Errata: EP9301 - Silicon revision: E1

Reference EP9301 Data Sheet revision DS636PP5 dated March 2005.

Determining the Silicon Revision of the Integrated Circuit

On the front of the integrated circuit, directly under the part number, is an alpha-numeric line. Characters 5 and 6 in this line represent the silicon revision of the chip. For example, this line indicates that the chip is a "E1" revision chip:

EFWA**E1**AM0340

This Errata is applicable only to the E1 revision of the chip.

Please refer to AN273, *"EP93xx Rev E0 & E1 Design Guidelines"* for additional information.

AC'97

Description

Disabling audio transmit by clearing the TEN bit in one of the AC97TXCRx registers will not clear out any remaining bytes in the TX FIFO. If the number of bytes left in the FIFO is not equal to a whole sample or samples, this will throw off subsequent audio playback causing distortion or channel swapping.

Workaround

To stop audio playback, do the following:

1) Pause DMA

2) Poll the AC97SRx register until either TXUE or TXFE is set.

3) Clear the TEN bit.

This ensures that the TX FIFO is empty before the transmit channel is disabled.

Ethernet

Description 1

The Ethernet controller does not correctly receive frames that have a size of 64 bytes.

Workaround

In order to receive frames of 64 bytes, enable the RCRCA bit in RxCTL. This will prevent the Ethernet controller from discarding the 64-byte-long frames.

Description 2

When there is inadequate AHB bus bandwidth for data to be transferred from the Ethernet controller FIFO to the receive descriptor, the Ethernet FIFO will overflow and cause the Ethernet controller to fail to receive any more packets.

This problem will also occur if the processor is too busy to service incoming packets in a timely manner. By the time that new receive descriptors are available, the data in the FIFO will contain frames that are corrupted.

It is the job of the system designer to ensure that there is adequate bandwidth for the applications being run.

Workaround

This is a rare occurrence, however at a system level it is important to reserve adequate bandwidth for the Ethernet controller. This can be accomplished by some of the following:

- Reducing the bandwidth use of other bus masters in the system.
- Lowering Ethernet rate to half duplex or 10Mbit if higher bandwidth is not required.
- **E**nsuring that the Ethernet controller receive descriptor processing is given a high enough priority to ensure that the controller never runs out of receive descriptors.

HDLC

Description

When the final byte of a received packet is read into the DMA controller's buffer, the software will be notified by an HDLC RFC interrupt. However, the DMA controller may not have written the currently buffered part of the packet to memory, so that the last one to fifteen bytes of a packet may not be accessible.

Workaround

To insure that the DMA channel empties the buffer, do the following (in the HDLC interrupt handler, for example):

1) Note the values in the MAXCNTx and REMAIN registers for the DMA channel. The difference is the number of bytes read from the UART/HDLC, which is the size of the HDLC packet. Call this number N. Note that the BC field of the UART1HDLCRXInfoBuf register should also be N.

2) Temporarily disable the UART DMA RX interface by clearing the RXDMAE bit in the UART1DMACtrl register.

3) Wait until the difference between the CURRENTx and BASEx registers in the DMA channel is equal to N + 1.

At this point, the rest of the packet is guaranteed to have been written to memory. Using this method will cause an extra byte to be read from the UART by the DMA channel and also written to memory. This last byte should be ignored.

SDRAM Controller

Description 1

Using the SDRAM controller in auto-precharge mode will produce system instability at external bus speeds greater than 50MHz.

Workaround

Do not turn on the auto-precharge feature of the SDRAM controller if the external bus speed will be greater than 50 MHz.

Description 2

When the SDRAM controller is configured for PRECHARGE ALL command, the actual sequence is not always issued to the SDRAM device(s).

Workaround

Do a read from each SDRAM bank so that a PRECHARGE command is issued to each bank of the SDRAM device. This will satisfy the required SDRAM initialization sequence.

Due to the effectiveness and simplicity of the software workaround, no silicon fix is planned.

RTC

Description

The internal RTC oscillator is susceptible to noise which can lead to extra clocks on the internal 32.768-kHz signal.

Workaround

Please refer to application note AN265, *“EP93xx RTC Oscillator Circuit”*, which can be found at http://www.cirrus.com/en/pubs/appNote/AN265REV2.pdf

No fix of this bug is planned for future silicon revisions.

Software Reset

Description

When configured for Sync Boot mode, the EP93xx device may lock up during the boot process after a watchdog reset has been issued.

Workaround

Boot the device in Async Boot mode.

USB

Description 1

Outgoing USB DMA transfers may be corrupted if the data buffer is not quad word aligned or if the data buffer length is not an integer number of quad words. USB quad word and single word transfers are not affected.

Workaround

Make the transmit buffers used by the USB device aligned on a quad word boundary, i.e. the start address ends in a 0x0 nibble, and make the buffer length an integer number of quad words, i.e. the length in bytes ends in a 0x0 nibble. This can be achieved by padding the transmit buffer structure by the appropriate number of bytes.

Incoming USB data is not affected.

A fix for this bug has been implemented for silicon revision E2.

Description 2

USB clock divider logic operates at a maximum rate of 288MHz under worst case conditions.

Workaround

When using USB, make sure the clock frequency supplied to the USB clock divider does not exceed 288MHz. The clock supplied to the USB clock divider is sourced by PLL2. The USB clock divider is controlled by the USBDIV setting in the CLKSET2 register. For example, configure PLL2 to output 192MHz and USBDIV to divide by four. Ensure that the new PLL2 setting does not adversely affect any other block using PLL2 as its clock source.

***NOTE:** PLL2 is completely functional. This is only an issue with the USB clock divider logic.*

Reset

Description

SDRAM controller gets stuck in a busy state after resetting in sync mode. When this condition occurs the processor will not complete its internal boot sequence.

Workaround

Implement the recommended external reset circuit described in AN258, "EP93xx Power-up and Reset Lockup Workaround", which can be found at http://www.cirrus.com/en/pubs/appNote/AN258REV2.pdf

A fix for this bug has been implemented for silicon revision E2.

ExtensionID Register

Description

The PartID field in register 0x8083_2714, ExtensionID, is not programmed.

Workaround

None, this register has been de-featured from the chip.