OCTEON II CN66XX Known Issues

Version: 1.5
Last modified: 31 May 2012

1 Introduction

The following list documents the known issues with OCTEON II CN66XX documentation and silicon.

All issues in this document apply to all members of the CN66XX family unless stated otherwise. The term "OCTEON II" in this document refers to those parts.

A number of these issues have software workarounds that are implemented in the software development kit (SDK). Please check the SDK Release Notes for more information.

1.1 Applicability and History

Id	Section	Last Updated	Block	Summary	Revisions Affected
14874	2.1.2.1	1.0	USB	USB REXT Resistor Value	HRM < 0.8E
15933	2.2.1.1	1.0	Core	WatchHi0[I] During Deferred Load or Store Watch	1.0+
15940	2.2.2.1	1.0	CIU	CIU_SOFT_PRST2 / CIU_SOFT_PRST3	1.0+
15954	2.2.3.1	1.0	SLI	SLI Relaxed Ordering	1.0+
15976	2.2.4.1	1.0	GMX	GMX_DROP Interrupt	< 1.2
15982	2.2.1.2	1.0	Core	Execution From Low XKSEG and XSSEG	< 1.2
15999	2.2.1.3	1.0	Core	KUSEG and CVMSEG in Debug Mode	< 1.2
16018	2.1.3.1	1.0	HRM	SRIO Maintenance Error Status Register Documentation	HRM 0.92E
16018	2.2.5.1	1.0	SRIO	SRIO Maintenance Error Status Register Behavior	< 1.2
16057	2.2.1.4	1.4	Core	TLB Parity Errors	< 1.2
16061	2.2.6.1	1.0	PEM	PCI Express Completion Tags	< 1.2
16068	2.2.5.2	1.0	SRIO	SRIO MAC Status Bits	< 1.2
16074	2.2.5.3	1.0	SRIO	SRIO Multi-Lane Alignment	< 1.2
16075	2.2.5.4	1.0	SRIO	SRIO Maintenance Register Lane Status Bits	< 1.2
16079	2.2.1.4	1.0	Core	CVMSEG LM Stores with IOBDMA Activity	< 1.2
16082	2.2.2.2	1.0	MIO	MIO_RST_CNTL Read Value	< 1.2
16094	2.2.6.2	1.0	QLM	PCI Express Equalizer Settings	< 1.2
16111	2.2.5.5	1.1	SRIO	SRIO Switching Control	< 1.2
16149	2.2.5.6	1.1	SRIO	SRIO Switched Packets	< 1.2
16174	2.2.6.3	1.1	QLM	PCI Express Idle Threshold	< 1.2
16259	2.2.7.1	1.2	UART	UART0_DTR_N Operation	< 1.2
16347	2.2.8.1	1.2	GPIO	GPIO_CLK_QLM0[QLM_SEL] Read	1.0+
16357	2.2.4.2	1.2	XAUI	XAUI External Loopback	1.0+
16371	2.2.1.6	1.3	Core	Outstanding Prefetch to L1 Affects Load Order	1.0+
16419	2.2.1.7	1.3	Core	ERET/DERET after Mispredicted Branch	1.0+
16490	2.1.1.1	1.3	LMC	DIMM Parity Handling	HRM
16693	2.2.4.1	1.5	ZIP	Compression of Certain Data	1.0+

The Id column is the Cavium identifier for the issue.

The Section column identifies and links to which section of this document describes the issue.

The Last Updated column identifies the version of this document in which the issue was most recently revised.

The Summary column provides a summary of the issue.

The Revisions Affected column describes the HRM or silicon revisions that display the issue. For forward-looking information about open-ended items, contact your Cavium sales representative for details.

1.2 Identifying Part Revisions

The revision number for CN66XX-family parts are marked on the case as part of the date code. The information is also available from the PrID register, and from the PCI Express device revision ID.

1.2.1 Date Code Markings

The date code shown is indicating a pass 1.2 part. Pass 1.0 parts have a date code with no letter prefix, similar to "1108-G".

1.2.2 Processor ID

Processor revisions may be identified through software in various ways. The PrID coprocessor 0 register is one source; another is the PCI Express configuration header RID field, `PCIEEPi_CFG002[RID]`.

Chip Revision	COP0 PrID	PCIE RID
1.0	0x000D9200	0x0
1.2	0x000D9202	0x2

2 Open Issues List

2.1 CN66XX Hardware Reference Manual

2.1.1 DRAM Controller (LMC) Chapter

2.1.1.1 DIMM Parity Handling 16490

The JEDEC DDR3 DIMM pinout has a pin for address and control parity, named PAR_IN. OCTEON does not directly support address parity, and instead will change DDR_WE_L during the register initialization sequence to ensure the parity is correct. Only registered DIMMs use this pin, as it does not appear in the JEDEC DDR3 chip pinout.

The PAR_IN input of a registered DIMM must be terminated; either pulled high or pulled low. The LMC must be notified of which termination was used by programming `LMC0_DIMM_CTL[PARITY]`. If PAR_IN is pulled high, `LMC0_DIMM_CTL[PARITY]` should be set to 1, if PAR_IN is pulled low, the register should be set to 0.

2.1.1.1.1 Workaround

Be aware of the proper handling of PAR_IN on registered DIMMs, and configure `LMC0_DIMM_CTL[PARITY]` accordingly so the correct initialization sequence can be used.

2.1.2 USB Chapter

2.1.2.1 USB REXT Resistor Value 14874

Early revisions of the CN66XX Hardware Reference Manual specified an incorrect value for the USB REXT resistor. The correct value is 37.4 Ω.

Additionally, Cavium recommends configuring `UCTLX_UPHY_PORTX_CTL_STATUS[TXREFTUNE] = 15` for optimal USB eye quality.

2.1.2.1.1 Workaround

Populate the USB REXT resistor with the proper value. The version of u-boot supplied with the OCTEON SDK 2.0 and later already configures `TXREFTUNE` appropriately.

2.1.3 Serial Rapid I/O Chapter

2.1.3.1 SRIO Maintenance Error Status Register Documentation 16018

The CN63XX Hardware Reference Manual version 0.92 shows the `SRIOMAINT`*`i`*`_PORT_0_ERR_STAT` fields `PT_ERR`, `PT_WRITE`, `I_ERROR`, `O_ERROR`, `O_RETRY`, `O_DGRAD`, `O_FAIL`, and `PKT_DROP` as R/W. While this is correct for CN66XX parts earlier than pass 1.2 (see also section 2.2.5.1 of this document), it is not correct for pass 1.2 and later parts. These fields are R/W1C in pass 1.2 and later.

2.1.3.1.1 Workaround

Be aware of the difference between chip versions. Write a 0 to these fields to clear their associated state on pass 1.0 parts, write a 1 to clear the state on pass 1.2 and later.

2.2 Silicon

2.2.1 cnMIPS II Cores

2.2.1.1 WatchHi0[I] During Deferred Load or Store Watch 15933

As part of software debugging support, the cnMIPS II core implements two watchpoints, one each for instructions and data, and they are configured via the `WatchLo`*`n`* and `WatchHi`*`n`* registers, where *n* is 0 for the instruction watchpoint and 1 for the data watchpoint.

When a deferred load or store matches the data watchpoint (and `Cause[WP]` is set as a result), it is possible for `WatchHi0[I]` (the instruction watchpoint match bit) to be set.

2.2.1.1.1 Workaround

If `WatchHi1[R]` or `WatchHi1[W]` are set, ignore and clear `WatchHi0[I]`.

2.2.1.2 Execution From Low XKSEG and XSSEG 15982

The MIPS64 virtual memory map is divided up into various segments to allow different protection when the processor is in the various modes (kernel, supervisor, or user).

Part of the XKSEG range, from address `0xC000 0000 0000 0000` to `0xC001 FFFF 7FFF FFFF` and all of the XSSEG address range, from `0x4000 0000 0000 0000` to `0x4001 FFFF FFFF FFFF`, can not be used for instruction space. Instruction fetches will stall and the core will enter a low-power state until the next interrupt or other exception arrives to pull the program counter out of the affected address range.

The affected address space is not commonly used by operating systems.

2.2.1.2.1 Workaround

Do not execute instructions in the lower XKSEG or XSSEG virtual address range.

2.2.1.3 KUSEG and CVMSEG in Debug Mode 15999

The MIPS architecture specifies that instructions executing in Debug Mode (CP0 `Debug[DM]` is set) should have full access to all resources that are available in Kernel Mode operation.

However, when in Debug Mode, and CP0 `Status[ERL]` is set, the KUSEG virtual address range will be mapped, when it should be unmapped. Additionally, CVMSEG is not addressable, but should be.

2.2.1.3.1 Workaround

Switch to Kernel Mode by clearing CP0 `Status[KSU]` before accessing the affected addresses from debug mode.

2.2.1.4 TLB Parity Errors 16057

The TLB memory is protected by parity to potentially detect the occurrence of single-bit errors in the memory array.

Under certain circumstances, it is possible for a TLB instruction to cause a TLB parity error to fire improperly. The circumstances relate to the timing of a prefetch or demand L1 instruction cache fill after a `TLBW`, `TLBWI`, `TLBWR`, or `TLBR` operation.

The TLB will report a spurious machine check exception, indicating a TLB parity error, and the `EPC` register will be corrupted, rendering the system non-recoverable.

This issue is fixed in pass 1.2 and later parts.

2.2.1.4.1 Workaround

Set `CvmCtl[IPREF]` to 1 on each core.

2.2.1.5 CVMSEG LM Stores with IOBDMA Activity 16079

There is a small region of memory local to each core known as the CVMSEG LM. This region is commonly used for scratchpad space, and for IOBDMA operations. The region is carved out of the L1 data cache by the `CvmMemCtl[LMEMSZ]` register.

If there is a pending IOBDMA transaction whose destination lies in the same cache line (128 byte region with 128-byte alignment) as a core-issued store instruction, it is possible that the core store may be lost. No indication that the store has been lost will be given.

This issue is fixed in pass 1.2 and later parts.

2.2.1.5.1 Workaround

Avoid the situation. There are several ways to do this.

One way is to arrange usage of the CVMSEG LM so that all IOBDMAs are on a different cache line than any core stores.

Another option, if core stores are necessary to the same line as contains an IOBDMA target, is to execute a SYNC, SYNCS, or SYNCIOBDMA instruction before issuing the potentially-conflicting store. This could have some performance impact, depending on the application usage of IOBDMA.

2.2.1.6 Outstanding Prefetch to L1 Affects Load Order 16371

The `PREF` instructions allow software to give an indication to the memory system that a particular piece of memory will be accessed in the near future. Prefetches can be made into either of the L1 caches or into the L2 cache.

There is a short-duration race condition that could cause loads to complete out of order when a prefetch to the L1 cache is underway.

2.2.1.6.1 Workaround

Insert a `SYNC` or `SYNCS` instruction between loads that have a data dependency. This will delay the second load until the prefetch has completed and avoid the race condition.

Alternately, do not issue a PREF instruction that fetches to L1, i.e. with hint values of 0-4 or 6-24. Affected prefetches may be converted into a prefetch to L2, PREF 28 with a negligible (0-2%) performance impact.

The toolchain included SDK versions 2.3 and later employ the workaround of replacing the affected prefetch instructions. Please see the SDK release notes for details.

2.2.1.7 ERET/DERET after Mispredicted Branch 16419

The cnMIPS II architecture has branch prediction logic, and does speculative execution of the instruction after jump and branch instructions until the prediction has been verified.

If an ERET or DERET instruction is encountered during speculative execution, it is possible for address exceptions to be reported where no address exception occurred.

The J and BEQ $0, $0 instructions will never mispredict, and are immune to this.

2.2.1.7.1 Workaround

Ensure that there is no ERET or DERET instruction in the four cycles following the issue of the branch delay slot. Note that this applies to both the branch target as well as the fall through code. No branch or jump target should contain an ERET or DERET within 8 instructions from the entry point, nor should there be an ERET or DERET instruction within 9 instructions of a branch.

2.2.2 Central Interrupt Unit (CIU)

2.2.2.1 CIU_SOFT_PRST2 / CIU_SOFT_PRST3 15940

The CIU_SOFT_PRST2 and CIU_SOFT_PRST3 registers are used to reset sRIO controllers 2 and 3.

Reading back these registers does not work, and will return unpredictable data.

Additionally, note that when QLM0 is in sRIO x1 or x2 mode, QLM0 and all four sRIO controllers will come out of reset immediately upon chip reset deassertion. The unused controllers should be placed back into reset.

2.2.2.1.1 Workaround

Do not read the CIU_SOFT_PRST2 or CIU_SOFT_PRST3 registers.

When operating in sRIO x1 mode, set CIU_SOFT_PRST1[SOFT_PRST], CIU_SOFT_PRST2[SOFT_PRST] and CIU_SOFT_PRST3[SOFT_PRST] early in the boot sequence.

When operating in sRIO x2 mode, set CIU_SOFT_PRST2[SOFT_PRST] and CIU_SOFT_PRST3[SOFT_PRST] early in the boot sequence.

2.2.2.2 MIO Reset Control Reads 16082

The MIO_RST_CNTL*p*[PRST_LINK] field controls whether a controller link-down event places the corresponding MAC (for PCIE or SRIO) into reset.

This register field does not read back correctly – it will always read as zero. Writes will have the intended effect on the chip behavior.

This issue is fixed in pass 1.2 and later parts.

2.2.2.2.1 Workaround

Be aware of the limitation.

2.2.3 DMA Packet Interface / Switch Logic Interface (DPI/SLI)

2.2.3.1 SLI Relaxed Ordering 15954

The SLI_CTL_PORT*i*[CTLP_RO] and SLI_CTL_PORT*i*[PTLP_RO] fields control whether relaxed ordering is enabled for completion and posted TLPs (respectively). When set to 1, relaxed ordering is enabled for the specific subtype.

However, individually setting these bits to zero does not correctly disable relaxed ordering for a port.

2.2.3.1.1 Workaround

To disable read relaxed ordering, choose either one of these options:

(a) If the port is in PCI Express mode: Clear all SLI_MEM_ACCESS_SUBID*[RTYPE<0>] used on the PCIe port, disabling PCIe relaxed-order reads on the port. If port *p* is in sRIO mode, clear SRIO*p*_MEM_OP_CTRL[RR_RO].

(b) Set all SLI_CTL_PORT*[CTLP_RO] to the same value for all 4 MACs, whether the MAC is used or not.

To disable write relaxed ordering control, either:

(a) Clear SLI_CTL_PORT*i*[WAIT_COM] so that no inbound stores wait for a commit on port *i*.

(b) Set all SLI_CTL_PORT*[PTLP_RO] to the same value for all 4 MACs, whether the MAC is used or not.

2.2.4 Compression / Decompression Unit (ZIP)

2.2.4.1 Compression of Certain Data 16693

The ZIP unit can compress data with a varying degree of tradeoff of speed for compression ratio.

In very rare cases, it is possible for a compression job to hang a ZIP engine when IWORD0[SS] is 0, 1, or 2. With the same input and the same IWORD0[SS] value, the ZIP engine will consistently hang given the same input.

When the hang occurs, the ZIP instruction and the engine that processed the ZIP instruction are stuck, and will not proceed until ZIP is reset. Input queues that use the other engine will continue processing ZIP instructions.

2.2.4.1.1 Workaround

None needed if ZIP compression is not used.

Otherwise, set IWORD0[SS] = 3. This will prevent the hang condition from occurring, but there will be a small reduction in compression ratio. The reduction in ratio is dependent upon the uncompressed data but is estimated at approximately 10% compared to the IWORD0[SS] = 0 setting.

2.2.5 Gigabit Ethernet MAC (GMX)

2.2.5.1 GMX Drop Interrupt 15976

The CIU_INT*_SUM*[GMX_DRP] interrupt is intended to fire whenever GMX drops packet data.

However, it does not trigger in all possible circumstances, depending on the precise coprocessor clock frequency and the process variation.

2.2.5.1.1 Workaround

Do not depend on CIU_INT*_SUM*[GMX_DRP] to debug GMX packet drops. They can be detected via either the GMX*i*_RX*p*_STATS_PKTS_DRP register (for packets completely dropped by GMX due to FIFO overflows), or by the WQE error indication of WORD2[RE] set, that have WORD2[Opcode] = 1.

2.2.5.2 XAUI External Loopback 16357

The GMX*i*_XAUI_EXT_LOOPBACK_EN[EN] register enables XAUI external loopback.

When a packet with a size divisible by 4 arrives for loopback, it will be corrupted upon transmission.

2.2.5.2.1 Workaround

Use different sized packets, or do the loopback via software.

2.2.6 Serial Rapid I/O MAC (SRIO)

2.2.6.1 SRIO Maintenance Error Status Register Behavior 16018

There are various SRIO port error and status information bits available from the SRIOMAINT*i*_PORT_0_ERR_STAT register. Some of the bits are latching versions of other bits. The

latching bits are PKT_DROP, O_FAIL, O_DGRAD, O_RETRY, O_ERROR, I_ERROR, PT_WRITE, and PT_ERROR. Each of the latching bits is supposed to be write-1-to-clear.

However, these fields are instead read/write, meaning a zero must be written to clear them.

This issue is fixed in pass 1.2 and later parts.

2.2.6.1.1 Workaround

For pass 1.2 and later parts, write a 1 in the latched status bits to reset the latch. For earlier parts, write a 0.

2.2.6.2 SRIO MAC Status Bits 16068

The sRIO MACs provide a status register, SRIO*i*_STATUS_REG[SRIO], to allow software to determine whether the SRIO port is enabled.

This status for SRIO interfaces 2 and 3 (i.e. the status reported by SRIO2_STATUS_REG[SRIO] and SRIO3_STATUS_REG[SRIO]) is not correctly reported.

This issue is fixed in pass 1.2 and later parts.

2.2.6.2.1 Workaround

Use the MIO_QLM*i*_CFG[QLM_CFG] register to determine which SRIO MACs are available.

2.2.6.3 SRIO Multi-Lane Alignment 16074

The SRIO specification has accommodations made for rate matching, when one end of a link is faster than the other. This is handled through the clock compensation sequence, which the receiver should use to re-align.

When there are multiple lanes in a link, and the link partner is transmitting faster than OCTEON II's transmit rate, the multi-lane alignment algorithm will lose synchronization. Received data will overflow a FIFO, causing the link to come down. The link may come up in 4 lane mode, but once data is flowing, will not stay up. The link will always function correctly in single-lane mode.

If the remote device is transmitting at the same or slower rate than OCTEON II, there will be no issue.

This issue is fixed in pass 1.2 and later parts.

2.2.6.3.1 Workaround

If possible, use the same reference clock source for OCTEON II and the sRIO link partner.

Alternately, ensure that the link partner has a slower reference clock than the OCTEON II, either by slowing the reference clock to the link partner slightly or increasing the reference clock to the OCTEON II slightly.

Finally, one-lane mode can be forced by either SRIOMAINT*i*_PORT_0_CTL[OV_WIDTH] or CIU_QLM0[LANE_EN] bits.

2.2.6.4 SRIO Maintenance Register Lane Status Bits 16075

The SRIOMAINT*i*_LANE*l*_STATUS_0[XSYNC] and [XTRAIN] fields provide information about whether the sRIO lane has undergone a lane sync or training status change since the last time the field was read.

However, the clear-after-read logic does not clear the read lane. Reading lane 0 status will clear the status for lane 3. Reading lane 1 will clear lane 2, lane 2 will clear lane 1, and lane 3 reads will clear lane 0.

This issue is fixed in pass 1.2 and later parts.

2.2.6.4.1 Workaround

Be aware of the unexpected clearing pattern. Note that it is not possible to read the status for all four lanes.

2.2.6.5 SRIO Switching Control 16111

The CN66XX parts can forward SRIO packets between SRIO interfaces, somewhat like an SRIO switch does.

When an interface is disabled by setting SRIOMANT*i*_PORT_0_CTL[O_ENABLE] to zero, packets may still be switched to that port and transmitted.

2.2.6.5.1 Workaround

Be aware that packets may still be transmitted on disabled SRIO ports.

This issue is fixed in pass 1.2 and later parts.

2.2.6.6 SRIO Switched Packets 16149

The CN66XX parts can forward SRIO packets between SRIO interfaces, somewhat like an SRIO switch does.

Packets being switched can incorrectly contain two extra bytes. This can occur when the received packet has both CRC-16 and two pad zero bytes plus one or more control symbols following the packet data and preceding the EOP delimiter. The CRC-16 from the received packet will be included in the transmitted packet, with an additional CRC-16 that includes the erroneous 2 bytes.

2.2.6.6.1 Workaround

None. Do not use the SRIO switching functionality, or be aware of the potential for extra bytes in certain circumstances.

This issue is fixed in pass 1.2 and later parts.

2.2.7 PCI Express MAC (PCIE)

2.2.7.1 PCI Express Completion Tags 16061

Each memory read (MRd) request in PCIe has a transaction ID that consists of the requestor ID (bus number / device number / function number) and a tag controlled by the requesting device. The transaction ID is used to differentiate responses when there are multiple reads outstanding from an initiator. A PCI Express multi-function device is one that has multiple internal peripherals, all operating independently.

When requests from multiple initiators are outstanding to a single OCTEON II PEM, all with the same tag value, only the last request will receive a response. The other responses will not be sent. The other initiators should then experience a completion time out, and should send an ERR_NONFATAL or ERR_FATAL message to the root complex.

Multiple initiators must be connected to one PEM, possibly via a PCIE bridge. Note that a single multi-function device may still encounter this issue, as there are multiple initiators within the device that may use the same tag value.

Note that if both PEMs are in use, and each is connected to only one other single-function device, this issue can not arise.

The most likely conditions for this to occur are when a single PEM in root complex mode is connected to a bridge, and multiple DMA-capable devices are beyond the bridge accessing OCTEON II memory.

However, this is not limited to root complex mode – if OCTEON II is an endpoint, and there are multiple devices in the system that can initiate MRd requests from OCTEON, the issue may still arise.

This issue is fixed in pass 1.2 and later parts.

2.2.7.1.1 Workaround

Avoid system configurations where multiple PCIe devices capable of issuing memory read requests are connected to a single PEM.

2.2.7.2 PCI Express Equalizer Settings 16094

To improve performance over marginal channels, OCTEON II has receive equalization in the SERDES path.

The default equalization settings for PCIe Gen2 are somewhat too strong, and may compromise timing performance in systems with shorter trace lengths and low loss between the PCIe partners.

This issue is fixed in pass 1.2 and later parts.

2.2.7.2.1 Workaround

Change the PCIe receive equalization settings to RxCap = 0x1 and RxEq = 0x8. Example code is provided in SDK 2.1 to do this.

2.2.7.3 PCIE Idle Threshold 16174

The PCI Express specification version 2.0 defines a receive idle threshold $V_{RX\text{-}IDLE\text{-}DET\text{-}DIFFp\text{-}p}$, that determines the electrical level at which a receiver should detect electrical idle. Additionally, the PCI Express specification defines the receiver sensitivity, $V_{RX\text{-}EYE}$, which determines the minimum amplitude of the received eye.

The idle threshold for OCTEON II is 175 mV. However, some environments may not be able to reach that level.

2.2.7.3.1 Workaround

Change the value for the idle detector DAC (`idle_dac`) from 4 to 2. Code is included in SDK 2.1 to demonstrate this.

This issue is fixed in pass 1.2 and later parts.

2.2.8 UART Interface

2.2.8.1 UART0_DTR_N Operation 16259

The UART0_DTR_N pin is a modem control pin for UART 0. It is shared with the SPI_DO pin when the MPI/SPI controller is enabled, controlled via the `MPI_CFG` register.

When the SPI controller is not enabled, UART0_DTR_N is not driven.

This issue is fixed in pass 1.2 and later parts.

2.2.8.1.1 Workaround

Be aware of the limitation on functionality in UART mode.

2.2.9 General-Purpose Input/Outputs (GPIO)

2.2.9.1 GPIO_CLK_QLM0[QLM_SEL] Read 16347

The GPIO pins can be configured to transmit a divided version of the clock recovered from a QLM lane via the `GPIO_CLK_QLM0` register and the `GPIO_BIT_CFG`*b*`[SYNCE_SEL]` register.

Reading back the value written to the QLM_SEL field in GPIO_CLK_QLM0 does not work properly, and returns the value programmed to `GPIO_CLK_QLM1[QLM_SEL]`.

2.2.9.1.1 Workaround

If it is necessary to determine the value programmed to `GPIO_CLK_QLM0[QLM_SEL]`, it must be saved somewhere that software can read it back.