Altera Triple-Speed Ethernet MAC driver

This is the driver for the Altera Triple-Speed Ethernet (TSE) controllers using the SGDMA and MSGDMA soft DMA IP components. The driver uses the platform bus to obtain component resources. The designs used to test this driver were built for a Cyclone(R) V SOC FPGA board, a Cyclone(R) V FPGA board, and tested with ARM and NIOS processor hosts separately. The anticipated use cases are simple communications between an embedded system and an external peer for status and simple configuration of the embedded system.

For more information visit www.altera.com and www.rocketboards.org. Support forums for the driver may be found on www.rocketboards.org, and a design used to test this driver may be found there as well. Support is also available from the maintainer of this driver, found in MAINTAINERS.

The Triple-Speed Ethernet, SGDMA, and MSGDMA components are all soft IP components that can be assembled and built into an FPGA using the Altera Quartus toolchain. Quartus 13.1 and 14.0 were used to build the design that this driver was tested against. The sopc2dts tool is used to create the device tree for the driver, and may be found at rocketboards.org.

The driver probe function examines the device tree and determines if the Triple-Speed Ethernet instance is using an SGDMA or MSGDMA component. The probe function then installs the appropriate set of DMA routines to initialize, setup transmits, receives, and interrupt handling primitives for the respective configurations.

The SGDMA component is to be deprecated in the near future (over the next 1-2 years as of this writing in early 2014) in favor of the MSGDMA component. SGDMA support is included for existing designs and reference in case a developer wishes to support their own soft DMA logic and driver support. Any new designs should not use the SGDMA.

The SGDMA supports only a single transmit or receive operation at a time, and therefore will not perform as well compared to the MSGDMA soft IP. Please visit www.altera.com for known, documented SGDMA errata.

Scatter-gather DMA is not supported by the SGDMA or MSGDMA at this time. Scatter-gather DMA will be added to a future maintenance update to this driver.

Jumbo frames are not supported at this time.

The driver limits PHY operations to 10/100Mbps, and has not yet been fully tested for 1Gbps. This support will be added in a future maintenance update.

1. Kernel Configuration

The kernel configuration option is ALTERA_TSE:

> Device Drivers ---> Network device support ---> Ethernet driver support ---> Altera Triple-Speed Ethernet MAC support (ALTERA_TSE)

2. Driver parameters list

- debug: message level (0: no output, 16: all);
- dma_rx_num: Number of descriptors in the RX list (default is 64);
- dma_tx_num: Number of descriptors in the TX list (default is 64).

3. Command line options

Driver parameters can be also passed in command line by using:

```
altera_tse=dma_rx_num:128,dma_tx_num:512
```

4. Driver information and notes

4.1. Transmit process

When the driver's transmit routine is called by the kernel, it sets up a transmit descriptor by calling the underlying DMA transmit routine (SGDMA or MSGDMA), and initiates a transmit operation. Once the transmit is complete, an interrupt is driven by the transmit DMA logic. The driver handles the transmit completion in the context of the interrupt handling chain by recycling resource required to send and track the requested transmit operation.

4.2. Receive process

The driver will post receive buffers to the receive DMA logic during driver initialization. Receive buffers may or may not be queued depending upon the underlying DMA logic (MSGDMA is able queue receive buffers, SGDMA is not able to queue receive buffers

to the SGDMA receive logic). When a packet is received, the DMA logic generates an interrupt. The driver handles a receive interrupt by obtaining the DMA receive logic status, reaping receive completions until no more receive completions are available.

4.3. Interrupt Mitigation

The driver is able to mitigate the number of its DMA interrupts using NAPI for receive operations. Interrupt mitigation is not yet supported for transmit operations, but will be added in a future maintenance release.

4.4) Ethtool support

Ethtool is supported. Driver statistics and internal errors can be taken using: ethtool -S ethX command. It is possible to dump registers etc.

4.5) PHY Support

The driver is compatible with PAL to work with PHY and GPHY devices.

4.7) List of source files:

- Kconfig
- Makefile
- altera_tse_main.c: main network device driver
- altera_tse_ethtool.c: ethtool support
- altera_tse.h: private driver structure and common definitions
- altera_msgdma.h: MSGDMA implementation function definitions
- altera_sgdma.h: SGDMA implementation function definitions
- altera_msgdma.c: MSGDMA implementation
- altera_sgdma.c: SGDMA implementation
- altera_sgdmahw.h: SGDMA register and descriptor definitions
- altera_msgdmahw.h: MSGDMA register and descriptor definitions
- altera_utils.c: Driver utility functions
- altera_utils.h: Driver utility function definitions

5. Debug Information

The driver exports debug information such as internal statistics, debug information, MAC and DMA registers etc.

A user may use the ethtool support to get statistics: e.g. using: ethtool -S ethX (that shows the statistics counters) or sees the MAC registers: e.g. using: ethtool -d ethX

The developer can also use the "debug" module parameter to get further debug information.

6. Statistics Support

The controller and driver support a mix of IEEE standard defined statistics, RFC defined statistics, and driver or Altera defined statistics. The four specifications containing the standard definitions for these statistics are as follows:

- IEEE 802.3-2012 - IEEE Standard for Ethernet.
- RFC 2863 found at http://www.rfc-editor.org/rfc/rfc2863.txt.
- RFC 2819 found at http://www.rfc-editor.org/rfc/rfc2819.txt.
- Altera Triple Speed Ethernet User Guide, found at http://www.altera.com

The statistics supported by the TSE and the device driver are as follows:

"tx_packets" is equivalent to aFramesTransmittedOK defined in IEEE 802.3-2012, Section 5.2.2.1.2. This statistics is the count of frames that are successfully transmitted.

"rx_packets" is equivalent to aFramesReceivedOK defined in IEEE 802.3-2012, Section 5.2.2.1.5. This statistic is the count of frames that are successfully received. This count does not include any error packets such as CRC errors, length errors, or alignment errors.

"rx_crc_errors" is equivalent to aFrameCheckSequenceErrors defined in IEEE 802.3-2012, Section 5.2.2.1.6. This statistic is the count of frames that are an integral number of bytes in length and do not pass the CRC test as the frame is received.

"rx_align_errors" is equivalent to aAlignmentErrors defined in IEEE 802.3-2012, Section 5.2.2.1.7. This statistic is the count of frames that are not an integral number of bytes in length and do not pass the CRC test as the frame is received.

"tx_bytes" is equivalent to aOctetsTransmittedOK defined in IEEE 802.3-2012, Section 5.2.2.1.8. This statistic is the count of data and pad bytes successfully transmitted from the interface.

"rx_bytes" is equivalent to aOctetsReceivedOK defined in IEEE 802.3-2012, Section 5.2.2.1.14. This statistic is the count of data

and pad bytes successfully received by the controller.

"tx_pause" is equivalent to aPAUSEMACCtrlFramesTransmitted defined in IEEE 802.3-2012, Section 30.3.4.2. This statistic is a count of PAUSE frames transmitted from the network controller.

"rx_pause" is equivalent to aPAUSEMACCtrlFramesReceived defined in IEEE 802.3-2012, Section 30.3.4.3. This statistic is a count of PAUSE frames received by the network controller.

"rx_errors" is equivalent to ifInErrors defined in RFC 2863. This statistic is a count of the number of packets received containing errors that prevented the packet from being delivered to a higher level protocol.

"tx_errors" is equivalent to ifOutErrors defined in RFC 2863. This statistic is a count of the number of packets that could not be transmitted due to errors.

"rx_unicast" is equivalent to ifInUcastPkts defined in RFC 2863. This statistic is a count of the number of packets received that were not addressed to the broadcast address or a multicast group.

"rx_multicast" is equivalent to ifInMulticastPkts defined in RFC 2863. This statistic is a count of the number of packets received that were addressed to a multicast address group.

"rx_broadcast" is equivalent to ifInBroadcastPkts defined in RFC 2863. This statistic is a count of the number of packets received that were addressed to the broadcast address.

"tx_discards" is equivalent to ifOutDiscards defined in RFC 2863. This statistic is the number of outbound packets not transmitted even though an error was not detected. An example of a reason this might occur is to free up internal buffer space.

"tx_unicast" is equivalent to ifOutUcastPkts defined in RFC 2863. This statistic counts the number of packets transmitted that were not addressed to a multicast group or broadcast address.

"tx_multicast" is equivalent to ifOutMulticastPkts defined in RFC 2863. This statistic counts the number of packets transmitted that were addressed to a multicast group.

"tx_broadcast" is equivalent to ifOutBroadcastPkts defined in RFC 2863. This statistic counts the number of packets transmitted that were addressed to a broadcast address.

"ether_drops" is equivalent to etherStatsDropEvents defined in RFC 2819. This statistic counts the number of packets dropped due to lack of internal controller resources.

"rx_total_bytes" is equivalent to etherStatsOctets defined in RFC 2819. This statistic counts the total number of bytes received by the controller, including error and discarded packets.

"rx_total_packets" is equivalent to etherStatsPkts defined in RFC 2819. This statistic counts the total number of packets received by the controller, including error, discarded, unicast, multicast, and broadcast packets.

"rx_undersize" is equivalent to etherStatsUndersizePkts defined in RFC 2819. This statistic counts the number of correctly formed packets received less than 64 bytes long.

"rx_oversize" is equivalent to etherStatsOversizePkts defined in RFC 2819. This statistic counts the number of correctly formed packets greater than 1518 bytes long.

"rx_64_bytes" is equivalent to etherStatsPkts64Octets defined in RFC 2819. This statistic counts the total number of packets received that were 64 octets in length.

"rx_65_127_bytes" is equivalent to etherStatsPkts65to127Octets defined in RFC 2819. This statistic counts the total number of packets received that were between 65 and 127 octets in length inclusive.

"rx_128_255_bytes" is equivalent to etherStatsPkts128to255Octets defined in RFC 2819. This statistic is the total number of packets received that were between 128 and 255 octets in length inclusive.

"rx_256_511_bytes" is equivalent to etherStatsPkts256to511Octets defined in RFC 2819. This statistic is the total number of packets received that were between 256 and 511 octets in length inclusive.

"rx_512_1023_bytes" is equivalent to etherStatsPkts512to1023Octets defined in RFC 2819. This statistic is the total number of packets received that were between 512 and 1023 octets in length inclusive.

"rx_1024_1518_bytes" is equivalent to etherStatsPkts1024to1518Octets define in RFC 2819. This statistic is the total number of packets received that were between 1024 and 1518 octets in length inclusive.

"rx_gte_1519_bytes" is a statistic defined specific to the behavior of the Altera TSE. This statistics counts the number of received good and errored frames between the length of 1519 and the maximum frame length configured in the frm_length register. See the Altera TSE User Guide for More details.

"rx_jabbers" is equivalent to etherStatsJabbers defined in RFC 2819. This statistic is the total number of packets received that were longer than 1518 octets, and had either a bad CRC with an integral number of octets (CRC Error) or a bad CRC with a non-integral number of octets (Alignment Error).

"rx_runts" is equivalent to etherStatsFragments defined in RFC 2819. This statistic is the total number of packets received that were less than 64 octets in length and had either a bad CRC with an integral number of octets (CRC error) or a bad CRC with a non-integral number of octets (Alignment Error).