DDRDATA[31]		AC23	Vcc SI/O	
A8	MDATA[15]		D18	V_{ss}		P26	DDRDATA[30]		AC24	Vcc SI/O	
A9	MDATA[14]		D19	V_{ss}		R1	INST		AC25	DDROEN[00]	
A10	MDATA[10]		D20	V_{cc} I/O		R2	EJTAG_TMS		AC26	DDRADDR[00]	
A11	MDATA[07]		D21	Vcc SI/O		R3	V_{ss}		AD1	JTAG_TRST_N	
A12	MDATA[06]		D22	Vcc SI/O		R4	V_{cc} I/O		AD2	JTAG_TMS	
A13	GPIO[29]		D23	Vcc SI/O		R23	Vcc SI/O		AD3	GPIO[15]	1
A14	GPIO[22]	1	D24	Vcc SI/O		R24	DDRDATA[29]		AD4	SDA	
A15	MADDR[21]		D25	DDRDATA[11]		R25	DDRADDR[13]		AD5	GPIO[27]	1
A16	MADDR[19]		D26	DDRDATA[10]		R26	DDRCSN[01]		AD6	PCIAD[30]	
A17	MADDR[16]		E1	MII0TXD[02]		T1	NC		AD7	PCIAD[26]	
A18	MADDR[13]		E2	MII0TXD[00]		T2	GPIO[03]	1	AD8	PCICBEN[03]	
A19	MADDR[10]		E3	MII0TXD[01]		T3	CPU		AD9	PCIAD[21]	
A20	MADDR[07]		E4	V_{ss}		T4	V_{cc} I/O		AD10	PCIAD[18]	
A21	MADDR[05]		E23	Vcc SI/O		T23	Vcc SI/O		AD11	PCIREQN[01]	
A22	MADDR[02]		E24	DDRDATA[09]		T24	DDRCSN[00]		AD12	PCICLK	
A23	RSTN		E25	DDRDATA[12]		T25	DDRADDR[10]		AD13	PCIGNTN[00]	
A24	DDRDATA[02]		E26	DDRDM[01]		T26	DDRADDR[12]		AD14	PCIIRDYN	
A25	DDRDATA[04]		F1	MII0TXER		U1	JTAG_TDI		AD15	PCISTOPN	
A26	DDRDATA[05]		F2	MII0TXD[03]		U2	JTAG_TCK		AD16	PCIPERRN	
B1	MII0CRS		F3	MII0TXENP		U3	JTAG_TDO		AD17	PCIAD[15]	
B2	WAITACKN		F4	V_{ss}		U4	V_{cc} I/O		AD18	PCIAD[11]	
B3	RWN		F23	Vcc SI/O		U23	V_{ss}		AD19	PCIAD[08]	
B4	CSN[04]		F24	DDRDQS[01]		U24	DDRADDR[11]		AD20	PCIAD[06]	
B5	CSN[01]		F25	DDRDATA[15]		U25	DDRWEN		AD21	PCIGNTN[03]	
B6	BWEN[00]		F26	DDRDATA[14]		U26	DDRADDR[09]		AD22	PCIAD[00]	
B7	BGN		G1	MII0RXER		V1	SDO		AD23	PCIAD[04]	
B8	MDATA[13]		G2	MII0RXDV		V2	SDI		AD24	DDRDM[05]	

Table 20 RC32438 416-pin Signal Pin-Out (Part 1 of 3)

Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt	Pin	Function	Alt
B9	MDATA[11]		G3	MII0TXCLK		V3	GPIO[05]	1	AD25	DDROEN[02]	
B10	MDATA[03]		G4	V_{ss}		V4	V_{ss}		AD26	DDROEN[01]	
B11	MDATA[08]		G23	V_{ss}		V23	V_{ss}		AE1	N/C	
B12	MDATA[02]		G24	DDRCKP[00]		V24	DDRADDR[08]		AE2	GPIO[13]	1
B13	GPIO[23]	1	G25	DDRDATA[16]		V25	DDRRASN		AE3	GPIO[18]	1
B14	MADDR[20]		G26	DDRDATA[13]		V26	DDRCASN		AE4	GPIO[24]	1
B15	GPIO[20]	1	H1	MII1CRS		W1	GPIO[04]	1	AE5	GPIO[26]	1
B16	MADDR[17]		H2	MII1CL		W2	SCK		AE6	PCIAD[31]	
B17	MADDR[14]		H3	MII1RXCLK		W3	CLK		AE7	PCIAD[28]	
B18	MADDR[12]		H4	V_{ss}		W4	V_{ss}		AE8	PCIAD[25]	
B19	MADDR[09]		H23	V_{ss}		W23	V_{ss}		AE9	GPIO[30]	1
B20	MADDR[06]		H24	DDRCKN[00]		W24	DDRADDR[07]		AE10	PCIAD[22]	
B21	MADDR[03]		H25	DDRDATA[18]		W25	DDRADDR[06]		AE11	PCIAD[19]	
B22	MADDR[00]		H26	DDRVREF		W26	DDRBA[01]		AE12	PCIAD[16]	
B23	DDRDATA[01]		J1	MII1RXD[01]		Y1	GPIO[06]	1	AE13	PCIRSTN	
B24	DDRDQS[00]		J2	MII1RXD[00]		Y2	V_{cc} PLL		AE14	PCIREQN[02]	
B25	DDRDM[00]		J3	MII1RXD[03]		Y3	GPIO[08]	1	AE15	PCIFRAMEN	
B26	DDRDATA[06]		J4	V_{ss}		Y4	V_{ss}		AE16	PCIDEVSELN	
C1	MII0RXD[00]		J23	V_{ss}		Y23	V_{ss}		AE17	PCILOCKN	
C2	MII0RXCLK		J24	DDRDATA[17]		Y24	DDRCKN[01]		AE18	PCICBEN[01]	
C3	EXTCLK		J25	DDRDATA[21]		Y25	DDRBA[00]		AE19	PCIAD[13]	
C4	COLDRSTN		J26	DDRDATA[19]		Y26	DDRADDR[05]		AE20	PCIAD[10]	
C5	OEN		K1	MII1RXDV		AA1	V_{ss} PLL		AE21	PCICBEN[00]	
C6	CSN[03]		K2	MII1RXD[02]		AA2	GPIO[07]	1	AE22	PCIAD[05]	
C7	CSN[00]		K3	MII1TXCLK		AA3	V_{cc} PLL		AE23	PCIAD[02]	
C8	BRN		K4	V_{cc} Core		AA4	V_{ss}		AE24	PCIGNTN[01]	
C9	BDIRN		K23	V_{ss}		AA23	V_{ss}		AE25	DDRDM[07]	
C10	MDATA[12]		K24	DDRDATA[20]		AA24	DDRCKP[01]		AE26	DDRDM[04]	
C11	MDATA[09]		K25	DDRDQS[02]		AA25	DDRADDR[03]		AF1	GPIO[16]	1
C12	MDATA[01]		K26	DDRCKE		AA26	DDRADDR[04]		AF2	GPIO[17]	1
C13	MDATA[05]		L1	MII1TXD[00]		AB1	GPIO[09]	1	AF3	GPIO[19]	1
C14	MDATA[04]		L2	MII1RXER		AB2	GPIO[14]	1	AF4	SCL	
C15	MDATA[00]		L3	MII1TXD[03]		AB3	GPIO[11]	1	AF5	GPIO[28]	1
C16	GPIO[21]	1	L4	V_{cc} Core		AB4	V_{ss}		AF6	PCIAD[29]	
C17	MADDR[18]		L23	V_{cc} Core		AB23	V_{ss}		AF7	PCIAD[27]	
C18	MADDR[15]		L24	DDRDM[02]		AB24	V_{cc} SI/O		AF8	PCIAD[24]	
C19	MADDR[11]		L25	DDRDATA[24]		AB25	DDRADDR[01]		AF9	PCIAD[23]	

Table 20 RC32438 416-pin Signal Pin-Out (Part 2 of 3)

Signal Name	I/O Type	Location	Signal Category
DDRADDR[00]	O	AC26	DDR Bus
DDRADDR[01]	O	AB25	
DDRADDR[02]	O	AB26	
DDRADDR[03]	O	AA25	
DDRADDR[04]	O	AA26	
DDRADDR[05]	O	Y26	
DDRADDR[06]	O	W25	
DDRADDR[07]	O	W24	
DDRADDR[08]	O	V24	
DDRADDR[09]	O	U26	
DDRADDR[10]	O	T25	
DDRADDR[11]	O	U24	
DDRADDR[12]	O	T26	
DDRADDR[13]	O	R25	
DDRBA[00]	O	Y25	
DDRBA[01]	O	W26	
DDRCASN	O	V26	
DDRCKE	O	K26	
DDRCKN[00]	O	H24	
DDRCKN[01]	O	Y24	
DDRCKP[00]	O	G24	
DDRCKP[01]	O	AA24	
DDRCSN[00]	O	T24	
DDRCSN[01]	O	R26	
DDRDATA[00]	I/O	C23	
DDRDATA[01]	I/O	B23	
DDRDATA[02]	I/O	A24	
DDRDATA[03]	I/O	C24	
DDRDATA[04]	I/O	A25	
DDRDATA[05]	I/O	A26	
DDRDATA[06]	I/O	B26	
DDRDATA[07]	I/O	C26	
DDRDATA[08]	I/O	C25	
DDRDATA[09]	I/O	E24	
DDRDATA[10]	I/O	D26	

Table 24 RC32438 Alphabetical Signal List (Part 2 of 9)

Signal Name	I/O Type	Location	Signal Category
PCIAD[21]	I/O	AD9	PCI Bus
PCIAD[22]	I/O	AE10	
PCIAD[23]	I/O	AF9	
PCIAD[24]	I/O	AF8	
PCIAD[25]	I/O	AE8	
PCIAD[26]	I/O	AD7	
PCIAD[27]	I/O	AF7	
PCIAD[28]	I/O	AE7	
PCIAD[29]	I/O	AF6	
PCIAD[30]	I/O	AD6	
PCIAD[31]	I/O	AE6	
PCICBEN[00]	I/O	AE21	
PCICBEN[01]	I/O	AE18	
PCICBEN[02]	I/O	AF14	
PCICBEN[03]	I/O	AD8	
PCICLK	I	AD12	
PCIDEVSELN	I/O	AE16	
PCIFRAMEN	I/O	AE15	
PCIGNTN[00]	I/O	AD13	
PCIGNTN[01]	I/O	AE24	
PCIGNTN[02]	I/O	AF24	
PCIGNTN[03]	I/O	AD21	
PCIIRDYN	I/O	AD14	
PCILOCKN	I/O	AE17	
PCIPAR	I/O	AF17	
PCIPERRN	I/O	AD16	
PCIREQN[00]	I/O	AF13	
PCIREQN[01]	I/O	AD11	
PCIREQN[02]	I/O	AE14	
PCIREQN[03]	I/O	AF12	
PCIRSTN	I/O	AE13	
PCISERRN	I/O	AF16	
PCISTOPN	I/O	AD15	
PCITRDYN	I/O	AF15	
RSTN	I/O	A23	System
RWN	O	B3	Memory and Peripheral Bus

Table 24 RC32438 Alphabetical Signal List (Part 8 of 9)

RC32438 Package Drawing — 416-pin BGA