Specification Update

AMD Alchemy™ Au1100™ Processor

Document Revision History

Date	Revision	Description
November 2002	1	First release of this document
December 2002	A	Errata #10-13, Spec Changes #1-6, Assigned Publication ID (PID) 27353 rev A
February 2003	B	• Errata #2, #5, and #13 wording modified • Added information on Silicon Revision 1.2 • Errata #14-20 added • Used chip order number in DC parameter headings • Chip ordering information added to Specification Changes
September 2003	C	• Errata #21-24 added • Errata updated for BE/1.3 silicon • Specification Changes update on: — SI bit in usbd_enable — Tcase specification — I2S Registers — Thermal Characteristics • Data Book Errata information added
March 2004	D	• Device Errata #25-28 added • Device Errata #2 and #27 wording modified • Device Errata #19, reference to other Errata corrected • Specification Changes: — Added data for Extended grade device for #2 — Corrected **i2s_config** register format for #3 — Updated ordering information for BF silicon stepping — New voltage numbers for processor running at 500 MHz • Data Book Errata #9-18 added
June 2005	E	• Device Errata #29 added • Data Book Errata #19-23 added

The AMD Alchemy™ Au1100™ 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 document 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	BA	0x02030201	0x1020228F
Rev 1.1	BC	0x02030202	0x2020228F
Rev 1.2	BD	0x02030203	0x3020228F
Rev 1.3	BE, BF[a]	0x02030204	0x4020228F

a. Silicon steppings BE and BF have no functional differences; however, BF has dual passivation and improved ESD protection.

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 No.	Silicon Stepping					Errata
	BA	BC	BD	BE	BF	
1	X	X	X	X	X	The USB Host Enable register (0xB017FFFC) may not return the correct value after a single read.
2	X	X	X	X	X	The time scale option (**mem_stcfg*n***[TS] = 1) restricts the Tcs_oe timing parameter on the static bus interface.
3	X	X	X			TLB does not correctly handle the physical address bit corresponding to the even/odd PFN select bit for page sizes greater than 4KB.
4	X	X				A duplicate store operation can occur when the internal Write Buffer is full.
5	X	X	X			The USB host controller will generate corrupted packets when attempting to transmit isochronous packets larger than 67 bytes.
6	X					Programmable Counter Control Register can become unwritable.
7	X	X	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.
8	X	X	X	X	X	The AC97 CODEC command response is only valid for one frame time after a read request.
9	X	X	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.
10	X	X	X			Actual voltage values for the Au1100™ processor running at 333 MHz are different than those specified. (Voltage values are correct in revision A and later of the data book.)
11	X	X	X			Actual voltage values for the Au1100™ processor running at 400 MHz are different than those specified. (Voltage values are correct in revision A and later of the data book.)
12	X	X	X			USB device interrupt latency may cause status register content to be lost.
13	X	X	X			Under certain conditions, the USB host will write an extra byte to system memory.
14	X	X	X			USB host disconnect event from the root hub while a device is transmitting can hang USB host controller.
15	X	X	X			USB software reset during USB Host master transaction can corrupt master transfer.
16	X	X	X			USB host receiving two back-to-back zero length IN packets can treat second packet incorrectly as a DataOverrun Error.
17	X	X	X			IN transactions when interleaved with certain control IN transactions to the USB device will receive incorrect data.
18	X	X	X			USB device returns additional byte on Bulk IN transfers, if a particular sequence occurs.
19	X	X	X			USB device changes, workarounds and software requirements for reliable operation.

Errata No.	Silicon Stepping					Errata
	BA	BC	BD	BE	BF	
20	X	X	X			EJTAG debug exception return executing from cacheable space can hang processor.
21	X	X	X	X	X	I^2S always outputs I^2S mode regardless of the mode specified by configuration register.
22	X	X	X			Incorrect phase relationship between I2SWRD and I2SCLK could cause I2S data shift.
23	X	X	X			A random audio channel swap may occur after a transmit underrun or a receive overrun.
24	X	X	X			Enabling I^2S Receive without enabling Transmit can corrupt received data.
25	X	X	X	X	X	The Ethernet MAC cannot detect underflow.
26	X	X	X			Peripheral cache snoop read may return erroneous data.
27	X	X	X	X	X	Ethernet MAC may drop multicast hash filtered packets on receive.
28	X	X	X	X	X	After an asynchronous debug exception (DINT), the DEPC may point to the incorrect address during an eret.
29	X	X	X	X	X	System bus masters (USB host, LCD, MAC, IrDA, DMA) may receive stale data.

Device Errata

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

Description

After only a single read, the USB Host Enable register (0xB017FFFC) may not return the correct value.

Affected Step

BA, BC, BD, BE, BF

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

2. The time scale option (mem_stcfg*n*[TS] = 1) restricts the Tcs_oe timing parameter on the static bus interface.

Description

If the time-scale bit **mem_stcfg*n***[TS] is set, the effective value of Tcs_oe is zero regardless of the actual values programmed in **mem_sttime*n***[T1] and **mem_sttime*n***[T0]. This means that the address setup time before asserting output enable is always one (Tcs_oe + 1) scaled clock when **mem_stcfg*n***[TS] is set.

Affected Step

BA, BC, BD, BE, BF

Workaround

None

Status

Not Fixed

3. 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 4. Values for Page Size and PageMask Register" in the Au1100™ processor 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

BA, BC, BD

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

4. 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

BA, BC

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 Au1core 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

5. 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

BA, BC, BD

Workaround

None

Status

Fixed

6. 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

BA

Workaround

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

Status

Fixed

7. 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

BA, BC, BD

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] != 0) are not mapped such that the lower 32 bits overlap cacheable system memory (i.e. SDRAM) that system bus masters will access.

Note: If memory pages for the LCD controller are marked uncacheable in the processor and cache coherent by the LCD controller, disable LCD snooping by clearing the coherency bit (bit 15) in the LCD Control Register.

Status

Fixed

8. 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

BA, BC, BD, BE, BF

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

9. 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

BA, BC, BD, BE, BF

Workaround

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

Status

Not Fixed

10. Voltage values for the Au1100™ processor running at 333 MHz are different than those specified.

Description

The minimum internal core voltage, Vddi, is 1.1V (not specified in data book).

The maximum internal core voltage, Vddi, is 1.3V (not specified in data book).

The typical internal core voltage, Vddi, is 1.2V (specified in data book as 1.0V).

Affected Step

BA, BC, BD

Workaround

None

Status

Fixed - Information provided in revision A and later of the data book is correct.

11. Voltage values for the Au1100™ processor running at 400 MHz are different than those specified.

Description

The minimum internal core voltage, Vddi, is 1.1V (not specified in data book).

The maximum internal core voltage, Vddi, is 1.3V (not specified in data book).

The typical internal core voltage, Vddi, is 1.2V (specified in data book as 1.0V).

(Information on power values has been removed since it is correctly stated in revision A and later of the data book and is not part of the voltage information covered by this errata.)

Affected Step

BA, BC, BD

Workaround

None

Status

Fixed- Information provided in revision A and later of the data book is correct.

12. 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 Au1100™ 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

BA, BC, BD

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

13. 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

BA, BC, BD

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

14. 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

BA, BC, BD

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

15. 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

BA, BC, BD

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

16. 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

BA, BC, BD

Workaround

Set software to ignore DataOverRuns caused by Zero length packets.

Status

Fixed

17. 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

BA, BC, BD

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

18. 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

BA, BC, BD

Workaround

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

Status

Fixed

19. 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 #12], 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

BA, BC, BD

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

Fixed

20. 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

BA, BC, BD

Workaround

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

Status

Fixed

21. I²S always outputs I²S mode regardless of the mode specified by the configuration register.

Description

I²S always outputs in I²S mode, regardless of the mode specified for it in the configuration register.

Affected Step

BA, BC, BD, BE, BF

Workaround

Software can shift audio data bits to output the desired format.

Status

Not Fixed

22. Incorrect phase relationship between I2SWRD and I2SCLK could cause I2S data shift.

Description

I2SWRD transitions on the active edge of I2SCLK instead of the inactive edge. This may cause a problem with an external CODEC determining the correct start of a new subframe. Consequently, data being read or written to the CODEC may become shifted.

Affected Step

BA, BC, BD

Workaround

A hardware work around to effectively push I2SCLK and I2SDIO (as an output) forward by 1/2 of an I2SCLK period can be described in verilog as follows:

```
****
assign i2sclk2 = ~i2sclk;
regi2sdio2;
always @(posedge i2sclk)
begin
   i2sdio2 <= i2sdio;
end
****
```

For this fix, the signals with the suffix '2' are from the CPLD to the CODEC, the signals without the '2' are from the Au1100™ processor to the CPLD.

In words this is inverting the I2SCLK and re-clocking the I2SDIO (as an output to the CODEC) on the rising edge of the I2SCLK. This will make it valid on the rising edge on I2SCLK2 at the CODEC. This fix will only work for data output.

Status

Fixed

23. A random audio channel swap may occur after a transmit underrun or a receive overrun.

Description

Channel playback becomes unpredictable when audio playback/receive is resumed after a transmit underrun or a receive overrun. For the transmit case, this is due to the data being shipped out on the first edge of I2SWORD instead of waiting for the edge that corresponds to initial channel. For the receive case, channel orientation cannot be determined after an overrun.

Affected Step

BA, BC, BD

Workaround

If transmit (playback) and receive are going to be enabled at random overlapping times, it is suggested that both be enabled from startup.

The driver should be programmed to feed an even number of 0's to the transmit FIFO and read an even number of samples from the receive FIFO whenever valid audio is not being transmitted or received. This will insure that channel playback is predictable.

Status

Fixed

24. Enabling I^2S Receive without enabling Transmit can corrupt received data.

Description

When Receive is enabled and Transmit is not enabled the received data can be corrupted. This is evident on playback or transmission of the data in symptoms such as a perceived "speeded-up" audio.

Affected Step

BA, BC, BD

Workaround

The following startup sequence must be used every time Receive is to be enabled without enabling Transmit:

- Write 2 words to Transmit FIFO (value does not matter)
- Enable Transmit & Receive
- Poll unit Transmit Empty (TE) is set
- Disable Transmit

Status

Fixed

25. The Ethernet MAC cannot detect underflow.

Description

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

Affected Step

BA, BC, BD, BE, BF

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

26. 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

BA, BC, BD

Workaround

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

Status

Fixed

27. Ethernet MAC may drop multicast hash filtered packets on receive.

Description

When receiving, the Ethernet MAC 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

BA, BC, BD, BE, BF

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

28. 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

AA

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

29. System bus masters (USB host, LCD, MAC, IrDA, DMA) may receive stale data.

Description

System bus masters (USB host controller, LCD controller, MAC, IrDA controller, 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

BA, BC, BD, BE, BF

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 to the device specification with reference to the June 2003 Revision 30362B issue of the *AMD Alchemy™ Au1100™ Processor Data Book*.

1. Page 124, Device Controller Enable Register

Old

Bits	Name	Description	R/W	Default
2	SI	Streaming Isochronous mode - Not Available	W	0

New

Bits	Name	Description	R/W	Default
2	SI	Streaming Isochronous mode - **Clearing this bit allows isochronous endpoints to service IN and OUT transactions when the endpoint interrupt is pending. This mode is enabled by default.** Setting this bit (not recommended) forces ISO endpoints to wait for pending interrupts to be cleared before accepting further data	W	0

2. Table 89, DC Parameters, page 332

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 (Au1100-333MBI) is also added.

Old

Param	Description	Min	Nom	Max	Unit
T_{case}	Package operating temperature			TBD	ºC
T_a	Operating temperature (ambient)	0		70	ºC

New

Param	Description	Min	Nom	Max	Unit
T_{case} for Au1100-333MBC Au1100-400MBC and Au1100-500MBC	Package operating temperature	**0**		**85**	ºC
T_{case} for Au1100-333MBI		**-40**		**100**	ºC

3. 6.6.1 I²S Register Descriptions, Bit 12, pages 176-177

Old Information

i2s_config - Configuration and Status Offset = 0x0004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							XU	XO	RU	RO	TR	TE	TF	RR	RE	RF					PD	LB	IC	FM		TN	RN	SZ					
Def.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

New Information

i2s_config - Configuration and Status Offset = 0x0004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							XU	XO	RU	RO	TR	TE	TF	RR	RE	RF				ICK	PD	LB	IC	FM		TN	RN	SZ					
Def.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Old Values

Bit	Name	Description	Read/ Write	Default
12	—	Reserved, should be cleared.	R	0

New Values

Bit	Name	Description	Read/ Write	Default
12	**ICK**	**Invert Clock** **0Data is valid on falling edge.** **1Data is valid on rising edge.**	**R/W**	0

4. Table 88. Thermal Characteristics, page 331

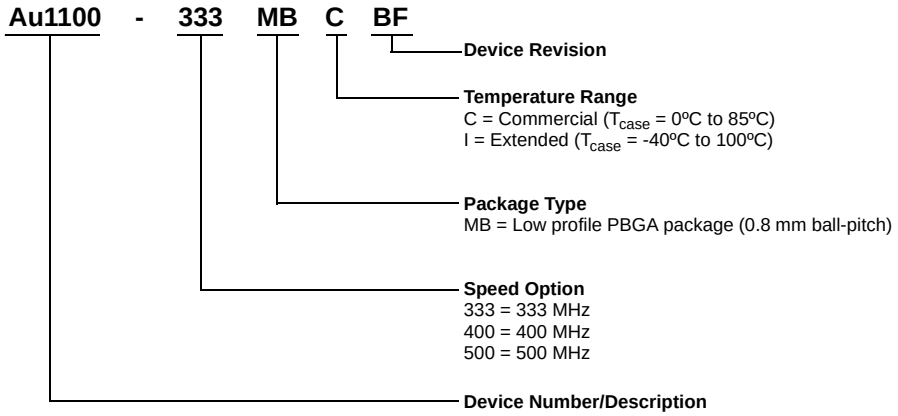
The following table replaces “Table 88. Thermal Characteristics” in the *AMD Alchemy™ Au1100™ Processor Data Book*, 30362B.

Table 88. 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}	37.8	36.3	35.3	34.4	33.2	—	ºC/Watt
Ψ_{JT}	—	—	—	—	—	5.0[a]	ºC/Watt

a. Air Flow does not apply to this specification.

5. Ordering Information, page 372



Valid Combinations

Au1100-333 Au1100-400 Au1100-500	MB	C
Au1100-333	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.

6. Table 92. Voltage and Power Parameters for 500-MHz Au1100 Part, page 334

Old Values

Parameter	Min	Typical	Max	Unit
VDDI	1.12	1.22	1.32	V

New Values

Parameter	Min	Typical	Max	Unit
VDDI	**1.14**	**1.20**	**1.30**	V

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		Change USB device endpoint 1 & 2 **receive** to **transmit** in Table 20.
2	X	X		Change first occurrence of **4096** to **4096 x 12** in *Table 110. Basic Au1100™ Processor Physical Memory Map* on page A-1.
3	X	X		Change **32-bit word** to **cache line (32 Bytes)** under *DMA Starting Address Registers* on page 91.
4	X	X		Change **MAC1** to **MAC0**, *6.5.1.3 MAC DMA Registers*, page 162.
5	X	X		Change **MAC1** to **MAC0** and add **GPIO[215]**, *6.5.2 Hardware Connections*, page 173.
6	X	X		Update section *7.2 Time of Year Clock and Real Time Clock*.
7	X	X		Change descriptions for SD Configuration Register bits 31 and 30.
8	X	X		Change description for SD Status Register bit 30.
9	X	X		Change description for LCD Control Register.
10	X	X		Change description for LCD Vertical Timing Register.
11	X	X		Change descriptions for LCD Clock Control Register.
12	X	X		For the LCD controller, correct Figure 32.
13	X	X		For the static bus controller, replace Figure 8.
14	X	X		For the static bus controller, correct the **mem_stcfg*n***[AS] description.
15	X	X		For PCMCIA, correct the $\overline{\text{ROE}}$ description.
16	X	X		Remove incorrect statement from AC97 Controller section.
17	X	X		For multiplexing on UART3, change GPIO[23] to GPIO[214].
18	X	X		Missing signal name in Figure 53, “MII Timing Interface.”
19	X	X		Correct the description of **sys_pininputen**.
20	X	X		Correct the description for the IrDA ring size register.
21	X	X		Correct the programming notes for IrDA.
22	X	X		Add exclusive transmit and receive enable note for IrDA.
23	X	X		Correct the default value of **macdma_rxaddr**[CB].

Data Book Errata

1. Change *receive* to *transmit*

Description

Table 20. Peripheral Addresses and Selectors

Incorrect

Peripheral Device	Dev ID Select	Dev ID	Xfer Size	FIFO Width	FIFO Phys Addr
USB device endpoint 1 receive	0	10	4	8	0x0 1020 0008
USB device endpoint 2 receive	0	11	4	8	0x0 1020 000C

Correct

Peripheral Device	Dev ID Select	Dev ID	Xfer Size	FIFO Width	FIFO Phys Addr
USB device endpoint 1 **transmit**	0	10	4	8	0x0 1020 0008
USB device endpoint 2 **transmit**	0	11	4	8	0x0 1020 000C

Affected Data Book Revs

A, B

Status

Fixed

2. Change *4096* to *4096 x 12*

Description

Table 110. Basic Au1100™ Processor Physical Memory Ma*p*

Incorrect

Start Address	End Address	Size (MB)	Function
0x1 F0000000	0xC FFFFFFFF	4096	Reserved

Correct

Start Address	End Address	Size (MB)	Function
0x1 F0000000	0xC FFFFFFFF	**4096 x 12**	Reserved

Affected Data Book Revs

A, B

Status

Fixed

3. Change *32-bit word* to *cache line (32 bytes*)

Description

DMA Buffer Starting Address Registers, page 91- First Paragraph, Last Sentence.

Incorrect

The starting address must be 32-bit word aligned.

Correct

The starting address must be **cache line (32 bytes)** aligned.

Affected Data Book Revs

A, B

Status

Fixed

4. Change *MAC1* to *MAC0*

Description

6.5.1.3 MAC DMA Registers, page 162 - Paragraph immediately following Table 48

Incorrect	Correct
To calculate the address of a specific MACDMA buffer all offsets should be combined. For example the physical address of the MAC1 receive buffer 3 address register is calculated as follows: macdma1_rx3addr = mac1dma_base + rx3 + addr = 0x0 1400 4200 + 0x130 + 0x4 = 0x0 1400 4334	To calculate the address of a specific MACDMA buffer all offsets should be combined. For example the physical address of the **MAC0** receive buffer 3 address register is calculated as follows: **macdma0_rx3addr** = **macdma0_base** + rx3 + addr = 0x0 1400 **4000** + 0x130 + 0x4 = 0x0 1400 **4134**

Affected Data Book Revs

A, B

Status

Fixed

5. Change *MAC1* to *MAC0* and add *GPIO[215]*

Description

6.5.2 Hardware Connections, page 173 - Last Paragraph, First Sentence

Incorrect	Correct
MAC1 shares its pins with GPIO[28:24]; those pins must be assigned to MAC1 in order to use MAC1.	**MAC0** shares its pins with GPIO[28:24] **and GPIO[215]**; those pins must be assigned to **MAC0** in order to use **MAC0**.

Affected Data Book Revs

A, B

Status

Fixed

6. Update section 7.2 Time of Year Clock and Real Time

Description

Replace Figure 37. TOY and RTC Block Diagram and 7.2.1 Time of Year Clock and Real Time Clock Registers with the following new material.

7.2 Time of Year Clock and Real Time Clock

Figure 37. shows the functional block diagram of both the TOY and the RTC. The registers used to implement the block, including the counter control register (**sys_cntrctrl**), are described in the following section.

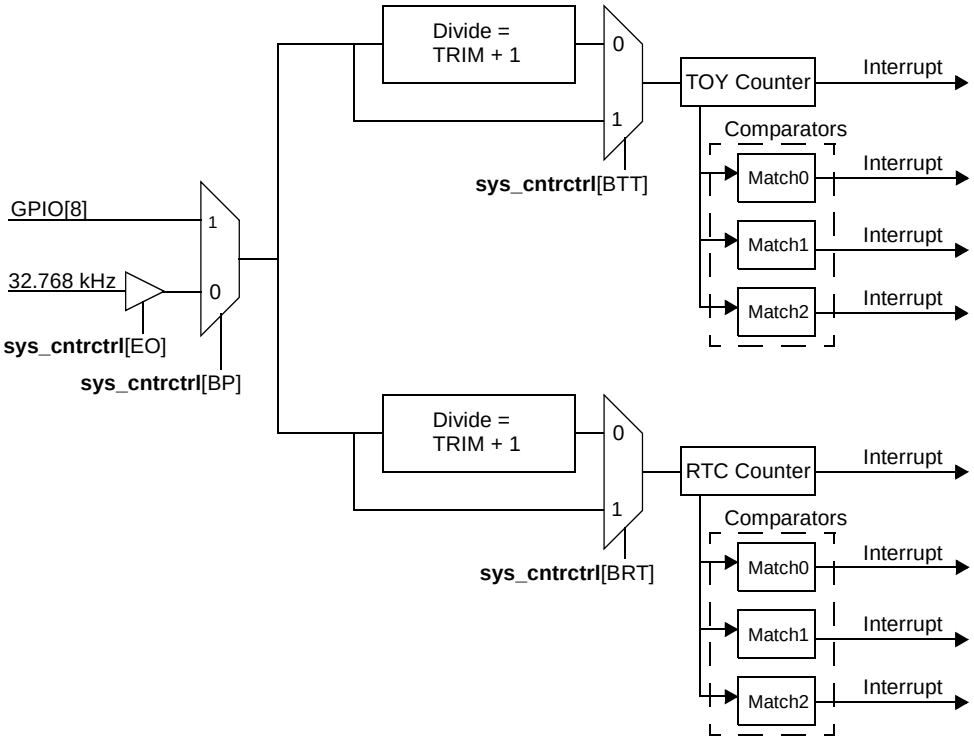


Figure 37. TOY and RTC Block Diagram

7.2.1 Time of Year Clock and Real Time Clock Registers

Counter Write

The TOY value write status bit (**sys_cntrctrl**[TS]) must be clear before writing **sys_toywrite**.

TOY and RTC Counter Control

sys_cntrctrl Offset = 0x0014

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												RTS	RM2	RM1	RM0	RS		BP		BRT		BTT		EO	CCS		32S	TTS	TM2	TM1	TM0	TS
Def.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	0	0	0	0	0

Bits	Name	Description	Read/Write	Default
31:21	—	Reserved, should be cleared.	R	0

Bits	Name	Description	Read/Write	Default
20	RTS	**sys_rtctrim** Write status **sys_rtcmatch2** write status **sys_rtcmatch1** write status **sys_rtcmatch0** write status **sys_rtcwrite** write status 0 No write is pending. It is safe to write to the register. 1 A write is pending. Do not write to the register.	R	0
19	RM2		R	0
18	RM1		R	0
17	RM0		R	0
16	RS		R	R
15	—	Reserved, should be cleared.	R	0
14	BP	Bypass the 32.768-kHz OSC 0 Select Oscillator Input (XTI32, XTO32) 1 GPIO[8] drives the counters. This is a test mode where GPIO[8] can drive the counters from an external source or through software using the GPIO controller.	R/W	0
13	—	Reserved, should be cleared.	R/W	0
12	BRT	Bypass RTC Trim 0 Normal operation 1 The RTC is driven directly by the 32.768-kHz clock, bypassing the trim.	R/W	0
11	—	Reserved, should be cleared.	R/W	0
10	BTT	Bypass TOY Trim 0 Normal operation 1 The TOY is driven directly by the 32.768-kHz clock, bypassing the trim.	R/W	0
9	—	Reserved, should be cleared.	R	0
8	EO	Enable 32.768-kHz Oscillator Enables the clock for the RTC/TOY block. 0 Disable the clock. 1 Enable the clock. Regardless of the clock source (crystal or overdriven clock through XTI32/XTO32, or bypass through GPIO[8]), the EO bit must be set to enable the RTC/TOY counters. After enabling the clock by setting EO, poll the oscillator status bit (32S) until it returns a ‘1’. Once 32S is set, wait an additional one second to allow for frequency stabilization within the block before accessing other RTC/TOY registers (not including **sys_cntrctrl**). Note: If the oscillator is being overdriven or bypassed through GPIO[8], be sure to set EO only *after* a stable clock is being driven into the part.	R/W	0
7	CCS	**sys_cntrcntrl** write status	R	0
6	—	Reserved, should be cleared.	R	0
5	32S	32.768-kHz Oscillator Status Detects two consecutive 32-kHz cycles from the clock source for the RTC/TOY block. 0 Clock is not running. 1 Clock is running. Note: Be sure to wait 1 second after 32S is set to allow for frequency stabilization within the block before accessing RTC/TOY registers.	R	UNPRED
4	TTS	**sys_toytrim** write status **sys_toymatch2** write status **sys_toymatch1** write status **sys_toymatch0** write status **sys_toywrite** write status 0 No write is pending. It is safe to write to the register. 1 A write is pending. Do not write to the register.	R	0
3	TM2		R	0
2	TM1		R	0
1	TM0		R	0
0	TS		R	0

Affected Data Book Revs

A, B

Status

Fixed

7. Change descriptions for SD Configuration Register bits 31 and 30

Description

SD Configuration Register table, page 223

Incorrect

Bits	Name	Description	Read/ Write	Default
31	SI	Slot *n* device insertion interrupt enable	R/W	0
30	CD	Slot *n* card insertion/removal detect interrupt enable	R/W	0

Correct

Bits	Name	Description	Read/ Write	Default
31	SI	SDIO device interrupt enable	R/W	0
30	CD	Card insertion/removal detect interrupt enable	R/W	0

Affected Data Book Revs

A, B

Status

Fixed

8. Change description for SD Status Register bit 30

Description

SD Status Register table, page 226

Incorrect

Bits	Name	Description	Read/ Write	Default
30	CD	SDIO card insertion/removal detect interrupt (ET)	R/W	0

Correct

Bits	Name	Description	Read/ Write	Default
30	CD	Card insertion/removal detect interrupt (ET)	R/W	0

Affected Data Book Revs

A, B

Status

Fixed

9. Change description for LCD Control Register.

Description

On page 208, add the following note to the description for **lcd_control**[BPP]:

16 bits per pixel is not supported for DSTN panels.

Affected Data Book Revs

A, B

Status

Fixed

10. Change description for LCD Vertical Timing Register.

Description

On page 212, add the following note to the description for **lcd_verttiming**[LPP]:

For DSTN panels, program LPP to be one less than half the total number of lines per panel.

Affected Data Book Revs

A, B

Status

Fixed

11. Change descriptions for LCD Clock Control Register.

Description

On page 212, change the description for **lcd_clkcontrol**[19] as follows:

Incorrect

Bits	Name	Description	R/W	Default
31:19	–	Reserved	R	0

Correct

Bits	Name	Description	R/W	Default
31:20	–	Reserved	R	0
19	PCDD	Pixel Clock Divisor Disable 0 Use the Pixel Clock Divisor as described in bits 9:0 1 Pixel Clock frequency = LCD Clock frequency	R/W	0

On page 213, add the following note to the description for **lcd_clkcontrol**[PCD]:

Note: To run the pixel clock at the same frequency as the LCD clock, disable the divisor by setting PCDD (bit 19).

Affected Data Book Revs

A, B

Status

Fixed

12. For the LCD controller, correct Figure 32, “TFT (Active Mode) Timing.”

Description

On page 220, correct Figure 32, “TFT (Active Mode) Timing,” as follows:

Incorrect

lcd_clkcontrol[IV:IH:IC:IO]

Correct

lcd_clkcontrol[IB:IC:IH:IV]

Affected Data Book Revs

A, B

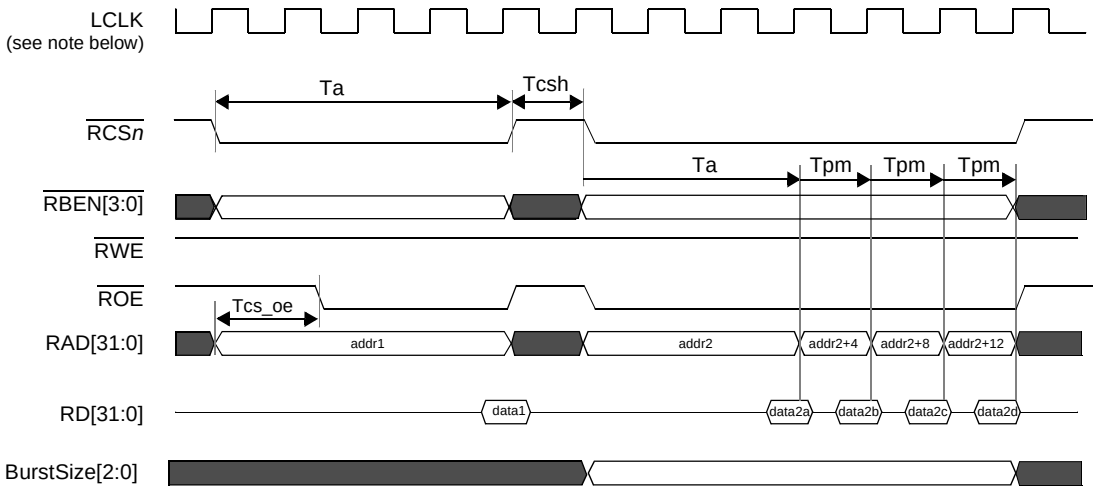
Status

Fixed

13. For the static bus controller, replace Figure 8.

Description

On page 73, replace Figure 8 with the following:



NOTE: The external clock LCLK is available only in synchronous mode (**mem_stcfg**[S]=1). For *asynchronous* chip selects, the timing parameters are based on a separate internal-only clock.

Figure 8. Static Memory Read Timing (Single Read Followed by Burst)

Affected Data Book Revs

A, B

Status

Fixed

14. For the static bus controller, correct the mem_stcfg*n*[AS] description.

Description

On page 65, change the description for **mem_stcfg**[AS] as follows:

Incorrect	Correct
Setup address before **chip select** on reads.	Setup address before **output enable** on reads.The setup duration is programmed in **mem_sttime*n***[T1, T0] and is shown as Tcs_oe in Figure 8. 0 Do not setup address. 1 Setup address.

Affected Data Book Revs

A, B

Status

Fixed

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

Description

On page 76 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

16. Remove incorrect statement in AC97 Controller section.

Description

The last sentence in section “6.1 AC97 Controller” on page 104 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

17. For multiplexing on UART3, change GPIO[23] to GPIO[214].

Description

7.3.3 Hardware Considerations, page 266 - 2nd Paragraph, change GPIO[23] to GPIO[214]:

Incorrect	Correct
For example, if sys_pinfunc[U3] is cleared configuring the pin as U3TXD, GPIO[23] can not be used as a GPIO nor can the GPIO be configured as an interrupt. Conversely if sys_pinfunc[U3] is set configuring the pin as GPIO[23], U3TXD (and thus the UART3 interface) is not usable. GPIO[23] can be used as a GPIO and to generate interrupts.	For example, if sys_pinfunc[U3] is cleared configuring the pin as U3TXD, **GPIO[214]** cannot be used as a GPIO nor can the GPIO be configured as an interrupt. Conversely if sys_pinfunc[U3] is set configuring the pin as **GPIO[214]**, U3TXD (and thus the UART3 interface) is not usable. **GPIO[214]** can be used as a GPIO and to generate interrupts.

Affected Data Book Revs

A, B

Status

Fixed

18. Missing signal name in Figure 53, “MII Interface Timing.”

Description

On page 344 in Figure 53, add the N0DIO signal name to the second waveform from the bottom.

Affected Data Book Revs

A, B

Status

Fixed

19. Correct the description of sys_pininputen.

Description

When describing **sys_pininputen**, the data book (on pages 264, 265, 266, and 271) states that *clearing* the register enables GPIO[31:0] to be used as inputs. Modify the descriptions to state that *any write* to **sys_pininputen** (0 or 1) is sufficient.

Affected Data Book Revs

A, B

Status

Fixed

20. Correct the description for the IrDA ring size register.

Description

On page 131, the description for **ir_ringsize**[TRBS, RRBS] is not correct. The choices for ring size should refer to *entries*, not *bytes*.

Affected Data Book Revs

A, B

Status

Fixed

21. Correct the programming notes for IrDA.

Description

In Table 41 (Fast Infrared Mode), for step 10 on page 141, the notes read "Set bit E to enable the peripheral, then read register again for correct status (should equal 0xC7FF)." This should read "Set bit E to enable the peripheral, then read register again for correct status (should equal **0xA6FF)."**

In Table 42 (Medium Infrared Mode), for step 10 on page 141, the notes read "Set bit E to enable the peripheral, then read register again for correct status (should equal 0xA7FF)." This should read "Set bit E to enable the peripheral, then read register again for correct status (should equal **0x96FF**)."

In Table 43 (Slow Infrared Mode), for step 10 on page 142, the notes read "Set bit E to enable the peripheral, then read register again for correct status (should equal 0xA7FF)." This should read "Set bit E to enable the peripheral, then read register again for correct status (should equal **0x8EFF)."**

Affected Data Book Revs

A, B

Status

Fixed

22. Add exclusive transmit and receive enable note for IrDA.

Description

On page 133, add the following note to the TE (transmit-enable) and RE (receive-enable) bit descriptions

Unless in loopback mode, only one transfer direction (transmit or receive) can be enabled at one time.

Affected Data Book Revs

A, B

Status

Fixed

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

Description

On page 167, 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