CN54XX/CN55XX/CN56XX/CN57XX Pass 2 Known Issues

Version: 1.4

Last modified: 25 May 2010

1 Introduction

The following list documents the known issues with OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX documentation and silicon pass 2. The change bars in this document indicate differences from version 1.3 of this document.

All issues in this document apply to all members of the CN54XX, CN55XX, CN56XX, and CN57XX families unless stated otherwise. The term “OCTEON Plus” in this document refers to those parts.

2 Open Issues List

2.1 CN54XX/CN55XX/CN56/57XX Hardware Reference Manual

2.1.1 cnMIPS™ Cores Chapter

2.1.1.1 CvmMemCtl[CVMSEGENAK] Enable Requires L1 DCache Invalidate

CVMSEG is an OCTEON feature that provides core-local scratchpad memory and space to issue IOBDMA instructions. CVMSEG is located at virtual addresses `0xFFFFFFFFFFFF8000` through `0xFFFFFFFFFFFFBFFF`.

Previous revisions of the Hardware Reference Manuals did not completely describe the steps necessary when enabling CVMSEG. It is required that the L1 cache be invalidated if any CVMSEG address might appear in the cache when CVMSEG is first enabled.

Failure to invalidate the L1 cache may result in spurious Parity, Cache Error or Data Access exceptions.

Invalidate the L1 cache by issuing either an L1 DCache Virtual Tag Invalidate or an L1 DCache Invalidate, using the CACHE instruction described in section A.4 of the Hardware Reference Manual.

2.1.1.1.1 Workaround

Invalidate any L1 lines overlapping CVMSEG when enabling CVMSEG.

2.1.2 Electrical Specifications Chapter

2.1.2.1 CN56XX / CN57XX 800 MHz Core Voltage

In versions 2.4 and newer of the OCTEON Plus Hardware Reference Manuals, the voltage for the core power supply at 800 MHz has been changed to 1.225 V. The maximum current consumption has not changed.

2.1.2.1.1 Workaround

Be aware of the increased core voltage specification.

2.2 Silicon

2.2.1 SGMII Interfaces

No known issues.

2.2.2 DDR Controller

2.2.2.1 DDR2 PLL During Resets

The default setting of the `LMC_PLL_CTL[FASTEN_N]` field for OCTEON Plus is 0. This default value minimizes the time required for initialization of the DDR2 interface internal clocking mechanism, specifically, the time required to lock the PLL for the DDR2 interface into the targeted DDR2 clock frequency. This PLL locking only takes place during reset, including soft reset, of the OCTEON Plus.

When the LMC_PLL_CTL[FASTEN_N] field is configured as 0, occasionally (a few percent) designs that use CLKF values greater 80 have seen a DDR2 PLL lock-issue rate of less than 0.5% on a soft-reset. Upon repeated soft-resets during extended QA or burn-in, this issue occurs on approximately 7% of the units after 100 to 200 soft-resets and is corrected after a further soft-reset.

When the LMC_PLL_CTL[FASTEN_N] field is configured as 1, PLL locking for the DDR2 interface works robustly without failure. The time required to lock the PLL during DDR2 interface initialization is still below the Hardware Reference Manual-specified value of 500 × (LMC_PLL_CTL[CLKR] + 1) cycles, and thus it will have no impact whatsoever on any system aspect.

2.2.2.1.1 Workaround

Cavium Networks recommends that customers set the LMC_PLL_CTL[FASTEN_N] field to 1 for conservative and robust operation, if the system usage model involves many soft resets of the system.

Note that the value programmed into LMC_PLL_CTL[CLKR] does not need to be in the range 128 to 256 as implied by current versions of the Hardware Reference Manual.

2.2.2.2 Accesses to Nonexistent DDR Memory with ECC_ADDR

If software makes an access, either through a TLB mapping or through an xkphys access within the DDR address range to memory that is not connected, when LMC*n*_CTL1[ECC_ADDR] is set, unpredictable results will occur. Those results include single- and/or double-bit ECC errors being reported on reads, or data being corrupted on writes. On writes, the data that is corrupted will be an entire cache line in size, and will be at the physical address truncated to the memory connected to chip select 0. ECC errors may be reported during the write, if the L2C reads the line to combine with the written data.

Because of the inclusion of address information in the ECC word when LMC*n*_CTL1[ECC_ADDR] is set, the data read through the truncated address into the L2 cache will most likely be corrupted by the ECC logic attempting to correct the data.

Note that if LMC*n*_CTL1[ECC_ADDR] is not set, no ECC errors will be reported, reads will return data from the truncated address, and writes will overwrite data at the truncated address.

The DDR physical address ranges, expressed in xkphys addresses, are:

```
0x8000 0000 0000 0000 to 0x8000 0000 0FFF FFFF
0x8000 0004 1000 0000 to 0x8000 0004 1FFF FFFF
0x8000 0000 2000 0000 to 0x8000 0003 FFFF FFFF
```

Note that unless 16 GB of memory is populated, not all of these address will be valid.

2.2.2.2.1 Workaround

Ensure that software does not make any accesses to non-existent DDR addresses, either through xkphys for supervisor mode, or through user-mode mappings.

2.2.3 JTAG and EJTAG

2.2.3.1 EJTAG PC Sampling

The MIPS EJTAG specification has a feature known as “PC sampling”, where the program counter of a core is sampled periodically.

This sampling is not reliable under most circumstances. It functions for a short period of time regardless of the requested sample rate, and then the new bit will no longer be set and the value will no longer be updated.

2.2.3.1.1 Workaround

None. Do not use the PCSAMPLE feature of EJTAG if reliable results are expected.

2.2.4 SERDES Interfaces

No known issues.

2.2.5 PCI Express Interface

2.2.5.1 PCIe Memory Space Writes in Swap Mode 3

The PCI Express standard has rules with regard to first and last data word byte enables. Those rules basically require that the enables need to be contiguous, except under some very specific circumstances.

When the OCTEON Plus DMA engines are generating PCIe memory space writes in swap mode 3, non-contiguous enables are generated in some cases where they are not allowed by the PCIe spec. There are no problems with swap modes 0, 1, and 2. To create the condition from the DMA engines, they must be making an OUTBOUND or EXTERNAL-ONLY transfer.

This issue is not present with PCIe transactions generated via IOBDMA or conventional loads and stores.

2.2.5.1.1 Workaround

If it is possible to avoid using swap mode 3, do so.

Alternatively, the problem will not occur if all PCIe writes are a multiple of 8 bytes long and start on an 8-byte-aligned address.

Software can break up the writes submitted to the DMA engines at the beginning and end of the transfer so that they can use 8B transfers for the bulk of the data and use swap mode 3, and then use another swap mode for the few starting and ending bytes. Each PCIe DMA instruction can have its own independent swap mode.

2.2.5.2 Per-Ring Input Backpressure

OCTEON Plus PCIe packet input supports both per-port and per-ring backpressure. There are 4 ports and 8 rings per port providing a total of 32 rings (the ring number is denoted with an italic *r* in the register names below). This backpressure can prevent the PCIe packet input logic from reading in too many packets. This issue affects only per-ring backpressure.

When software writes the NPEI_PKT*r*_IN_BP[CNT] fields, OCTEON Plus is supposed to read the old value of the count field, subtract the written value, and then write NPEI_PKT*r*_IN_BP[CNT] with the result.

This update only happens correctly when both of the following are true:

- The write to NPEI_PKT*r*_IN_BP[CNT] is immediately preceded by a read to the same NPEI_PKT*r*_IN_BP_CSR, and
- Without any intervening CSR accesses to any of the following CSRs:
  NPEI_PKT*r*_SLIST_BADDR, NPEI_PKT*r*_SLIST_BAOFF_DBELL,
  NPEI_PKT*r*_SLIST_FIFO_RSIZE, NPEI_PKT_IN_DONE*r*_CNTS, NPEI_PKT*r*_CNTS,
  NPEI_PKT*r*_INSTR_BADDR, NPEI_PKT*r*_INSTR_BAOFF_DBELL,
  NPEI_PKT*r*_INSTR_FIFO_RSIZE, NPEI_PKT*r*_INSTR_HEADER, NPEI_PKT*r*_IN_BP

2.2.5.2.1 Workaround

One choice is to use only per-port backpressure, instead of per-ring backpressure (that is, clear NPEI_CTL_STATUS[PKT_BP]).

Alternatively, precede all writes to the NPEI_PKT*r*_IN_BP[CNT] field with a read and make no intervening CSR accesses to the listed CSRs, above. This may require using software locks if multiple threads or cores may make writes to any of these registers.

2.2.5.3 Mixing Direct and Indirect PCIe Gather Packets

PCIe instructions send packets into OCTEON Plus for processing. PCIe instructions may be any of Direct Data, Direct Gather, or Indirect Gather.

It is possible for the packet input logic to hang when fetching both Direct Gather and Indirect Gather type instructions.

2.2.5.3.1 Workaround

Use exclusively Direct Gather or Indirect Gather instruction types. Do not submit both types of instructions.

2.2.5.4 PCIe DMA Request Count Register

The `NPEI_DMA_PCIE_REQ_NUM` register contains a number of fields, `PKT_CNT`, `DMA0_CNT`, `DMA1_CNT`, `DMA2_CNT`, `DMA3_CNT`, and `DMA4_CNT` that are intended to control the number of requests the packet I/O engine or a particular DMA engine (respectively) can have outstanding. The `DMA_CNT` field is the per-interface maximum for outstanding requests among all DMA engines and the packet I/O engine.

In CN54XX, CN55XX, CN56XX, and CN57XX pass 2.0, the counters underlying the individual engine fields can become corrupted. This can cause the DMA engine or packet I/O engine to stall and not make further progress. The overall limit, as set in `NPEI_DMA_PCIE_REQ_NUM[DMA_CNT]` works correctly as described.

In pass 2.1, the `PKT_CNT`, `DMA0_CNT`, `DMA1_CNT`, `DMA2_CNT`, `DMA3_CNT` and `DMA4_CNT` fields have no effect.

2.2.5.4.1 Workaround

On pass 2.0, allow only a single outstanding request per interface among all of the units. This can be accomplished by setting `NPEI_DMA_PCIE_REQ_NUM[DMA_CNT]` to 1.

On pass 2.1, be aware that the `PKT_CNT`, `DMA0_CNT`, `DMA1_CNT`, `DMA2_CNT`, `DMA3_CNT` and `DMA4_CNT` fields in this register do not have any effect on the maximum I/Os that may be outstanding.

2.2.5.5 Multiple PCIe Port Usage

OCTEON Plus provides two PCIe ports (controllers), and also provides DMA engines and a packet I/O engine that can transfer data on either PCIe port. The DMA engines generate read requests on the PCIe ports in response to an INBOUND or EXTERNAL-ONLY transfer. The packet I/O engine generates read requests in response to a doorbell write, as described in section 9.4, “PCIe Instruction Input from an External Host” of the Hardware Reference Manual.

When both ports are enabled, and both the DMA engines and the packet I/O engine are simultaneously making read requests, it is possible for those requests to go out the wrong port, or other internal state corruption to happen inside OCTEON Plus.

This is not an issue when only one PCIe port is in use. It is also not an issue when only the DMA engines or the packet I/O engine is active. Also, this is not an issue if the only traffic being generated by either the DMA engines or the packet engine is write activity, for example a DMA engine executing an OUTBOUND transfer, or a packet sent to the PCIe host.

Finally, this is not an issue for traffic generated by core loads, stores, and IOBDMA transactions. Core transactions will not affect the issue, nor does this issue affect them.

2.2.5.5.1 Workaround

No workaround is required if the system does not have both PCIe ports enabled, and both DMA and packet I/O transactions running.

Use only DMA or packet I/O, but not both.

If both DMA and packet I/O is required, DMA INBOUND and EXTERNAL-ONLY instructions must go to the same port that packet I/O is being used on. Note that DMA OUTBOUND instructions may be used on either port at any time.

2.2.5.6 Packet Input or DMA 8-Byte Read

When completions for two outstanding PCIe packet input/output or DMA reads to the same PCIe port return near in time, and the second read is 8 bytes or smaller and returned in order, it is possible for a hang or data corruption to occur when OCTEON issues a third PCIe read to the same port from the same engine soon thereafter. The timing is particular, and the case is uncommon.

2.2.5.6.1 Workaround

None needed if packet input is not used and if INBOUND and EXTERNAL-ONLY DMAs are not used.

PCIe packet output alone can not generate the condition because it can not generate reads that are 8 bytes or less.

Three outstanding reads are necessary, so one workaround is to limit packet input/output and DMA engines to two outstanding reads by selecting a value for `NPEI_DMA_PCIE_REQ_NUM[DMA_CNT]` less than the default value of 16. If only one of the two PCIe ports is active, `NPEI_DMA_PCIE_REQ_NUM[DMA_CNT] = 2` will work. If both ports are used for packet input/output, then DMA_CNT may need to be set to one because the engine associated with PCIe packet input/output could have up to two outstanding reads per port when `NPEI_DMA_PCIE_REQ_NUM[DMA_CNT] = 2`.

Note that limiting the number of outstanding reads may limit performance of all of the PCIe DMA engines and packet I/O. Also note that the individual engines can not be controlled due to the issue described in section 2.2.5.4.

Another workaround is to limit all DMA engine or packet pointers to a 16-byte boundary, and ensure that the length of each transfer modulo 16 is 0 or greater than 8.

2.2.5.7 PCIe DMA Doorbell Ordering

Each of the PCIe DMA engines has a doorbell register (`NPEI_DMA`*n*`_DBELL[DBELL]`) that instructs the DMA engine to fetch instructions from the instruction queue.

If the doorbell register is incremented by a value that is less than the size of the next instruction, the DMA engine may hang. For example, a single core could cause this by generating two doorbell writes for a single DMA instruction.

Alternatively, two cores could cause this if they are both submitting differently-sized instructions to the same DMA engine, but their doorbell register writes occur in the order opposite the instructions in the queue. To amplify, core A atomically adds a 5-word instruction to the queue, then core B atomically adds an 8-word instruction to the queue, then core B writes 8 to the doorbell register followed by core A writing 5 to the doorbell register. While the final doorbell increment value (13) is correct, each doorbell register write did not cover a single instruction, and the DMA engine might hang.

2.2.5.7.1 Workaround

Software must ensure that the write to the queue and the write to the doorbell register for a particular DMA engine are completed atomically, by extending the critical section that protects the DMA instruction queue to also cover the doorbell write. Alternatively, ensure that all instructions for a particular DMA engine are the same size.

2.2.5.8 PCIe Packet Output with ISIZE = 0

The PCIe packet output logic supports both info-pointer and buffer-pointer-only modes. With info-pointer mode, `NPEI_PKT_SLIST_ID_SIZE[ISIZE]` indicates the number of bytes from the packet to write at the first info pointer, with the remaining packet bytes written at the buffer pointer(s).

PCIe packet output malfunctions when info-pointer mode is used with NPEI_PKT_SLIST_ID_SIZE[ISIZE] set to zero.

2.2.5.8.1 Workaround

Don't enable PCIe packet output with NPEI_PKT_SLIST_ID_SIZE[ISIZE] set to zero. Use a different PCIe packet output mode or another method to move the data.

2.2.5.9 TLP Ordering after Soft Reset

The PCI Express transaction layer defines certain requirements for when Transaction Layer Packets (TLPs) may or must not be reordered. OCTEON Plus implements these rules in accordance with section 2.4 of the PCI Express Base Specification, revision 1.1. Internally, the ordering for posted TLPs is managed by two eight-bit counters, one for input and one for output.

When only the PCIe link is reset, either through CIU_SOFT_PRST (when in root-complex mode) or through CIU_SOFT_RST (in end-point mode), one of those counters is cleared, but the other one is not. If the number of posted TLPs issued before the reset was not a multiple of 256, the two counters will no longer be synchronized, and TLP ordering may not be correctly preserved.

Note that if the PCIe link is reset as part of a complete chip reset, both counters will be cleared, so there will be no issue.

Which counter is cleared depends on the source of the reset and whether OCTEON Plus is operating as an endpoint or a root complex. A reset provoked by CIU_SOFT_PRST will reset only the input counter when OCTEON Plus is in endpoint mode, and a reset provoked by CIU_SOFT_RST will reset only the output counter.

The counters can be read out through NPEI_DBG_SELECT[DBG_SEL] and NPEI_DBG_DATA[DATA] to determine whether they are synchronized. Sending memory write transactions will increment both counters. If they are found to be out of synchronization, sending enough memory write transactions until the counter is zero and resetting the interface again will put the counters back in to synchronization.

The debug registers are as follows:

NPEI_DBG_SELECT[DBG_SEL]		**NPEI_DBG_DATA[DATA]**	
PCIe Port 0	**PCIe Port 1**	**Bit**	**Comments**
0xC80F	0xD00F	<0>	P_PRES
		<8:1>	OUT_P_COUNT – the output counter
0xCFFC	0xD7FC	<7:0>	IN_FIF_P_COUNT – the input counter minus one, valid only under certain circumstances (see below).

The following procedure is required to get a valid reading of the input counter:

1. Verify that there has been at least one PCIe memory store and that P_PRES is zero.
2. Set `x = (IN_FIF_P_COUNT + 1) & 0xFF`
3. Issue another PCIe memory store, wait for it to complete, and verify that P_PRES is zero.
4. Set `y = (IN_FIF_P_COUNT + 1) & 0xFF`
5. If `y != (x + 1) & 0xFF`, set `x = y`, and return to step 3.

After step 5, y is the input counter.

2.2.5.9.1 Workaround

When OCTEON Plus is the root complex, during the PCIe initialization code, verify that the input counter equals the output counter. If it does not, issue enough (`256 - y`, where y is as determined above) memory writes to make the input counter zero, check that P_PRES is zero, and then reset the interface again. Note

that when OCTEON Plus is the root complex, if CIU_SOFT_PRST is not toggled except during first initialization, then the problem will not arise.

When OCTEON Plus is the endpoint, before issuing any PCIe memory space stores, verify that the input counter equals the output counter. If it does not, issue enough (256 – OUT_P_COUNT) memory writes to make the output counter zero, and then strobe CIU_SOFT_RST again. When OCTEON Plus is in endpoint mode, systems that do not strobe CIU_SOFT_RST will not encounter any problems, and do not need further workarounds.

These workarounds will be included in the next release of the SDK, and a software patch will be available for supported SDKs; check the Cavium Networks Technical Support Site or your sales representative to acquire the patch for your SDK version.

2.2.5.10 PCIe Memory Space Store Combining

The internal PCIe switch logic attempts to merge sequential PCIe memory-space store operations before forwarding the combined operation to the appropriate PCIe controller. This will reduce the number of PCIe transactions required when writing to a large area of memory.

Under some very specific circumstances, it is possible for stores to be combined in circumstances where they should not be combined. One such case is when two writes arrive for the adjacent increasing addresses and SubDID, but with different SubDID extender (SE) values; they will be incorrectly combined. Another case is when two stores that should combine are issued with an intermediate load and store that should not be combined in between, and arrive at the combining unit with timing boundaries; the combining logic will incorrectly combine the requests.

2.2.5.10.1 Workaround

Disable PCIe store combining by setting NPEI_MEM_ACCESS_CTL[MAX_WORD] to 1.

2.2.5.11 PCIe Packet I/O TLP Generation

There are two types of PCIe transaction layer protocol packet headers: A 3 double-word one for use when the PCIe address of the transaction is within the bottom 4 GB of address space, and a 4 DW one for when the address of the transaction is above 4 GB. The PCIe specification requires that the 3 DW TLP header be used for all transaction in the lower 4 GB of memory.

When using the PCIe packet output feature of OCTEON Plus, if the destination address is below 0x0000000080000000, a 3 DW TLP will be generated. For all other addresses, including those between 0x0000000080000000 and 0x00000000FFFFFFFF, a 4 DW TLP will be generated.

Certain PCIe bridges will reject a 4DW TLP with the upper 32 bits of address set to all 0, flagging a malformed TLP error.

This only applies to writes generated from the PCIe packet output as described in section 9.5 of the CN54/5/6/7XX HRM. All other writes (DMA engines, IOBDMA, and core generated) are correct.

2.2.5.11.1 Workaround

None required if the system does not contain any bridges that flag 4 DW TLPs between 2 GB and 4 GB as an error.

Another possibility is to suppress reporting of the error for the bridges in the system, if possible.

Finally, avoid supplying addresses that are >= 2 GB and < 4 GB for the buffer or info pointers that are used during packet output.

2.2.5.12 PCIe Packet Output Blocking

During periods of congestion, backpressure is used in various places inside OCTEON and between OCTEON and external devices to reduce the admission rate of traffic to prevent overflows. One of those places is packet input from the PCIe interface. Because PCIe is a pull-mode interface (i.e. PIP/IPD must take

action to retrieve data), PIP/IPD will only stop processing input packets when the FPA runs out of packet and/or WQE buffers.

If PIP/IPD stops processing input packets on the PCI express interface, it is possible for packet output to be blocked on the same interface, potentially leading to a deadlock if the blocked packet output is needed to free up buffers.

2.2.5.12.1 Workaround

PIP/IPD usually only stops processing input packets when FPA runs out of packet and/or WQE buffers. The workaround is thus to prevent FPA from running out of these buffers when PCIe packet input is fetching instructions.

Note that a combination of these suggested workarounds may be needed. Multiple packet input sources share the PIP/IPD packet input path, and each may be affected by this. For example, you might do the first item below for GMX to prevent that source of exhaustion, and the second for PCIe.

One workaround is to set PIP/IPD to drop packets before buffers get too low.

Another workaround is to have the OCTEON cores monitor packet and/or WQE buffers, and write a location on the remote host when they are too low. The driver on the remote host would query this location and stop ringing any PCIe instruction input doorbells when told to. Eventually, when the OCTEON core notices increased packet and/or WQE buffers, the code would write the location on the remote host again, the driver would notice, and would again be allowed to ring doorbells.

2.2.6 Random Number Generator

No known issues.

2.2.7 General

No known issues.

2.2.8 Packet Input (PIP/IPD)

2.2.8.1 Readability of Certain PIP_PORT_CFG*n* Fields

The HRM specifies that `PIP_PORT_CFGn[HG_QOS]` and `PIP_PORT_CFGn[QOS_VSEL]` are supposed to be read/write, that is, they should return the value they were configured with.

However, these two fields of `PIP_PORT_CFGn` read as zero.

2.2.8.1.1 Workaround

Be aware that these bits when read back do not reflect the configured state.

2.2.9 XAUI Interface

No known issues.

2.2.10 Media Independent Interface

No known issues.

2.2.11 Central Interrupt Unit

No known issues.

2.2.12 Boot Bus

No known issues.

2.2.13 Universal Serial Bus (USB)

No known issues.

2.2.14 Universal Asynchronous Receiver / Transmitter (UART)

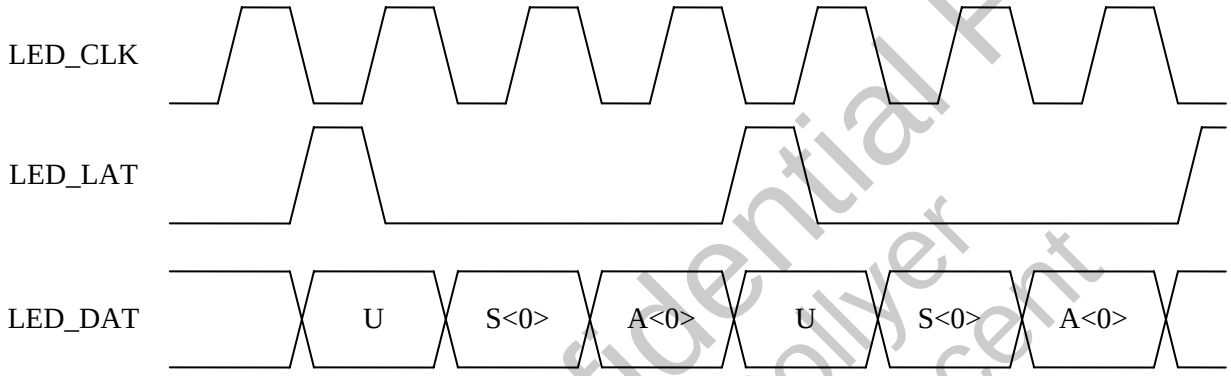
No known issues.

2.2.15 LED Interface

2.2.15.1 LED Traffic Blink Function

The LED interface provides a flexible output that, among many other features, can provide traffic activity information. The traffic activity display is configured through the `LED_PRT_STATUS`*n* registers. The blink rate is configured through the `LED_BLINK[RATE]` register.

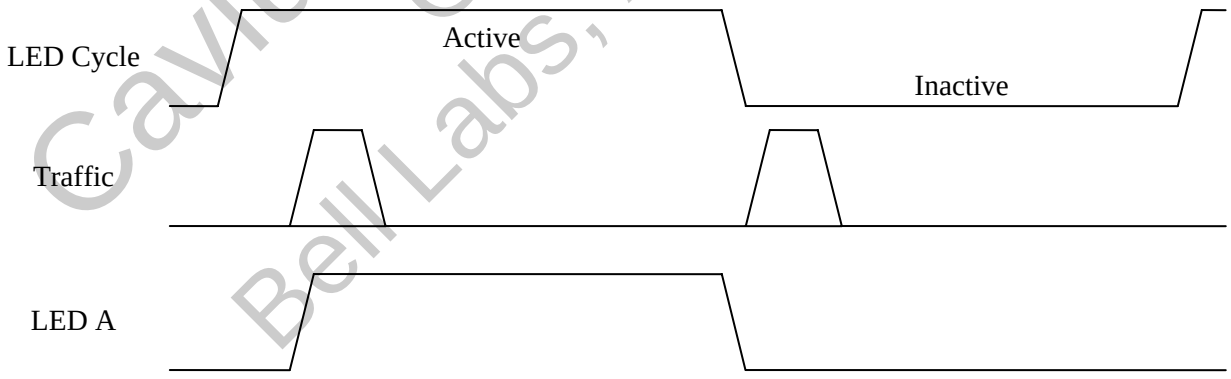
Consider the following diagram, which is the LED stream for a single-port configuration with `LED_PRT_FMT[FORMAT] = 4`, and no user data. S<0> is the port status for port 0, A<0> is the activity status for port 0. The data in the U cycle is indeterminate.



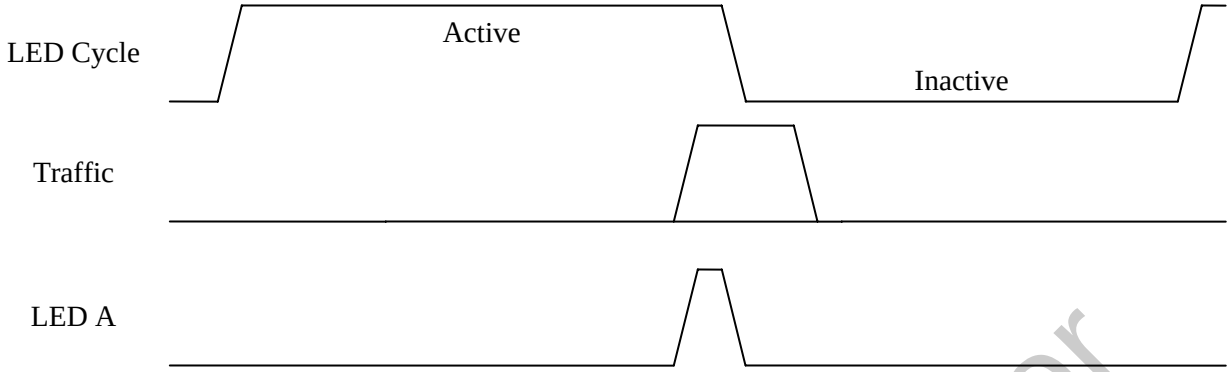
The `LED_BLINK[RATE]` period is defined as a number of LED_LAT pulses.

The LED blink cycle is split into two phases. During the active phase, any receive, transmit, or both types of traffic on the port in question (depending on configuration) will cause the activity LED to be turned on in the next LED_LAT cycle, and the LED will remain on until the inactive phase. During the inactive phase, the activity LED will be forced off.

The figure below shows the case where traffic arrives shortly after the LED cycle enters the active state and more traffic arrives shortly after the inactive state. The LED will be on for most of the active period, and then off.



If port activity occurs during the inactive portion of the blink cycle, no LED indication will be provided at all. Further, traffic very late in the active portion of the blink cycle, while it may indeed turn on the LED, the LED will probably not be on for long enough to be detected by the eye, as shown in the figure below.



2.2.15.1.1 Workaround

Be aware of the potential for low duty-cycle blink displays.

2.2.16 Free Pool Allocator Unit (FPA)

2.2.16.1 FPA Allocate when Internal Pool Low

The FPA maintains an internal cache of free pointers so that many allocation requests can be serviced without going to memory to read a page of pointers. It refills the internal cache as needed from memory.

If an allocation request is made when there are sufficient pointers in the overall pool (that is, `FPA_QUE`*n*`_PAGES_AVAILABLE[QUE_SIZ]` is greater than or equal to the number of requested pointers) but the internal cache does not have sufficient pointers, NULL will be returned for the allocation request, as though the pool was empty.

2.2.16.1.1 Workaround

If a NULL response is received from an FPA allocate request, check to see if `FPA_QUE`*n*`_PAGES_AVAILABLE[QUE_SIZ]` indicates sufficient pointers to satisfy the request. If sufficient pointers are available, retry the operation. Only consider the pool empty after both the allocate request returns NULL and `FPA_QUE`*n*`_PAGES_AVAILABLE[QUE_SIZ]` is too low.

3 Issues Resolved in Pass 2

The following issues have been resolved in pass 2 (or earlier) of OCTEON Plus, and no longer apply. They are listed in the same order as the pass 1 known issues document to ease cross-referencing.

3.1 CN54XX/CN55XX/CN56/57XX Hardware Reference Manual

3.1.1 Coherent Memory Bus, Level-2 Cache Controller, DRAM Controllers Chapter

3.1.1.1 LMC_PLL_CTL[CLKF] Range

This issue is fixed in newer versions of the Hardware Reference Manual.

3.1.2 Signal Descriptions Chapter

3.1.2.1 SMI Interface Power

This issue is fixed in newer versions of the Hardware Reference Manual.

3.1.3 USB Chapter

3.1.3.1 USBN_CLK_CTL Bits <16:13>

This issue is fixed in newer versions of the Hardware Reference Manual.

3.1.4 System Management Interface (SMI) Chapter

3.1.4.1 SMI/MDIO Read Sampling

This issue is fixed in newer versions of the Hardware Reference Manual.

3.1.5 Electrical Specifications Chapter

3.1.5.1 Core Voltage for 600 MHz Parts

This issue is fixed in newer versions of the Hardware Reference Manual.

3.2 Silicon

3.2.1 SGMII Interfaces

3.2.1.1 CRC Generation when Preambles Disabled

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.1.2 HiGig2 Backpressure Messages Not Supported

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.1.3 Sample Point Selection in 10 Mbps Mode

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.1.4 SGMII/1000BASE-X to Core Clock Frequency Ratio

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.1.5 External Loopback Inoperative

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.1.6 GMX Enable Sequence

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.2 DDR Controller

3.2.2.1 LMC*_DDR2_CTL[ODT_ENA] Non-functional

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.2.2 Accesses to Nonexistent DDR Memory

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.3 JTAG

3.2.3.1 DC JTAG on SerDes Wires

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.4 SERDES Interfaces

3.2.4.1 Transmit Resistor

This issue is fixed in pass 1.1 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5 PCI Express Interface

3.2.5.1 PCIE DMA Engines

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.2 PCIE Inbound DMA

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.3 PCIE_DMA_CNTn Byte Counting Mode

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.4 Configuration Request Retry

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.5 UR/CA Response without ECRC

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.6 PCIE Packet Input and Output

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.7 NPEI_LAST_WIN_RDATAn Registers

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.8 ECRC Destroyed in Peer-to-Peer UR/CA Completions

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.9 UR/CA to PCIe DMA Engines

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.10 DMA Engines Create Only 64-bit TLPs

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.11 DMA Engines can't be Reset

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.12 PCIe Deskew with XAUI

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.13 Detecting PCIe Root Complex Reset Completion

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.14 Expansion ROM Access

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.15 PCIe Soft Reset from Remote

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.5.16 Unused PCIe Controllers

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.6 Random Number Generator

3.2.6.1 RNG Output Constant

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.7 General

3.2.7.1 MSC_CLKOUT

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.7.2 3.3 V Power Consumption

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.8 Packet Input (PIP/IPD)

3.2.8.1 Multi-Buffer Packet Corruption

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.9 XAUI Interface

3.2.9.1 XAUI Internal Loopback

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.9.2 XAUI Transmit Lane Swapping

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.9.3 Packets Received when Interface Shut Down

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.10 Media Independent Interface

3.2.10.1 Input Packet Alignment

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.11 Central Interrupt Unit

3.2.11.1 Watchdog Timeout Frequency

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.12 Boot Bus

3.2.12.1 Configurable DMACK Polarity

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.13 Universal Serial Bus (USB)

3.2.13.1 USBN_CLK_CTL[DIVIDE] Range

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.

3.2.14 Universal Asynchronous Receiver / Transmitter (UART)

3.2.14.1 UART Line Status Interrupt

This issue is fixed in pass 2 OCTEON Plus CN54XX / CN55XX / CN56XX / CN57XX.