Specification Update

AMD Alchemy™ Au1500™ Processor

Document Revision History

Date	Revision	Description
	1	Early version
January 2002	2	Added issue #5
May 2002	3	Added issue #6
October 2002	4	Added issue #7-13 and updates for rev 1.1 silicon
December 2002	A	Added issue #14-21 and Specification changes, Assigned Publication ID (PID) 27362 rev A
January 2003	B	• Errata #20 and #21 expanded • Errata #22 through #29 added • Modifications to DC Parameters added to Specification Changes • Labeling corrections for *Figure 34. Processor External Signals* • CSBA default value corrected under *Static Chip Address Registers* • Information on Silicon Rev 1.2 added • Chip ordering information added to Specification Changes
March 2003	C	• Title modified • Errata #15 and #29 modified • Errata #30 added • Chip ordering number used in DC Parameter headings
September 2003	D	• Errata #29 fixed in Silicon Stepping AD • Specification Changes added: — Tcase Specification — Thermal Characteristics
June 2005	E	**Device Errata** • Add #31-36 **Specification Changes** • Data for Extended grade device added **Data Book Errata** • Add #1-8

The AMD Alchemy™ Au1500™ processor may contain design defects or errors known as errata that cause its behavior to differ from the published specification. Characterized *Device Errata* are documented in this Specification Update.

As needed, this Specification Update provides a summary table and more detailed descriptions of permanent *Specification Changes* made to the published product documentation.

This Specification Update also provides a summary table and more detailed descriptions of known *Data Book Errata*.

To establish Specification Update information, AMD tests to errata, specification changes, and current documentation. Specification Update information supersedes current published documentation.

Product Identification

Software is able to determine the silicon revision and stepping by reading the Processor ID and Revision (PRid) Register of Coprocessor 0. The following table lists the supported silicon steppings. Errata are reported by silicon stepping.

Silicon Revision	Silicon Stepping	Processor ID Register (PRId)	EJTAG Device ID (IDCODE)
Rev 1.0	AB	0x01030200	0x0010228F
Rev 1.1	AC	0x01030201	0x1010228F
Rev 1.2	AD	0x01030202	0x2010228F

Device Errata Summary Table

The following table lists the status of all characterized errata. A blank field under a particular Silicon Stepping implies that the listed erratum does not apply to that stepping.

Errata Number	Silicon Stepping			Errata
	AB	AC	AD	
1	X			Enabling multiple buffers in the DMA controller can cause the DMA mode configuration to become corrupted.
2	X			PCMCIA reads incorrect byte lane on odd byte transfers from 8-bit cards.
3	X			Overlapped system bus accesses by USB and PCI can result in corrupted data.
4	X			OD bit in the Config0 register is write only.
5	X			Interrupts from the GPIO2 block are generated incorrectly.
6	X			Cacheable accesses by the core to PCI can interact with external master writes.
7	X	X	X	USB Host Enable register (0xB017FFFC) may not return the correct value after a single read.
8	X			Static bus configured as 16-bit bus drives all 32 bits.
9	X			PCI target access to Au1500™ processor may not stop at MBAR boundary.
10	X	X	X	PCI block register or master accesses during a PCI reset may hang the Au1500™ processor.
11	X			TLB does not correctly handle the physical address bit corresponding to the even/odd PFN select bit for page sizes greater than 4KB.
12	X			PCI target accesses which hit in the internal cache may not return correct data.
13	X			Programmable Counter Control Register can become unwritable.
14	X	X		A duplicate store operation can occur when the internal Write Buffer is full.
15	X	X		The USB host controller will generate corrupted packets when attempting to transmit isochronous packets larger than 67 bytes.

Errata Number	Silicon Stepping			Errata
	AB	AC	AD	
16	X			Snoop may return incorrect data and in the worst case stall the system bus for coherent access by a peripheral and non-cacheable access by the processor.
17	X			DMA Configuration Register Write and DMA channel post collision can result in bad data in Configuration Register.
18	X	X	X	The AC97 CODEC command response is only valid for one frame time after a read request.
19	X	X	X	When using a DMA buffer address that is not cacheline aligned, the number of datums in the last transfer of a transaction may not match the Transfer Size set in the dma_mode register.
20	X	X		USB device interrupt latency may cause status register content to be lost.
21	X	X		Under certain conditions, the USB host will write an extra byte to system memory.
22	X	X		USB host disconnect event from the root hub while a device is transmitting can hang USB host controller.
23	X	X		USB software reset during USB Host master transaction can corrupt master transfer.
24	X	X		USB host receiving two back-to-back zero length IN packets can treat second packet incorrectly as a DataOverrun Error.
25	X	X		IN transactions when interleaved with certain control IN transactions to the USB device will receive incorrect data.
26	X	X		USB device returns additional byte on Bulk IN transfers, if a particular sequence occurs.
27	X	X	X	USB device changes, workarounds and software requirements for reliable operation.
28	X	X		EJTAG debug exception return executing from cacheable space can hang processor.
29	X	X		PCI Master Reads/Writes less than 32-bits to Memory or Configuration Space are not supported.
30	X	X		Automatic hardware flow control is only available in silicon revision 1.2 (silicon stepping AD) and later.
31	X	X	X	Ethernet MACs cannot detect underflow.
32	X	X	X	The AMD Alchemy™ Au1500™ processor does not support PCI-to-PCI bridges.
33	X	X		Peripheral cache snoop read may return erroneous data.
34	X	X	X	Ethernet MACs may drop multicast hash filtered packets on receive.
35	X	X	X	After an asynchronous debug exception (DINT), the DEPC may point to the incorrect address during an eret.
36	X	X	X	System bus masters (USB host, PCI, MAC0, MAC1, DMA) may receive stale data.

Device Errata

1. When simultaneously enabling 2 buffers in a DMA channel, the DMA mode configuration register can become inadvertently corrupted.

Description

When simultaneously enabling 2 buffers in a DMA channel, the DMA mode configuration register can become corrupted if software updates the mode register at the same time hardware is posting an update to the register.

Affected Step

AB

Workaround

Two workarounds are available.

1. If only one buffer is enabled at any one time, this problem will not occur. This forces the software to use the buffers serially and will decrease the amount of latency that the software has to enable the next buffer.

2. Setting the OD bit (bit19) in the coprocessor register Config0 will fix this problem.

Note: The OD bit is write only (See #4). Care must be taken that read modify write operations to the config register do not clear this bit.

Status

Fixed

2. PCMCIA controller reads incorrect byte lane on odd byte transfers from 8-bit cards.

Description

The PCMCIA controller directs data to or from the wrong byte lane when doing a read or write of an odd byte from an 8-bit peripheral.

Affected Step

AB

Workaround

Requires external hardware to echo PCMCIA DATA[7:0] to RD[15:8] when a single byte read from an odd byte lane occurs and to modify the CE encoding during both reads and writes. See application note: “Au1000 8-bit PCMCIA Host Adaptor”.

Status

Fixed

3. Overlapped system bus accesses by USB and PCI can result in corrupted data.

Description

Simultaneous activity on USB and PCI that generates overlapped system bus accesses can result in corrupted data.

Affected Step

AB

Workaround

This problem will only occur if external masters are accessing the Au1500™ system bus during USB activity. If the Au1500 processor is the only PCI master accessing the Au1500 processor resources this problem will not occur. If alternate master access is required, setting the OD bit (bit 19) in the coprocessor register Config0 will fix this problem by inhibiting overlapped system bus activity.

Note: The OD bit is write only (See #4). Care must be taken that read modify write operations to the config register do not clear this bit.

Status

Fixed

4. The OD bit in the Config0 register is write only.

Description

The OD bit in register Config0 is write only (it is read as a zero) so software performing a read/modify/write to this register could inadvertently clear this bit.

Affected Step

AB

Workaround

Software should keep a shadow register of the config 0 register so that it will not inadvertently clear the OD bit.

Status

Fixed

5. Interrupts from the GPIO2 block are generated incorrectly.

Description

The GPIO2 block generates interrupts based on bits 15:8 of the GPIO2 Data Output Register, rather than on the state of the GP215-GP208 pins. Therefore, pins GP215-GP208 cannot generate interrupts and unexpected interrupts can occur by writing the Data Output Register.

Affected Step

AB

Workaround

None

Status

Fixed

6. Cacheable accesses by the core to PCI can interact with external master writes.

Description

If the core does a cacheable read from PCI at the same time that an external master does a partial word write to the Au1500 processor's SDRAM the byte masks for the write can be incorrectly associated with the processor read resulting in an incorrect cache update.

Affected Step

AB

Workaround

If external PCI based peripherals are capable of generating partial word writes then the CMEM region should not be used.

Status

Fixed

7. The USB Host Enable register (0xB017FFFC) may not return the correct value after a single read.

Description

The USB Host Enable register (0xB017FFFC) may not return the correct value after a single read.

Affected Step

AB, AC, AD

Workaround

When reading the USB Host Enable register (0xB017FFFC), it should be read twice, back to back, to insure that the correct value is read.

Status

Not Fixed

8. Static Bus configured as 16-bit bus drives all 32 bits.

Description

The upper 16 bits of the 32-bit static bus is driven on all accesses (reads and writes) when configured as a 16-bit bus, such as PCMCIA.

Affected Step

AB

Workaround

If designing with 16-bit bus, insure that upper 16 bits of data bus are not in contention with other signals for any access to 16-bit bus (reads and writes).

Status

Fixed

9. PCI target access to Au1500™ processor may not stop at MBAR boundary.

Description

If an external PCI master attempts to write 8 words starting at the last 4 words of the MBAR window, the Au1500™ processor will incorrectly write all 8 words into memory.

Affected Step

AB

Workaround

Current software drivers are aware of the MBAR boundary and do not exceed it. Custom drivers need to not overflow the boundary.

Status

Fixed

10. PCI block register or master accesses during a PCI reset may hang the Au1500™ processor.

Description

PCI block register or master accesses during PCI_RESET# may hang the Au1500 processor.

Affected Step

AB, AC, AD

Workaround

Always reset the Au1500 processor if PCI_RST# is asserted by tying PCI_RST# and $\overline{\text{RESETIN}}$ (Au1500 system reset) together in the board design. Or, control the assertion of PCI_RST# from Au1500 software with $\overline{\text{PCI\_RSTO}}$ (GPIO[200]) to indicate when there are no PCI transactions from the processor.

Status

Not Fixed

11. TLB does not correctly handle the physical address bit corresponding to the even/odd PFN select bit for page sizes greater than 4KB.

Description

When a TLB access occurs with a page size greater than 4KB, the physical address bit corresponding to the even/odd PFN select bit is masked off when EntryLo0/EntryLo1 is updated. During translation of a virtual address, if it hits in the TLB with a page size greater than 4KB, the address bit that corresponds to the even/odd PFN select bit comes from the virtual address, rather than the physical address in EntryLo0/EntryLo1. For example, with a page size of 16MB, the even/odd PFN select bit is bit 24 of the virtual address. When updating the TLB, physical address bit 24 is masked for EntryLo0/Entrylo1. During address translation, virtual address bit 24 is propagated into bit 24 of the translated physical address. See "Table 3. Values for Page Size and

PageMask Register" in the Au1500 databook for additional information on the even/odd PFN select bit. A side effect is TLBR instructions also have the lowest PFN bit incorrectly masked on the tlb read result.

Affected Step

AB

Workaround

In software, always set the tlb page frame number (PFN) bit corresponding to the odd/even TLB array select bit the same as the virtual address. Or, only use 4KB page sizes in the TLB. Software must ensure that for TLB updates with a page size greater than 4KB, the corresponding even/odd PFN select bit in the EntryLo0 physical address is zero, and the corresponding even/odd PFN select bit in the EntryLo1 physical address is one. While the TLB does not record these bits in the EntryLo0/EntryLo1, the even/odd PFN select bit of the virtual address that hits in this TLB will propagate into the translated address and produce the correct physical address. For contiguous memory regions mapped using a TLB entry (e.g. two adjacent physical memory regions mapped by a single TLB), the problem does not manifest itself. In this situation, the EntryLo0/EntryLo1 physical addresses differ only by the corresponding even/odd PFN select bit. As such, during the TLB update, EntryLo0 and EntryLo1 are written with the same physical address. Then during translation, the virtual address even/odd PFN select bit propagates to produce the correct physical address. For non-contiguous memory regions, software must abide by the workaround described above.

Status

Fixed

12. PCI Target accesses which hit in the internal cache may not return correct data.

Description

When an external PCI master initiates a target read access to the Au1500™ processor and that access hits in the internal cache and crosses a cacheline boundary, incorrect data can be returned.

Affected Step

AB

Workaround

Always set the PCI block to do non-cacheable accesses, or always set the TLB pages in the MBAR window in Au1500 processor memory space as uncacheable CCA=2.

Status

Fixed

13. Programmable Counter Control Register can become unwritable.

Description

If bit 8 of the Programmable Counter Control Register is set to enable the 32K oscillator and software then clears this same bit, subsequent writes to the Programmable Counter Control Register have no effect.

Affected Step

AB

Workaround

Once the 32K oscillator is enabled, it can be disabled, but no further writes will have an effect.

Status

Fixed

14. A duplicate store operation (with an identical address and data) can occur within a specific timing window, when the internal Write Buffer is full.

Description

All non-cacheable processor stores and data cache store misses are routed through the write buffer. The write buffer is a 16-word deep first-in-first-out (FIFO) queue.

When the write buffer is full, there is a possibility of duplicating the last store entry, which eventually appears as a duplicate store transaction. This effect is considered harmless for RAM, but can create problems for memory or peripherals where write operations may have side effects beyond updating a specific storage location; such as a FIFO or Flash memory. This will apply to both on-chip and off-chip devices.

The conditions needed to create this situation are:

1. 15 valid write buffer entries plus a valid store in the write buffer merge latch.

2. The write buffer is shifting entries due to an internal system bus grant.

3. The merge latch store is pushed into the write buffer at the same time as the shift due to one of the following conditions:

a) merge latch data is marked non-mergeable

b) load hit to merge latch

c) dcache flush (sync operation)

When these three events happen, the merge latch store will be entered into both of the last two entries of the write buffer and will eventually result in two identical stores.

Affected Step

AB, AC

Workaround

Note: Use either Workaround #1 or Workaround #2.

Workaround #1:

Add a SYNC instruction before every non-mergeable (meaning non-cacheable) store instruction to devices that would be adversely affected by identical stores to the same location and also, add a SYNC instruction before returns from all exceptions.

Note: Software drivers for the processor's internal peripherals which use the internal DMA controllers to move data into and out of peripheral FIFOs will not need modifications. The problem occurs only when the Au1 core is performing stores.

The integrated peripherals AC97, USB Device, I^2S, UART and SecureDigital controllers contain embedded, software-accessible FIFOs within the peripheral. Therefore, any programmed-IO access to these FIFOs must accommodate the workaround for the write-buffer double-write. If DMA is utilized to access these embedded FIFOs, such is typically the case with the AC97, USB Device, I^2S and SecureDigital controllers, then the workaround is not needed.

The remaining integrated peripherals do not contain software accessible FIFOs and thus are unaffected by this erratum: USB Host, IrDA, Ethernet MAC, SSI, LCD, GPIO2, System Control, PCI, and DMA controllers. The design of the interrupt controllers is such that this double write has no affect on the operation of the interrupt controllers; interrupts will not be lost.

If an AMD Alchemy™ processor-based design incorporates external FIFOs or logic sensitive to a double-write (i.e. there are undesirable side effects), and if the FIFO/logic is accessed via programmed-I/O, then the workaround must be implemented for access to these devices. The ATA interface contained on a PCMCIA or CompactFlash disk drive is an example of a common external FIFO that is accessed via programmed-I/O; all accesses to the ATA interface must incorporate the workaround. Similarly, Flash memory devices rely on an exact sequence of write operations (commands) to initiate a program or erase operation and will require the workaround.

In cases where the driver for an affected peripheral must implement the workaround, simply issuing a SYNC instruction prior to the store to the FIFO is all that is necessary. For example, consider the transmission of a character utilizing the integrated UART, the code for such an activity is typically:

```
*uart_txdata = ch;

asm("sync");
```

The SYNC instruction after the write to uart_txdata is to ensure that the data is written to the peripheral and prevents the write from remaining in the write-buffer indefinitely. Since this code is doing programmed-IO, the write-buffer workaround is needed, and the code example becomes:

```
asm("sync");

*uart_txdata = ch;

asm("sync");
```

In this situation, the SYNC instruction preceding the write to uart_txdata ensures that the write-buffer is empty before this store occurs, thus avoiding the duplicate write-buffer store conditions.

If an exception occurs between the SYNC instruction and the write, then it is possible for the exception handling software to cause the write buffer to fill, satisfying condition a) of the description. To avoid this, the return from exception must issue a SYNC before the ERET instruction as well. Since in many cases it is not feasible to make changes to the exception handlers (e.g. commercial RTOS), the workaround is then to prevent exceptions, specifically interrupts, from occurring between the SYNC and the store. Thus the example UART code becomes:

```
disable interrupts

asm("sync");

*uart_txdata = ch;

asm("sync");

enable interrupts
```

Workaround # 2:

Disable the write buffer (bit 22 in CP0_config0) entirely.

Since Workaround #1 only needs changes in a few focused areas, it is the recommended workaround for this erratum. Workaround #2 is not recommended except in the case where developers want a quick fix in order to continue development in parallel with implementing Workaround #1. Workaround #2 can cause a significant reduction in performance since it forces the CPU to wait for each previous write to complete before any additional write may be issued.

Status

Fixed

15. The USB host controller will generate corrupted packets when attempting to transmit isochronous packets larger than 67 bytes.

Description

Isochronous OUT packets may not exceed 67 bytes. If an OUT is sent with an isochronous payload greater than 67 bytes, the internal FIFO will be written erroneously and a corrupted packet will result.

Affected Step

AB, AC

Workaround

None

Status

Fixed

16. Snoop may return incorrect data and in the worst case stall the system bus for coherent access by a peripheral and non-cacheable access by the processor.

Description

A data cache snoop may return bad data, and in the worst case stall the system bus when:

1. A peripheral makes a coherent access on the System Bus (i.e. the Data Cache will be snooped), and

2. The processor does a non-cacheable load instruction such that the lower 32 bits of the address map to the same cacheline as the peripheral access

Affected Step

AB

Workaround

1. Set up the software so that it does not have different cache-ability for the same cacheline address.

2. Insure that off chip peripherals utilizing the upper address space (ADDR[35:32-03= 0) are not mapped such that the lower 32 bits overlap cacheable system memory (i.e. SDRAM) that system bus masters will access.

Note: In typical systems, the implication of this errata is that the system programmer must insure that the lower 32 bits of the PCI addresses are uniquely mapped relative to system memory.

Status

Fixed

17. DMA Configuration Register Write and DMA channel post collision can result in bad data in Configuration Register.

Description

There is a timing window where if a DMA channel is posting status on an operation at the same time a Processor configuration register write occurs, the write data and value in the configuration register is corrupted.

Affected Step

AB

Workaround

Set the OD bit (bit 19) in the coprocessor Register Config0.

Status

Fixed

18. The AC97 CODEC command response is only valid for one frame time after a read request.

Description

When reading an AC97 CODEC register (by issuing a CODEC command through the ac97_cmmd register), the response in ac97_cmmdresp will only be valid for 1 frame time (~20uS) after the Command Pending (CP bit in the ac97_status) is cleared to indicate that the command is completed for a read request.

Affected Step

AB, AC, AD

Workaround

The CODEC response register, ac97_cmmdresp, must be read within 1 frame time (~20uS) after Command Pending (CP bit in the ac97_status) is cleared.

Status

Not Fixed

19. When using a DMA buffer address that is not cacheline aligned, the number of datums in the last transfer of a transaction may not match the Transfer Size set in the dma_mode register.

Description

When using a DMA buffer address that is not cacheline aligned, the number of datums (four or eight) in the last transfer of a transaction may not match the Transfer Size (TS) bit set in the dma_mode register. In the above explanation, transaction is defined as multiple transfers; the size of a transfer is defined as a specific amount of datum (four or eight); the width of a datum is defined the DW bit in dma_setmode register.

Affected Step

AB, AC, AD

Workaround

To ensure that the last transfer has the correct number of datums, the DMA transfer buffer must be cacheline aligned.

Status

Not Fixed

20. USB device interrupt latency may cause status register content to be lost.

Description

Each event on the USB bus is capable of changing the contents of the USB device status register. This means that the status of a particular transfer is only valid until the next bus event. Bus events include tokens, errors, successful or unsuccessful transfers, and status changes like connect/disconnect.

To illustrate the effect of this, consider a successful data transfer to the host, followed by an ACK. This generates an interrupt to the Au1500™ processor. At the time the processor is interrupted, the status register indicates a successful transfer. If the interrupt is not serviced before another IN token is received and the FIFO is empty, the device hardware will send a NAK (normal response to an IN when no data is ready). Unfortunately, this event will change the status register to show the NAK, making it look like the previous transfer failed.

In a related problem, packet boundaries can be lost if interrupts are not serviced quickly enough. A short OUT transaction that indicates the end of a longer data stream can be concatenated with new data if the data starts arriving before the interrupt is serviced. This is because DMA is not inhibited or signaled to switch buffers at the end of packet.

Affected Step

AB, AC

Workaround

In order to service the device reliably, the status for a USB event must be read from the USB device status register before the next USB event. To accomplish this, the interrupt latency must be less than 1µs, or a higher layer protocol must be used to verify packet contents.

Status

Fixed

21. Under certain conditions the USB host will write an extra byte to system memory.

Description

The USB host can include an extra byte on IN packets that are less than the maximum packet size (MPS) for the endpoint and less than the buffer size indicated by the transfer descriptor. In these cases, the last byte of the packet is actually the first byte of the generated CRC, bit reversed. The trigger for this event is that the last bit of the CRC generates a bit stuff which implies that the last byte of the CRC is either 0x3f or 0xbf.

Affected Step

AB, AC

Workaround

A packet that meets these criteria can be detected by looking for packets that return less than the buffer size indicated by the transfer descriptor. Then calculate a CRC over all the bytes except the last. If the low byte of the CRC is 0x3f or 0xbf and the high byte of the CRC matches the bit-reversed last byte then the last byte should be discarded. This still leaves a small possibility (0.03%) that a bad packet will be accepted (when the actual size is one less than the maximum packet size) or, alternately, that a good byte will be thrown away (for a full sized packet that matches the syndrome).

Status

Fixed

22. USB host disconnect event from the root hub while the device is transmitting can hang the USB host controller.

Description

A disconnect event from the root hub while a device is active and sending data can hang the USB host controller. The window for this event is about 3 bit times and will result in about 1 hang in 50 disconnects in average use. This bug is more common with low speed devices connected to the root hub than with full speed devices.

Affected Step

AB, AC

Workaround

If the system software looks for frame interrupts after a disconnect event it can detect the hung controller by a lack thereof. At this point the host controller should be reset, by writing to the module control register, and re-enumerated.

Status

Fixed

23. USB software reset during USB Host master transaction can corrupt master transfer.

Description

If a Software Reset is issued while the USB Host is performing a DMA write, the internal statemachines can leave data in the internal FIFO's, the DMA transfer may not complete correctly, and there can be a deadlock situation between the application interface waiting for the FIFO's to empty and the host controller thinking it is reset.

Affected Step

AB, AC

Workaround

In normal operation, a software reset (this occurs by setting HcCommandStatus.HCR bit) is not generated. It is typically generated in extreme cases where the host controller is stuck. However as per the problem described above, if the software generates a software reset and it happens to hit the USB Host controller at the critical point during DMA, then there could be a deadlock situation between the application interface logic and the host controller.

To avoid this situation, software drivers should follow the steps below whenever it issues a software reset:

1. Disable all the lists to be processed by host controller.

2. Wait until the next frame boundary by enabling/monitoring SOF Interrupt

3. Wait for about ½ frame (0.5 ms) to make sure the host controller finishes writing FrameNumber back to the HCCA area of system memory.

4. Issue a Software Reset.

Status

Fixed

24. USB host receiving two back-to-back zero length IN packets can treat second packet incorrectly as a DataOverrun Error.

Description

If a zero length IN packets is immediately followed by a second zero length IN packet, the second packet can be treated as a DataOverRun error by the USB host controller.

Affected Step

AB, AC

Workaround

Set software to ignore DataOverRuns caused by Zero length packets.

Status

Fixed

25. IN transactions when interleaved with certain control IN transactions to the USB device will receive incorrect data.

Description

When an IN token is received after the setup phase for a control read transaction, the USB device will send the data requested by the control read instead of data for the IN transaction.

Affected Step

AB, AC

Workaround

There is no workaround from the device side. From the host, the problem may be avoided by not interleaving endpoint IN transactions with control read transactions.

Status

Fixed

26. USB device returns additional byte on Bulk IN transfers, if a particular sequence occurs.

Description

If a particular sequence of transactions occurs to the USB device, an additional byte is sent during Bulk IN transfers. This occurs under the following scenario:

- software has stalled the control endpoint by setting the USB device register USBD_EP0CS[FS]

- another control transfer starts before software can clear the stall condition

During the subsequent Bulk IN transfer, 1+MaxPktSize bytes are returned by the USB device.

Affected Step

AB, AC

Workaround

Do not stall the control endpoint, or ensure that the interrupt latency is less than 1µs to clear the stall condition.

Status

Fixed

27. USB device changes, workarounds and software requirements for reliable operation.

Description

In general, it is difficult to make the USB device controller function correctly in a real operating system. If the system has a very low latency RTOS, or is just an event driven program with no OS then it is possible to get a functional device. For normal OS environments (Linux, WinCE, VxWorks, etc) the device can be made to work for some applications if the effect of these problems is taken into account.

Along with the interrupt latency issue [Errata #20], there are two other areas that require special software handling:

1. Setting the STALL bit in the control register causes an endpoint to stall indefinitely. This should really only cause the current transaction to stall.
2. If the endpoint zero FIFO is not emptied immediately, a following SETUP packet can be discarded with no indication. Hosts, in general, consider this very bad behavior and will sometimes cease enumeration when this happens.

Affected Step

AB, AC, AD

Workaround

Each issue has its workarounds. To better help customers a more detailed software driver guide is planned.

For issue #1 in the Description: Clear the STALL bit when the next interrupt occurs.

For issue #2 in the Description: Always have a DMA channel enabled to capture data from endpoint zero. It is recommended that endpoint zero be received into a continuously available circular buffer that is parsed by the interrupt service routine.

Status

Not Fixed

28. EJTAG debug exception return executing from cacheable space can hang processor.

Description

If an EJTAG debug exception return (deret) instruction is executed while in debug mode, and the deret instruction resides in Cacheable space, the MIPS processor instruction fetch logic can become confused. This is not usually a problem since deret handlers are normally in non-cacheable space.

Affected Step

AB, AC

Workaround

Debug exception handler should reside in non-cacheable address space.

Status

Fixed

29. PCI Master Reads/Writes less than 32-bits to Memory or Configuration Space are not supported.

Description

Only word (32-bit) reads and writes from the MIPS processor to PCI memory or configuration space are supported. The on-chip PCI interface asserts all byte lanes from the processor to these address regions, so all accesses to PCI memory and configuration spaces are word oriented.

Affected Step

AB, AC

Workaround

Software drivers should expect all master PCI memory and configuration space accesses to be 32 bits. Byte and halfword accesses are fully supported in I/O space, so PCI peripherals that require these accesses should be placed in I/O space.

Status

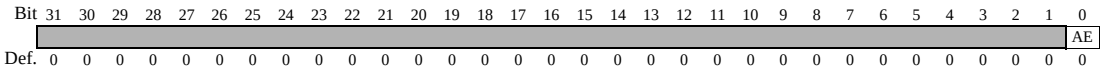
Fixed

30. Automatic hardware control is only available in silicon revision 1.2 (silicon stepping AD) and later.

Description

The **uart_autoflow** register controls automatic hardware flow control using modem control signals CTS and RTS on UART 3. Upon enabling this mode RTS becomes an output signal, and CTS becomes an input signal. RTS is asserted low to request data until the internal receive FIFO reaches its preset fullness threshold. The UART transmits data from its internal transmit FIFO while the CTS signal is low.

uart_autoflow - Automatic Hardware Flow Control Register Offset = 0x0024



Bits	Name	Description	R/W	Default
31:1	—	Reserved, should be cleared.	R	0
0	AE	Autoflow Enable Setting this bit enables automatic hardware flow control on UART3. Enabling this mode overrides software control of the signals.	R/W	0

Affected Step

AB, AC

Workaround

None

Status

Fixed

31. Ethernet MACs cannot detect underflow.

Description

After initiating a transmit, if the MAC encounters an underrun because the MAC_DMA cannot service the MAC FIFO request, the resulting underrun will go undetected and an erroneous packet will be transmitted.

Affected Step

AB, AC, AD

Workaround

The following techniques can help minimize the effect of undetected underflow conditions.

- Analyze and work to reduce system bus usage and raw latencies
- Use smaller packets where possible

Status

Not Fixed

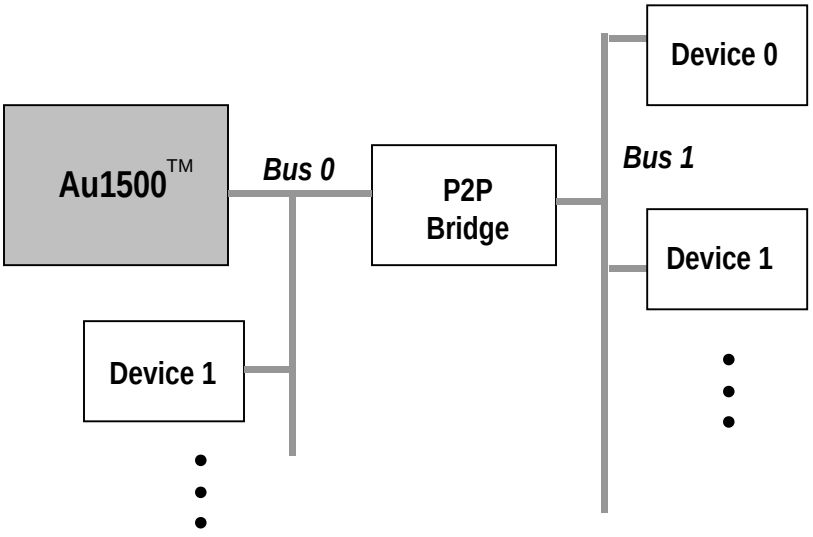
32. The AMD Alchemy™ Au1500™ processor does not support PCI-to-PCI bridges.

Description

The Au1500 processor does not support the use of mastering PCI devices "behind" PCI-to-PCI (P2P) bridges. A P2P bridge is a device which extends the PCI bus, allowing the addition of more devices than the timing or loading of one PCI bus segment can allow. This affects the Au1500 processor in both host and satellite modes.

Most Au1500 processor users can achieve their goals without considering a P2P bridge. Those that do consider a bridge should be aware that any mastering device on the other side of a P2P bridge, where the target is the Au1500 processor, will eventually lead to a lock-up condition of the PCI bus or even the entire system.

The following diagram illustrates a multi-bus system where the Au1500 processor is the host, and devices exist on bus 0 and bus 1.



Configurations that are NOT Supported:

- Bus master devices on Bus 1 which target the Au1500 processor.
- P2P Bus Bridges cannot be used with the Au1500 processor in satellite mode. This may preclude the use of the Au1500 processor in some PCI board applications

Host Mode Configurations that ARE supported:

- All PCI devices which reside on the single Au1500 processor PCI bus (bus 0)
- All PCI mastering devices which reside on the Au1500 processor PCI bus (bus 0). The Au1500 processor includes a five-way arbiter supporting the Au1500 processor and four external mastering devices.
- All PCI slave devices which reside behind a bridge (bus 1)

Affected Step

AB, AC, AD

Workaround

None

Status

Not Fixed

33. Peripheral cache snoop read may return erroneous data.

Description

While the Au1 core is doing an Index CacheOp (load tag, store tag, index invalidate or index write-back invalidate), a peripheral does a cache snoop read. If the snoop hits the cache any way other than the way specified in the CacheOp, the first half of the snoop read will be wrong.

Affected Step

AB, AC

Workaround

No work around, other than preventing peripheral snoops of cache during Index CacheOp.

Status

Fixed

34. Ethernet MACs may drop multicast hash filtered packets on receive.

Description

When receiving, the Ethernet MACs may drop multicast hash filtered packets in applications running above 180 MHz system bus speed when the toss bit **macen_mac*n***[TS] is set. The problem applies to all Ethernet speed/duplex modes.

This problem does not affect applications running at or below a system bus frequency of 174 MHz (348 MHz or lower CPU frequency).

Affected Step

AB, AC, AD

Workaround

This problem can be avoided by using either of the following options:

- Turn on the Pass All Multicast (PM) bit, or
- Set all of the bits in both hash table registers to 1 (0xFFFFFFFF).

Status

Not Fixed

35. After an asynchronous debug exception (DINT), the DEPC may point to the incorrect address during an eret.

Description

When an asynchronous debug exception (DINT) is recognized by the ibox at the same time that an eret is to begin execution, the exception may be taken in the middle of the eret with the DEPC incorrectly pointing to the delay slot of the eret and with the status register updated as if the eret occurred.

Affected Step

AB, AC, AD

Workaround

- The debug handler can determine that the error has occurred if on an asynchronous DINT the DEPC is pointing to an instruction following an eret. To avoid the problem, the debug handler can replace the DEPC with the EPC before doing the deret.

- If there is valid code at the sequential address after an eret and there is an asynchronous DINT on the sequential address, the EJTAG probe is not able to distinguish whether the DEPC points to the eret delay slot in error because of the bug, or because it actually took an exception on the code at that address. As another option the probe handler can:

1) Replace all eret with sdbbp.

2) When the sdbbp occurs, replace the sdbbp with eret and set single step.

3) Single step through eret, not allowing DINT during this time.

4) Replace eret with sdbbp again and clear single step.

Status

Not Fixed

36. System bus masters (USB host, PCI, MAC0, MAC1, DMA) may receive stale data.

Description

System bus masters (USB host controller, PCI controller, MAC0, MAC1, DMA controller), when performing coherent reads, may incorrectly receive stale data from memory instead of valid modified data from the Au1 data cache. If the request for data arrives within a 3-clock window prior to the cache line castout to memory, the cache snoop response is incorrect and stale data is retrieved from memory instead of the correct data from the cache. The cache line castout then completes, and memory is updated.

Cache/memory data is not corrupted, but the specific bus read in not valid.

Affected Step

AB, AC, AD

Workaround

Do not enable cacheable master reads if the core modifies data in cache.

Status

Not Fixed

Specification Changes

The following Specification Changes are permanent changes with reference to June 2003 Revision 30361B issue of the *AMD Alchemy™ Au1500™ Processor Data Book*.

1. Table 74, DC Parameters, page 280

This change identifies case temperature (T_{case}) as the operating temperature specification. Information on ambient temperature (T_a) is no longer provided as part of the device specification. Specification information on the Extended grade of the processor (Au1500-333MBI) is also added.

Old Values

Param	Description	Min	Nom	Max	Unit
T_{case}	Package operating temperature			TBD	ºC
T_a for Au1500-333MBC & Au1500-400MBC	Operating temperature (ambient)	0		70	ºC
T_a for Au1500-500MBC		0		50	ºC

New Values

Param	Description	Min	Nom	Max	Unit
T_{case} for Au1500-333MBC Au1500-400MBC and Au1500-500MBC	Package operating temperature	**0**		**85**	ºC
T_{case} for Au1500-333MBI		**-40**		**100**	ºC

2. Table 73. Thermal Characteristics, page 279

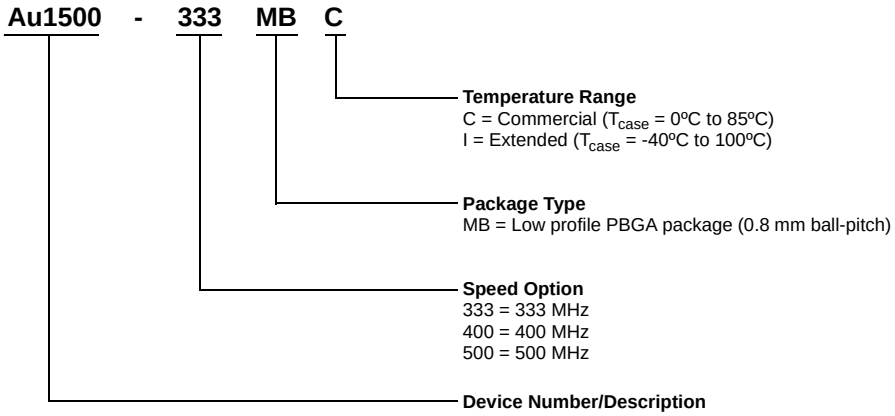
The following table replaces “Table 73. Thermal Characteristics” in the *AMD Alchemy™ Au1500™ Processor Data Book*, 30361B.

Table 77. Thermal Characteristics

Parameter	Air Flow						Unit
	0 m/sec 0 LFPM	0.25 m/sec 50 LFPM	0.5 m/sec 100 LFPM	0.75 m/sec 150 LFPM	1.0 m/sec 200 LFPM		
Θ_{JA}	33.1	31.8	30.7	30.1	28.6	—	ºC/Watt
Ψ_{JT}	—	—	—	—	—	5.0 [1]	ºC/Watt

1. Air Flow does not apply to this specification.

3. Ordering Information, page 372



Valid Combinations

Au1500-333 Au1500-400 Au1500-500	MB	C
Au1500-333	MB	I

Valid Combinations

Valid Combinations lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

Data Book Errata Summary Table

The following table lists problems discovered in the data book. A blank field under a particular Data Book Revision implies that the listed erratum does not apply to that version.

Errata No.	Data Book Revision			Description
	30362A	30362B	30362C	
1	X	X		Remove incorrect statement in AC97 Controller section.
2	X	X		Missing signal name in Figure 47, “MII Timing Interface.”
3	X	X		For PCMCIA, correct the $\overline{\text{ROE}}$ description.
4	X	X		For PCMCIA, add a qualification about compliance.
5	X	X		Remove step #5 from Section 5.2, “Using GPIO as External DMA Requests (DMA_REQ*n*).”.
6	X	X		Change the IP bit description in the UART interrupt cause register **uart_intcause**.
7	X	X		Correct the default values for the PCI timeout register.
8	X	X		Correct the default value of **macdma_rxaddr**[CB].

Data Book Errata

1. Remove incorrect statement in AC97 Controller section

Description

The last sentence in section “7.1 AC97 Controller” on page 124 reads:

All data being sent and received through the AC97 controller must be 48kHz.

This statement is incorrect and will be removed from future versions of the data book, since the processor does support variable sample rates.

Affected Data Book Revs

A, B

Status

Fixed

2. Missing signal name in Figure 49, “MII Interface Timing.”

Description

On page 292 in Figure 47, add the N*n*DIO signal name to the second waveform from the bottom.

Affected Data Book Revs

A, B

Status

Fixed

3. For PCMCIA, correct the $\overline{\text{ROE}}$ description.

Description

On page 74 in Table 16, “PCMCIA Interface Signals,” change the description for $\overline{\text{ROE}}$ as follows:

Incorrect	Correct
Output Enable - This output enable is intended to be used as a data transceiver control. It remains **high** voltage for reads and **low** voltage for writes during the entire PCMCIA transaction.	Output Enable - This output enable is intended to be used as a data transceiver control. During a PCMCIA transaction, $\overline{\text{ROE}}$ remains **asserted (low)** as configured in the timing registers (**mem_sttime*n***) for reads and **negated (high)** for writes.

Affected Data Book Revs

A, B

Status

Fixed

4. For PCMCIA, add a qualification about compliance.

Description

On page 73, add the following sentence to the first paragraph of Section 3.2.3, “PCMCIA/Compact Flash Device Type”:

The PCMCIA peripheral is designed to the PCMCIA2.1 specification—but only for the bus transactions as described in this section.

Affected Data Book Revs

A, B

Status

Fixed

5. Remove step #5 from Section 5.2, “Using GPIO as External DMA Request (DMA_REQ*n*).”

Description

On page 110 in Section 5.2, remove step #5 “Be sure that the corresponding interrupt is also enabled....” The DMA request is a separate function and does *not* need to be enabled in the interrupt controller.

Affected Data Book Revs

A, B

Status

Fixed

6. Change the IP bit description in the UART interrupt cause register.

Description

On page 180 for the UART interrupt cause register **uart_intcause**, change the default value of the IP bit to '1' and change the bit description as follows:

Incorrect

Bits	Name	Description	R/W	Default
0	IP	Interrupt Pending The IP bit is set when an interrupt is pending.	R	1

Correct

Bits	Name	Description	R/W	Default
0	IP	No interrupt pending 0 An interrupt is pending. 1 No interrupts are pending.	R	1

Affected Data Book Revs

A, B

Status

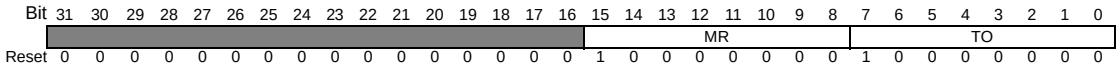
Fixed

7. Correct the default values for the PCI timeout register.

Description

On page 94, correct the default values for **pci_timeout** as follows:

pci_timeout - PCI configuration timeout Offset = 0x0140



Bit(s)	Name	Description	R/W	Default
31:16	—	Reserved	R/W	0
15:8	MR	Max retries Contains the maximum number of retries to be attempted per transaction. Valid values range from 1 to 255. Note that a value of 0 disables this function, which means no limit is placed on the number of retries attempted.	R/W	0x80
7:0	TO	Target-ready ($\overline{\text{PCI\_TRDY}}$) timeout Contains the maximum number of PCI clocks to wait for $\overline{\text{PCI\_TRDY}}$ assertion before terminating a transfer. Valid values range from 1 to 255. Note that a value of 0 disables this function, which means no time limit is placed on waiting for $\overline{\text{PCI\_TRDY}}$.	R/W	0x80

Affected Data Book Revs

A, B

Status

Fixed

8. Correct the default value of macdma_rxaddr[CB].

Description

On page 168, the default value for macdma_rxaddr[CB] should be *UNPRED* (not 0).

Affected Data Book Revs

A, B

Status

Fixed

Contacts

www.amd.com pcs.support@amd.com

Trademarks