Specification Update

AMD Alchemy™ Au1000™ Processor

Document Revision History

Date	Revision	Description
	1	Early version
November 2002	2	Added issue #19-36, silicon stepping HC updates, and Specification changes
December 2002	A	Added Errata #36-38, reformatted Specification changes, Assigned Publication ID (PID) 27348A
March 2003	B	• Deleted Errata on mac0_enable and mac1_enable. The data book was changed to show these registers as write only. • Errata #29 modified • Errata #37 and #38 expanded • Errata #39 through #46 added • Chip ordering information added to Specification Changes • Modified *Table 82 DC Parameters* added to Specification Changes
June 2003	C	• Updated VDDX rise time in Specification Changes section
September 2003	D	• Specification Changes added: — Tcase Specification — Thermal Characteristics
June 2005	E	• Device Errata #47-51 added • Device Errata #44, reference to other Errata corrected • Specification Change: AC97 variable rate data sampling supported by HC and later silicon. • Data Book Errata #1-10 added

The AMD Alchemy™ Au1000™ 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.

Permanent *Specification Changes* remain in this Specification Update until the appropriate changes are made to the published product specification or documentation.

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

To establish Specification Update information, we test 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 1.1	DA	0x00030100	0x003E828F
Rev 2.1	HA	0x00030201	0x1000228F
Rev 2.2	HB	0x00030202	0x2000228F
Rev 2.3	HC	0x00030203	0x3000228F

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
	DA	HA	HB	HC		
1	X					Ethernet MAC transmit packets must have data payload with a minimum size of 46 bytes.
2	X					Ethernet MDC clock will exceed the 2.5MHz maximum of the IEEE802.3 specification when running Au1000™ Processor at high speeds.
3	X					Coprocessor 0 Config1 register misreports the number of sets in both I-Cache and D-Cache.
4	X					LCD interface byte strobes are incorrect when processor is in big endian mode for byte access.
5	X					Reads of TOY1 Match1 or Match2 registers returns the wrong value.
6	X					USB OHCI control structures swapped when processor is in big endian mode.
7	X					Snooped reads return incorrect data if immediately following a data cache fill at the same address.
8	X					Core PLL powers up at other than the default 192 MHz.
9	X					Operating at Vddi below 1.4 Volts produces unpredictable results on SDRAM pins.
10	X	X	X			Static bus configured for 16 Bit will always access lower 16 Bits of 32-bit system data bus if RAD[1]=0.
11	X	X	X			Static bus does not support devices that do not use byte enables.
12	X	X				USB host controller is not properly reset by the USB Host Enable register.
13	X	X				USB host controller does not properly reset ports.
14	X	X				USB host HccaDoneHead register may become corrupted.
15	X	X	X	X		Ethernet minimum inter-frame gap between transmitted packets exceeds 960 nSec specification minimum at 100 Mbps.
16	X					Discarded load miss return data incorrectly bypassed into subsequent zero displacement load/store base address.
17	X	X				Discarded load miss return data incorrectly bypassed into subsequent instruction causing corrupted register value.
18	X	X	X			Incorrect phase relationship between I2SWORD and I2SCLK could cause I2S data shift.
19	X	X	X			When simultaneously enabling 2 buffers in a DMA channel, the DMA mode configuration register can become inadvertently corrupted.
20	X	X	X			The OD bit in the Config0 register is write only.
21	X	X	X			All registers in the IrDA peripheral may not return the correct value after a single read.
22	X	X	X	X		The USB Host Enable register (0xB017FFFC) may not return the correct value after a single read.
23	X	X	X			The USB Host will not operate at all frequencies.
24	X	X	X			The USB Device may replicate data in stream during out transactions.
25	X	X	X			Command Pending bit in the AC97 status register may return wrong value during a read command.
26	X	X	X	X		PCMCIA reads incorrect byte lane on odd byte transfers from 8-bit cards.

Errata Number	Silicon Stepping					Errata
	DA	HA	HB	HC		
27	X	X	X	X		A duplicate store operation can occur when the internal Write Buffer is full.
28	X	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.
29	X	X	X	X		The USB host controller will generate corrupted packets when attempting to transmit isochronous packets larger than 67 bytes.
30	X	X	X	X		Programmable Counter Control Register can become unwritable.
31	X	X	X	X		Static bus configured as 16-bit bus drives all 32 bits.
32	X	X	X	X		I^2S always outputs I^2S mode regardless the mode specified by configuration register.
33	X	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.
34	X	X	X	X		DMA Configuration Register Write and DMA channel post collision can result in bad data in Configuration Register.
35	X	X	X	X		The AC97 CODEC command response is only valid for one frame time after a read request.
36	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.
37	X	X	X	X		USB device interrupt latency may cause status register content to be lost.
38	X	X	X	X		Under certain conditions, the USB host will write an extra byte to system memory.
39	X	X	X	X		USB host disconnect event from the root hub while a device is transmitting can hang USB host controller.
40	X	X	X	X		USB software reset during USB Host master transaction can corrupt master transfer.
41	X	X	X	X		USB host receiving two back-to-back zero length IN packets can treat second packet incorrectly as a DataOverrun Error.
42	X	X	X	X		IN transactions when interleaved with certain control IN transactions to the USB device will receive incorrect data.
43	X	X	X	X		USB device returns additional byte on Bulk IN transfers, if a particular sequence occurs.
44	X	X	X	X		USB device changes, workarounds and software requirements for reliable operation.
45	X	X	X	X		EJTAG debug exception return executing from cacheable space can hang processor.
46	X	X	X			Automatic hardware flow control is only available in silicon revision 2.3 (silicon stepping HC) and later.
47	X	X	X	X		Ethernet MACs cannot detect underflow.
48	X	X	X	X		Peripheral cache snoop read may return erroneous data.
49	X	X	X	X		Ethernet MACs may drop multicast hash filtered packets on receive.
50	X	X	X	X		After an asynchronous debug exception (DINT), the DEPC may point to the incorrect address during an eret.
51	X	X	X	X		System bus masters (USB host, MAC, IrDA, DMA) may receive stale data.

Device Errata

Note: Unless stated otherwise, all steps listed in a Workaround are required.

1. Ethernet MAC transmit packets must have data payload with a minimum size of 46 bytes.

Description

Short Ethernet DMA transfer packets will not initiate transmit FIFO draining. Hence, the Ethernet MAC will not transmit and DMA will stall if a transmit packet of less than 46 bytes is placed into the transmit FIFO.

Affected Step

DA

Workaround

Software should zero pad short transmit packets such that data payload length is at least 46 bytes. This will generate a minimum package size of 64 bytes.

Status

Fixed

2. Ethernet MDC clock will exceed the 2.5MHz maximum of the IEEE802.3 specification when running Au1000™ Processor at high speeds.

Description

MDC clock may exceed the maximum frequency allowed and prevent correct operation of MII Management Functions.

Affected Step

DA

Workaround

Core frequency and/or system bus divider should be set such that the MDC clock frequency will not exceed the maximum allowed by the specific PHY chip in use when performing MII Management Functions. Generally this will be at initialization only. Core frequency and system bus divider can then be adjusted without impacting MAC operation. MDC Clock frequency can be determined by SysClk/26.

Status

Fixed

3. Coprocessor 0 Config1 register misreports the number of sets in both I-Cache and D-Cache.

Description

Reads of Coprocessor 0 Config1 Register incorrectly reports the number of sets for each cache as 256 rather than the correct 128 for each. The result is that the caches appear twice their size to software.

Affected Step

DA

Workaround

Hardcode the CONFIG1 IS and DS bit fields to be 128.

Status

Fixed

4. LCD interface byte strobes are incorrect when processor is in big endian mode for byte access.

Description

LRDn[1:0] and LWRn[1:0] bits are reversed for byte access when the processor is in big endian mode. Hence, byte reads and writes to LCD controller registers or frame buffer appear to be to the wrong address. Halfword accesses behave correctly.

Affected Step

DA

Workaround

Operate processor in little endian mode.

Status

Fixed

5. Reads of TOY1 Match1 or Match2 registers returns the wrong value.

Description

The READ datapath for TOY1 Match Registers is incorrect. Attempting to Read TOY1 Match1 Register returns the value of Match2 Register and attempting to Read TOY1 Match2 Register returns the value of Match1 Register. The WRITE datapath and compare logic use the correct registers.

Affected Step

DA

Workaround

To READ Match1 register use the address of Match2 Register and to READ Match2 register use the address of Match1 Register.

Status

Fixed

6. USB OHCI control structures swapped when processor is in big endian mode.

Description

The USB Host internal DMA engine only supports little endian mode transfers.

Affected Step

DA

Workaround

Operate processor in little endian mode.

Status

Fixed

7. Snooped reads return incorrect data if immediately following a data cache fill at the same address.

Description

If a Snoop Read occurs as first System Bus operation following a Data Cache Fill and the Snoop address matches that of the cache fill, then the processor core responds shared but puts unpredictable data on the System Bus in response to the Snoop Read.

Affected Step

DA

Workaround

None

Status

Fixed

8. Core PLL powers up at other than the default 192 MHz.

Description

The Core PLL frequency may exceed the maximum operating frequency of the processor core at initial power-on. This will prevent the core from correctly fetching and issuing the proper boot sequence to set the processor speed.

Affected Step

DA

Workaround

Use a Test Mode for setting the PLL frequency independent of the cpu_pll_control Register.

1. Set TC[3:0]=1000 prior to asserting VDDX_OK.
2. Use GPIO[14:9] pins (instead of cpu_pll_control[5:0] register) to control core pll multiplier.
3. TC pins should be returned to GND after asserting VDDX_OK.

Status

Fixed

9. Operating at Vddi below 1.4 volts produces unpredictable results on SDRAM pins.

Description

Error in the voltage translation circuit for SDRAM pins prevents driving address, data, or control pins to SDRAM if Vddi is below 1.4 Volts.

Affected Step

DA

Workaround

Operate Au1000™ processor with Vddi voltage at 1.5V or above.

Status

Fixed

10. Static bus configured for 16 bit will always access lower 16 Bits of a 32-bit system data bus if RAD[1]=0.

Description

When configured for 16 bit mode the static bus always performs two 16 bit accesses. For Reads, 16 bits of data are fetched from memory with RAD[1]=0 and placed on the lower 16 bits of the internal system bus, followed by additional 16 bits fetched with RAD[1]=1 and placed on the upper 16 bits of the system bus. The requestor then extracts the correct data from the system bus. For Writes, the data on the lower 16 bits of the system bus is presented to memory with RAD[1]=0 and the data on the upper 16 bits of the system bus is presented with RAD[1]=1. Byte Enables are only asserted for the valid write data. For RAM accesses this will have no affect since reads and writes occur in a consistent manner. For preprogrammed Flash or ROM access with systems operating in Big Endian mode, the wrong halfword will be accessed on halfword accesses unless the memory is programmed as described below.

Affected Step

DA, HA, HB

Workaround

1. For Little Endian systems: All memory accesses should be 32-bit accesses until the boot code has set up the processor for little endian mode. That is, software should not execute Load Byte, Load Halfword, Store Byte, or Store Halfword instructions until after programming the BE bit of Config0 Register.

2. For Big Endian systems: Data should be halfword swapped before programming ROM or Flash. Example code is available from AMD to perform this task. An alternative solution for Big Endian systems is for the system designer to invert address line RAD[1] that is routed to flash or ROM. With this fix, all data can be stored in flash in a standard big endian format.

Status

Fixed

11. Static bus does not support devices which do not use byte enables.

Description

When the Au1000™ processor performs 16-bit halfword writes while the Static Bus is programmed for 16 Bit Mode, two 16-bit halfword transactions are generated with the byte enables, BE[3:0], deasserted for the halfword access which should not take place. Write Enable, WE, remains asserted for both transactions. This can cause problems for devices that do not support byte enables, e.g., flash. In Flash, the second access could corrupt the write protocol.

Affected Step

DA, HA, HB

Workaround

System design can logically 'OR' RWE and RBE0 to form the write enable to be used for 16-bit flash. The system designer must be aware of Errata 11 as this could also affect 16-bit flash programming.

Status

Fixed

12. USB host controller is not properly reset by the USB Host Enable register.

Description

Clearing the USB_HOST_ENABLE enable bit (usbh_enable[2]) does not reset all of the host controller logic. This may result in erroneous USB host controller memory accesses. Communication to connected USB devices while the controller is expected to be in reset may also result.

Affected Step

DA, HA

Workaround

1. Initialization of Disabled controller
   Set usbh_enable register CE bit
   Set usbh_enable register E bit
   Clear HcControl register HCFS bits
   Wait for usbh_enable register RD bit to set
2. Shutdown of Running Controller
   Clear HcControl register HCFS bits
   Clear usbh_enable register E bit
   Clear usbh_enable register CE bit
3. Reinitialize Running Controller
   Clear HcControl register HCFS bits
   Clear usbh_enable register E bit
   Set usbh_enable register E bit

Status

Fixed

13. USB host controller does not properly reset ports.

Description

When writing a "1" to SetPortReset (HcRhPortStatus register, PRS bit) to reset the host port, the required 10 mSec reset hold time is not generated. This may result in unexpected operation of connected devices.

Affected Step

DA, HA

Workaround

Software must not issue SetPortReset commands but instead should perform a complete Host Controller Reset as described in Errata 12 above. This will reset both ports.

Status

Fixed

14. USB host HccaDoneHead register may become corrupted.

Description

Asynchronous DMA FIFO transition can prevent proper completion of USB transaction causing the HcPeriodCurrentED register to update in the wrong cycle and latch incorrect data. The error propagates until the HccaDoneHead value is corrupted resulting in erroneous addresses being used when processing Transfer Descriptors.

Affected Step

DA, HA

Workaround

Operate the Core PLL at even multiple (sys_cpupll register value is even) and program clock generator block such that USB Host Clock is derived from the cpu_pll.

Status

Fixed

15. Ethernet minimum inter-frame gap between transmitted packets exceeds 960 nSec specification minimum at 100 Mbps.

Description

For transmit packets at 100Mbps rate, the minimum inter-frame gap for the Au1000™ processor is either 1000ns in full-duplex or 1080ns in half-duplex. However, the Au1000™ processor is capable of receiving packets at the full frame rate (minimum inter-frame gap of 960ns). Hence, for applications that receive at the full frame rate and re-transmit all packets, an eventual buffer memory overrun could occur. In general, it is highly unlikely that one could achieve 100% receive saturation indefinitely outside of a controlled test environment.

Affected Step

DA, HA, HB, HC, HD

Workaround

Limit input bandwidth to be less than 96% of the maximum if all packets are retransmitted. One way to ensure this is to use 1000ns inter-frame gap for full duplex on all network packets.

Status

Not Fixed

16. Discarded load miss return data incorrectly bypassed into subsequent zero displacement load/store base address.

Description

We define a Discarded Load Miss as a case where a load instruction misses in the D-Cache and the target register is subsequently overwritten by some other instruction prior to the load miss data return. In this case the return load data is discarded and not written to the register file. However, a one-cycle window exists where this discarded data may be bypassed into a zero displacement load or store base address resulting in incorrect memory address calculation. The following instruction sequence is capable of exhibiting this problem:

LW Rx, offset(base) (cache miss with no outstanding miss)

… some number of intervening instructions (1)

ADD Rx,rs,rt (can be any instruction with Rx as destination)

… one instruction (2)

LW rt, 0(Rx) (can be any Load or Store with zero displacement

which uses Rx as base address register)

If the number of instructions in sequence (1) is such that the load data is returned while the second load is in the pipeline Issue stage, the target register, rt, of the second load will be corrupted.

If either instruction sequence (1) or (2) meet any one of the following conditions, this bypass bug will not occur:

- Causes a cache miss
- Causes a pipeline synchronization
- Use of register Rx

Affected Step

DA

Workaround

Prior to issue of zero displacement load or store instructions, use the base register as a register source. This can be a NOP of the form:

OR r0,Rx,Rx

This forces a dependency stall on Rx prior to the load and will prevent corruption of the load or store target.

Status

Fixed

17. Discarded load miss return data incorrectly bypassed into subsequent instruction causing corrupted register value.

Description

We define a Discarded Load Miss as a case where a load instruction misses in the D-Cache and the target register is subsequently overwritten by some other instruction prior to the load miss data return. In this case the return load data is discarded and not written to the register file. If the load miss data is returned just as an instruction is updating the same target register the correct value is written into the register file but the load miss data is incorrectly forwarded to the following instruction. The following instruction sequence is capable of exhibiting this problem:

LW Rx, offset(base) (cache miss with no outstanding miss)

… some number of intervening instructions (1)

ADD Rx,rs,rt (can be any instruction with Rx as destination)

ADD rd,Rx,rt (can be any instruction using Rx as rs or rt)

If the number of instructions in sequence (1) is such that the load data is returned while the first add is in the pipeline Issue stage, Rx is correctly updated in the register file but the load miss data is incorrectly bypassed into the second add causing the destination register, rd, of the second add to be corrupted.

If instruction sequence (1) meets any one of the following conditions, this bypass bug will not occur:

- Causes a cache miss

- Causes a pipeline synchronization

- Use of register Rx without destination of Rx

A more concise way to force the proper instruction alignment is to both use and update the load target register in the same instruction as in the following sequence:

LW Rx, offset(base) (cache miss with no outstanding miss)

ADD Rx,Rx,rt (any instruction with Rx as destination and source)

ADD rd,Rx,rt (can be any instruction using Rx as rs or rt)

Again, Rx is correctly updated in the register file but the load miss data is incorrectly bypassed into the second add causing the destination register, rd, of the second add to be corrupted.

Affected Step

DA, HA

Workaround

Under evaluation

Status

Fixed

18. Incorrect phase relationship between I2SWORD and I2SCLK could cause I2S data shift.

Description

I2SWORD 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

DA, HA, HB

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 Au1000™ 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

19. 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

DA, HA, HB

Workaround

Two workarounds exist for this problem.

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. It should be noted that the OD bit is write only and will be read as a zero. (see #21 below).

Status

Fixed

20. 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

DA, HA, HB

Workaround

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

Status

Fixed

21. All registers in the IrDA peripheral may not return the correct value after a single read.

Description

All registers in the IrDA peripheral may not return the correct value after a single read.

Affected Step

DA, HA, HB

Workaround

When reading any IrDA register, it should be read twice, back to back, to insure that the correct value is read.

Status

Fixed

22. 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

DA, HA, HB, HC, HD

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

23. The USB Host will not operate at all frequencies.

Description

When brought up at frequencies other than those stated below, and without the specific bringup sequence, the USB may function unpredictably.

Affected Step

DA, HA, HB

Workaround

The USB host can only be run with a CPU frequency of 384MHz. The workaround has been propagated into the supported operating systems. The specific code for the workaround can be obtained from AMD technical support. Support for more frequencies are under evaluation.

Status

Fixed

24. The USB Device may replicate data in stream during out transactions.

Description

The USB Device may replicate data in stream during out transactions. This may cause unpredictable results when using the USB device.

Affected Step

DA, HA, HB

Workaround

Under Evaluation

Status

Fixed

25. Command Pending bit in the AC97 status register may return wrong value during a read command.

Description

Command Pending bit in the AC97 status register may return the wrong value during a read command.

Affected Step

DA, HA, HB

Workaround

On a read command from the AC97 CODEC, the AC97 command pending bit in the AC97 status register must have the same value twice in a row before being considered valid.

Status

Fixed

26. 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

DA, HA, HB, HC

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™ Processor 8-bit PCMCIA Host Adaptor”.

Status

Not Fixed

27. 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

DA, HA, HB, HC

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 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

Not Fixed

28. 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 Au1000™ 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

DA, HA, HB, HC

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

Not Fixed

29. 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

DA, HA, HB, HC

Workaround

None

Status

Not Fixed

30. 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

DA, HA, HB, HC

Workaround

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

Status

Not Fixed

31. 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

DA, HA, HB, HC, HD

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

Not Fixed

32. I^2S always outputs I^2S mode regardless of the mode specified by the configuration register.

Description

I^2S always outputs in I^2S mode, regardless of the mode specified for it in the configuration register.

Affected Step

DA, HA, HB, HC, HD

Workaround

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

Status

Not Fixed

33. 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

DA, HA, HB, HC

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.

Status

Not Fixed

34. 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

DA, HA, HB, HC

Workaround

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

Status

Not Fixed

35. 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

DA, HA, HB, HC, HD

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

36. 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

DA, HA, HB, HC, HD

Workaround

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

Status

Not Fixed

37. 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 Au1000™ 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

DA, HA, HB, HC

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

Not Fixed

38. 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

DA, HA, HB, HC

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

Not Fixed

39. 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

DA, HA, HB, HC, HD

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

Not Fixed

40. 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 state machines 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

DA, HA, HB, HC, HD

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

Not Fixed

41. 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

DA, HA, HB, HC, HD

Workaround

Set software to ignore DataOverRuns caused by Zero length packets.

Status

Not Fixed

42. 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

DA, HA, HB, HC

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

Not Fixed

43. 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

DA, HA, HB, HC

Workaround

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

Status

Not Fixed

44. 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, Windows® CE, 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 #37], 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

DA, HA, HB, HC

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

45. 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

DA, HA, HB, HC

Workaround

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

Status

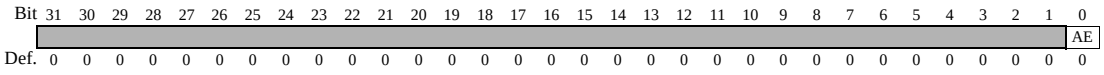
Not Fixed

46. Automatic hardware control is only available in silicon revision 2.3 (silicon stepping HC) 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

DA, HA, HB

Workaround

None

Status

Fixed

47. 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

DA, HA, HB, HC, HD

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

48. 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

DA, HA, HB, HC

Workaround

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

Status

Not Fixed

49. 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

DA, HA, HB, HC

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

50. 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

DA, HA, HB, HC

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

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

Description

System bus masters (USB host controller, MAC0, MAC1, 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 with reference to the June 2003, 30360B issue of the *AMD Alchemy™ Au1000™ Processor Data Book*.

1. DC Parameters – Add Minimum Vddx Rise Time of 200µS

An extremely fast turn-on of Vddx may cause the Vddx ESD protection devices to turn on.

2. Table 78, DC Parameters, pages 297-298

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.

Old Values

Param	Description	Min	Nom	Max	Unit
T_{case}	Package operating temperature			TBD	ºC
T_a for Au1000-266MCI	Operating temperature (ambient)	-40		85	ºC
T_a for Au1000-266MCC & Au1000-400MCC		0		70	ºC
T_a for Au1000-500MCC		0		60	ºC

New Values

Param	Description	Min	Nom	Max	Unit
T_{case} for Au1000-266MCC, Au1000-400MCC & Au1000-500MCC	Package operating temperature	0		85	ºC
T_{case} for Au1000-266MCI		-40		100	ºC

3. Table 88. Thermal Characteristics, page 297

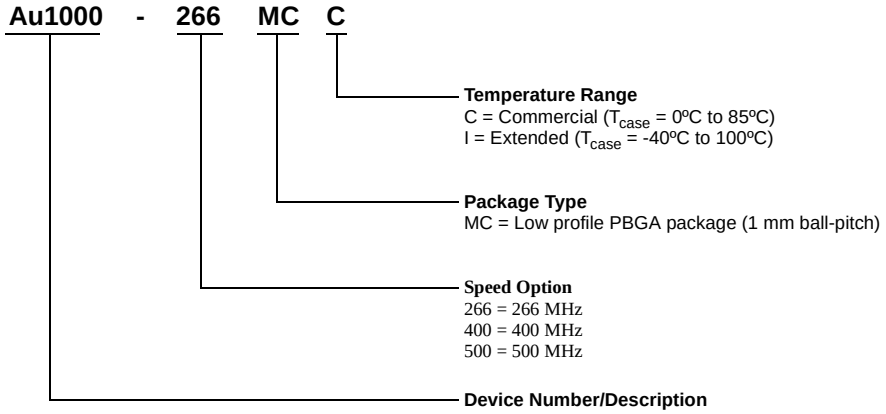
The following table replaces “Table 77. Thermal Characteristics” in the *AMD Alchemy™ Au1000™ Processor Data Book*, 30360B.

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}	28.0	27.1	26.4	25.7	25.1	—	ºC/Watt
Ψ_{JT}	—	—	—	—	—	4.4[a]	ºC/Watt

a. Air Flow does not apply to this specification.

4. Ordering Information, page 336



Valid Combinations

Au1000-266 Au1000-400 Au1000-500	MC	C
Au1000-266	MC	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.

5. AC97 Controller, page 104

Replace the last sentence in section “6.1 AC97 Controller” with the following paragraph:

“All data being sent and received through the AC97 controller must be 48kHz when using HB silicon and earlier. Variable rate data sampling is supported by HC and later silicon.”

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
	30360A	30360B	30360C	
1	X	X		Static Bus Configuration Register bit 6 should be cleared for LCD device type.
2	X	X		Missing signal name in Figure 49, “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 4.2, “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 description for the IrDA ring size register.
8	X	X		Correct the programming notes for IrDA.
9	X	X		Add exclusive transmit and receive enable note for IrDA.
10	X	X		Correct the default value of **macdma_rxaddr**[CB].

Data Book Errata

1. Static Bus Configuration Register bit 6 should be cleared for LCD device type

Description

Static Bus Configuration Registers table, page 65

Incorrect

Bits	Name	Description	R/W	Default
6	H	Half Bus Selects the data bus width for the chip select. 0 32-bit bus 1 16-bit bus using bits 15:0 of the data bus. For PCMCIA device type, clear this bit. For LCD device type, set this bit.	R/W	0, except for **mem_stcfg0** where the default value is determined by ROMSEL and ROMSIZE out of reset. See Table 68

Correct

Bits	Name	Description	R/W	Default
6	H	Half Bus Selects the data bus width for the chip select. 0 32-bit bus 1 16-bit bus using bits 15:0 of the data bus. For PCMCIA device type, clear this bit. For LCD device type, **clear** this bit.	R/W	0, except for **mem_stcfg0** where the default value is determined by ROMSEL and ROMSIZE out of reset. See Table 68

Affected Data Book Revs

A, B

Status

Fixed

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

Description

On page 309 in Figure 49, 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 75 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 74, 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 4.2, “Using GPIO as External DMA Request (DMA_REQ*n*).”

Description

On page 92 in Section 4.2, 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 187 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 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

8. 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

9. 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

10. 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