BlazePPS

A Packet Processing System Implemented in an Altera Cyclone V

CSEE W4840
Embedded Systems

Valeh Valiollahpour Amiri (vv2252)
Christopher Campbell (cc3769)
Sheng Qian (sq2168)
Yuanpei Zhang (yz2727)

May 14th, 2015

Contents

1 Overview

The goal of this project was to implement a 10 Gigabit Ethernet (10GbE) packet processing system in an Altera Cyclone V SoC FPGA. Implementing such a system on the Cyclone V would provide a relatively inexpensive, flexible, and extensible 10GbE packet processing system. The flexibility and extensibility are derived from the fact that the system is mostly implemented in FPGA fabric. Anyone wishing to adapt the system to their needs could easily do so by modifying the existing modules or adding additional modules. This is especially appealing for industries that wish to combine high-speed networking with hardware accelerated tasks.

We utilized a Dual XAUI to SFP+ HSMC daughter card to interface the Cyclone V with a 10G network. In the FPGA we utilized an Altera XAUI PHY IP core, Altera 10GbE MAC IP core, dual-clock FIFOs, and on-chip FIFOs. We interfaced the hardware with Linux using a custom network driver. Originally, we built our project on top of RocketBoards' Golden System Reference Design (GSRD). The GSRD is a set of software and hardware components that can be used as a starting point for custom designs. It includes a Linux image meant to boot from an SD card, a base hardware design called the Golden Hardware Reference Design (GHRD), and various development tools. In response to various constraints and issues with the GSRD we moved to a custom hardware/software system. The custom system uses U-Boot to boot Linux on the Cyclone V from a UImage and uses Linux to configure the FPGA from a Raw Binary File (rbf).

Due to technical difficulties associated with getting the Altera 10GbE MAC to work, we created a second version of the project using a simple MAC from a previous project. The project using the Altera 10GbE MAC is referred to as BlazePPS_v1 and the project using the simple MAC is referred to as BlazePPS_v2.

We discuss the architectural and timing details of our project in the Design section, provide tutorials on how to recreate our project in the Tutorials, and detail who did what and lessons learned in the Credits & Lessons Learned section. Additionally, we've included all relevant files to our project in the Files section.

2 Design

2.1 Architectual Design

2.1.1 Overall Design

Our design primarily consists of 5 modules, which are interconnected as illustrated in figure 1. Starting from the Avalon Memory-Mapped (Avalon-MM), we use a set of Avalon On-Chip (OC) FIFOs to interface with the Avalon Streaming (Avalon-ST). The TX OC FIFO takes as input an Avalon-MM write slave interface, and as output an Avalon-ST source. On the other hand, the RX OC FIFO has as input an Avalon-ST sink interface, and its output is an Avalon-MM read slave interface.

Each of the OC FIFOs is connected to an Avalon Dual-Clock FIFO (DCFIFO) which is synchronized to the HPS clock on the one side (input for the TX DCFIFO and output for the RX DCFIFO) and to the XAUI PHY clock on the other side (input for the RX DCFIFO and output for the TX DCFIFO).

Before being directed to the MAC module, the data outputted by the TX DCFIFO is first buffered in a TX buffer module which is used in order to make sure that the 64 bits ST data is written in one clock cycle along with the valid signal (as expected by the XAUI PHY module).

The next stage is the MAC module itself. This is Alteras 10-Gbps Ethernet (10GbE) Media Access Controller (MAC) IP core, which has on one side the 64-bit wide Avalon-ST interface, and on the other side the 72-bit wide SDR XGMII interface. Both interfaces run at a 156.25 MHz frequency.

The last module in the chain is the XAUI PHY module. This is Altera's XAUI PHY IP Core which has a 72-bit data (single data rate - SDR XGMII) interface to the application layer, at 156.25 Mbps and a serial interface that consists of 4 x 3.125 Gbps lanes connected to the Dual XAUI board through the HSMC connectors.

Mounted on the Altera Cyclone V board, we have a daughter card which is the Dual XAUI to SFP+ High Speed Mezzanine (HSMC) board. This board interfaces with the XAUI module on the FPGA through the HSMC connectors on the one side, and with the network through SFP+ optical modules, on the other side.

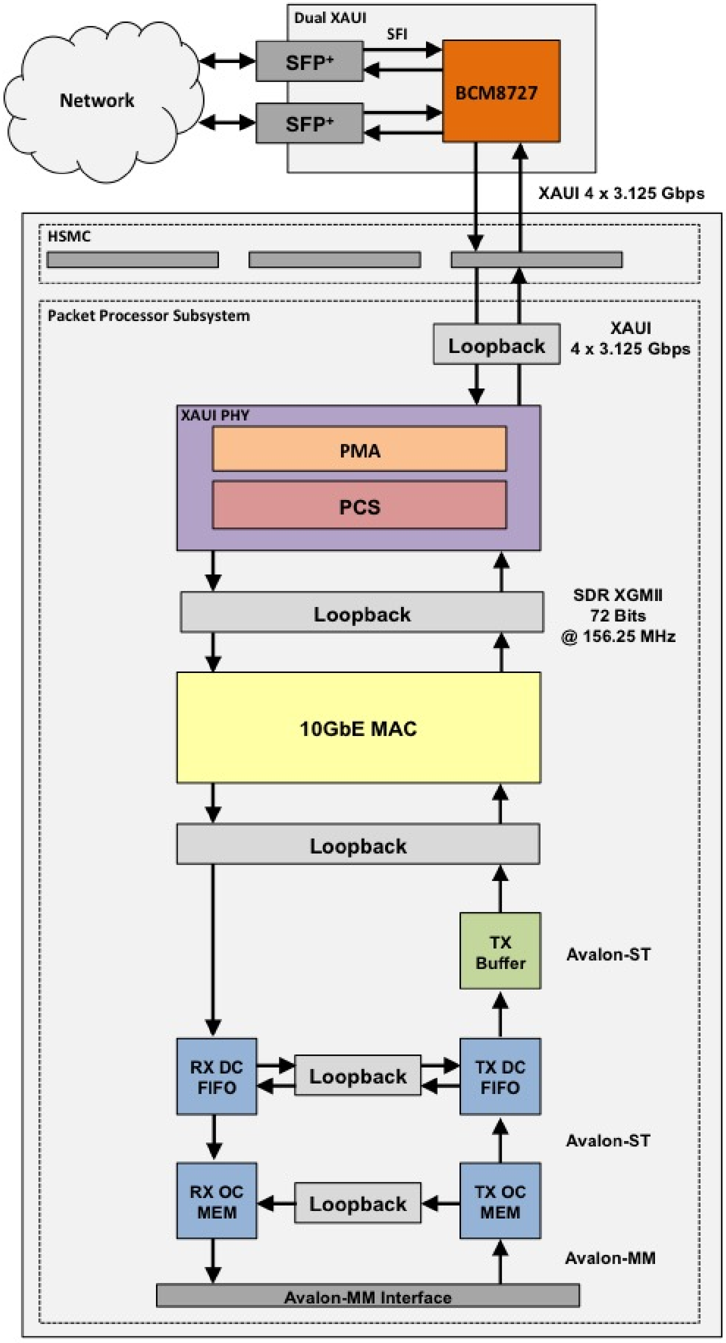


figure 1 - Hardware Design Schematic

2.1.2 Dual XAUI HSMC Board

We use the Dual XAUI to SFP+ HSMC for the actual communication with the network. This board contains 2 full duplex 10G SFP+ channels with a XAUI backend interface. The XAUI interfaces will be attached to the HSMC side of the card and the SFI side of the interface will be attached to the SFP+ optical modules on opposite side of the board. The main device on the daughter board is the BCM8727, a dual-channel 10-GbE SFI-to-XAUI transceiver. It is fully compliant with the 10-GbE IEEE 802.3aq standard and incorporates an Electronic Dispersion Compensation (EDC) equalizer which supports SFP+ line-card applications. Figure 2 describes the connections between the BCM8727 and the Cyclone V board through the HSMC connectors.

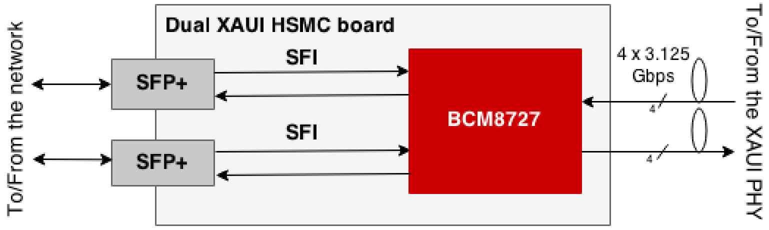


figure 2 - Dual XAUI board and its connections

The BCM8727 interfaces, on the one hand, with the XAUI PHY through a 4-lane XAUI interface (4 x 3.125 Gbps). On the other hand, it uses the SFI protocol to connect and communicate with the SFP+ optical modules.

2.1.3 XAUI PHY

The Altera XAUI PHY IP Core implements the IEEE 802.3 Clause 48 specification to extend the operational distance of the XGMII interface and reduce the number of interface signals. The XAUI PHY module accepts 72-bit data (single data rate - SDR XGMII) from the application layer at either 156.25 Mbps or 312.5 Mbps. The serial interface runs at either 4 x 3.125 Gbps or 4 6.25 Gbps (DDR XAUI option).

In our design, the XAUI PHY module communicates through the HSMC connectors with the daughter board to receive 4 lanes of serial data at 3.125 Gbps each and outputs a 72-bit SDR XGMII signal at 156.25 MHz towards the MAC module (or the other way round, for transmission). The 72-bit data are obtained by adding a one-bit control to each byte of the arriving packets. Figure 3 represents the XAUI PHY module and its interconnections with the rest of our system. As we can see, there are two main components inside the XAUI PHY IP core, which it uses to manipulate data at the low-level physical layer: the PMA (Physical Medium Attachment) and the PCS (Physical Coding Sublayer).

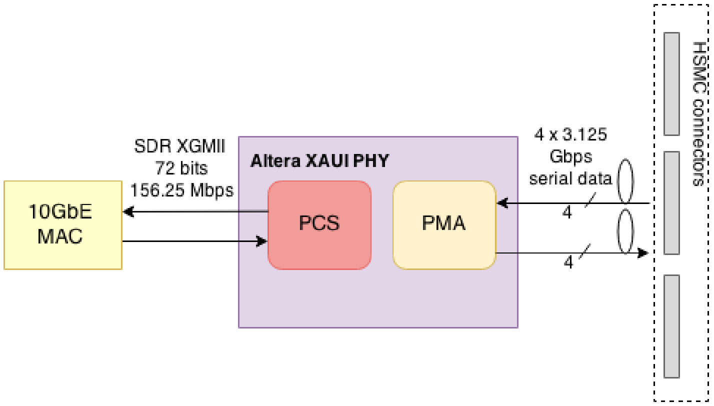


figure 3 - XAUI PHY module and its interfaces

The XAUI PHY has two in/out interfaces:

1. The XAUI PCS interface to the FPGA fabric uses a SDR XGMII interface. This interface implements a simple version of Avalon-ST protocol. The interface does not include ready or valid signals; consequently, the sources always drive data and the sinks must always be ready to receive data. According to the configuration parameters that we chose, the application interface in our case runs at 156.25 Mbps and the data in only driven on the rising edge of clock. To meet the bandwidth requirements, the datapath is eight bytes wide with eight control bits (one per data byte), hence the 72 bits obtained (instead of 64 bits).

2. The other interface of the XAUI module is the Transceiver Serial Data Interface. This interface has 4 lanes of serial data for both the TX and RX interfaces. It runs at 3.125 GHz in the case of our design. There is no separate clock signal because it is encoded in the data.

2.1.4 10GbE MAC

The 10-Gbps Ethernet (10GbE) Media Access Controller (MAC) IP core is a configurable component that implements the IEEE 802.3-2008 specification. The IP core offers several modes (according to the Ethernet speed). We use the “10M-10G mode” which uses the Avalon-ST interface on the client side and SDR XGMII (for the 10G mode) on the network side. The 10GbE MAC IP core handles the flow of data between a client and Ethernet network through a 10-Gbps Ethernet PHY. On the transmit path, the MAC accepts client frames and constructs Ethernet frames by inserting various control fields, such as checksums before forwarding them to the PHY. Similarly, on the receive path, the MAC accepts Ethernet frames via a PHY, performs checks, and removes the relevant fields before forwarding the frames to the client. Figure 4 summarizes the internal blocks and the interfaces to the core.

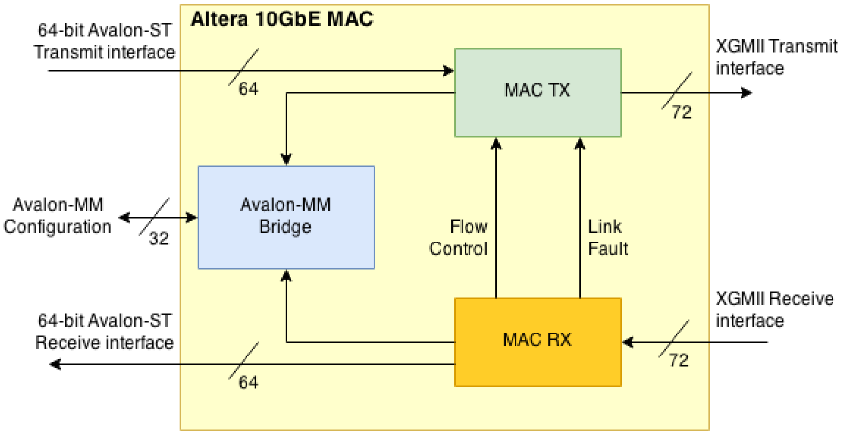


figure 4 - MAC module - its internal organization and its interfaces

The core is a composition of three blocks: MAC receiver (MAC RX), MAC transmitter (MAC TX), and Avalon-MM bridge. The MAC RX and MAC TX handle data flow between the client and Ethernet network. The Avalon-MM bridge provides a single interface to all Avalon-MM interfaces within the MAC, which allows a host to access 32-bit configuration and status registers, and statistics counters.

On the client-side interface of the MAC, we have the ST data coming from/going to the DC-FIFOs. In the MAC, the Avalon-ST interface acts as a sink in the transmit datapath and source in the receive datapath.

The network-side interface of the 10GbE MAC implements the SDR version of the XGMII protocol. On this side, the MAC is connected to XAUI PHY module. The data bus carries the MAC frame; the most significant byte occupies the least significant lane.

Note: It is worth mentioning that in our design, the Altera 10GbE MAC IP core worked to some extent only. More precisely, when using any loopback system that did not include the OC FIFOs, the system worked properly, meaning we could use `tcpdump` and/or `tcpreplay` commands (on the desktop) to transmit and receive packets. In this configuration, we managed to transmit/receive the expected data, using the Altera 10GbE MAC IP core, and the whole system as long as it did not inolve the OC FIFOs, i.e. the interface with the Avalon-MM.

In spite of our attempts to troubleshoot the problem (by using several TCL scripts for faster testing, signal tapping, trying other possible simple configurations for the MAC, etc.) we could not manage to solve the issue. For this reason, we used the FAST MAC module from the previous project instead of the Altera IP core. When the FAST MAC module is used, the system works front to back, both in transmission and reception (i.e. starting from the Avalon-MM interface and receiving the packet with `tcpdump` on the desktop, for transmission, and sending a packet from the desktop using `tcpreplay` and receiving the packet on the Avalon-MM interface).

2.1.5 TX buffer

The XAUI PHY module expects to receive 64 bits data at once (in one clock cycle), along with one valid signal for all of the 64 bits. On the other hand, the Altera MM_to_ST converter takes some clock cycles to organize the 64 bits stream data when the input is 32 bits. As a result (and also because of the higher latency of software compared to hardware), the 64-bit data outputted from the TX DC FIFO is sometimes broken down into chunks and the valid pulse signal is only sent at the end. But this scenario is not accepted by the XAUI PHY. For this reason, we use the TX FIFO module, from the previous project, which holds the output data from the TX DC FIFO and transmits the 64 bits when ready, in one clock cycle, along with the valid signal.

2.1.6 DC FIFOs

Altera provides FIFO functions through the parameterizable single-clock FIFO (SCFIFO) and dual-clock FIFO (DCFIFO) megafunction IP cores. The FIFO functions are mostly applied in data buffering applications that comply with the first-in-first-out data flow in synchronous or asynchronous clock domains.

For our design, we used two DCFIFOs - one for transmit and one for receive - to prepare the 64 bits input data arriving at the input clock frequency, for a 64 bits output data transmitted at the output clock frequency.

We also enabled the **Use packets** feature for the DCFIFOs in conformance with the OC FI-

FOs that they communicate with, which are configured to receive and transmit packet-oriented data.

2.1.7 On-Chip Memory FIFOs

The on-chip FIFO memory core is a configurable component used to buffer data as well as provide flow control. The FIFO can operate with a single clock or with separate clocks for the input and output ports. The input interface to the FIFO may be an Avalon-MM write slave or an Avalon-ST sink. The output interface can be an Avalon-ST source or an Avalon-MM read slave. The data is delivered to the output interface in the same order that it was received at the input interface, regardless of the value of channel, packet, frame, or any other signals.

The on-chip FIFO has four possible configurations:

1. Avalon-MM write slave to Avalon-MM read slave
2. Avalon-ST sink to Avalon-ST source
3. Avalon-MM write slave to Avalon-ST source
4. Avalon-ST sink to Avalon-MM read slave

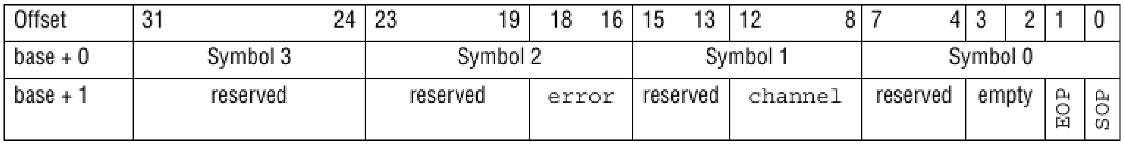
In our system, we used the last two configuration, to interface, from the Avalon-MM, with the Avalon-ST.

2.1.8 Avalon-MM write slave to Avalon-ST source

The TX OC MEM (as noted on figure 1) is an on-chip FIFO with the "Avalon-MM write slave to Avalon-ST source" configuration. In this mode, the FIFO's input is an Avalon-MM write slave with a width of 32 bits. The Avalon-ST output (source) data width must also be 32 bits. Several output interface parameters are configurable, including: **bits per symbol**, **symbols per beat**, **width of the channel**, and **error signals**. In the Tutorials section, we mention the settings for these parameters in the case of our design.

The FIFO performs the endian conversion to conform to the output interface protocol.

In this mode ("Avalon-MM write slave to Avalon-ST source"), the on-chip FIFO has an additional parameter - **Enable packet data** - that allows to specify whether or not we are using packet data.



Offset	31 24	23 19	18 16	15 13	12 8	7 4	3 2	1	0
base + 0	Symbol 3	Symbol 2		Symbol 1		Symbol 0			
base + 1	reserved	reserved	error	reserved	channel	reserved	empty	EOP	SOP

figure 5 - Avalon-MM to Avalon-ST Memory Map

If **Enable packet data** is off, the Avalon-MM write master writes all data at address offset 0 repeatedly to push data into the FIFO.

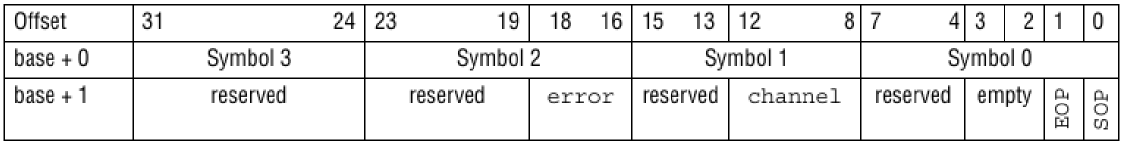
If **Enable packet data** is on, the Avalon-MM write master starts by writing the SOP, error (optional), channel (optional), EOP, and empty packet status information at address offset 1. Writing to address offset 1 does not push data into the FIFO. The Avalon-MM master then writes packet data to the FIFO repeatedly at address offset 0, pushing 8-bit symbols into the FIFO. Whenever a valid write occurs at address offset 0, the data and its respective packet information is pushed into the FIFO. Subsequent data is written at address offset 0 without the need to clear the SOP. Rewriting to address offset 1 is not required each time if the subsequent data to be pushed into the FIFO is not the end-of-packet data, as long as error and channel do not change.

At the end of each packet, the Avalon-MM master writes to the address at offset 1 to set the EOP bit to 1, before writing the last symbol of the packet at offset 0. The write master uses the empty field to indicate the number of unused symbols at the end of the transfer. If the last packet data is not aligned with the **symbols per beat**, the empty field indicates the number of empty symbols in the last packet data. For example, if the Avalon-ST interface has **symbols per beat** of 4, and the last packet only has 3 symbols, the empty field will be 1, indicating that one symbol (the least significant symbol in the memory map) is empty.

2.1.9 Avalon-ST sink to Avalon-MM read slave

The RX OC MEM (as noted on figure 1) is an on-chip FIFO with the "Avalon-ST sink to Avalon-MM read slave" configuration. In this mode, the FIFO's input is an Avalon-ST sink and the output is an Avalon-MM read slave with a width of 32 bits. The Avalon-ST input (sink) data width must also be 32 bits. Similarly to the previous on-chip FIFO configuration, the same ST interface parameters (this time as input) are configurable. Refer to the Tutorials section for a detailed specification of these parameters in the case of our design. The FIFO performs the endian conversion to conform to the output interface protocol.

Just as before, in the "Avalon-ST sink to Avalon-MM read slave", the on-chip FIFO has an additional parameter - **Enable packet data** - that allows to specify whether we are using packet data.



Offset	31 24	23 19	18 16	15 13	12 8	7 4	3 2	1	0
base + 0	Symbol 3	Symbol 2		Symbol 1		Symbol 0			
base + 1	reserved	reserved	error	reserved	channel	reserved	empty	EOP	SOP

figure 6 - Avalon-ST to Avalon-MM Memory Map

If **Enable packet data** is off, read data repeatedly at address offset 0 to pop the data from the FIFO.

If **Enable packet data** is on, the Avalon-MM read master starts reading from address offset 0. If the read is valid, that is, the FIFO is not empty, both data and packet status information are popped from the FIFO. It is important to highlight that *before* reading status information at address offset 1, it is necessary to *first* do the data read at address offset 0 to make sure it is valid (i.e. FIFO not empty), otherwise the packet status information is not yet popped from the FIFO so it is too early to try to read its correct value. The packet status information is obtained by reading at address offset 1. Reading from address offset 1 does not pop data from the FIFO. The `error`, `channel`, `SOP`, `EOP` and `empty` fields are available at address offset 1 to determine the status of the packet data read from address offset 0. The empty field indicates the number of empty symbols in the data field. For example, if the Avalon-ST interface has symbols-per-beat of 4, and the last packet data only has 1 symbol, then the empty field will be 3 to indicate that 3 symbols (the 3 least significant symbols in the memory map) are empty.

In our design, we enabled the packet data feature for both OC FIFOs in order to have a simpler design and avoid the use of Alteras packet-to-byte and byte-to-packet modules.

2.1.10 Loopbacks

In order to incrementally test each subset of the system, we extensively used the loopback mechanism, meaning if the output of the system matches its inputs then we can confirm that that system works correctly.

Figure 1 illustrates how many loopback modules we used and where in the whole design we inserted them in order to test basically each stage before moving to the next one.

2.2 Timing Design

2.2.1 XAUI PHY

On the network-side (the side that interfaces the Dual XAUI daughter card through the HSMC interfce) it has transmit and receive serial interfaces that run at 4 3.125 Gbps. That is, it has 4 serial transmit lanes and 4 serial receive lanes that run at 3.125 Gbps.

The XAUI PHY is fed a 156.25 MHz reference clock produced by an external PLL. It uses this clock as a reference for clock data recovery (the clock is recovered from the serial input data) and for its transmit PLL. It outputs a 156.25 MHz clock (produced by clock recovery) associated with the client-side receive data and accepts a 156.25 MHz clock associated with the client-side transmit data. In our project we used the XAUI's 156.25 MHz output clock as input to its transmist clock, as well as input to the transmist and receive clock of the MAC. This configuration ensures synchronization between all interfaces. A block diagram showing the various clocks associated withe XAUI PHY is shown in figure 7.

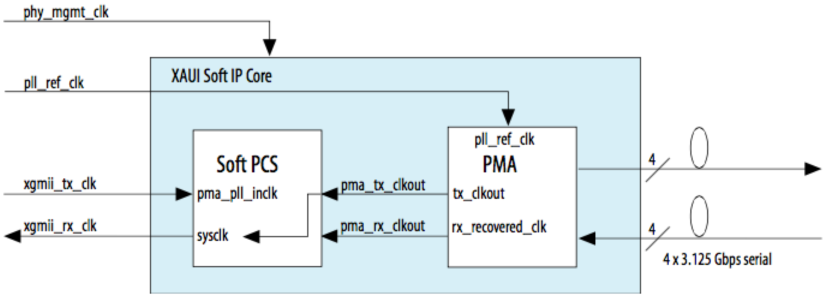


figure 7. XAUI Input and Output Clocks

On the client-side (the side that interfaces with the 10GBE MAC), it has SDR XGMII transmit and receive interfaces that provide 72 bits at 156.25 Mbps. The XGMII interface is an implementation of the Avalon-ST interface that lacks valid and ready signals, so data is transmitted/received on every rising clock edge. The XAUI PHY treats the datapath as two interleaved 32-bit data buses. An example of the transformation from the original data to the interleaved data is shown in figure 8.

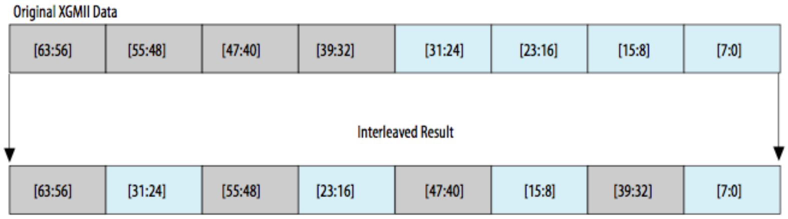


figure 8. XAUI Data Transformation

Because of the interleaving the start of frame transmission can start with byte 0 or byte 5. The timing diagram for byte 0 start of frame transmission is shown in figure 9, while the timing diagram for byte 5 start of frame transmission is shown in figure 10.[1]

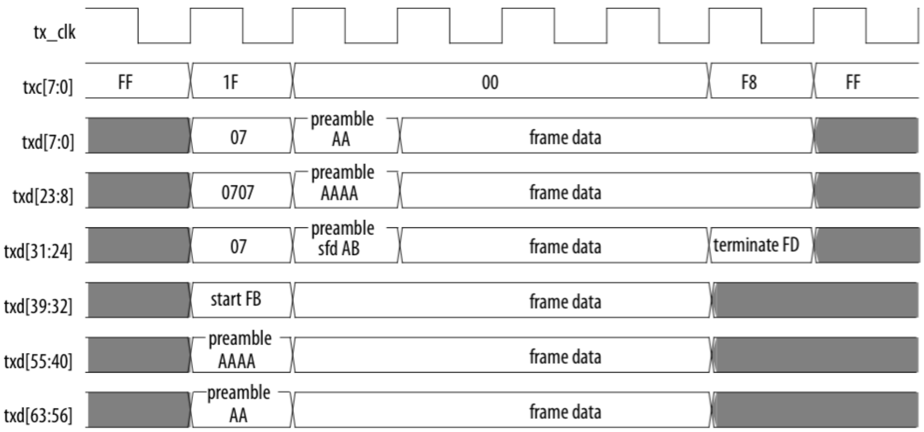


figure 9. XAUI Timing Diagram (byte 0 start of frame)

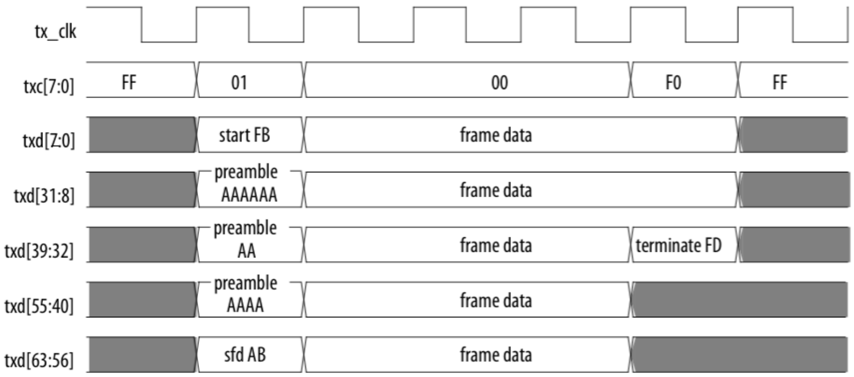


figure 10. XAUI Timing Diagram (byte 5 start of frame)

2.2.2 10GbE MAC

As mentioned above, the input to the transmit and receive clock of the MAC is the 156.25MHz clock produced by the XAUI PHY. The network-side (the side that interfaces with the XAUI PHY) has SDR XGMII transmit and receive interfaces that transmit/receive 72 bits at 156.25 MHz. Timing diagrams showing the transmission of 8 values on the network-side is shown in figure 11 and figure 12. Timing diagrams showing the receiving of 8 values on the client-side is shown in figure 13 and figure 14.[2]

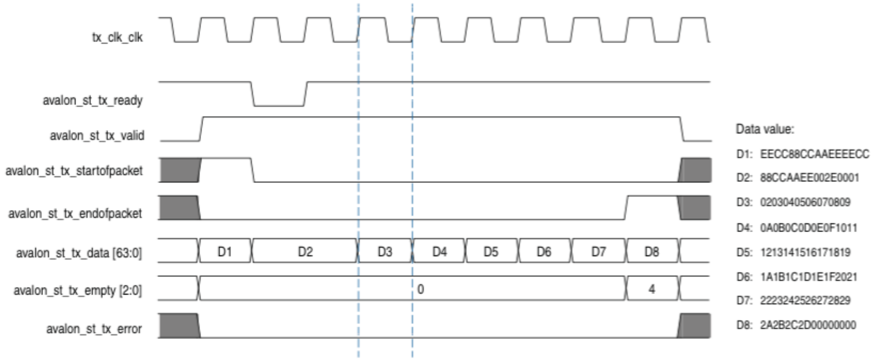


figure 11. MAC Network-Side Timing Diagram

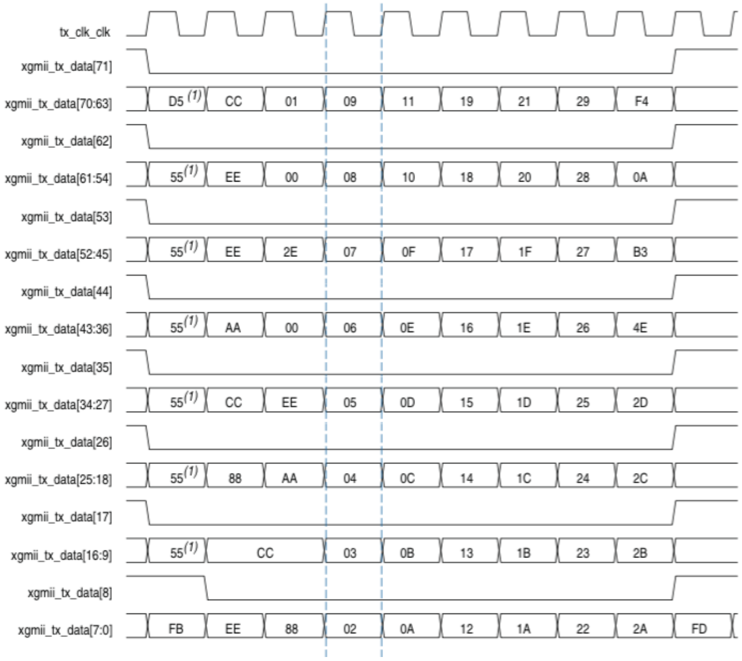


figure 12. MAC Network-Side Timing Diagram (detailed)

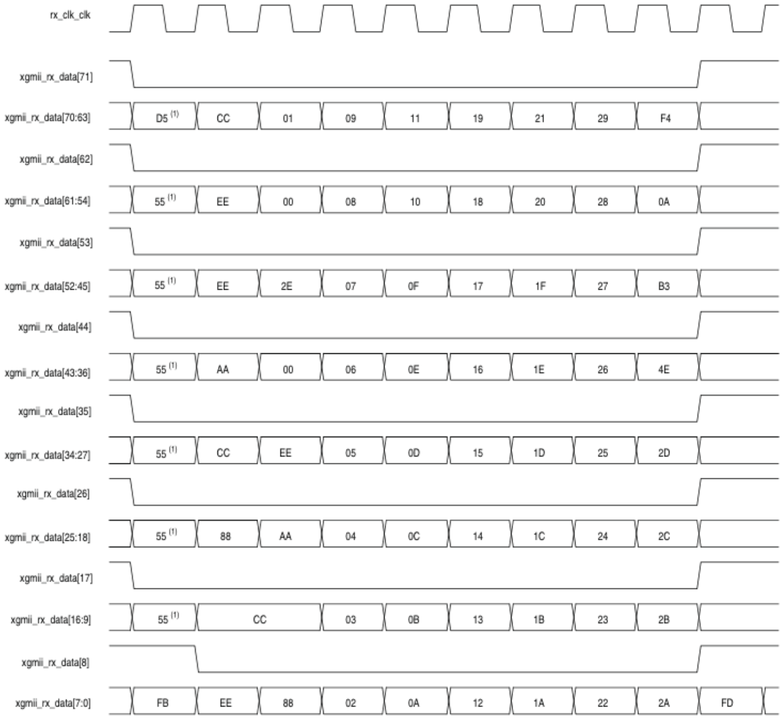


figure 13. MAC Client-Side Timing Diagram

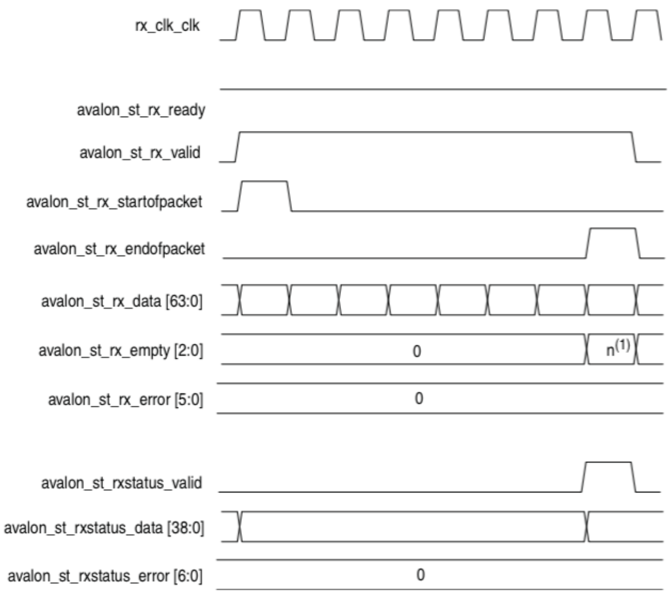


figure 14. MAC Client-Side Timing Diagram (detailed)

The interfaces on the client-side (the side that interfaces with the dual-clock FIFOs) has Avalon-ST transmit and receive interfaces that transmit/receive 64 bits at 156.25 MHz. The Avalon-ST is high bandwidth, low latency, and has (optional) packet support. It uses a ready signal to indicate when it is capable of receiving data and a valid signal to indicate when it is ready to transmit data. Its packet support uses a signal to indicate the start of packet (SOP) as well as a signal to indicate end of packet (EOP). A timing diagram for the Avalon-ST interface is shown in figure 15.^{3}

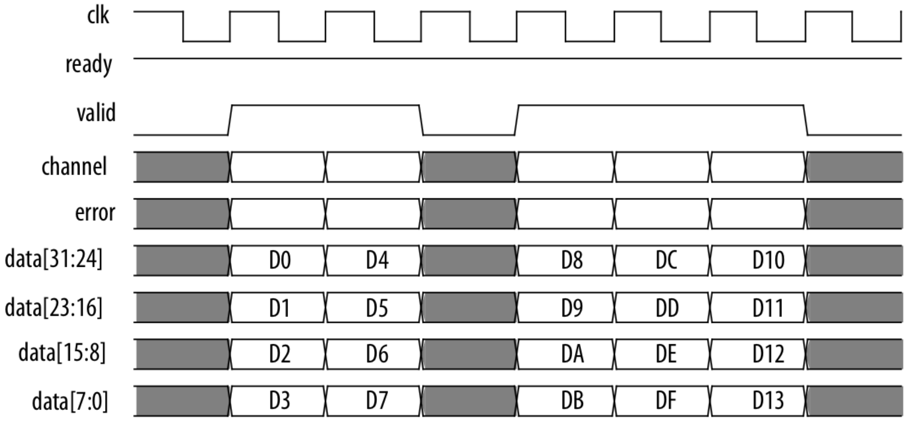


figure 15. Avalon-ST Timing Diagram

2.2.3 Tx Buffer

The input clock to the Tx buffer is also the 156.25 MHz output clock from the XAUI PHY. The Tx buffer uses the Avalon-ST interface to transmit and receive data at 156.25 MHz. See figure 15 for an Avalon-ST timing diagram.

2.2.4 DC FIFOs

The transmit DC FIFO receives data from the client-side (the side connected to the transmit on-chip FIFO) at 50 MHz on an Avalon-ST interface and transmits data to the network-side (the side connected to the MAC) at 156.25 MHz on an Avalon-ST interface. The receive DC FIFO receives data from the network-side (the side connected to the MAC) at 156.25 MHz on an Avalon-ST interface and transmits data to the client-side (the side connected to the receive on-chip FIFO) at 50 MHz on an Avalon-ST interface. For a discussion of the Avalon-ST interface consult the MAC section.

2.2.5 On-Chip Memory FIFOs

The On-Chip FIFOs interface with HPS/JTAG Master through the Lightweight HPS-to-FPGA Bridge, as well as interface with the DC FIFOs. The client-side of the transmit on-chip FIFO (the side that interfaces with the HPS/JTAG Master) has an Avalon-MM interface and receives 32 bits at 50 MHz. The network-side of the transmit on-chip FIFO (the side connected to the transmit DC FIFO) has an Avalon-ST interface and transmits 32 bits at 50 MHz. The network-side of the receive on-chip FIFO (the side connected to the receive DC FIFO) has an Avalon-ST interface and receives 32 bits at 50 MHz. The client-side of the receive on-chip FIFO (the side that interfaces with the HPS/JTAG Master) has an Avalon-MM interface and transmits 32 bits at 50 MHz. For a discussion of the Avalon-ST interface consult the MAC section. The Avalon-MM interface provides memory-mapped read and write interfaces for master and slave components. The Avalon-MM interface associated with the both the transmit and receive FIFOs are considered slaves because the HPS/JTAG Master act as the master. Like the use of the Avalon-ST interface, the Avalon-MM interface provides packet support. Given that it is memory-mapped, status bits in registers are set to indicate start of packet (SOP) and end of packet (EOP). The SOP bit is set when the first 4 bytes (32 bits) of a packet are written or read and the EOP bit is set when the last bytes of a packet are written or read. Because the last bytes of a packet may not be aligned to 4 bytes, two additional status bits (EMPTY) are used to indicate the actual number of bytes being written or read. A timing diagram for the Avalon-MM interface is shown in figure 16.

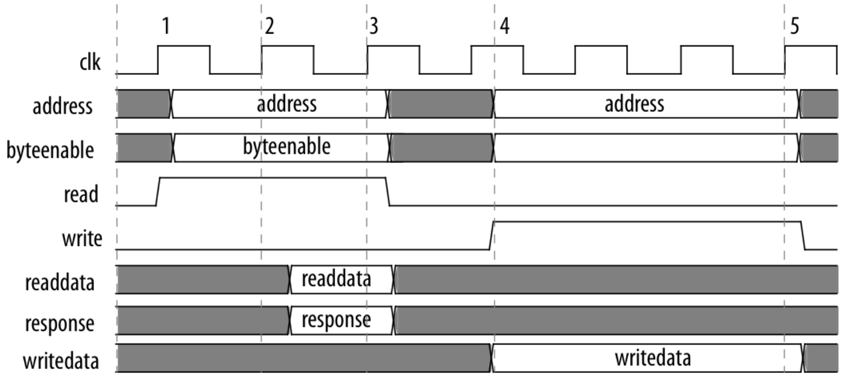


figure 16. Avalon-MM Timing Diagram

2.3 Network Driver

We wrote our driver in both an incremental and iterative manner. We started by writing a simple Linux character/miscellaneous driver capable of writing to and reading from memory mapped interfaces on the FPGA. This driver was heavily influenced by a driver from CSEE W4840 Lab 3 at Columbia University. The goal of this driver was to familiarize us with Linux drivers and provide the framework for writing to and reading from transmit and receive buffers in our final network driver. We then wrote a simple Linux network driver that was devoid of hardware details (it simulated all hardware in software). This driver was heavily influenced by snull.c, a driver developed by O'Reilly & Associates as part of a Linux network driver tutorial. The goal of this driver was to familiarize us with Linux network drivers and provide the overall framework for our final network driver. We then used the simple character driver and the simple network driver to write a more complete network driver. In addition to using our previous drivers as reference, we also used Altera's Triple-Speed Ethernet MAC driver.

At load time several parameters of the driver can be set, including the pool size and timeout length. The pool size represents the size of the packet pool, which is a linked lists of in-flight packet structs used by the driver for transmit and receive and the timeout length represents the time the kernel will wait on the hardware before timing out. If the parameters are not set at load time they are set to default values. Additionally, the values of the parameters are always exposed through the sysfs so that the user can review them. An example of loading the driver with and without setting the parameters is shown below.

```
Without Parameters
$ insmod blazepps_ethernet

With Parameters
$ insmod blazepps_ethernet pool_size="16" timeout="10"
```

The network driver is loaded as a platform driver, so when it is loaded its probe function is called. The probe function allocates and network device and then initializes and registers it. The MAC address and transmit/receive buffer resources are obtained from the driver's entry in the device tree.

```
blazepps_ethernet: blazepps_ethernet@20{
compatible = "altr,blazepps_ethernet";
reg = < 0x20 0x8 0x28 0x8>;
reg-names = "tx_fifo", "rx_fifo";
local-mac-address = [ 48 40 48 40 48 40 ];
};
```

The network device is actually allocated as an Ethernet device, which means that the kernel initializes many of its fields. This results in the interface automatically being named ”ethx,” were x is the next available eth interface number. For example, if the system already has an eth0 interface, the interface associated with the network device registered by our driver will be eth1. Below we show the output of ifconfig after loading the network driver and upping the interface.

```
eth1      Link encap:Ethernet  HWaddr 48:40:48:40:48:40
```

```
inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

When an application, such as ping, uses the interface to send a packet the kernel calls the network driver transmit function. The transmit function passed an sk_buff, which contains the contents of the packet. The network driver then writes the packet to the packet to the transmit buffer, setting the SOP, EOP, and EMPTY status bits as necessary. As a debugging feature, the driver writes all data that it writes to the transmit buffer to the system message buffer (to see the system message buffer simply use the dmesg command). When it is done writing the packet to the transmit buffer it simulates an interrupt indicating that packet transmission is complete. The reason this interrupt is simulated is because the hardware does not currently have interrupt capabilities.

Because the hardware does not have interrupt capabilities it is difficult to test true receiving, so a combination of hardware loopback and receive interrupt simulation was to test the receive pathway of the driver. More specifically, we enabled loopback between the transmit and receive buffers in hardware and simulate a receive interrupt after transmission (the process of transmission was described above). This means that soon after the driver writes the packet to the transmit buffer, it will arrive at the receive buffer and the driver will be notified that there is a packet to receive. When the network driver receives the receive interrupt it first reads the packet in from the receive buffer. As a debugging feature, the driver writes all data that it reads from the receive buffer to the system message buffer. After reading in the packet it packages it in an sk_buff and passes it to the upper layers.

The network driver also keeps statistics on the packets it transmits and receives, which are stored in a net_device_stats struct. Applications, such as ping can access and report the stats.

3 Tutorials

3.1 Hardware

3.1.1 Cyclone V Configuration

The following configuration is necessary for booting Linux from a server and having Linux configure the FPGA (see the Boot Server section to learn how to setup the boot server).

JTAG CHAIN SW (selects devices for the JTAG scan chain)
4.1 = 1 (HSMC off)
4.2 = 0 (HPS on)
SW6
MSEL[0:4] = 0000
CODEC_SEL = 1
J15-J19
CLKSEL[0:1] = 00
BOOTSEL[0:2] = 111

3.1.2 Dual XAUI Configuration

Jumpers
The following configuration is necessary for the board to work correctly.

Shunt J13 (SCL)
Shunt J14 (CLK_OE)

3.1.3 Dual XAUI Pin Assignments

The Cyclone V signals/pins necessary to connect the FPGA to the Dual XAUI through the HSMC are listed below to (some are not necessary, but recommended). To see the corresponding location and I/O standard values please consult section A in the appendix.

```
input                HSMC_CLKIN_p;
input  logic [3:0]   SMC_XAUI_RX_p0;
output logic [3:0]   HSMC_XAUI_TX_p0;
output logic         MDC2;
inout  logic         MDIO2;
output logic         MDC1;
inout  logic         MDIO1;
output logic         PRTAD02;
output logic         PRTAD01;
output logic         PRTAD4;
output logic         PRTAD3;
output logic         PRTAD2;
output logic         PRTAD1;
output logic         TXONOFF1;
output logic        TXONOFF2;
output logic        OPINLVL;
output logic        OPOUTLVL;
output logic        PHYRESET;
output logic [7:0]  USER_LED_G;
output logic [7:0]  USER_LED_R;
output logic        CONFIG1_1;
output logic        CONFIG0_1;
output logic        CONFIG1_2;
output logic        CONFIG0_2;
output logic        SS338_CLKIN;

inout  logic        GPIO0_1;
inout  logic        GPIO1_1;
inout  logic        GPIO0_2;
inout  logic        GPIO1_2;
output logic        SER_BOOT;
output logic        SMBSPDSEL1;
output logic        SMBSPDSEL2;
inout  logic        SMBWEN;
inout  logic        NVMA1SEL;
inout  logic        NVMPROT;
input  logic        OPRXLOS2;
input  logic        OPTXFLT2;
input  logic        SFP_TXDIS2;
input  logic        SFP_TXRS20;
input  logic        LASI2;
```

3.1.4 Dual XAUI Control Signal Assignments

The following control signal assignments are necessary for the Dual XAUI to work correctly (some are not strictly necessary, but recommended).

```
assign CONFIG0_1 = 1'b1;
assign CONFIG1_1 = 1'b0;
assign CONFIG0_2 = 1'b1;
assign CONFIG1_2 = 1'b0;
assign SS338_CLKIN = 1'b0;
assign SER_BOOT = 1'b0;
assign SMBSPDSEL1 = 1'b0;
assign SMBSPDSEL2 = 1'b0;
assign SMBWEN = 1'b1;
assign GPIO0_1 = 1'b0;
assign GPIO1_1 = 1'b0;
assign GPIO0_2 = 1'b0;
assign GPIO1_2 = 1'b0;
assign NVMA1SEL = 1'b1;
assign NVMPROT = 1’b0;
assign MDIO1 = !MDOEN1? MDO1 : 1’bz;
assign MDIN1 = MDIO1;
assign MDC2 = 1’bz;
assign MDIO2 = 1’bz;
assign PHYRESET =  KEY[1];
assign STOPMON = ~KEY[2];
assign {PRTAD4,PRTAD3,PRTAD2,PRTAD1,PRTAD01} = 5’b00000;
assign PRTAD02 = 1’b0;
assign TXONOFF1 = 1’b1;
assign TXONOFF2 = 1’b1;
assign OPOUTLVL = 1b0;
assign OPINLVL = 1’b1;
assign USER_LED_R = 8b00001111;
assign USER_LED_G = 8b11110000;
```

3.1.5 XAUI PHY

1. Using the MegaWizard Plug-In Manager in Quartus (13.1 64-bit), create a new XAUI PHY v13.3 IP core (Installed Plugins ¿ Interfaces ¿ Ethernet ¿ XAUI PHY v13.3)

2. Use the following potentially non-default settings:

   XAUI Interface Type: Soft XAUI
   PLL Type: CMU
   Base Data Rate: 3125 Mbps
   (see xaui.v in the project files for a full description of the settings)

3.1.6 PLL

1. Using the MegaWizard Plug-In Manager in Quartus (13.1 64-bit), create a new Altera PLL v13.1 IP core (Installed Plugins ¿ PLL ¿ Altera PLL v13.1) using the following potentially non-default settings:

   PLL Mode: Fractional-N PLL
   Reference Clock Frequency: 50.0 MHz
   Channel Spacing: 0 kHz
   Operation Mode: direct
   Enable locked output port: selected
   Desired Frequency: 156.2 MHz
   Actual Frequency: 156.2 MHz
   Duty Cycle: 50(see pll.v in the project files for a full description of the settings)

3.1.7 Timing Constraints

1. In Quartus, open the TimeQuest Timing Analyzer(tools ¿ TimeQuest Timing Analyzer)

2. Create Timing Netlist(Netlist ¿ Create Timing Netlist)

3. Select post fit, if you have done full compilation; select post map, if you have done only analysis and synthesis (for this project, either is OK).

4. In TimeQuest, open TimeQuest Timing Analyzer Wizard and use the following settings:

   Clock Name=clk_buff0
   Input_Pin=OSC_50_B4A
   Period=6.4ns
   Rising=0ns
   Falling=3.2ns

5. Update Timing Netlist (Netlist ¿ Create Timing Netlist)
   (see blazepps
   textunderscore lab.sdc for a full description of the settings)

3.1.8 10GbE MAC

3.1.9 On-Chip Memory FIFOs

In Qsys, create two On-Chip Memory FIFOs (Memories and Memory Controllers ¿ On-Chip ¿ On-Chip FIFO Memory) with the following settings:

Tx On-Chip Memory FIFO
SingledResetMode selected
Depth = 1024
Allow backpressure selected
Clock setting: Single clock mode
FIFO implementation: construct FIFO from embedded memory blocks
Enable IRQ for status ports selected
Input type: AVALONMM_WRITE
Output type: AVALONST_SOURCE
Data width = 32
Bits per symbol: 8
Symbols per beat: 4
Error width: 0
Channel width: 0
Enabled packet data selected

Rx On-Chip Memory FIFO
SingledResetMode selected

Depth = 1024
Allow backpressure selected
Clock setting: Single clock mode FIFO implementation: construct FIFO from embedded memory blocks
Enable IRQ for status ports selected
Input type: AVALONST_SINK
Output type: AVALONMM_READ
Data width = 32
Bits per symbol: 8
Symbols per beat: 4
Error width: 0
Channel width: 0
Enabled packet data selected

Tx Dual-Clock FIFO
Symbols per beat: 8
Bit per symbol: 8
FIFO depth: 8192
Channel width: 0
Error width: 0
Use packets selected
Write pointer synchronizer length: 2
Read pointer synchronizer length: 2
Explicit maxChannel: 0

Rx Dual-Clock FIFO
Symbols per beat: 8
Bit per symbol: 8
FIFO depth: 8192
Channel width: 0
Error width: 0
Use packets selected
Write pointer synchronizer length: 2
Read pointer synchronizer length: 2
Explicit maxChannel: 0

The clocks and clock resets for the FIFOs should be a connected to the main clock and clock reset.

For the Tx OC FIFO, the data out should be connected to the data in of the Tx OC FIFO.

For the Rx OC FIFO, the data in should be connected to the data out of the Rx DC FIFO.

3.1.10 Dual-Clock FIFOs

In Qys, create two Dual-Cock FIFOs (Memories and Memory Controllers ¿ On-Chip ¿ Avalon-ST DC FIFO) with the following settings:

Tx DC FIFO
Symbols per beat: 8
Bit per symbol: 8
FIFO depth: 8192
Channel width: 0
Error width: 0
Use packets selected
Write pointer synchronizer length: 2
Read pointer synchronizer length: 2
Explicit maxChannel: 0

Rx DC FIFO
Symbols per beat: 8
Bit per symbol: 8
FIFO depth: 8192
Channel width: 0
Error width: 0
Use packets selected
Write pointer synchronizer length: 2
Read pointer synchronizer length: 2
Explicit maxChannel: 0

For the Tx DC FIFO, the in clock and clock reset should be connected to the main clock and clock reset and the out clock and clock reset should be connected to the XAUI clock and clock reset. Additionally, the data in should be connected to the data out of the Tx OC FIFO and the data out should be exported so it can connect to the Tx Buffer.

For the Rx DC FIFO, the in clock and clock reset should be connected to the xaui clock and clock reset and the out clock should be connected to the mail clock and clock reset. Additionally, the data in should be exported so it can be connected to the data out of the MAC Rx and the data out should be connected to the data in of the Rx OC FIFO. Memories and Memory Controllers ¿ On-Chip ¿ Avalon-ST DC FIFO

3.1.11 Loopbacks

3.1.12 Resets

The following resets are very important for making the Dual XAUI and FPGA components work correctly.

PHYRESET
Resets the Dual XAUI, more specifically the BCM8727 (see Dual XAUI Logic & Pin Assignments section to determine which pin it should be connected to)
Active low
KEY[1] is assigned to it.

RESET_N
Resets XAUI PHY, MAC, XGMII Swap, Tx Buffer, and xaui clock.
Active low (XAUI PHY, XGMII Swap, and Tx Buffer are reset with active high, so !RESET_N was used).
Logic is used so that KEY[0] is effectively assigned to it.

3.2 Software

3.2.1 Boot Server

A boot server can be used to load and boot a Linux image which, in turn, configures the FPGA board.

Setting Up Boot Server

1. The first stage bootloader loads from onboard serial flash, resets the processor, configures the memory controller, and loads the second stage bootloader.
2. The second stage bootloader sets the boards Ethernet address, uses dhcp to obtain the boards ip address and the address of the tftp/nfs server. The second stage bootloader uses tftp to download pxelinux.cfg/default, which contains paths to kernel images, fpga files, and device tree files.
3. The user selects one of these groups by typing on the console.
4. The second stage bootloader uses tftp to download the kernel image, the device tree, and fpga bitstream file, programs the FPGA, and passes control to the Linux kernel.
5. The Linux kernel enumerates and initializes devices, mounts the root directory with nfs, then runs /sbin/init, which runs the initialization scripts in /etc/init.d/, eventually running a getty instance, which prompts for a username and password.

In our case, we have one server machine that stores the Linux images and FPGA configuration files, and another machine which actually communicates with the board through USB connection for JTAG and debug purposes. [TODO: add schematic of the connections with the two machines - marvin and thyme (server) when you have a good mouse to draw the schematic!]

Part of the setting up of the communication between the server and the board involves using the NFS protocol in order to remotely mount the filesystems. This allows the SoCKit boards to store their root directories (i.e. the entire filesystem) on the server, and also allows us to share that filesystem across several machines. In particular, in our case, we used this protocol to share the filesystem existing on the server machine with the machine which our board is connected to (via JTAG connection). The steps to follow for this setup are the as follows (for a Linux CentOS distribution):

1. On both host (server) and client machines, install the nfs-utils and nfs-utils-lib (yum install on CentOS) packages and run the following commands:

```
$ chkconfig nfs on
$ service rpcbind start
$ service nfs start
```

The configuration file for NFS is /etc/exports. We export the root filesystems to the SoCKit boards and allow the workstations to also mount the relevant directory so that files, such as the FPGA configuration file, can be updated.

```
# Export root filesystems to the SoCKit board
/export/labs 192.168.1.*(rw,no_root_squash,no_subtree_check)

# Export to the workstation (client machine)
/export/labs 128.59.19.118/255.255.248.0(rw,sync,no_root_squash)
```

In the above example 128.59.19.118 is the IP address of our client machine with netmask 255.255.248.0.

2. After updating this configuration file, run the exportfs command (on the server):

```
$ exportfs -a
```

3. At this step, run the mount command on the client machine in order to mount NFS on the client side:

```
$ mkdir -p /nfs
$ mount -t nfs 128.59.20.169:/export/labs /nfs/
```

In the above example 128.59.20.169 is the IP address of the sever and /export/labs is the remote (i.e. on the server) directoy to share with the client. Finally /nfs is a local (i.e. on the client) directory that we create to hold the shared file system on the our local (client) machine.

In case the mount command leads to a Connection timed out error it is good practice to think of checking that the different services (especially nfs) are running.
If the problem persists, it can be useful to consider switching on and off the server and/or the client machines and repeat the process.

```
$ service nfs status
```

Booting From the Server
Once the root directories (i.e. the entire filesystem) is shared between the SoCKit board and our client machine, then we can change the files in the /nfs directory and see these changes directly reflected on the board. At this point, we can connect to the board using the screen command and compile the files that we have created/modified on the shared filesystem on the client machine.

```
$ screen /dev/ttyUSB01 57600
$ BlazePPS
```

3.3 Golden System Reference Design

An alternative to booting Linux and configuring the FPGA from the network is to boot it from an SD card and configure/program the FPGA manually (i.e. through Quartus). Rocketboards provides what is known as the Golden System Reference Design (GSRD) to facilitate this. Rocketboards describes the GSRD as a set of essential hardware and software system components that can be used as a starting point for various custom user designs. Directions related to obtaining and using the GSRD can be found at http://www.rocketboards.org/foswiki/Documentation/GSRD131#Introduction.

Often, the partition of the SD card used for the Linux image is too small to store the kernel source tree, which means that you cannot compile drivers on it. To deal with this you can cross-compile the driver on anothe Linux system using the proper kernel source tree. Currently, the GSRD uses Linux 3.9.0. Below are the directions for cross-compilation.

Obtaining Linux 3.9.0 source tree
1) In a terminal, navigate to the directory you want to keep the source tree
2) Clone the source tree

```
$ git clone git://git.rocketboards.org/linux-socfpga.git source
```

3) Checkout the 3.9.0 branch

```
$ git checkout 5f4c1c02c48f8685c0bd777395bad97112389df0
```

Preparing source tree for cross compilation 4) In a terminal, navigate to the root of the source tree (going forward, we refer to the root directory as Linux)
5) Run make clean
6) Run make ARCH=arm socfpga_defconfig
7) Run make ARCH=arm CROSS_COMPILE=/usr/local/DS-5/bin/arm_linux_gnueabinf modules
8) In Linux/include, create a symbolic link to asm-generic called asm

```
$ sudo ln -s asm-generic asm
```

9) In Linux/include/uapi, create a symbolic link to asm-generic called asm

```
$ sudo ln -s asm-generic asm
```

Program Cross Compilation

```
$ /usr/local/DS-5/bin/arm_linux_gnueabinf-gcc -c test.c
$ /usr/local/DS-5/bin/arm_linux_gnueabinf-gcc -o test test.o
```

Driver Cross Compilation

```
${MAKE} -C ${KDIR} ${PWD} ARCH=arm \
CROSS_COMPILE=/usr/local/DS-5/bin/arm_linux_gnueabinf modules
```

The MAKE variable is used to make a recursive call to make. KDIR should be the root directory of the Linux 3.9.0 source tree. PWD should be your current directory.

3.4 Useful Resources

Terasic SoCKit

```
http://www.terasic.com.tw/cgi-bin/page/archive.pl?
Language=English&CategoryNo=165&No=816&PartNo=4
```

Altera Cyclone V

```
https://www.altera.com/content/dam/altera-www/global/en\textunderscore US/pdfs/
literature/hb/cyclone-v/cyclone5\textunderscore handbook.pdf

https://www.altera.com/content/dam/altera-www/global/en\textunderscore US/pdfs/
literature/hb/cyclone-v/cv\textunderscore 5v3.pdf

https://www.altera.com/en\textunderscore US/pdfs/literature/hb/cyclone-v/cv\textunderscore 5v4
```

Dual XAUI HSMC Card

```
http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=71&No=597&PartNo=
```

HSMC Interface

```
https://www.altera.com/en\textunderscore US/pdfs/literature/ds/hsmc\textunderscore spec.pdf
```

Altera Transceiver IP Cores

```
https://www.altera.com/en\textunderscore US/pdfs/literature/ug/xcvr\textunderscore user\textunc
\end{Verbatim}
\textbf{Altera 10GbE}
\begin{verbatim}
https://www.altera.com/content/dam/altera-www/global/en\textunderscore US/pdfs/literature/ug/1
```

4 Lessons Learned

4.1 Valeh Valiollahpour Amiri

Our project was rather ambitious and challenging on multiple points... and this is actually what I appreciated most about it since it made me learn and explore a lot both in terms of hardware design (and extensive use of Quartus CAD tools), and also in terms of software. Especially, apart from practicing more in depth what we already learned in class, I particularly appreciated gaining more in-depth knowledge about network-oriented designs and Linux network drivers.

In terms of challenges we faced, I think one of the main challenges that our project involved was to start off of an existing project which worked only to some extent. This required to take considerable time to understand their design and trying to figure out what made it not work on some points. Plus, spending a lot of time on debugging (or trying to) another design turned out to be too time-consuming, somewhat complicated and little useful to us. This is why we (thankfully) decided to instead concentrate mostly on our design from then on. This made me remember how difficult and time-consuming it can to be to try to fix others' designs.

The second challenge (which I guess is more specific to our project) is that it involved a large amount of new concepts and which required reading a lot of technical documents first in order to learn how things worked but also (and especially) why they sometimes did not (which is more painful in term of finding the relevant information)! But personally I appreciated this feature most since it made me explore new concepts and learn a lot of useful knowledge and know-how.

Finally, In terms of teamwork, I guess it goes without saying that human-human communication is sometimes much more complex than human-machine communication! It can be hard to make oneself understood, to split tasks, and to agree on what to do and what not to do. In spite of all this, I really think it is much more pleasant to work in team than individually! Also, I appreciate learning from the others on some points. And I believe the earlier we get to deal with teamwork the more we get prepared for what we will be facing soon in "real-world".

In terms of contribution, I contributed to the three deliverables (proposal, design, final report), to the hardware design and implementation, and to the implementation of the Linux drivers (memory-mapped char driver and base network driver).

4.2 Christopher Campbell

Many aspects of this project were challenging and there were many lessons learned. Our initial idea was to extend a project from a previous semester, but that turned out to be very difficult. Interpreting their documentation and determining when they were right and when they were wrong was extremely time consuming and challenging. In fact, it was so challenging that we eventually decided to re-implement the system from scratch. Although this was also challenging, I found it much less difficult than attempting to understand and interface with someone's elses work. Additionally, it gave me a much better understanding of the system, which, of course, proved to be useful when we tried to extend it. I truly learned that it is much better to try to fix your own work than it is to try to fix somebody else's.

Another challenge we had with the project was sticking with a goal. Our goal changed several times throughout the project in response to our changing interests, difficulties we encountered, and outside advice. This taught me that you really need to spend a substantial amount of time evaluating a goal before you attempt to achieve it. And, once you decide on a goal you should be confident in it and see it through unless there are extremely compelling reasons to do otherwise.

A third challenge that we had, often, was coming to a group consensus. Four people is enough to create enormous differences in opinion and it was often difficult to agree on project goals, methods, etc. In fact, differences in opinion played a direct role in the problems we had sticking with a goal. The challenges we had coming to a consensus made me learn that even small groups may be benefit from a benevolent dictator.

I contributed to the proposal, design, hardware implementation, network driver (as well as the simple MM and network driver preceding it), and the final report.

4.3 Sheng Qian

The Programming on FPGA is quite a difficult thing, especially the debugging. Every module of the system can cause the failure. In my view, the on-chip FIFO should be one of the simplest modules in our system. But we faced a lot of failures to combine it with other modules, such as double clock FIFO and the 10G EMAC module. It worked fine with the simple MAC module, but it caused failure of sending packets when we tried to use the 10G EMAC module for some reason. After trying a lot of debugging, the problem still couldn't be fixed. In the project, the biggest lesson I learn is the debugging method. It is always difficult to combine the modules together. When the system doesn't work, it is good to debug from simplest part step by step to get the cause of the problem, like using loopback to test the FIFOs, XAUI PHY module and EMAC module with different combinations to get which parts work well and which parts cause the problem.

Another lesson is to have a feasible plan. In the project, we tried many versions of system. When one version didn't work and we couldn't fix the problem, we used another version. It is quite time-consuming.

In the project, I contributed to the ethernet driver, simple memory-mapped driver, some part of the documentation and helped with debugging the system.

4.4 Yuanpei Zhang

Through working on the 10Gb/s packet processing project and doing relevant research, I gained a basic knowledge on networking systems(data transfer between physical layer and MAC layer). I focused on our hardware design which is based on Altera Golden Hardware Reference Design(GHRD) in the project, and provided assistance in utilizing Altera IP cores (especially PHY IP and 10GbE MAC) in our design. In order to continue the 10Gb/s packet processing project done by a team in previous semester, for the hardware part of our project, we have spend lots of time reading Altera IP core documents as well as other relevant documentations, from which I gained deep understanding of the PHY IP interfaces and configurations. Since in the previous project a customized MAC controller was used to transfer data between MAC layer and physical layer, we also worked on improving the networking system by using 10GbE MAC IP core with enabled data transfer feature. Our debugging process involves several techniques such as making loopbacks and using signaltab, which is significant learning experience for me. During this project I also get familiar with using Qsys and MegaWizard to create, instantiate and connect modules.

5 Files

5.1 BlazePPS v1 (Altera 10GbE MAC currently using loopback)

SoCKit_top.sv

```
// ============================================================================
// Copyright (c) 2013 by Terasic Technologies Inc.
// ============================================================================
//
// Permission:
//
//   Terasic grants permission to use and modify this code for use
//   in synthesis for all Terasic Development Boards and Altera Development
//   Kits made by Terasic.  Other use of this code, including the selling
//   ,duplication, or modification of any portion is strictly prohibited.
//
// Disclaimer:
//
//   This VHDL/Verilog or C/C++ source code is intended as a design reference
//   which illustrates how these types of functions can be implemented.
//   It is the user's responsibility to verify their design for
//   consistency and functionality through the use of formal
//   verification methods.  Terasic provides no warranty regarding the use
//   or functionality of this code.
//
// ============================================================================
//
//  Terasic Technologies Inc
//  9F., No.176, Sec.2, Gongdao 5th Rd, East Dist, Hsinchu City, 30070. Taiwan
//
//
//                     web: http://www.terasic.com/
//                     email: support@terasic.com
//
// ============================================================================
// ============================================================================
//
// Major Functions: SoCKit_Default
//
// ============================================================================
// Revision History :
// ============================================================================
//  Authors: Valeh Valiollahpour Amiri (vv2252@columbia.edu)
//           Christopher Campbell (cc3769@columbia.edu)
//           Sheng Qian (sq2168@columbia.edu)
//           Yuanpei Zhang (yz2727@columbia.edu)
//
```

```
//  Last Mod. Date: May 14th, 2015
// ============================================================================

//‘define ENABLE_HPS
‘define ENABLE_MDIO

module SoCKit_Top(

‘ifdef ENABLE_HPS
    /////////HPS/////////
    HPS_CLOCK_25,
    HPS_CLOCK_50,
    HPS_CONV_USB_n,
    HPS_DDR3_A,
    HPS_DDR3_BA,
    HPS_DDR3_CAS_n,
    HPS_DDR3_CKE,
    HPS_DDR3_CK_n,
    HPS_DDR3_CK_p,
    HPS_DDR3_CS_n,
    HPS_DDR3_DM,
    HPS_DDR3_DQ,
    HPS_DDR3_DQS_n,
    HPS_DDR3_DQS_p,
    HPS_DDR3_ODT,
    HPS_DDR3_RAS_n,
    HPS_DDR3_RESET_n,
    HPS_DDR3_RZQ,
    HPS_DDR3_WE_n,
    HPS_ENET_GTX_CLK,
    HPS_ENET_INT_n,
    HPS_ENET_MDC,
    HPS_ENET_MDIO,
    HPS_ENET_RESET_n,
    HPS_ENET_RX_CLK,
    HPS_ENET_RX_DATA,
    HPS_ENET_RX_DV,
    HPS_ENET_TX_DATA,
    HPS_ENET_TX_EN,
    HPS_FLASH_DATA,
    HPS_FLASH_DCLK,
    HPS_FLASH_NCSO,
    HPS_GSENSOR_INT,
    HPS_I2C_CLK,
    HPS_I2C_SDA,
    HPS_KEY,
    HPS_LCM_D_C,
    HPS_LCM_RST_N,
    HPS_LCM_SPIM_CLK,
    HPS_LCM_SPIM_MISO,
    HPS_LCM_SPIM_MOSI,
    HPS_LCM_SPIM_SS,
    HPS_LED,
    HPS_LTC_GPIO,
    HPS_RESET_n,
    HPS_SD_CLK,
    HPS_SD_CMD,
    HPS_SD_DATA,
    HPS_SPIM_CLK,
    HPS_SPIM_MISO,
    HPS_SPIM_MOSI,
    HPS_SPIM_SS,
    HPS_SW,
    HPS_UART_RX,
    HPS_UART_TX,
    HPS_USB_CLKOUT,
    HPS_USB_DATA,
    HPS_USB_DIR,
    HPS_USB_NXT,
    HPS_USB_RESET_PHY,
    HPS_USB_STP,
    HPS_WARM_RST_n,
‘endif /*ENABLE_HPS*/

    /////////KEY/////////
    KEY,

    /////////LED/////////
    LED,

    /////////OSC/////////
    OSC_50_B3B,
    OSC_50_B4A,
    OSC_50_B5B,
    OSC_50_B8A,

    /////////RESET/////////
    RESET_n,

    /////////SI5338/////////
    SI5338_SCL,
    SI5338_SDA,

    ///////// SW /////////
    SW,
    ///////////hps//////////
    memory_mem_a,
    memory_mem_ba,
    memory_mem_ck,
    memory_mem_ck_n,
    memory_mem_cke,
    memory_mem_cs_n,
    memory_mem_ras_n,
    memory_mem_cas_n,
    memory_mem_we_n,
    memory_mem_reset_n,
    memory_mem_dq,
    memory_mem_dqs,
    memory_mem_dqs_n,
    memory_mem_odt,
    memory_mem_dm,
    memory_oct_rzqin,
    hps_io_hps_io_emac1_inst_TX_CLK,
    hps_io_hps_io_emac1_inst_TXD0,
    hps_io_hps_io_emac1_inst_TXD1,
    hps_io_hps_io_emac1_inst_TXD2,
    hps_io_hps_io_emac1_inst_TXD3,
    hps_io_hps_io_emac1_inst_RXD0,
    hps_io_hps_io_emac1_inst_MDIO,
    hps_io_hps_io_emac1_inst_MDC,
    hps_io_hps_io_emac1_inst_RX_CTL,
    hps_io_hps_io_emac1_inst_TX_CTL,
    hps_io_hps_io_emac1_inst_RX_CLK,
    hps_io_hps_io_emac1_inst_RXD1,
    hps_io_hps_io_emac1_inst_RXD2,
    hps_io_hps_io_emac1_inst_RXD3,
    hps_io_hps_io_qspi_inst_IO0,
    hps_io_hps_io_qspi_inst_IO1,
    hps_io_hps_io_qspi_inst_IO2,
    hps_io_hps_io_qspi_inst_IO3,
    hps_io_hps_io_qspi_inst_SS0,
    hps_io_hps_io_qspi_inst_CLK,
    hps_io_hps_io_sdio_inst_CMD,
    hps_io_hps_io_sdio_inst_D0,
    hps_io_hps_io_sdio_inst_D1,
    hps_io_hps_io_sdio_inst_CLK,
    hps_io_hps_io_sdio_inst_D2,
    hps_io_hps_io_sdio_inst_D3,
    hps_io_hps_io_usb1_inst_D0,
    hps_io_hps_io_usb1_inst_D1,
    hps_io_hps_io_usb1_inst_D2,
    hps_io_hps_io_usb1_inst_D3,
    hps_io_hps_io_usb1_inst_D4,
    hps_io_hps_io_usb1_inst_D5,
    hps_io_hps_io_usb1_inst_D6,
    hps_io_hps_io_usb1_inst_D7,
    hps_io_hps_io_usb1_inst_CLK,
    hps_io_hps_io_usb1_inst_STP,
    hps_io_hps_io_usb1_inst_DIR,
    hps_io_hps_io_usb1_inst_NXT,
    hps_io_hps_io_spim0_inst_CLK,
    hps_io_hps_io_spim0_inst_MOSI,
    hps_io_hps_io_spim0_inst_MISO,
    hps_io_hps_io_spim0_inst_SS0,
    hps_io_hps_io_spim1_inst_CLK,
    hps_io_hps_io_spim1_inst_MOSI,
    hps_io_hps_io_spim1_inst_MISO,
    hps_io_hps_io_spim1_inst_SS0,
    hps_io_hps_io_uart0_inst_RX,
    hps_io_hps_io_uart0_inst_TX,
    hps_io_hps_io_i2c1_inst_SDA,
    hps_io_hps_io_i2c1_inst_SCL,
    hps_io_hps_io_gpio_inst_GPIO00,

    ////////////HMSC///////////////////////////
    HSMC_CLKIN_p,
 HSMC_XAUI_RX_p0,
 HSMC_XAUI_TX_p0,
    MDC2,
    MDIO2,
    MDC1,
    MDIO1,
    PRTAD02,
    PRTAD01,
    PRTAD4,
    PRTAD3,
    PRTAD2,
    PRTAD1,
    TXONOFF1,
    TXONOFF2,
    OPINLVL,
    OPOUTLVL,
    PHYRESET,
    USER_LED_G,
    USER_LED_R,
    CONFIG1_1,
    CONFIG0_1,
    CONFIG1_2,
    CONFIG0_2,
    SS338_CLKIN,
    GPIO0_1,
    GPIO1_1,
    GPIO0_2,
    GPIO1_2,
    SER_BOOT,
    SMBSPDSEL1,
    SMBSPDSEL2,
    SMBWEN,
    NVMA1SEL,
    NVMPROT,
    OPRXLOS2,
    OPTXFLT2,
    SFP_TXDIS2,
    SFP_TXRS20,
    LASI2
);


//=======================================================
//  PORT declarations
//=======================================================

‘ifdef ENABLE_HPS
    ///////// HPS /////////
    input         HPS_CLOCK_25;
    input         HPS_CLOCK_50;
    input         HPS_CONV_USB_n;
    output [14:0] HPS_DDR3_A;
    output [2:0]  HPS_DDR3_BA;
    output        HPS_DDR3_CAS_n;
    output        HPS_DDR3_CKE;
    output        HPS_DDR3_CK_n;
    output        HPS_DDR3_CK_p;
    output        HPS_DDR3_CS_n;
    output [3:0]  HPS_DDR3_DM;
    inout  [31:0] HPS_DDR3_DQ;
    inout  [3:0]  HPS_DDR3_DQS_n;
    inout  [3:0]  HPS_DDR3_DQS_p;
    output        HPS_DDR3_ODT;
    output        HPS_DDR3_RAS_n;
    output        HPS_DDR3_RESET_n;
    input         HPS_DDR3_RZQ;
    output        HPS_DDR3_WE_n;
    input         HPS_ENET_GTX_CLK;
    input         HPS_ENET_INT_n;
    output        HPS_ENET_MDC;
    inout         HPS_ENET_MDIO;
    output        HPS_ENET_RESET_n;
    input         HPS_ENET_RX_CLK;
    input  [3:0]  HPS_ENET_RX_DATA;
    input         HPS_ENET_RX_DV;
    output [3:0]  HPS_ENET_TX_DATA;
    output        HPS_ENET_TX_EN;
    inout  [3:0]  HPS_FLASH_DATA;
    output        HPS_FLASH_DCLK;
    output        HPS_FLASH_NCSO;
    input         HPS_GSENSOR_INT;
    inout         HPS_I2C_CLK;
    inout         HPS_I2C_SDA;
    inout  [3:0]  HPS_KEY;
    output        HPS_LCM_D_C;
    output        HPS_LCM_RST_N;
    input         HPS_LCM_SPIM_CLK;
    inout         HPS_LCM_SPIM_MISO;
    output        HPS_LCM_SPIM_MOSI;
    output        HPS_LCM_SPIM_SS;
    output [3:0]  HPS_LED;
    inout         HPS_LTC_GPIO;
    input         HPS_RESET_n;
    output        HPS_SD_CLK;
    inout         HPS_SD_CMD;
    inout  [3:0]  HPS_SD_DATA;
    output        HPS_SPIM_CLK;
    input         HPS_SPIM_MISO;
    output        HPS_SPIM_MOSI;
    output        HPS_SPIM_SS;
    input  [3:0]  HPS_SW;
    input         HPS_UART_RX;
    output        HPS_UART_TX;
    input         HPS_USB_CLKOUT;
    inout  [7:0]  HPS_USB_DATA;
    input         HPS_USB_DIR;
    input         HPS_USB_NXT;
    output        HPS_USB_RESET_PHY;
    output        HPS_USB_STP;
    input         HPS_WARM_RST_n;
`endif /*ENABLE_HPS*/

    ///////// KEY /////////
    input [3:0] KEY;

    ///////// LED /////////
    output [3:0] LED;

    ///////// OSC /////////
    input OSC_50_B3B;
    input OSC_50_B4A;
    input OSC_50_B5B;
    input OSC_50_B8A;


    ///////// RESET /////////
    input RESET_n;

    ///////// SI5338 /////////
    inout SI5338_SCL;
    inout SI5338_SDA;

    ///////// SW /////////
    input [3:0] SW;

    /////////hps pin///////
    output wire [14:0] memory_mem_a;
    output wire [2:0]  memory_mem_ba;
    output wire        memory_mem_ck;
    output wire        memory_mem_ck_n;
    output wire        memory_mem_cke;
    output wire        memory_mem_cs_n;
    output wire        memory_mem_ras_n;
    output wire        memory_mem_cas_n;
    output wire        memory_mem_we_n;
    output wire        memory_mem_reset_n;
    inout  wire [31:0] memory_mem_dq;
    inout  wire [3:0]  memory_mem_dqs;
    inout  wire [3:0]  memory_mem_dqs_n;
    output wire        memory_mem_odt;
    output wire [3:0]  memory_mem_dm;
    input  wire        memory_oct_rzqin;
    output wire        hps_io_hps_io_emac1_inst_TX_CLK;
    output wire        hps_io_hps_io_emac1_inst_TXD0;
    output wire        hps_io_hps_io_emac1_inst_TXD1;
    output wire        hps_io_hps_io_emac1_inst_TXD2;
    output wire        hps_io_hps_io_emac1_inst_TXD3;
    input  wire        hps_io_hps_io_emac1_inst_RXD0;
    inout  wire        hps_io_hps_io_emac1_inst_MDIO;
    output wire        hps_io_hps_io_emac1_inst_MDC;
    input  wire        hps_io_hps_io_emac1_inst_RX_CTL;
    output wire        hps_io_hps_io_emac1_inst_TX_CTL;
    input  wire        hps_io_hps_io_emac1_inst_RX_CLK;
    input  wire        hps_io_hps_io_emac1_inst_RXD1;
    input  wire        hps_io_hps_io_emac1_inst_RXD2;
    input  wire        hps_io_hps_io_emac1_inst_RXD3;
    inout  wire        hps_io_hps_io_qspi_inst_IO0;
    inout  wire        hps_io_hps_io_qspi_inst_IO1;
    inout  wire        hps_io_hps_io_qspi_inst_IO2;
```

```
    inout  wire        hps_io_hps_io_qspi_inst_IO3;
    output wire        hps_io_hps_io_qspi_inst_SS0;
    output wire        hps_io_hps_io_qspi_inst_CLK;
    inout  wire        hps_io_hps_io_sdio_inst_CMD;
    inout  wire        hps_io_hps_io_sdio_inst_D0;
    inout  wire        hps_io_hps_io_sdio_inst_D1;
    output wire        hps_io_hps_io_sdio_inst_CLK;
    inout  wire        hps_io_hps_io_sdio_inst_D2;
    inout  wire        hps_io_hps_io_sdio_inst_D3;
    inout  wire        hps_io_hps_io_usb1_inst_D0;
    inout  wire        hps_io_hps_io_usb1_inst_D1;
    inout  wire        hps_io_hps_io_usb1_inst_D2;
    inout  wire        hps_io_hps_io_usb1_inst_D3;
    inout  wire        hps_io_hps_io_usb1_inst_D4;
    inout  wire        hps_io_hps_io_usb1_inst_D5;
    inout  wire        hps_io_hps_io_usb1_inst_D6;
    inout  wire        hps_io_hps_io_usb1_inst_D7;
    input  wire        hps_io_hps_io_usb1_inst_CLK;
    output wire        hps_io_hps_io_usb1_inst_STP;
    input  wire        hps_io_hps_io_usb1_inst_DIR;
    input  wire        hps_io_hps_io_usb1_inst_NXT;
    output wire        hps_io_hps_io_spim0_inst_CLK;
    output wire        hps_io_hps_io_spim0_inst_MOSI;
    input  wire        hps_io_hps_io_spim0_inst_MISO;
    output wire        hps_io_hps_io_spim0_inst_SS0;
    output wire        hps_io_hps_io_spim1_inst_CLK;
    output wire        hps_io_hps_io_spim1_inst_MOSI;
    input  wire        hps_io_hps_io_spim1_inst_MISO;
    output wire        hps_io_hps_io_spim1_inst_SS0;
    input  wire        hps_io_hps_io_uart0_inst_RX;
    output wire        hps_io_hps_io_uart0_inst_TX;
    inout  wire        hps_io_hps_io_i2c1_inst_SDA;
    inout  wire        hps_io_hps_io_i2c1_inst_SCL;
    inout  wire        hps_io_hps_io_gpio_inst_GPIO00;


    ////////HSMC///////////////////////
    input              HSMC_CLKIN_p;
    input  logic [3:0] HSMC_XAUI_RX_p0;
output logic [3:0] HSMC_XAUI_TX_p0;
    output logic       MDC2;
    inout  logic       MDIO2;
    output logic       MDC1;
    inout  logic       MDIO1;
    output logic       PRTAD02;
    output logic       PRTAD01;
    output logic       PRTAD4;
    output logic       PRTAD3;
    output logic       PRTAD2;
    output logic       PRTAD1;
    output logic       TXONOFF1;
    output logic       TXONOFF2;
    output logic       OPINLVL;
    output logic       OPOUTLVL;
    output logic       PHYRESET;
    output logic [7:0] USER_LED_G;
    output logic [7:0] USER_LED_R;
    output logic       CONFIG1_1;
    output logic       CONFIG0_1;
    output logic       CONFIG1_2;
    output logic       CONFIG0_2;
    output logic       SS338_CLKIN;
    inout  logic       GPIO0_1;
    inout  logic       GPIO1_1;
    inout  logic       GPIO0_2;
    inout  logic       GPIO1_2;
    output logic       SER_BOOT;
    output logic       SMBSPDSEL1;
    output logic       SMBSPDSEL2;
    inout  logic       SMBWEN;
    inout  logic       NVMA1SEL;
    inout  logic       NVMPROT;
    input  logic       OPRXLOS2;
    input  logic       OPTXFLT2;
    input  logic       SFP_TXDIS2;
    input  logic       SFP_TXRS20;
    input  logic       LASI2;


    //=======================================================
    //  REG/WIRE declarations
    //=======================================================

    // MAC
    logic mac_tx_ready;
    logic mac_rx_ready;

    // Make the FPGA reset cause an HPS reset
    reg [19:0]   hps_reset_counter = 20'h0;
    reg          hps_fpga_reset_n = 0;
    wire         MDIN1;
    wire         MDOEN1;
    wire         MDO1;
    logic        RESET_N;                    // S5 active low
    wire         clk_buff0;
    wire         clk_buff1;
    wire         clk_buff2;
    wire  [71:0]  link;
    wire         link_clk;
    logic [71:0] xgmii_buff ;
    logic [71:0] xgmii_rx_dc;
    logic [71:0] xgmii_tx_dc;
    logic [71:0] xgmiialigned;
    logic [31:0] crclink;
    logic        eop;
    logic        sop;
    logic        valid;
    logic        ready;
    logic        checksum_err;
    logic [2:0]  empty;
    logic [63:0] data;
    logic [1:0]  state;
    logic        eoptx;
    logic        soptx;
    logic        readytx;
    logic [2:0]  emptytx;
    logic [63:0] stdata;
    logic        validtx;
    logic [31:0] crclinktx;
    logic [31:0] cctx;
    logic [63:0] xgmiirevtx;
    logic [71:0] xgmiitx;
    logic [71:0] tx_buff;
    logic [63:0] databuff;
    logic        validbuff;
    logic        readybuff;
    logic        sopbuff;
    logic        eopbuff;
    logic [2:0]  emptybuff;
    parameter idle = 0, trig =1, lock =2;


    //=======================================================
    //  Structural coding
    //=======================================================


    // +-------------------------------------------------------------------
    // +
    assign CONFIG0_1 = 1’b1; // Configure BCM8727 from EEPROM
    assign CONFIG1_1 = 1’b0;
    assign CONFIG0_2 = 1’b1;
    assign CONFIG1_2 = 1’b0;
    assign SS338_CLKIN = 1’b0;
```

```
    // BCM8272C allow spi-rom to be removed
    // 1 = Boot microcode from spi proms
    assign SER_BOOT  = 1’b0;
    assign SMBSPDSEL1 = 1’b0;
    assign SMBSPDSEL2 = 1’b0;
    assign SMBWEN    = 1’b1;
    assign GPIO0_1   = 1’b0;
    assign GPIO1_1   = 1’b0;
    assign GPIO0_2   = 1’b0;
    assign GPIO1_2   = 1’b0;


    // ------------------------------------------------------------------
    // 1: EEPROM Slave Addr 52, 0: 50 addr:
    // During deassertion of BCM8727 reset,
    // latched into bit 10 of register 1.8002h
    assign NVMA1SEL = 1’b1;

    // when high protect non volatile memory
    assign NVMPROT  = 1’b0;

    // MDIO ports connection
‘ifdef ENABLE_MDIO
    assign MDIO1 = !MDOEN1? MDO1 : 1’bz;
    assign MDIN1 = MDIO1;
‘else
    assign MDC1  = 1’bz;
    assign MDIO1 = 1’bz;
‘endif
    assign MDC2  = 1’bz;
    assign MDIO2 = 1’bz;

    // +-------------------------------------------------------------------
    // +TXONOFF2


    assign PHYRESET =  (KEY[1]); // S4 active low
    // assign STOPMON  = ~KEY[2]; // S3 active high

    assign {PRTAD4,PRTAD3,PRTAD2,PRTAD1,PRTAD01} = 5’b00000;
    assign  PRTAD02 = 1’b0;
    // +
    assign TXONOFF1 = 1’b1;
    assign TXONOFF2 = 1’b1;
    assign OPOUTLVL = 1’b0; // 0 for active low OPTXENB/OPTXRST
    assign OPINLVL  = 1’b1; // 1 for active high OPRXLOS/TXONOFF
    assign USER_LED_R = 8’b00001111;
    assign USER_LED_G = 8’b11110000;

    //assign LED[0] = SFP_TXRS20;
    //assign LED[1] = SFP_TXDIS2;
    assign LED[0] = mac_tx_ready;
    assign LED[1] = mac_rx_ready;

    always @(posedge OSC_50_B4A) begin
        if (hps_reset_counter == 20’h ffffff) hps_fpga_reset_n <= 1;
            hps_reset_counter <= hps_reset_counter + 1;
end

    // Logic for resetting modules using KEY[0]
always @(posedge OSC_50_B4A)
        begin
            case(state)
                idle:
                    if (KEY[0]==0)
                        state <= trig;
                    else
                        state <= idle;
                trig:
                    if (KEY[0]==0)
                        state <= lock;
                    else
                        state <= idle;
                lock:
                    if (KEY[0]==0)
                        state <= lock;
                    else
                        state <= idle;
endcase
end
always @(state)
        begin
            case (state)
                idle:
                    RESET_N <= 1;
                trig:
RESET_N <= 0;
lock:
RESET_N <= 1;
endcase
end

//=======================================================
//  Modules Connection
```

```
//======================================================

// PLL REFERNCE  CLOCK FOR  XAUI
pll p1(.refclk(OSC_50_B4A), .rst(0), .outclk_0(clk_buff0), .locked());

// XAUI  PHY
xaui test(
.pll_ref_clk          (clk_buff0),
.xgmii_tx_clk         (link_clk),
.xgmii_rx_clk         (link_clk),
.xgmii_rx_dc          (xgmii_rx_dc),
.xgmii_tx_dc          (tx_buff),
.xaui_rx_serial_data  (HSMC_XAUI_RX_p0),
.xaui_tx_serial_data  (HSMC_XAUI_TX_p0),
.rx_ready       (LED[2]),
.tx_ready   (LED[3]),
.phy_mgmt_clk         (OSC_50_B4A),
.phy_mgmt_clk_reset   (!RESET_N),
.phy_mgmt_address     (3'h000),
.phy_mgmt_read        (0),
.phy_mgmt_readdata    (8'hFFFFFFFF),
.phy_mgmt_write       (0),
.phy_mgmt_writedata   (8'h00000000),
.phy_mgmt_waitrequest (0),
.reconfig_from_xcvr   (),
.reconfig_to_xcvr     ()
);


// TX  BUFFER
txbuffer buffer(
.clk(link_clk),
.reset(!RESET_N),
.stdata(stdata),
.valid(validtx),
.sopin(soptx),
.eopin(eoptx),
.emptyin(emptytx),

.databuff(databuff),
.validout(validbuff),
.readybuff(readybuff),
.sopout(sopbuff),
.eopout(eopbuff),
.emptybuff(emptybuff),
.readytx(readytx)
);
```

```
// LOOPBACKS
/*
// XAUI TX LOOPBACK
xaui_tx_loopback xaui_tx_loop (
    .clk(link_clk),
    .tx_out(HSMC_XAUI_TX_p0),
    .rx_in(HSMC_XAUI_RX_p0,)
);

// XAUI RX LOOPBACK
xaui_rx_loopback xaui_rx_loop (
    .clk(link_clk),
    .rx_out(xgmii_rx_dc),
    .tx_in(tx_buff)
);

// Tx Buffer TX Loopback
txbuff_tx_loopback tx_buff_loop (
    .clk(link_clk),
    txbuff_out(databuff),
    rx_dc_in(data)
);

// Tx Buffer RX Loopback
txbuff_tx_loopback tx_buff_loop (
    .clk(link_clk),
    rx_mac_out(data),
    txbuff_in(stdata)
);
*/

////////////////QSYS:MM/ST  CONVERTER & HPS//////////..////
blazepps u0 (
.clk_clk                                  (OSC_50_B4A),
.reset_reset_n                            (hps_fpga_reset_n),
.memory_mem_a                             (memory_mem_a),
.memory_mem_ba                            (memory_mem_ba),
.memory_mem_ck                            (memory_mem_ck),
.memory_mem_ck_n                          (memory_mem_ck_n),
.memory_mem_cke                           (memory_mem_cke),
.memory_mem_cs_n                          (memory_mem_cs_n),
.memory_mem_ras_n                         (memory_mem_ras_n),
.memory_mem_cas_n                         (memory_mem_cas_n),
.memory_mem_we_n                          (memory_mem_we_n),
.memory_mem_reset_n                       (memory_mem_reset_n),
.memory_mem_dq                            (memory_mem_dq),
.memory_mem_dqs                           (memory_mem_dqs),
.memory_mem_dqs_n                         (memory_mem_dqs_n),
.memory_mem_odt                           (memory_mem_odt),
.memory_mem_dm                            (memory_mem_dm),
.memory_oct_rzqin                         (memory_oct_rzqin),
.hps_io_hps_io_emac1_inst_TX_CLK       (hps_io_hps_io_emac1_inst_TX_CLK),
.hps_io_hps_io_emac1_inst_TXD0       (hps_io_hps_io_emac1_inst_TXD0),
.hps_io_hps_io_emac1_inst_TXD1         (hps_io_hps_io_emac1_inst_TXD1),
.hps_io_hps_io_emac1_inst_TXD2     (hps_io_hps_io_emac1_inst_TXD2),
.hps_io_hps_io_emac1_inst_TXD3     (hps_io_hps_io_emac1_inst_TXD3),
.hps_io_hps_io_emac1_inst_RXD0     (hps_io_hps_io_emac1_inst_RXD0),
.hps_io_hps_io_emac1_inst_MDIO     (hps_io_hps_io_emac1_inst_MDIO),
.hps_io_hps_io_emac1_inst_MDC      (hps_io_hps_io_emac1_inst_MDC),
.hps_io_hps_io_emac1_inst_RX_CTL   (hps_io_hps_io_emac1_inst_RX_CTL),

.hps_io_hps_io_emac1_inst_TX_CTL    (hps_io_hps_io_emac1_inst_TX_CTL),

.hps_io_hps_io_emac1_inst_RX_CLK    (hps_io_hps_io_emac1_inst_RX_CLK),

.hps_io_hps_io_emac1_inst_RXD1     (hps_io_hps_io_emac1_inst_RXD1),
.hps_io_hps_io_emac1_inst_RXD2     (hps_io_hps_io_emac1_inst_RXD2),
.hps_io_hps_io_emac1_inst_RXD3     (hps_io_hps_io_emac1_inst_RXD3),
.hps_io_hps_io_qspi_inst_IO0       (hps_io_hps_io_qspi_inst_IO0),
.hps_io_hps_io_qspi_inst_IO1       (hps_io_hps_io_qspi_inst_IO1),
.hps_io_hps_io_qspi_inst_IO2       (hps_io_hps_io_qspi_inst_IO2),
.hps_io_hps_io_qspi_inst_IO3       (hps_io_hps_io_qspi_inst_IO3),
.hps_io_hps_io_qspi_inst_SS0       (hps_io_hps_io_qspi_inst_SS0),
.hps_io_hps_io_qspi_inst_CLK       (hps_io_hps_io_qspi_inst_CLK),
.hps_io_hps_io_sdio_inst_CMD       (hps_io_hps_io_sdio_inst_CMD),
.hps_io_hps_io_sdio_inst_D0           (hps_io_hps_io_sdio_inst_D0),
.hps_io_hps_io_sdio_inst_D1        (hps_io_hps_io_sdio_inst_D1),
.hps_io_hps_io_sdio_inst_CLK          (hps_io_hps_io_sdio_inst_CLK),
.hps_io_hps_io_sdio_inst_D2         (hps_io_hps_io_sdio_inst_D2),
.hps_io_hps_io_sdio_inst_D3        (hps_io_hps_io_sdio_inst_D3),
.hps_io_hps_io_usb1_inst_D0        (hps_io_hps_io_usb1_inst_D0),
.hps_io_hps_io_usb1_inst_D1        (hps_io_hps_io_usb1_inst_D1),
.hps_io_hps_io_usb1_inst_D2           (hps_io_hps_io_usb1_inst_D2),
.hps_io_hps_io_usb1_inst_D3           (hps_io_hps_io_usb1_inst_D3),
.hps_io_hps_io_usb1_inst_D4           (hps_io_hps_io_usb1_inst_D4),
.hps_io_hps_io_usb1_inst_D5        (hps_io_hps_io_usb1_inst_D5),
.hps_io_hps_io_usb1_inst_D6           (hps_io_hps_io_usb1_inst_D6),
.hps_io_hps_io_usb1_inst_D7           (hps_io_hps_io_usb1_inst_D7),
.hps_io_hps_io_usb1_inst_CLK          (hps_io_hps_io_usb1_inst_CLK),
.hps_io_hps_io_usb1_inst_STP          (hps_io_hps_io_usb1_inst_STP),
.hps_io_hps_io_usb1_inst_DIR          (hps_io_hps_io_usb1_inst_DIR),
.hps_io_hps_io_usb1_inst_NXT          (hps_io_hps_io_usb1_inst_NXT),
.hps_io_hps_io_spim0_inst_CLK         (hps_io_hps_io_spim0_inst_CLK),
.hps_io_hps_io_spim0_inst_MOSI        (hps_io_hps_io_spim0_inst_MOSI),
.hps_io_hps_io_spim0_inst_MISO        (hps_io_hps_io_spim0_inst_MISO),
.hps_io_hps_io_spim0_inst_SS0         (hps_io_hps_io_spim0_inst_SS0),
.hps_io_hps_io_spim1_inst_CLK         (hps_io_hps_io_spim1_inst_CLK),
.hps_io_hps_io_spim1_inst_MOSI        (hps_io_hps_io_spim1_inst_MOSI),
.hps_io_hps_io_spim1_inst_MISO        (hps_io_hps_io_spim1_inst_MISO),
.hps_io_hps_io_spim1_inst_SS0        (hps_io_hps_io_spim1_inst_SS0),
.hps_io_hps_io_uart0_inst_RX       (hps_io_hps_io_uart0_inst_RX),
.hps_io_hps_io_uart0_inst_TX       (hps_io_hps_io_uart0_inst_TX),
.hps_io_hps_io_i2c1_inst_SDA       (hps_io_hps_io_i2c1_inst_SDA),
.hps_io_hps_io_i2c1_inst_SCL       (hps_io_hps_io_i2c1_inst_SCL),
.xaui_clk_clk                             (link_clk),
.xaui_reset_reset_n                       (RESET_N)
);
endmodule
```

txbuff_rx_loopback.sv

```
 module txbuff_rx_loopback(
    input   logic       clk,
    input   logic [3:0] rx_mac_out,
    output  logic [3:0] txbuff_in);

    always @(posedge clk) begin
        txbuff_in <= rx_mac_out;
    end
endmodule
```

txbuff_tx_loopback.sv

```
 module txbuff_tx_loopback(
    input   logic       clk,
    input   logic [3:0] txbuff_out,
    output  logic [3:0] rx_dc_in);

    always @(posedge clk) begin
        rx_dc_in <= txbuff_out;
    end
endmodule
```

txbuffer.sv

```
module txbuffer(
 input logic clk,
 input logic reset,
 input  logic  [63:0] stdata,
 input  logic valid,
 input  logic sopin,
 input  logic eopin,
 input  logic [2:0] emptyin,
 output  logic  [63:0] databuff,
 output  logic validout,
 output  logic readybuff,
 output  logic sopout,
 output  logic eopout,
 output  logic [2:0]emptybuff,
 output  logic readytx
);

 logic [7:0] cnt;
 logic [7:0] cnt2;
 logic [0:190][63:0] fifo;
 logic flag;
 logic sendflag;
 logic validbuff;
 logic sopbuff;
 logic eopbuff;
 assign readytx  = ! sendflag;

 always @(posedge clk)
 begin
   validout  <=validbuff;
   sopout    <=sopbuff;
   eopout    <=eopbuff;
 end
 initial
 begin
   databuff <= 64'b0;
   cnt <= 8'b0;
   cnt2<=8'b0;
   sendflag <= 0;
 end

 always @(posedge clk)
 begin
   if (reset == 1)
     begin
       databuff <= 64'b0;
       cnt <= 8'b0;
       cnt2<=8'b0;
       sendflag <= 0;
    end
   else
   begin
/////////////read from dc fifo/////////////
if (valid && readytx)
  begin
    fifo[cnt][63:0] <= stdata;
    cnt <= cnt +1;
    flag <=1;
  end
if (eopin && valid)
  begin
    endflag <= 1;
  end
/////////////////write to mac///////////////
else if (sendflag)
  begin
    if (flag)
      begin
        sopbuff <=1;
        flag <=0;
        databuff <= fifo[cnt2][63:0];
        cnt2 <= cnt2+1;
        cnt <= cnt -1;
        validbuff <=1;
      end

  if (cnt ==0)
    begin
      eopbuff  <=0;
      sendflag <=0;
      validbuff <=0;
      databuff <= fifo[cnt2][63:0];
    end
  else if (cnt ==8'b1)
    begin
      eopbuff<=1;
      databuff <= fifo[cnt2][63:0];
      cnt2 <= cnt2+1;
      cnt <= cnt -1;
    end
  else if (flag ==0)
    begin
    databuff <= fifo[cnt2][63:0];
    cnt2 <= cnt2+1;
    cnt <= cnt -1;
    sopbuff <=0;
  end
  else
    begin
      databuff <= 64'b0;
      cnt2 <= 8'b0;
    end
  end
end
```

```
end
endmodule
```

xaui_rx_loopback.sv

```
module xaui_rx_loopback(
    input  logic clk,
    input  logic [71:0] rx_out,
    output logic [71:0] tx_in);

    always @(posedge clk) begin
        tx_in <= rx_out;
    end
endmodule
```

xaui_tx_loopback.sv

```
module xaui_tx_loopback(
    input   logic       clk,
    input   logic [3:0] tx_out,
    output  logic [3:0] rx_in);

    always @(posedge clk) begin
        rx_in <= tx_out;
    end
endmodule
```

pll.v

```
// megafunction wizard: %Altera PLL v13.1%
// GENERATION: XML
// pll.v

// Generated using ACDS version 13.1 162 at 2015.04.28.12:21:37

‘timescale 1 ps / 1 ps
module pll (
input  wire  refclk,   //  refclk.clk
input  wire  rst,      //   reset.reset
output wire  outclk_0, // outclk0.clk
output wire  locked    //  locked.export
);

pll_0002 pll_inst (
.refclk   (refclk),   //  refclk.clk
.rst      (rst),      //   reset.reset
.outclk_0 (outclk_0), // outclk0.clk
.locked   (locked)    //  locked.export
);

endmodule
```

xaui.v

```
// megafunction wizard: %XAUI PHY v13.1%
// GENERATION: XML
// xaui.v

// Generated using ACDS version 13.1 162 at 2015.04.28.12:20:18

‘timescale 1 ps / 1 ps
module xaui (
input  wire         pll_ref_clk,          //         pll_ref_clk.clk
input  wire         xgmii_tx_clk,         //        xgmii_tx_clk.clk
output wire         xgmii_rx_clk,         //        xgmii_rx_clk.clk
output wire [71:0]  xgmii_rx_dc,          //         xgmii_rx_dc.data
input  wire [71:0]  xgmii_tx_dc,          //         xgmii_tx_dc.data
input  wire [3:0]   xaui_rx_serial_data,  // xaui_rx_serial_data.export
output wire [3:0]   xaui_tx_serial_data,  // xaui_tx_serial_data.export
output wire         rx_ready,             //            rx_ready.export
output wire         tx_ready,             //            tx_ready.export
input  wire         phy_mgmt_clk,         //        phy_mgmt_clk.clk
input  wire         phy_mgmt_clk_reset,   //  phy_mgmt_clk_reset.reset
input  wire [8:0]   phy_mgmt_address,     //            phy_mgmt.address
input  wire         phy_mgmt_read,        //                    .read
output wire [31:0]  phy_mgmt_readdata,    //                    .readdata
input  wire         phy_mgmt_write,       //                    .write
input  wire [31:0]  phy_mgmt_writedata,   //                    .writedata
output wire         phy_mgmt_waitrequest, //                    .waitrequest
output wire [229:0] reconfig_from_xcvr,   //  reconfig_from_xcvr.data
input  wire [349:0] reconfig_to_xcvr      //    reconfig_to_xcvr.data
);

altera_xcvr_xaui #(
.device_family                ("Cyclone V"),
.starting_channel_number      (0),
.interface_type               ("Soft XAUI"),
.data_rate                    ("3125 Mbps"),
.xaui_pll_type                ("CMU"),
.BASE_DATA_RATE               ("3125 Mbps"),
.en_synce_support             (0),
.use_control_and_status_ports (0),
.external_pma_ctrl_reconf     (0),
.recovered_clk_out            (0),
.number_of_interfaces         (1),
.reconfig_interfaces          (5),
.use_rx_rate_match            (0),
.tx_termination               ("OCT_100_OHMS"),
.tx_vod_selection             (4),
.tx_preemp_pretap             (0),
.tx_preemp_pretap_inv         ("false"),
.tx_preemp_tap_1              (0),
.tx_preemp_tap_2              (0),
.tx_preemp_tap_2_inv          ("false"),
.rx_common_mode               ("0.82v"),
.rx_termination               ("OCT_100_OHMS"),
.rx_eq_dc_gain                (0),
.rx_eq_ctrl                   (0),
.mgmt_clk_in_mhz              (150)
) xaui_inst (
.pll_ref_clk              (pll_ref_clk),
.xgmii_tx_clk             (xgmii_tx_clk),
.xgmii_rx_clk             (xgmii_rx_clk),
.xgmii_rx_dc              (xgmii_rx_dc),
.xgmii_tx_dc              (xgmii_tx_dc),
.xaui_rx_serial_data      (xaui_rx_serial_data),
.xaui_tx_serial_data      (xaui_tx_serial_data),
.rx_ready                 (rx_ready),
.tx_ready                 (tx_ready),
.phy_mgmt_clk             (phy_mgmt_clk),
.phy_mgmt_clk_reset       (phy_mgmt_clk_reset),
.phy_mgmt_address         (phy_mgmt_address),
.phy_mgmt_read            (phy_mgmt_read),
.phy_mgmt_readdata        (phy_mgmt_readdata),
.phy_mgmt_write           (phy_mgmt_write),
.phy_mgmt_writedata       (phy_mgmt_writedata),
.phy_mgmt_waitrequest     (phy_mgmt_waitrequest),
.reconfig_from_xcvr       (reconfig_from_xcvr),
.reconfig_to_xcvr         (reconfig_to_xcvr),
.rx_recovered_clk         (),
.tx_clk312_5              (),
.rx_digitalreset          (1’b0),
.tx_digitalreset          (1’b0),
.rx_channelaligned        (),
.rx_syncstatus            (),
.rx_disperr               (),
.rx_errdetect             (),
.rx_analogreset           (1’b0),
.rx_invpolarity           (4’b0000),
.rx_set_locktodata        (4’b0000),
.rx_set_locktoref         (4’b0000),
.rx_seriallpbken          (4’b0000),
.tx_invpolarity           (4’b0000),
.rx_is_lockedtodata       (),
.rx_phase_comp_fifo_error (),
.rx_is_lockedtoref        (),
.rx_rlv                   (),
.rx_rmfifoempty           (),
.rx_rmfifofull            (),
.tx_phase_comp_fifo_error (),
.rx_patterndetect         (),
.rx_rmfifodatadeleted     (),
.rx_rmfifodatainserted    (),
.rx_runningdisp           (),
.cal_blk_powerdown        (1’b0),
.pll_powerdown            (1’b0),
.gxb_powerdown            (1’b0),
.pll_locked               (),
.cdr_ref_clk              (1’b0)
);
endmodule
```

blazepps.qsys

```
<?xml version="1.0" encoding="UTF-8"?>
<system name="$${FILENAME}">
 <component
   name="$${FILENAME}"
   displayName="$${FILENAME}"
   version="1.0"
   description=""
   tags=""
   categories="System" />
 <parameter name="bonusData"><![CDATA[bonusData
{
   element $${FILENAME}
   {
   }
   element clk_0
   {
      datum _sortIndex
      {
         value = "6";
         type = "int";
      }
   }
   element eth_10g_mac
   {
      datum _sortIndex
      {
         value = "5";
         type = "int";
      }
   }
   element hps_0.f2h_axi_slave
   {
      datum baseAddress
      {
         value = "4294967296";
         type = "String";
      }
   }
   element hps_0
   {
      datum _sortIndex
      {
         value = "7";
         type = "int";
      }
   }
   element master_0
   {
      datum _sortIndex
      {
         value = "8";
         type = "int";
      }
   }
   element rx_dc_fifo
   {
      datum _sortIndex
      {
         value = "4";
         type = "int";
      }
   }
   element rx_dc_fifo_loopback
   {
      datum _sortIndex
      {
         value = "11";
         type = "int";
      }
   }
   element rx_oc_fifo
   {
      datum _sortIndex
      {
         value = "2";
```

```
         type = "int";
      }
   }
   element tx_dc_fifo
   {
      datum _sortIndex
      {
         value = "3";
         type = "int";
      }
   }
   element tx_dc_fifo_loopback
   {
      datum _sortIndex
      {
         value = "10";
         type = "int";
      }
   }
   element tx_oc_fifo
   {
      datum _sortIndex
      {
         value = "1";
         type = "int";
      }
   }
   element tx_oc_fifo_loopback
   {
      datum _sortIndex
      {
         value = "9";
         type = "int";
      }
   }
   element xaui_clk
   {
      datum _sortIndex
      {
         value = "0";
         type = "int";
      }
   }
}
]]></parameter>
 <parameter name="clockCrossingAdapter" value="HANDSHAKE" />
 <parameter name="device" value="5CSXFC6D6F31C8ES" />
 <parameter name="deviceFamily" value="Cyclone V" />
```

```
<parameter name="deviceSpeedGrade" value="8_H6" />
<parameter name="fabricMode" value="QSYS" />
<parameter name="generateLegacySim" value="false" />
<parameter name="generationId" value="0" />
<parameter name="globalResetBus" value="false" />
<parameter name="hdlLanguage" value="VERILOG" />
<parameter name="maxAdditionalLatency" value="1" />
<parameter name="projectName" value="blazepps.qpf" />
<parameter name="sopcBorderPoints" value="false" />
<parameter name="systemHash" value="0" />
<parameter name="timeStamp" value="0" />
<parameter name="useTestBenchNamingPattern" value="false" />
<instanceScript></instanceScript>
<interface name="clk" internal="clk_0.clk_in" type="clock" dir="end" />
<interface name="reset" internal="clk_0.clk_in_reset" type="reset" dir="end" />
<interface name="memory" internal="hps_0.memory" type="conduit" dir="end" />
<interface name="hps_io" internal="hps_0.hps_io" type="conduit" dir="end" />
<interface name="xaui_clk" internal="xaui_clk.clk_in" type="clock" dir="end" />
<interface
  name="xaui_reset"
  internal="xaui_clk.clk_in_reset"
  type="reset"
  dir="end" />
<interface name="eth_10g_mac_xgmii_tx" internal="eth_10g_mac.xgmii_tx" />
<interface name="eth_10g_mac_xgmii_rx" internal="eth_10g_mac.xgmii_rx" />
<interface name="eth_10g_mac_avalon_st_tx" internal="eth_10g_mac.avalon_st_tx" />
<interface
  name="eth_10g_mac_avalon_st_rxstatus"
  internal="eth_10g_mac.avalon_st_rxstatus" />
<interface
  name="eth_10g_mac_link_fault_status_xgmii_rx"
  internal="eth_10g_mac.link_fault_status_xgmii_rx" />
<interface name="stream_src" internal="tx_dc_fifo.out" />
<interface name="stream_sink" internal="rx_dc_fifo.in" />
<module kind="clock_source" version="13.1" enabled="1" name="clk_0">
 <parameter name="clockFrequency" value="50000000" />
 <parameter name="clockFrequencyKnown" value="true" />
 <parameter name="inputClockFrequency" value="0" />
 <parameter name="resetSynchronousEdges" value="NONE" />
</module>
<module kind="altera_hps" version="13.1" enabled="1" name="hps_0">
 <parameter name="MEM_VENDOR" value="Micron" />
 <parameter name="MEM_FORMAT" value="DISCRETE" />
 <parameter name="RDIMM_CONFIG" value="0000000000000000" />
 <parameter name="LRDIMM_EXTENDED_CONFIG">0x000000000000000000</parameter>
 <parameter name="DISCRETE_FLY_BY" value="true" />
 <parameter name="DEVICE_DEPTH" value="1" />
 <parameter name="MEM_MIRROR_ADDRESSING" value="0" />
```

```
<parameter name="MEM_CLK_FREQ_MAX" value="800.0" />
<parameter name="MEM_ROW_ADDR_WIDTH" value="15" />
<parameter name="MEM_COL_ADDR_WIDTH" value="10" />
<parameter name="MEM_DQ_WIDTH" value="32" />
<parameter name="MEM_DQ_PER_DQS" value="8" />
<parameter name="MEM_BANKADDR_WIDTH" value="3" />
<parameter name="MEM_IF_DM_PINS_EN" value="true" />
<parameter name="MEM_IF_DQSN_EN" value="true" />
<parameter name="MEM_NUMBER_OF_DIMMS" value="1" />
<parameter name="MEM_NUMBER_OF_RANKS_PER_DIMM" value="1" />
<parameter name="MEM_NUMBER_OF_RANKS_PER_DEVICE" value="1" />
<parameter name="MEM_RANK_MULTIPLICATION_FACTOR" value="1" />
<parameter name="MEM_CK_WIDTH" value="1" />
<parameter name="MEM_CS_WIDTH" value="1" />
<parameter name="MEM_CLK_EN_WIDTH" value="1" />
<parameter name="ALTMEMPHY_COMPATIBLE_MODE" value="false" />
<parameter name="NEXTGEN" value="true" />
<parameter name="MEM_IF_BOARD_BASE_DELAY" value="10" />
<parameter name="MEM_IF_SIM_VALID_WINDOW" value="0" />
<parameter name="MEM_GUARANTEED_WRITE_INIT" value="false" />
<parameter name="MEM_VERBOSE" value="true" />
<parameter name="PINGPONGPHY_EN" value="false" />
<parameter name="REFRESH_BURST_VALIDATION" value="false" />
<parameter name="MEM_BL" value="OTF" />
<parameter name="MEM_BT" value="Sequential" />
<parameter name="MEM_ASR" value="Manual" />
<parameter name="MEM_SRT" value="Normal" />
<parameter name="MEM_PD" value="DLL off" />
<parameter name="MEM_DRV_STR" value="RZQ/7" />
<parameter name="MEM_DLL_EN" value="true" />
<parameter name="MEM_RTT_NOM" value="RZQ/4" />
<parameter name="MEM_RTT_WR" value="RZQ/4" />
<parameter name="MEM_WTCL" value="8" />
<parameter name="MEM_ATCL" value="Disabled" />
<parameter name="MEM_TCL" value="11" />
<parameter name="MEM_AUTO_LEVELING_MODE" value="true" />
<parameter name="MEM_USER_LEVELING_MODE" value="Leveling" />
<parameter name="MEM_INIT_EN" value="false" />
<parameter name="MEM_INIT_FILE" value="" />
<parameter name="DAT_DATA_WIDTH" value="32" />
<parameter name="TIMING_TIS" value="180" />
<parameter name="TIMING_TIH" value="140" />
<parameter name="TIMING_TDS" value="30" />
<parameter name="TIMING_TDH" value="65" />
<parameter name="TIMING_TDQSQ" value="125" />
<parameter name="TIMING_TQHS" value="300" />
<parameter name="TIMING_TQH" value="0.38" />
<parameter name="TIMING_TDQSCK" value="255" />
```

```
<parameter name="TIMING_TDQSCKDS" value="450" />
<parameter name="TIMING_TDQSCKDM" value="900" />
<parameter name="TIMING_TDQSCKDL" value="1200" />
<parameter name="TIMING_TDQSS" value="0.25" />
<parameter name="TIMING_TDQSH" value="0.35" />
<parameter name="TIMING_TQSH" value="0.4" />
<parameter name="TIMING_TDSH" value="0.2" />
<parameter name="TIMING_TDSS" value="0.2" />
<parameter name="MEM_TINIT_US" value="500" />
<parameter name="MEM_TMRD_CK" value="4" />
<parameter name="MEM_TRAS_NS" value="35.0" />
<parameter name="MEM_TRCD_NS" value="13.75" />
<parameter name="MEM_TRP_NS" value="13.75" />
<parameter name="MEM_TREFI_US" value="7.8" />
<parameter name="MEM_TRFC_NS" value="260.0" />
<parameter name="CFG_TCCD_NS" value="2.5" />
<parameter name="MEM_TWR_NS" value="15.0" />
<parameter name="MEM_TWTR" value="4" />
<parameter name="MEM_TFAW_NS" value="30.0" />
<parameter name="MEM_TRRD_NS" value="7.5" />
<parameter name="MEM_TRTP_NS" value="7.5" />
<parameter name="POWER_OF_TWO_BUS" value="false" />
<parameter name="SOPC_COMPAT_RESET" value="false" />
<parameter name="AVL_MAX_SIZE" value="4" />
<parameter name="BYTE_ENABLE" value="true" />
<parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
<parameter name="CTL_DEEP_POWERDN_EN" value="false" />
<parameter name="CTL_SELF_REFRESH_EN" value="false" />
<parameter name="AUTO_POWERDN_EN" value="false" />
<parameter name="AUTO_PD_CYCLES" value="0" />
<parameter name="CTL_USR_REFRESH_EN" value="false" />
<parameter name="CTL_AUTOPCH_EN" value="false" />
<parameter name="CTL_ZQCAL_EN" value="false" />
<parameter name="ADDR_ORDER" value="0" />
<parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
<parameter name="CONTROLLER_LATENCY" value="5" />
<parameter name="CFG_REORDER_DATA" value="true" />
<parameter name="STARVE_LIMIT" value="10" />
<parameter name="CTL_CSR_ENABLED" value="false" />
<parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="CTL_ECC_ENABLED" value="false" />
<parameter name="CTL_HRB_ENABLED" value="false" />
<parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
<parameter name="MULTICAST_EN" value="false" />
<parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
<parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
<parameter name="DEBUG_MODE" value="false" />
<parameter name="ENABLE_BURST_MERGE" value="false" />
```

```
<parameter name="CTL_ENABLE_BURST_INTERRUPT" value="false" />
<parameter name="CTL_ENABLE_BURST_TERMINATE" value="false" />
<parameter name="LOCAL_ID_WIDTH" value="8" />
<parameter name="WRBUFFER_ADDR_WIDTH" value="6" />
<parameter name="MAX_PENDING_WR_CMD" value="16" />
<parameter name="MAX_PENDING_RD_CMD" value="32" />
<parameter name="USE_MM_ADAPTOR" value="true" />
<parameter name="USE_AXI_ADAPTOR" value="false" />
<parameter name="HCX_COMPAT_MODE" value="false" />
<parameter name="CTL_CMD_QUEUE_DEPTH" value="8" />
<parameter name="CTL_CSR_READ_ONLY" value="1" />
<parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
<parameter name="NUM_OF_PORTS" value="1" />
<parameter name="ENABLE_BONDING" value="false" />
<parameter name="ENABLE_USER_ECC" value="false" />
<parameter name="AVL_DATA_WIDTH_PORT" value="32,32,32,32,32,32" />
<parameter name="PRIORITY_PORT" value="1,1,1,1,1,1" />
<parameter name="WEIGHT_PORT" value="0,0,0,0,0,0" />
<parameter name="CPORT_TYPE_PORT">Bidirectional,Bidirectional,Bidirectional,
Bidirectional,Bidirectional,Bidirectional</parameter>
<parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
<parameter name="FORCE_SEQUENCER_TCL_DEBUG_MODE" value="false" />
<parameter name="ENABLE_SEQUENCER_MARGINING_ON_BY_DEFAULT" value="false" />
<parameter name="REF_CLK_FREQ" value="25.0" />
<parameter name="REF_CLK_FREQ_PARAM_VALID" value="false" />
<parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
<parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_DR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_DR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0" />
```

```
<parameter name="PLL_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_NIOS_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_HR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_HR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_HR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_HR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_HR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_HR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0" />
```

```
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CLK_PARAM_VALID" value="false" />
<parameter name="ENABLE_EXTRA_REPORTING" value="false" />
<parameter name="NUM_EXTRA_REPORT_PATH" value="10" />
<parameter name="ENABLE_ISS_PROBES" value="false" />
<parameter name="CALIB_REG_WIDTH" value="8" />
<parameter name="USE_SEQUENCER_BFM" value="false" />
<parameter name="DEFAULT_FAST_SIM_MODEL" value="true" />
<parameter name="PLL_SHARING_MODE" value="None" />
<parameter name="NUM_PLL_SHARING_INTERFACES" value="1" />
<parameter name="EXPORT_AFI_HALF_CLK" value="false" />
<parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false" />
<parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false" />
<parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false" />
<parameter name="USE_FAKE_PHY" value="false" />
<parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0" />
<parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false" />
<parameter name="TRACKING_ERROR_TEST" value="false" />
<parameter name="TRACKING_WATCH_TEST" value="false" />
<parameter name="MARGIN_VARIATION_TEST" value="false" />
<parameter name="EXTRA_SETTINGS" value="" />
<parameter name="MEM_DEVICE" value="MISSING_MODEL" />
<parameter name="FORCE_SYNTHESIS_LANGUAGE" value="" />
<parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0" />
<parameter name="SEQUENCER_TYPE" value="NIOS" />
<parameter name="ADVERTIZE_SEQUENCER_SW_BUILD_FILES" value="false" />
<parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CK_INVERT" value="false" />
<parameter name="PHY_ONLY" value="false" />
<parameter name="SEQ_MODE" value="0" />
<parameter name="ADVANCED_CK_PHASES" value="false" />
<parameter name="COMMAND_PHASE" value="0.0" />
<parameter name="MEM_CK_PHASE" value="0.0" />
<parameter name="P2C_READ_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="C2P_WRITE_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="ACV_PHY_CLK_ADD_FR_PHASE" value="0.0" />
<parameter name="MEM_VOLTAGE" value="1.5V DDR3" />
<parameter name="PLL_LOCATION" value="Top_Bottom" />
<parameter name="SKIP_MEM_INIT" value="true" />
<parameter name="READ_DQ_DQS_CLOCK_SOURCE" value="INVERTED_DQS_BUS" />
<parameter name="DQ_INPUT_REG_USE_CLKN" value="false" />
<parameter name="DQS_DQSN_MODE" value="DIFFERENTIAL" />
<parameter name="AFI_DEBUG_INFO_WIDTH" value="32" />
<parameter name="CALIBRATION_MODE" value="Skip" />
<parameter name="NIOS_ROM_DATA_WIDTH" value="32" />
<parameter name="READ_FIFO_SIZE" value="8" />
<parameter name="PHY_CSR_ENABLED" value="false" />
```

```
<parameter name="PHY_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="USER_DEBUG_LEVEL" value="1" />
<parameter name="TIMING_BOARD_DERATE_METHOD" value="AUTO" />
<parameter name="TIMING_BOARD_CK_CKN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_AC_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_DQS_DQSN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_DQ_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_TIS" value="0.0" />
<parameter name="TIMING_BOARD_TIH" value="0.0" />
<parameter name="TIMING_BOARD_TDS" value="0.0" />
<parameter name="TIMING_BOARD_TDH" value="0.0" />
<parameter name="TIMING_BOARD_ISI_METHOD" value="AUTO" />
<parameter name="TIMING_BOARD_AC_EYE_REDUCTION_SU" value="0.0" />
<parameter name="TIMING_BOARD_AC_EYE_REDUCTION_H" value="0.0" />
<parameter name="TIMING_BOARD_DQ_EYE_REDUCTION" value="0.0" />
<parameter name="TIMING_BOARD_DELTA_DQS_ARRIVAL_TIME" value="0.0" />
<parameter name="TIMING_BOARD_READ_DQ_EYE_REDUCTION" value="0.0" />
<parameter name="TIMING_BOARD_DELTA_READ_DQS_ARRIVAL_TIME" value="0.0" />
<parameter name="PACKAGE_DESKEW" value="false" />
<parameter name="AC_PACKAGE_DESKEW" value="false" />
<parameter name="TIMING_BOARD_MAX_CK_DELAY" value="0.03" />
<parameter name="TIMING_BOARD_MAX_DQS_DELAY" value="0.02" />
<parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MIN" value="0.09" />
<parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MAX" value="0.16" />
<parameter name="TIMING_BOARD_SKEW_BETWEEN_DIMMS" value="0.05" />
<parameter name="TIMING_BOARD_SKEW_WITHIN_DQS" value="0.01" />
<parameter name="TIMING_BOARD_SKEW_BETWEEN_DQS" value="0.08" />
<parameter name="TIMING_BOARD_DQ_TO_DQS_SKEW" value="0.0" />
<parameter name="TIMING_BOARD_AC_SKEW" value="0.03" />
<parameter name="TIMING_BOARD_AC_TO_CK_SKEW" value="0.0" />
<parameter name="RATE" value="Full" />
<parameter name="MEM_CLK_FREQ" value="400.0" />
<parameter name="USE_MEM_CLK_FREQ" value="false" />
<parameter name="FORCE_DQS_TRACKING" value="AUTO" />
<parameter name="FORCE_SHADOW_REGS" value="AUTO" />
<parameter name="MRS_MIRROR_PING_PONG_ATSO" value="false" />
<parameter name="SYS_INFO_DEVICE_FAMILY" value="Cyclone V" />
<parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM_VALID" value="false" />
<parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM" value="" />
<parameter name="DEVICE_FAMILY_PARAM" value="" />
<parameter name="SPEED_GRADE" value="7" />
<parameter name="IS_ES_DEVICE" value="false" />
<parameter name="DISABLE_CHILD_MESSAGING" value="false" />
<parameter name="HARD_EMIF" value="true" />
<parameter name="HHP_HPS" value="true" />
<parameter name="HHP_HPS_VERIFICATION" value="false" />
<parameter name="HHP_HPS_SIMULATION" value="false" />
<parameter name="HPS_PROTOCOL" value="DDR3" />
```

```
<parameter name="CUT_NEW_FAMILY_TIMING" value="true" />
<parameter name="ENABLE_EXPORT_SEQ_DEBUG_BRIDGE" value="false" />
<parameter name="CORE_DEBUG_CONNECTION" value="EXPORT" />
<parameter name="ADD_EXTERNAL_SEQ_DEBUG_NIOS" value="false" />
<parameter name="ED_EXPORT_SEQ_DEBUG" value="false" />
<parameter name="ADD_EFFICIENCY_MONITOR" value="false" />
<parameter name="ENABLE_ABS_RAM_MEM_INIT" value="false" />
<parameter name="ABS_RAM_MEM_INIT_FILENAME" value="meminit" />
<parameter name="DLL_SHARING_MODE" value="None" />
<parameter name="NUM_DLL_SHARING_INTERFACES" value="1" />
<parameter name="OCT_SHARING_MODE" value="None" />
<parameter name="NUM_OCT_SHARING_INTERFACES" value="1" />
<parameter name="MPU_EVENTS_Enable" value="false" />
<parameter name="GP_Enable" value="false" />
<parameter name="DEBUGAPB_Enable" value="false" />
<parameter name="STM_Enable" value="false" />
<parameter name="CTI_Enable" value="false" />
<parameter name="TPIUFPGA_Enable" value="false" />
<parameter name="BOOTFROMFPGA_Enable" value="false" />
<parameter name="TEST_Enable" value="false" />
<parameter name="HLGPI_Enable" value="false" />
<parameter name="BSEL_EN" value="false" />
<parameter name="BSEL" value="1" />
<parameter name="CSEL_EN" value="false" />
<parameter name="CSEL" value="0" />
<parameter name="F2S_Width" value="2" />
<parameter name="S2F_Width" value="2" />
<parameter name="LWH2F_Enable" value="true" />
<parameter name="F2SDRAM_Type" value="" />
<parameter name="F2SDRAM_Width" value="" />
<parameter name="BONDING_OUT_ENABLED" value="false" />
<parameter name="S2FCLK_COLDRST_Enable" value="false" />
<parameter name="S2FCLK_PENDINGRST_Enable" value="false" />
<parameter name="F2SCLK_DBGRST_Enable" value="false" />
<parameter name="F2SCLK_WARMRST_Enable" value="false" />
<parameter name="F2SCLK_COLDRST_Enable" value="false" />
<parameter name="DMA_Enable">No,No,No,No,No,No,No,No</parameter>
<parameter name="F2SINTERRUPT_Enable" value="true" />
<parameter name="S2FINTERRUPT_CAN_Enable" value="false" />
<parameter name="S2FINTERRUPT_CLOCKPERIPHERAL_Enable" value="false" />
<parameter name="S2FINTERRUPT_CTI_Enable" value="false" />
<parameter name="S2FINTERRUPT_DMA_Enable" value="false" />
<parameter name="S2FINTERRUPT_EMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_FPGAMANAGER_Enable" value="false" />
<parameter name="S2FINTERRUPT_GPIO_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CEMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CPERIPHERAL_Enable" value="false" />
<parameter name="S2FINTERRUPT_L4TIMER_Enable" value="false" />
```

```
<parameter name="S2FINTERRUPT_NAND_Enable" value="false" />
<parameter name="S2FINTERRUPT_OSCTIMER_Enable" value="false" />
<parameter name="S2FINTERRUPT_QSPI_Enable" value="false" />
<parameter name="S2FINTERRUPT_SDMMC_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPIMASTER_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPISLAVE_Enable" value="false" />
<parameter name="S2FINTERRUPT_UART_Enable" value="false" />
<parameter name="S2FINTERRUPT_USB_Enable" value="false" />
<parameter name="S2FINTERRUPT_WATCHDOG_Enable" value="false" />
<parameter name="EMAC0_PinMuxing" value="Unused" />
<parameter name="EMAC0_Mode" value="N/A" />
<parameter name="EMAC1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="EMAC1_Mode" value="RGMII" />
<parameter name="NAND_PinMuxing" value="Unused" />
<parameter name="NAND_Mode" value="N/A" />
<parameter name="QSPI_PinMuxing" value="HPS I/O Set 0" />
<parameter name="QSPI_Mode" value="1 SS" />
<parameter name="SDIO_PinMuxing" value="HPS I/O Set 0" />
<parameter name="SDIO_Mode" value="4-bit Data" />
<parameter name="USB0_PinMuxing" value="Unused" />
<parameter name="USB0_Mode" value="N/A" />
<parameter name="USB1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="USB1_Mode" value="SDR" />
<parameter name="SPIM0_PinMuxing" value="HPS I/O Set 0" />
<parameter name="SPIM0_Mode" value="Single Slave Select" />
<parameter name="SPIM1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="SPIM1_Mode" value="Single Slave Select" />
<parameter name="SPIS0_PinMuxing" value="Unused" />
<parameter name="SPIS0_Mode" value="N/A" />
<parameter name="SPIS1_PinMuxing" value="Unused" />
<parameter name="SPIS1_Mode" value="N/A" />
<parameter name="UART0_PinMuxing" value="HPS I/O Set 0" />
<parameter name="UART0_Mode" value="No Flow Control" />
<parameter name="UART1_PinMuxing" value="Unused" />
<parameter name="UART1_Mode" value="N/A" />
<parameter name="I2C0_PinMuxing" value="Unused" />
<parameter name="I2C0_Mode" value="N/A" />
<parameter name="I2C1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="I2C1_Mode" value="I2C" />
<parameter name="I2C2_PinMuxing" value="Unused" />
<parameter name="I2C2_Mode" value="N/A" />
<parameter name="I2C3_PinMuxing" value="Unused" />
<parameter name="I2C3_Mode" value="N/A" />
<parameter name="CAN0_PinMuxing" value="Unused" />
<parameter name="CAN0_Mode" value="N/A" />
<parameter name="CAN1_PinMuxing" value="Unused" />
<parameter name="CAN1_Mode" value="N/A" />
<parameter name="TRACE_PinMuxing" value="Unused" />
```

```
<parameter name="TRACE_Mode" value="N/A" />
<parameter name="GPIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No
,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No</parameter>
<parameter name="LOANIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No</parameter>
<parameter name="S2FCLK_USER0CLK_Enable" value="false" />
<parameter name="S2FCLK_USER0CLK_FREQ" value="100" />
<parameter name="S2FCLK_USER1CLK_Enable" value="false" />
<parameter name="S2FCLK_USER1CLK_FREQ" value="100" />
<parameter name="S2FCLK_USER2CLK_Enable" value="false" />
<parameter name="S2FCLK_USER2CLK_FREQ" value="100" />
<parameter name="F2SCLK_PERIPHCLK_Enable" value="false" />
<parameter name="F2SCLK_PERIPHCLK_FREQ" value="100" />
<parameter name="F2SCLK_SDRAMCLK_Enable" value="false" />
<parameter name="F2SCLK_SDRAMCLK_FREQ" value="100" />
<parameter name="F2H_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="H2F_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="H2F_LW_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="F2H_SDRAM0_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM1_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM2_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM3_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM4_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM5_CLOCK_FREQ" value="100" />
<parameter name="H2F_CTI_CLOCK_FREQ" value="100" />
<parameter name="H2F_TPIU_CLOCK_IN_FREQ" value="100" />
<parameter name="H2F_DEBUG_APB_CLOCK_FREQ" value="100" />
<parameter
   name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC_PTP_REF_CLOCK"
   value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_RX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_TX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_MD_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_RX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_TX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_MD_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_QSPI_SCLK_OUT" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SDIO_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SDIO_CCLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB0_CLK_IN" value="100" />
```

```
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB1_CLK_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM0_SCLK_OUT" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM1_SCLK_OUT" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIS0_SCLK_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIS1_SCLK_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C0_SCL_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C0_CLK" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C1_SCL_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C1_CLK" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C2_SCL_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C2_CLK" value="100" />
 <parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C3_SCL_IN" value="100" />
 <parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C3_CLK" value="100" />
 <parameter name="device_name" value="5CSXFC6D6F31C8ES" />
 <parameter
    name="quartus_ini_hps_ip_enable_all_peripheral_fpga_interfaces"
    value="false" />
 <parameter
    name="quartus_ini_hps_ip_enable_emac0_peripheral_fpga_interface"
    value="false" />
 <parameter name="quartus_ini_hps_ip_enable_test_interface" value="false" />
 <parameter name="quartus_ini_hps_ip_fast_f2sdram_sim_model" value="false" />
 <parameter name="quartus_ini_hps_ip_suppress_sdram_synth" value="false" />
 <parameter
    name="quartus_ini_hps_ip_enable_low_speed_serial_fpga_interfaces"
    value="false" />
 <parameter name="quartus_ini_hps_ip_enable_bsel_csel" value="false" />
 <parameter name="quartus_ini_hps_ip_f2sdram_bonding_out" value="false" />
</module>
<module
  kind="altera_jtag_avalon_master"
  version="13.1"
  enabled="1"
  name="master_0">
 <parameter name="USE_PLI" value="0" />
 <parameter name="PLI_PORT" value="50000" />
 <parameter name="COMPONENT_CLOCK" value="0" />
 <parameter name="FAST_VER" value="0" />
 <parameter name="FIFO_DEPTHS" value="2" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
</module>
<module kind="clock_source" version="13.1" enabled="1" name="xaui_clk">
 <parameter name="clockFrequency" value="15625000" />
 <parameter name="clockFrequencyKnown" value="true" />
 <parameter name="inputClockFrequency" value="0" />
 <parameter name="resetSynchronousEdges" value="NONE" />
</module>
```

```
<module
  kind="altera_avalon_fifo"
  version="13.1"
  enabled="1"
  name="tx_oc_fifo">
 <parameter name="avalonMMAvalonMMDataWidth" value="32" />
 <parameter name="avalonMMAvalonSTDataWidth" value="32" />
 <parameter name="bitsPerSymbol" value="8" />
 <parameter name="channelWidth" value="0" />
 <parameter name="errorWidth" value="0" />
 <parameter name="fifoDepth" value="512" />
 <parameter name="fifoInputInterfaceOptions" value="AVALONMM_WRITE" />
 <parameter name="fifoOutputInterfaceOptions" value="AVALONST_SOURCE" />
 <parameter name="showHiddenFeatures" value="false" />
 <parameter name="singleClockMode" value="true" />
 <parameter name="singleResetMode" value="true" />
 <parameter name="symbolsPerBeat" value="4" />
 <parameter name="useBackpressure" value="false" />
 <parameter name="useIRQ" value="true" />
 <parameter name="usePacket" value="true" />
 <parameter name="useReadControl" value="false" />
 <parameter name="useRegister" value="false" />
 <parameter name="useWriteControl" value="false" />
 <parameter name="deviceFamilyString" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_fifo"
  version="13.1"
  enabled="1"
  name="rx_oc_fifo">
 <parameter name="avalonMMAvalonMMDataWidth" value="32" />
 <parameter name="avalonMMAvalonSTDataWidth" value="32" />
 <parameter name="bitsPerSymbol" value="8" />
 <parameter name="channelWidth" value="0" />
 <parameter name="errorWidth" value="0" />
 <parameter name="fifoDepth" value="512" />
 <parameter name="fifoInputInterfaceOptions" value="AVALONST_SINK" />
 <parameter name="fifoOutputInterfaceOptions" value="AVALONMM_READ" />
 <parameter name="showHiddenFeatures" value="false" />
 <parameter name="singleClockMode" value="true" />
 <parameter name="singleResetMode" value="false" />
 <parameter name="symbolsPerBeat" value="4" />
 <parameter name="useBackpressure" value="false" />
 <parameter name="useIRQ" value="true" />
 <parameter name="usePacket" value="true" />
 <parameter name="useReadControl" value="false" />
 <parameter name="useRegister" value="false" />
 <parameter name="useWriteControl" value="false" />
 <parameter name="deviceFamilyString" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_dc_fifo"
  version="13.1"
  enabled="0"
  name="tx_dc_fifo">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_IN_FILL_LEVEL" value="0" />
 <parameter name="USE_OUT_FILL_LEVEL" value="0" />
 <parameter name="WR_SYNC_DEPTH" value="2" />
 <parameter name="RD_SYNC_DEPTH" value="2" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_dc_fifo"
  version="13.1"
  enabled="0"
  name="rx_dc_fifo">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_IN_FILL_LEVEL" value="0" />
 <parameter name="USE_OUT_FILL_LEVEL" value="0" />
 <parameter name="WR_SYNC_DEPTH" value="2" />
 <parameter name="RD_SYNC_DEPTH" value="2" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_eth_10g_mac"
  version="13.1"
  enabled="0"
  name="eth_10g_mac">
 <parameter name="ENABLE_TIMESTAMPING" value="0" />
 <parameter name="ENABLE_PTP_1STEP" value="0" />
 <parameter name="TSTAMP_FP_WIDTH" value="4" />
 <parameter name="PREAMBLE_PASSTHROUGH" value="1" />
 <parameter name="ENABLE_PFC" value="0" />
 <parameter name="PFC_PRIORITY_NUM" value="8" />
 <parameter name="DATAPATH_OPTION" value="3" />
 <parameter name="ENABLE_SUPP_ADDR" value="0" />
 <parameter name="INSTANTIATE_TX_CRC" value="0" />
 <parameter name="INSTANTIATE_STATISTICS" value="0" />
 <parameter name="REGISTER_BASED_STATISTICS" value="0" />
 <parameter name="ENABLE_1G10G_MAC" value="0" />
 <parameter name="DEVICE_FAMILY" value="Cyclone V" />
 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="1"
  name="tx_oc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="4" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="0"
  name="tx_dc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="0"
  name="rx_dc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.h2f_axi_clock" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.f2h_axi_clock" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.h2f_lw_axi_clock" />
<connection kind="clock" version="13.1" start="clk_0.clk"
 end="master_0.clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="master_0.clk_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="tx_dc_fifo.out_clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="tx_dc_fifo.in_clk_reset" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="tx_dc_fifo.out_clk_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="rx_dc_fifo.in_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="rx_dc_fifo.in_clk_reset" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="rx_dc_fifo.out_clk_reset" />
<connection
  kind="avalon"
  version="13.1"
  start="master_0.master"
  end="rx_oc_fifo.out">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0028" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="avalon"
  version="13.1"
  start="master_0.master"
  end="tx_oc_fifo.in">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0020" />
 <parameter name="defaultConnection" value="false" />
```

```
</connection>
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.csr_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.csr_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.tx_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.tx_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.rx_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.rx_reset" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="tx_oc_fifo.clk_in" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="tx_oc_fifo.reset_in" />
<connection
  kind="avalon"
  version="13.1"
  start="hps_0.h2f_lw_axi_master"
  end="tx_oc_fifo.in">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0020" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="rx_dc_fifo.out_clk_reset" />
<connection
  kind="avalon"
  version="13.1"
  start="hps_0.h2f_lw_axi_master"
  end="rx_oc_fifo.out">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0028" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="clock" version="13.1" start="clk_0.clk" end="rx_oc_fifo.clk_in" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="tx_dc_fifo.in_clk" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="rx_dc_fifo.out_clk" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="tx_dc_fifo_loopback.clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="tx_dc_fifo_loopback.clk_reset" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="rx_oc_fifo.reset_in" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="tx_oc_fifo_loopback.clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="tx_oc_fifo_loopback.clk_reset" />
<connection
  kind="avalon_streaming"
  version="13.1"
  start="tx_oc_fifo.out"
```

```
   end="tx_oc_fifo_loopback.in" />
 <connection
   kind="avalon_streaming"
   version="13.1"
   start="tx_oc_fifo_loopback.out"
   end="rx_oc_fifo.in" />
 <interconnectRequirement for="$system" name="qsys_mm.clockCrossingAdapter" value="HANDSHAKE" /
 <interconnectRequirement for="$system" name="qsys_mm.maxAdditionalLatency" value="1" />
 <interconnectRequirement for="$system" name="qsys_mm.insertDefaultSlave" value="false" />
</system>
```

loopback_test.tcl

```
variable s_path ""
set s_path [lindex [get_service_paths master] 0]
open_service master $s_path

set tx_oc_fifo 0x20
set tx_oc_fifo_ctrl 0x24
set rx_oc_fifo 0x28
set rx_oc_fifo_ctrl 0x2C
set num 20

set SOP   0x00000001
set EOP   0x00000006
set EMPTY 0x0000000C

set sop_expr   "*** SOP ***"
set eop_expr   "*** EOP ***"
set empty_expr "*** empty bits ***"

puts "*** Begin Write ***";
master_write_32 $s_path $tx_oc_fifo_ctrl $SOP
for {set i 0} {$i < [expr $num]} {incr i} {
    master_write_32 $s_path $tx_oc_fifo $i;
}
master_write_32 $s_path $tx_oc_fifo_ctrl $EOP
master_write_32 $s_path $tx_oc_fifo 0xA916C57B
puts "*** End Write ***";

puts "*** Begin Read ***";
for {set i 0} {$i < [expr $num + 1]} {incr i} {
    puts "#######i######";
    puts $i;

    puts "Data bits";
    set a [master_read_32 $s_path $rx_oc_fifo 1]
    puts $a;
    set b [master_read_32 $s_path $rx_oc_fifo_ctrl 1]
    puts "Status bits"
    puts $b;
    if { $b & $SOP } {
puts $sop_expr;
    }
    if { $b & $EOP } {
puts $eop_expr;
set c [expr ($b & $EMPTY) >> 2]
puts $empty_expr;
puts $c;
    }
}
puts "*** End Read ***";

puts "End"
```

tx_stream.tcl

```
variable infile 0
variable s_path ""
set s_path [lindex [get_service_paths master] 0]
open_service master $s_path

set src_base 0x20
set src_base_control 0x24

puts "sop"
master_write_32 $s_path $src_base_control 0x00000001
master_write_32 $s_path $src_base 0x01005e00
master_write_32 $s_path $src_base 0x00010017
master_write_32 $s_path $src_base 0xf293c2df
master_write_32 $s_path $src_base 0x08004500
master_write_32 $s_path $src_base 0x0046417c
master_write_32 $s_path $src_base 0x00000111
master_write_32 $s_path $src_base 0x032d803b
master_write_32 $s_path $src_base 0x14c2e000
master_write_32 $s_path $src_base 0x00010272
master_write_32 $s_path $src_base 0x02720032
master_write_32 $s_path $src_base 0x2fd6534e
master_write_32 $s_path $src_base 0x51554552
master_write_32 $s_path $src_base 0x593a6468
master_write_32 $s_path $src_base 0x63703437
master_write_32 $s_path $src_base 0x2e63732e
master_write_32 $s_path $src_base 0x636f6c75
master_write_32 $s_path $src_base 0x6d626961
master_write_32 $s_path $src_base 0x2e656475
```

```
master_write_32 $s_path $src_base 0x3a725a77
master_write_32 $s_path $src_base 0x6472533a
master_write_32 $s_path $src_base 0x78737672
master_write_32 $s_path $src_base_control 0x00000002
master_write_32 $s_path $src_base 0xA9ffffff
```

rx_stream.tcl

```
variable s_path ""
set s_path [lindex [get_service_paths master] 0]
open_service master $s_path

set rx_oc_fifo 0x28
set rx_oc_fifo_ctrl 0x2C
set num 20

set SOP   0x00000001
set EOP   0x00000006
set EMPTY 0x0000000C

set sop_expr   "*** SOP ***"
set eop_expr   "*** EOP ***"
set empty_expr "*** empty bits ***"

for {set i 0} {$i < [expr $num + 1]} {incr i} {
    puts "#######i######";
    puts $i;

    puts "Data bits";
    set a [master_read_32 $s_path $rx_oc_fifo 1]
    puts $a;

    set b [master_read_32 $s_path $rx_oc_fifo_ctrl 1]
    puts "Status bits"
    puts $b;
    if { $b & $SOP } {
        puts $sop_expr;
    }
    if { $b & $EOP } {
        puts $eop_expr;
        set c [expr ($b & $EMPTY) >> 2]
        puts $empty_expr;
        puts $c;
    }
}
```

5.2 BlazePPS v2 (Simple MAC from previous project

(only new files and files with deltas between BlazePPS_v1 are shown)
SoCKit_top.sv

```
// ============================================================================
// Copyright (c) 2013 by Terasic Technologies Inc.
// ============================================================================
//
// Permission:
//
//   Terasic grants permission to use and modify this code for use
//   in synthesis for all Terasic Development Boards and Altera Development
//   Kits made by Terasic.  Other use of this code, including the selling
//   ,duplication, or modification of any portion is strictly prohibited.
//
// Disclaimer:
//
//   This VHDL/Verilog or C/C++ source code is intended as a design reference
//   which illustrates how these types of functions can be implemented.
//   It is the user's responsibility to verify their design for
//   consistency and functionality through the use of formal
//   verification methods.  Terasic provides no warranty regarding the use
//   or functionality of this code.
//
// ============================================================================
//
//  Terasic Technologies Inc
//  9F., No.176, Sec.2, Gongdao 5th Rd, East Dist, Hsinchu City, 30070. Taiwan
//
//
//                     web: http://www.terasic.com/
//                     email: support@terasic.com
//
// ============================================================================
// ============================================================================
//
// Major Functions: SoCKit_Default
//
// ============================================================================
// Revision History :
// ============================================================================
//  Authors: Valeh Valiollahpour Amiri (vv2252@columbia.edu)
//           Christopher Campbell (cc3769@columbia.edu)
//           Sheng Qian (sq2168@columbia.edu)
//           Yuanpei Zhang (yz2727@columbia.edu)
//
//  Last Mod. Date: May 14th, 2015
// ============================================================================

//‘define ENABLE_HPS
‘define ENABLE_MDIO

module SoCKit_Top(

‘ifdef ENABLE_HPS
    /////////HPS/////////
    HPS_CLOCK_25,
    HPS_CLOCK_50,
    HPS_CONV_USB_n,
    HPS_DDR3_A,
    HPS_DDR3_BA,
    HPS_DDR3_CAS_n,
    HPS_DDR3_CKE,
    HPS_DDR3_CK_n,
    HPS_DDR3_CK_p,
    HPS_DDR3_CS_n,
    HPS_DDR3_DM,
    HPS_DDR3_DQ,
    HPS_DDR3_DQS_n,
    HPS_DDR3_DQS_p,
    HPS_DDR3_ODT,
    HPS_DDR3_RAS_n,
    HPS_DDR3_RESET_n,
    HPS_DDR3_RZQ,
    HPS_DDR3_WE_n,
    HPS_ENET_GTX_CLK,
    HPS_ENET_INT_n,
    HPS_ENET_MDC,
    HPS_ENET_MDIO,
    HPS_ENET_RESET_n,
    HPS_ENET_RX_CLK,
    HPS_ENET_RX_DATA,
    HPS_ENET_RX_DV,
    HPS_ENET_TX_DATA,
    HPS_ENET_TX_EN,
    HPS_FLASH_DATA,
    HPS_FLASH_DCLK,
    HPS_FLASH_NCSO,
    HPS_GSENSOR_INT,
    HPS_I2C_CLK,
    HPS_I2C_SDA,
    HPS_KEY,
    HPS_LCM_D_C,
    HPS_LCM_RST_N,
    HPS_LCM_SPIM_CLK,
    HPS_LCM_SPIM_MISO,
    HPS_LCM_SPIM_MOSI,
    HPS_LCM_SPIM_SS,
    HPS_LED,
    HPS_LTC_GPIO,
    HPS_RESET_n,
    HPS_SD_CLK,
    HPS_SD_CMD,
    HPS_SD_DATA,
    HPS_SPIM_CLK,
    HPS_SPIM_MISO,
    HPS_SPIM_MOSI,
    HPS_SPIM_SS,
    HPS_SW,
    HPS_UART_RX,
    HPS_UART_TX,
    HPS_USB_CLKOUT,
    HPS_USB_DATA,
    HPS_USB_DIR,
    HPS_USB_NXT,
    HPS_USB_RESET_PHY,
    HPS_USB_STP,
    HPS_WARM_RST_n,
‘endif /*ENABLE_HPS*/

    /////////KEY/////////
    KEY,

    /////////LED/////////
    LED,

    /////////OSC/////////
    OSC_50_B3B,
    OSC_50_B4A,
    OSC_50_B5B,
    OSC_50_B8A,

    /////////RESET/////////
    RESET_n,

    /////////SI5338/////////
    SI5338_SCL,
    SI5338_SDA,

    ///////// SW /////////
    SW,

    ///////////hps//////////
```

```
memory_mem_a,
memory_mem_ba,
memory_mem_ck,
memory_mem_ck_n,
memory_mem_cke,
memory_mem_cs_n,
memory_mem_ras_n,
memory_mem_cas_n,
memory_mem_we_n,
memory_mem_reset_n,
memory_mem_dq,
memory_mem_dqs,
memory_mem_dqs_n,
memory_mem_odt,
memory_mem_dm,
memory_oct_rzqin,
hps_io_hps_io_emac1_inst_TX_CLK,
hps_io_hps_io_emac1_inst_TXD0,
hps_io_hps_io_emac1_inst_TXD1,
hps_io_hps_io_emac1_inst_TXD2,
hps_io_hps_io_emac1_inst_TXD3,
hps_io_hps_io_emac1_inst_RXD0,
hps_io_hps_io_emac1_inst_MDIO,
hps_io_hps_io_emac1_inst_MDC,
hps_io_hps_io_emac1_inst_RX_CTL,
hps_io_hps_io_emac1_inst_TX_CTL,
hps_io_hps_io_emac1_inst_RX_CLK,
hps_io_hps_io_emac1_inst_RXD1,
hps_io_hps_io_emac1_inst_RXD2,
hps_io_hps_io_emac1_inst_RXD3,
hps_io_hps_io_qspi_inst_IO0,
hps_io_hps_io_qspi_inst_IO1,
hps_io_hps_io_qspi_inst_IO2,
hps_io_hps_io_qspi_inst_IO3,
hps_io_hps_io_qspi_inst_SS0,
hps_io_hps_io_qspi_inst_CLK,
hps_io_hps_io_sdio_inst_CMD,
hps_io_hps_io_sdio_inst_D0,
hps_io_hps_io_sdio_inst_D1,
hps_io_hps_io_sdio_inst_CLK,
hps_io_hps_io_sdio_inst_D2,
hps_io_hps_io_sdio_inst_D3,
hps_io_hps_io_usb1_inst_D0,
hps_io_hps_io_usb1_inst_D1,
hps_io_hps_io_usb1_inst_D2,
hps_io_hps_io_usb1_inst_D3,
hps_io_hps_io_usb1_inst_D4,
hps_io_hps_io_usb1_inst_D5,
   hps_io_hps_io_usb1_inst_D6,
   hps_io_hps_io_usb1_inst_D7,
   hps_io_hps_io_usb1_inst_CLK,
   hps_io_hps_io_usb1_inst_STP,
   hps_io_hps_io_usb1_inst_DIR,
   hps_io_hps_io_usb1_inst_NXT,
   hps_io_hps_io_spim0_inst_CLK,
   hps_io_hps_io_spim0_inst_MOSI,
   hps_io_hps_io_spim0_inst_MISO,
   hps_io_hps_io_spim0_inst_SS0,
   hps_io_hps_io_spim1_inst_CLK,
   hps_io_hps_io_spim1_inst_MOSI,
   hps_io_hps_io_spim1_inst_MISO,
   hps_io_hps_io_spim1_inst_SS0,
   hps_io_hps_io_uart0_inst_RX,
   hps_io_hps_io_uart0_inst_TX,
   hps_io_hps_io_i2c1_inst_SDA,
   hps_io_hps_io_i2c1_inst_SCL,
   hps_io_hps_io_gpio_inst_GPIO00,

   ////////////HMSC///////////////////////////
   HSMC_CLKIN_p,
HSMC_XAUI_RX_p0,
HSMC_XAUI_TX_p0,
   MDC2,
   MDIO2,
   MDC1,
   MDIO1,
   PRTAD02,
   PRTAD01,
   PRTAD4,
   PRTAD3,
   PRTAD2,
   PRTAD1,
   TXONOFF1,
   TXONOFF2,
   OPINLVL,
   OPOUTLVL,
   PHYRESET,
   USER_LED_G,
   USER_LED_R,
   CONFIG1_1,
   CONFIG0_1,
   CONFIG1_2,
   CONFIG0_2,
   SS338_CLKIN,
   GPIO0_1,
   GPIO1_1,
    GPIO0_2,
    GPIO1_2,
    SER_BOOT,
    SMBSPDSEL1,
    SMBSPDSEL2,
    SMBWEN,
    NVMA1SEL,
    NVMPROT,
    OPRXLOS2,
    OPTXFLT2,
    SFP_TXDIS2,
    SFP_TXRS20,
    LASI2
);


//=======================================================
//  PORT declarations
//=======================================================

‘ifdef ENABLE_HPS
    ///////// HPS /////////
    input         HPS_CLOCK_25;
    input         HPS_CLOCK_50;
    input         HPS_CONV_USB_n;
    output [14:0] HPS_DDR3_A;
    output [2:0]  HPS_DDR3_BA;
    output        HPS_DDR3_CAS_n;
    output        HPS_DDR3_CKE;
    output        HPS_DDR3_CK_n;
    output        HPS_DDR3_CK_p;
    output        HPS_DDR3_CS_n;
    output [3:0]  HPS_DDR3_DM;
    inout  [31:0] HPS_DDR3_DQ;
    inout  [3:0]  HPS_DDR3_DQS_n;
    inout  [3:0]  HPS_DDR3_DQS_p;
    output        HPS_DDR3_ODT;
    output        HPS_DDR3_RAS_n;
    output        HPS_DDR3_RESET_n;
    input         HPS_DDR3_RZQ;
    output        HPS_DDR3_WE_n;
    input         HPS_ENET_GTX_CLK;
    input         HPS_ENET_INT_n;
    output        HPS_ENET_MDC;
    inout         HPS_ENET_MDIO;
    output        HPS_ENET_RESET_n;
    input         HPS_ENET_RX_CLK;
    input  [3:0]  HPS_ENET_RX_DATA;
    input         HPS_ENET_RX_DV;
    output [3:0]  HPS_ENET_TX_DATA;
    output        HPS_ENET_TX_EN;
    inout  [3:0]  HPS_FLASH_DATA;
    output        HPS_FLASH_DCLK;
    output        HPS_FLASH_NCSO;
    input         HPS_GSENSOR_INT;
    inout         HPS_I2C_CLK;
    inout         HPS_I2C_SDA;
    inout  [3:0]  HPS_KEY;
    output        HPS_LCM_D_C;
    output        HPS_LCM_RST_N;
    input         HPS_LCM_SPIM_CLK;
    inout         HPS_LCM_SPIM_MISO;
    output        HPS_LCM_SPIM_MOSI;
    output        HPS_LCM_SPIM_SS;
    output [3:0]  HPS_LED;
    inout         HPS_LTC_GPIO;
    input         HPS_RESET_n;
    output        HPS_SD_CLK;
    inout         HPS_SD_CMD;
    inout  [3:0]  HPS_SD_DATA;
    output        HPS_SPIM_CLK;
    input         HPS_SPIM_MISO;
    output        HPS_SPIM_MOSI;
    output        HPS_SPIM_SS;
    input  [3:0]  HPS_SW;
    input         HPS_UART_RX;
    output        HPS_UART_TX;
    input         HPS_USB_CLKOUT;
    inout  [7:0]  HPS_USB_DATA;
    input         HPS_USB_DIR;
    input         HPS_USB_NXT;
    output        HPS_USB_RESET_PHY;
    output        HPS_USB_STP;
    input         HPS_WARM_RST_n;
‘endif /*ENABLE_HPS*/

    ///////// KEY /////////
    input [3:0] KEY;

    ///////// LED /////////
    output [3:0] LED;

    ///////// OSC /////////
    input OSC_50_B3B;
    input OSC_50_B4A;
    input OSC_50_B5B;
    input OSC_50_B8A;


    ///////// RESET /////////
    input RESET_n;

    ///////// SI5338 /////////
    inout SI5338_SCL;
    inout SI5338_SDA;

    ///////// SW /////////
    input [3:0] SW;

    /////////hps pin///////
    output wire [14:0] memory_mem_a;
    output wire [2:0]  memory_mem_ba;
    output wire        memory_mem_ck;
    output wire        memory_mem_ck_n;
    output wire        memory_mem_cke;
    output wire        memory_mem_cs_n;
    output wire        memory_mem_ras_n;
    output wire        memory_mem_cas_n;
    output wire        memory_mem_we_n;
    output wire        memory_mem_reset_n;
    inout  wire [31:0] memory_mem_dq;
    inout  wire [3:0]  memory_mem_dqs;
    inout  wire [3:0]  memory_mem_dqs_n;
    output wire        memory_mem_odt;
    output wire [3:0]  memory_mem_dm;
    input  wire        memory_oct_rzqin;
    output wire        hps_io_hps_io_emac1_inst_TX_CLK;
    output wire        hps_io_hps_io_emac1_inst_TXD0;
    output wire        hps_io_hps_io_emac1_inst_TXD1;
    output wire        hps_io_hps_io_emac1_inst_TXD2;
    output wire        hps_io_hps_io_emac1_inst_TXD3;
    input  wire        hps_io_hps_io_emac1_inst_RXD0;
    inout  wire        hps_io_hps_io_emac1_inst_MDIO;
    output wire        hps_io_hps_io_emac1_inst_MDC;
    input  wire        hps_io_hps_io_emac1_inst_RX_CTL;
    output wire        hps_io_hps_io_emac1_inst_TX_CTL;
    input  wire        hps_io_hps_io_emac1_inst_RX_CLK;
    input  wire        hps_io_hps_io_emac1_inst_RXD1;
    input  wire        hps_io_hps_io_emac1_inst_RXD2;
    input  wire        hps_io_hps_io_emac1_inst_RXD3;
    inout  wire        hps_io_hps_io_qspi_inst_IO0;
    inout  wire        hps_io_hps_io_qspi_inst_IO1;
    inout  wire        hps_io_hps_io_qspi_inst_IO2;
    inout  wire        hps_io_hps_io_qspi_inst_IO3;
```

```
    output wire        hps_io_hps_io_qspi_inst_SS0;
    output wire        hps_io_hps_io_qspi_inst_CLK;
    inout  wire        hps_io_hps_io_sdio_inst_CMD;
    inout  wire        hps_io_hps_io_sdio_inst_D0;
    inout  wire        hps_io_hps_io_sdio_inst_D1;
    output wire        hps_io_hps_io_sdio_inst_CLK;
    inout  wire        hps_io_hps_io_sdio_inst_D2;
    inout  wire        hps_io_hps_io_sdio_inst_D3;
    inout  wire        hps_io_hps_io_usb1_inst_D0;
    inout  wire        hps_io_hps_io_usb1_inst_D1;
    inout  wire        hps_io_hps_io_usb1_inst_D2;
    inout  wire        hps_io_hps_io_usb1_inst_D3;
    inout  wire        hps_io_hps_io_usb1_inst_D4;
    inout  wire        hps_io_hps_io_usb1_inst_D5;
    inout  wire        hps_io_hps_io_usb1_inst_D6;
    inout  wire        hps_io_hps_io_usb1_inst_D7;
    input  wire        hps_io_hps_io_usb1_inst_CLK;
    output wire        hps_io_hps_io_usb1_inst_STP;
    input  wire        hps_io_hps_io_usb1_inst_DIR;
    input  wire        hps_io_hps_io_usb1_inst_NXT;
    output wire        hps_io_hps_io_spim0_inst_CLK;
    output wire        hps_io_hps_io_spim0_inst_MOSI;
    input  wire        hps_io_hps_io_spim0_inst_MISO;
    output wire        hps_io_hps_io_spim0_inst_SS0;
    output wire        hps_io_hps_io_spim1_inst_CLK;
    output wire        hps_io_hps_io_spim1_inst_MOSI;
    input  wire        hps_io_hps_io_spim1_inst_MISO;
    output wire        hps_io_hps_io_spim1_inst_SS0;
    input  wire        hps_io_hps_io_uart0_inst_RX;
    output wire        hps_io_hps_io_uart0_inst_TX;
    inout  wire        hps_io_hps_io_i2c1_inst_SDA;
    inout  wire        hps_io_hps_io_i2c1_inst_SCL;
    inout  wire        hps_io_hps_io_gpio_inst_GPIO00;


    ////////HSMC///////////////////////
    input              HSMC_CLKIN_p;
    input  logic [3:0] HSMC_XAUI_RX_p0;
output logic [3:0] HSMC_XAUI_TX_p0;
    output logic       MDC2;
    inout  logic       MDIO2;
    output logic       MDC1;
    inout  logic       MDIO1;
    output logic       PRTAD02;
    output logic       PRTAD01;
    output logic       PRTAD4;
    output logic       PRTAD3;
    output logic       PRTAD2;
    output logic       PRTAD1;
    output logic       TXONOFF1;
    output logic       TXONOFF2;
    output logic       OPINLVL;
    output logic       OPOUTLVL;
    output logic       PHYRESET;
    output logic [7:0] USER_LED_G;
    output logic [7:0] USER_LED_R;
    output logic       CONFIG1_1;
    output logic       CONFIG0_1;
    output logic       CONFIG1_2;
    output logic       CONFIG0_2;
    output logic       SS338_CLKIN;
    inout  logic       GPIO0_1;
    inout  logic       GPIO1_1;
    inout  logic       GPIO0_2;
    inout  logic       GPIO1_2;
    output logic       SER_BOOT;
    output logic       SMBSPDSEL1;
    output logic       SMBSPDSEL2;
    inout  logic       SMBWEN;
    inout  logic       NVMA1SEL;
    inout  logic       NVMPROT;
    input  logic       OPRXLOS2;
    input  logic       OPTXFLT2;
    input  logic       SFP_TXDIS2;
    input  logic       SFP_TXRS20;
    input  logic       LASI2;


    //=======================================================
    //  REG/WIRE declarations
    //=======================================================

    // MAC
    logic mac_tx_ready;
    logic mac_rx_ready;

    // Make the FPGA reset cause an HPS reset
    reg [19:0]   hps_reset_counter = 20’h0;
    reg          hps_fpga_reset_n = 0;
    wire         MDIN1;
    wire         MDOEN1;
    wire         MDO1;
    logic        RESET_N;                    // S5 active low
    wire         clk_buff0;
    wire         clk_buff1;
    wire         clk_buff2;
    wire  [71:0]  link;
    wire         link_clk;
    logic [71:0] xgmii_buff ;
    logic [71:0] xgmii_rx_dc;
    logic [71:0] xgmii_tx_dc;
    logic [71:0] xgmiialigned;
    logic [31:0] crclink;
    logic        eop;
    logic        sop;
    logic        valid;
    logic        ready;
    logic        checksum_err;
    logic [2:0]  empty;
    logic [63:0] data;
    logic [1:0]  state;
    logic        eoptx;
    logic        soptx;
    logic        readytx;
    logic [2:0]  emptytx;
    logic [63:0] stdata;
    logic        validtx;
    logic [31:0] crclinktx;
    logic [31:0] cctx;
    logic [63:0] xgmiirevtx;
    logic [71:0] xgmiitx;
    logic [71:0] tx_buff;
    logic [63:0] databuff;
    logic        validbuff;
    logic        readybuff;
    logic        sopbuff;
    logic        eopbuff;
    logic [2:0]  emptybuff;
    parameter idle = 0, trig =1, lock =2;


    //=======================================================
    //  Structural coding
    //=======================================================


    // +-------------------------------------------------------------------
    // +
    assign CONFIG0_1 = 1'b1; // Configure BCM8727 from EEPROM
    assign CONFIG1_1 = 1'b0;
    assign CONFIG0_2 = 1'b1;
    assign CONFIG1_2 = 1'b0;
    assign SS338_CLKIN = 1'b0;
```

```
    // BCM8272C allow spi-rom to be removed
    // 1 = Boot microcode from spi proms
    assign SER_BOOT  = 1’b0;
    assign SMBSPDSEL1 = 1’b0;
    assign SMBSPDSEL2 = 1’b0;
    assign SMBWEN    = 1’b1;
    assign GPIO0_1   = 1’b0;
    assign GPIO1_1   = 1’b0;
    assign GPIO0_2   = 1’b0;
    assign GPIO1_2   = 1’b0;


    // ------------------------------------------------------------------
    // 1: EEPROM Slave Addr 52, 0: 50 addr:
    // During deassertion of BCM8727 reset,
    // latched into bit 10 of register 1.8002h
    assign NVMA1SEL = 1’b1;

    // when high protect non volatile memory
    assign NVMPROT  = 1’b0;

    // MDIO ports connection
‘ifdef ENABLE_MDIO
    assign MDIO1 = !MDOEN1? MDO1 : 1’bz;
    assign MDIN1 = MDIO1;
‘else
    assign MDC1  = 1’bz;
    assign MDIO1 = 1’bz;
‘endif
    assign MDC2  = 1’bz;
    assign MDIO2 = 1’bz;

    // +-------------------------------------------------------------------
    // +TXONOFF2


    assign PHYRESET =  (KEY[1]); // S4 active low
    // assign STOPMON  = ~KEY[2]; // S3 active high

    assign {PRTAD4,PRTAD3,PRTAD2,PRTAD1,PRTAD01} = 5’b00000;
    assign  PRTAD02 = 1’b0;
    // +
    assign TXONOFF1 = 1’b1;
    assign TXONOFF2 = 1’b1;
    assign OPOUTLVL = 1’b0; // 0 for active low OPTXENB/OPTXRST
    assign OPINLVL  = 1’b1; // 1 for active high OPRXLOS/TXONOFF

    assign USER_LED_R = 8’b00001111;
    assign USER_LED_G = 8’b11110000;

    //assign LED[0] = SFP_TXRS20;
    //assign LED[1] = SFP_TXDIS2;
    assign LED[0] = mac_tx_ready;
    assign LED[1] = mac_rx_ready;

    always @(posedge OSC_50_B4A) begin
        if (hps_reset_counter == 20’h ffffff) hps_fpga_reset_n <= 1;
            hps_reset_counter <= hps_reset_counter + 1;
end

    // Logic for resetting modules using KEY[0]
always @(posedge OSC_50_B4A)
        begin
            case(state)
                idle:
                    if (KEY[0]==0)
                        state <= trig;
                    else
                        state <= idle;
                trig:
                    if (KEY[0]==0)
                        state <= lock;
                    else
                        state <= idle;
                lock:
                    if (KEY[0]==0)
                        state <= lock;
                    else
                        state <= idle;
endcase
end
always @(state)
        begin
            case (state)
                idle:
                    RESET_N <= 1;
                trig:
RESET_N <= 0;
lock:
RESET_N <= 1;
endcase
end

//=======================================================
//  Modules Connection
//======================================================
// PLL REFERNCE  CLOCK FOR  XAUI
pll p1(.refclk(OSC_50_B4A), .rst(0), .outclk_0(clk_buff0), .locked());

// XAUI  PHY
xaui test(
.pll_ref_clk          (clk_buff0),
.xgmii_tx_clk         (link_clk),
.xgmii_rx_clk         (link_clk),
.xgmii_rx_dc          (xgmii_rx_dc),
.xgmii_tx_dc          (tx_buff),
.xaui_rx_serial_data  (HSMC_XAUI_RX_p0),
.xaui_tx_serial_data  (HSMC_XAUI_TX_p0),
.rx_ready       (LED[2]),
.tx_ready   (LED[3]),
.phy_mgmt_clk         (OSC_50_B4A),
.phy_mgmt_clk_reset   (!RESET_N),
.phy_mgmt_address     (3'h000),
.phy_mgmt_read        (0),
.phy_mgmt_readdata    (8'hFFFFFFFF),
.phy_mgmt_write       (0),
.phy_mgmt_writedata   (8'h00000000),
.phy_mgmt_waitrequest (0),
.reconfig_from_xcvr   (),
.reconfig_to_xcvr     ()
);


// SWAP
xgmii swap (
.clk(link_clk),
.xgmiidata(xgmii_rx_dc),
.reset(!RESET_N),
.flag(),
.xgmii(xgmiialigned)
);

// RX FAST MAC
mac_rx rx (
.clk(link_clk),
.xgmiidata(xgmiialigned),
.reset(!RESET_N),
.c(crclink),

.data(data),//output data
.empty(empty),
.sop(sop),
.eop(eop),
.valid(valid),
.ready(ready),
.newcrc(crclink),
.checksum_err(checksum_err)
);

// TX  BUFFER
txbuffer buffer(
.clk(link_clk),
.reset(!RESET_N),
.stdata(stdata),
.valid(validtx),
.sopin(soptx),
.eopin(eoptx),
.emptyin(emptytx),

.databuff(databuff),
.validout(validbuff),
.readybuff(readybuff),
.sopout(sopbuff),
.eopout(eopbuff),
.emptybuff(emptybuff),
.readytx(readytx)
);

// TX FAST MAC
mac_tx tx (
.clk(link_clk),
.stdata(databuff),            // input data
.reset(!RESET_N),
.c(crclinktx),
.sopin(sopbuff),
.eopin(eopbuff),
.valid(validbuff),
.ready(readybuff),
.empty(emptybuff),
.xgmiirev(xgmiirevtx),
.xgmii(tx_buff),
.newcrc(crclinktx),
.cc(cctx)
);

// LOOPBACKS
/*
// XAUI TX LOOPBACK
xaui_tx_loopback xaui_tx_loop (
    .clk(link_clk),
    .tx_out(HSMC_XAUI_TX_p0),
    .rx_in(HSMC_XAUI_RX_p0,)
);

// XAUI RX LOOPBACK
xaui_rx_loopback xaui_rx_loop (
    .clk(link_clk),
    .rx_out(xgmii_rx_dc),
    .tx_in(tx_buff)
);

//  FAST MAC TX LOOPBACK
fast_mac_tx_loopback fast_mac_tx_loop (
    .clk(link_clk),
    .tx_out(tx_buff),
    .swap_in(xgmii_rx_dc)
);

//  FAST MAC RX LOOPBACK
fast_mac_rx_loopback fast_mac_rx_loop (
    .clk(link_clk),
    .rx_out(data),
    .txbuff_in(databuff)
);

// Tx Buffer TX Loopback
txbuff_tx_loopback tx_buff_loop (
    .clk(link_clk),
    txbuff_out(databuff),
    rx_dc_in(data)
);

// Tx Buffer RX Loopback
txbuff_tx_loopback tx_buff_loop (
    .clk(link_clk),
    rx_mac_out(data),
    txbuff_in(stdata)
);
*/

////////////////QSYS:MM/ST  CONVERTER & HPS//////////..////
blazepps u0 (
.clk_clk                                (OSC_50_B4A),
.reset_reset_n                          (hps_fpga_reset_n),
.memory_mem_a                           (memory_mem_a),
.memory_mem_ba                          (memory_mem_ba),
.memory_mem_ck                          (memory_mem_ck),
.memory_mem_ck_n                        (memory_mem_ck_n),
.memory_mem_cke                         (memory_mem_cke),
.memory_mem_cs_n                          (memory_mem_cs_n),
.memory_mem_ras_n                         (memory_mem_ras_n),
.memory_mem_cas_n                         (memory_mem_cas_n),
.memory_mem_we_n                          (memory_mem_we_n),
.memory_mem_reset_n                       (memory_mem_reset_n),
.memory_mem_dq                            (memory_mem_dq),
.memory_mem_dqs                           (memory_mem_dqs),
.memory_mem_dqs_n                         (memory_mem_dqs_n),
.memory_mem_odt                           (memory_mem_odt),
.memory_mem_dm                            (memory_mem_dm),
.memory_oct_rzqin                         (memory_oct_rzqin),
.hps_io_hps_io_emac1_inst_TX_CLK       (hps_io_hps_io_emac1_inst_TX_CLK),
.hps_io_hps_io_emac1_inst_TXD0       (hps_io_hps_io_emac1_inst_TXD0),
.hps_io_hps_io_emac1_inst_TXD1         (hps_io_hps_io_emac1_inst_TXD1),
.hps_io_hps_io_emac1_inst_TXD2     (hps_io_hps_io_emac1_inst_TXD2),
.hps_io_hps_io_emac1_inst_TXD3     (hps_io_hps_io_emac1_inst_TXD3),
.hps_io_hps_io_emac1_inst_RXD0     (hps_io_hps_io_emac1_inst_RXD0),
.hps_io_hps_io_emac1_inst_MDIO     (hps_io_hps_io_emac1_inst_MDIO),
.hps_io_hps_io_emac1_inst_MDC      (hps_io_hps_io_emac1_inst_MDC),
.hps_io_hps_io_emac1_inst_RX_CTL   (hps_io_hps_io_emac1_inst_RX_CTL),

.hps_io_hps_io_emac1_inst_TX_CTL    (hps_io_hps_io_emac1_inst_TX_CTL),

.hps_io_hps_io_emac1_inst_RX_CLK    (hps_io_hps_io_emac1_inst_RX_CLK),

.hps_io_hps_io_emac1_inst_RXD1     (hps_io_hps_io_emac1_inst_RXD1),
.hps_io_hps_io_emac1_inst_RXD2     (hps_io_hps_io_emac1_inst_RXD2),
.hps_io_hps_io_emac1_inst_RXD3     (hps_io_hps_io_emac1_inst_RXD3),
.hps_io_hps_io_qspi_inst_IO0       (hps_io_hps_io_qspi_inst_IO0),
.hps_io_hps_io_qspi_inst_IO1       (hps_io_hps_io_qspi_inst_IO1),
.hps_io_hps_io_qspi_inst_IO2       (hps_io_hps_io_qspi_inst_IO2),
.hps_io_hps_io_qspi_inst_IO3       (hps_io_hps_io_qspi_inst_IO3),
.hps_io_hps_io_qspi_inst_SS0       (hps_io_hps_io_qspi_inst_SS0),
.hps_io_hps_io_qspi_inst_CLK       (hps_io_hps_io_qspi_inst_CLK),
.hps_io_hps_io_sdio_inst_CMD       (hps_io_hps_io_sdio_inst_CMD),
.hps_io_hps_io_sdio_inst_D0           (hps_io_hps_io_sdio_inst_D0),
.hps_io_hps_io_sdio_inst_D1        (hps_io_hps_io_sdio_inst_D1),
.hps_io_hps_io_sdio_inst_CLK          (hps_io_hps_io_sdio_inst_CLK),
.hps_io_hps_io_sdio_inst_D2         (hps_io_hps_io_sdio_inst_D2),
.hps_io_hps_io_sdio_inst_D3        (hps_io_hps_io_sdio_inst_D3),
.hps_io_hps_io_usb1_inst_D0        (hps_io_hps_io_usb1_inst_D0),
.hps_io_hps_io_usb1_inst_D1        (hps_io_hps_io_usb1_inst_D1),
.hps_io_hps_io_usb1_inst_D2           (hps_io_hps_io_usb1_inst_D2),
.hps_io_hps_io_usb1_inst_D3           (hps_io_hps_io_usb1_inst_D3),
.hps_io_hps_io_usb1_inst_D4           (hps_io_hps_io_usb1_inst_D4),
.hps_io_hps_io_usb1_inst_D5        (hps_io_hps_io_usb1_inst_D5),
.hps_io_hps_io_usb1_inst_D6           (hps_io_hps_io_usb1_inst_D6),
.hps_io_hps_io_usb1_inst_D7           (hps_io_hps_io_usb1_inst_D7),
.hps_io_hps_io_usb1_inst_CLK          (hps_io_hps_io_usb1_inst_CLK),
.hps_io_hps_io_usb1_inst_STP          (hps_io_hps_io_usb1_inst_STP),
.hps_io_hps_io_usb1_inst_DIR          (hps_io_hps_io_usb1_inst_DIR),
.hps_io_hps_io_usb1_inst_NXT          (hps_io_hps_io_usb1_inst_NXT),
.hps_io_hps_io_spim0_inst_CLK         (hps_io_hps_io_spim0_inst_CLK),
.hps_io_hps_io_spim0_inst_MOSI        (hps_io_hps_io_spim0_inst_MOSI),
.hps_io_hps_io_spim0_inst_MISO        (hps_io_hps_io_spim0_inst_MISO),
.hps_io_hps_io_spim0_inst_SS0         (hps_io_hps_io_spim0_inst_SS0),
.hps_io_hps_io_spim1_inst_CLK         (hps_io_hps_io_spim1_inst_CLK),
.hps_io_hps_io_spim1_inst_MOSI        (hps_io_hps_io_spim1_inst_MOSI),
.hps_io_hps_io_spim1_inst_MISO        (hps_io_hps_io_spim1_inst_MISO),
.hps_io_hps_io_spim1_inst_SS0        (hps_io_hps_io_spim1_inst_SS0),
.hps_io_hps_io_uart0_inst_RX       (hps_io_hps_io_uart0_inst_RX),
.hps_io_hps_io_uart0_inst_TX       (hps_io_hps_io_uart0_inst_TX),
.hps_io_hps_io_i2c1_inst_SDA       (hps_io_hps_io_i2c1_inst_SDA),
.hps_io_hps_io_i2c1_inst_SCL       (hps_io_hps_io_i2c1_inst_SCL),
.xaui_clk_clk                             (link_clk),
.xaui_reset_reset_n                       (RESET_N)
   .stream_src_data                           (stdata),
   .stream_src_valid                          (validtx),
   .stream_src_ready                          (readytx),
   .stream_src_startofpacket                  (soptx),
   .stream_src_endofpacket                    (eoptx),
   .stream_src_empty                          (emptytx),
   .stream_sink_data                          (data),
   .stream_sink_valid                         (valid),
   .stream_sink_ready                         (ready),
   .stream_sink_startofpacket                 (sop),
   .stream_sink_endofpacket                   (eop),
   .stream_sink_empty                         (empty)
);
endmodule
```

mac_tx.sv (adapter from previous project)

```
module mac_tx(

  input logic clk,
  input logic [63:0] stdata,// input data
  input logic reset,
  input logic [31:0]c,
  input logic sopin,
  input logic eopin,
  input logic valid,
  input logic [2:0] empty,
  output logic ready,
  output logic [63:0]xgmiirev,//reverse endian for debuggin
  output logic [71:0] xgmii,
  output logic [31:0] newcrc,
  output logic [31:0] cc); /////cc is for debugging, inverted
  logic [63:0]swapxgmiidata;
  logic eop1;
  logic eop2;
  logic eop3;
  logic valid1;
  logic [63:0] stdata1;
  logic sop;
  logic eop;

  assign sop = sopin && valid;
  assign eop = eopin && valid;
  assign ready = 1;

  always @(posedge clk)
  begin
    eop1 <=eop;
    eop2 <=eop1;
    eop3 <=eop2;
    valid1 <= valid;
    stdata1 <= stdata;
  end

  integer i;
  integer g;
  integer k;

  initial begin
    newcrc<=32'hFFFFFFFF;
    xgmii <= {1'b1, 8'h07,1'b1, 8'h07,1'b1, 8'h07, 1'b1, 8'h07,1'b1,
              8'h07,1'b1, 8'h07,1'b1, 8'h07,1'b1, 8'h07};
end

    always@* begin
        for (i=0;i<8;i++)begin
          for(g=0;g<8;g++)begin
             xgmiirev[g+i*8]=stdata[(7-g)+i*8];
          end
        end


        for (k=0;k<32;k++)begin
           cc[k] = ~newcrc[31-k];
        end
     end
  always @(posedge clk) begin
    if (reset == 1)
    begin
      newcrc<=32’hFFFFFFFF;
    end else if (sop)
      begin
        xgmii <= {1’b0, 8’h55,1’b0, 8’h55,1’b0, 8’h55,1’b0,
        8’hD5,1’b1, 8’hFB,1’b0, 8’h55,1’b0, 8’h55, 1’b0, 8’h55};
      end else if (valid1) //middle cases
    begin
          xgmii <= {1’b0,  stdata1[31:24], 1’b0, stdata1[23:16], 1’b0,
           stdata1[15:8], 1’b0, stdata1[7:0], 1’b0, stdata1[63:56],
           1’b0, stdata1[55:48], 1’b0, stdata1[47:40], 1’b0, stdata1[39:32]};
   end else if (eop2)
     begin
           xgmii <= { 1’b1, 8’hFD,1’b1, 8’hBC,1’b1, 8’hBC, 1’b1,
           8’hBC,1’b0, cc[7:0], 1’b0, cc[15:8], 1’b0, cc[23:16], 1’b0,
            cc[31:24]};
        end else if ( eop3)
        begin
          xgmii <= {1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07, 1’b1,
          8’h07,1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07};
          newcrc <=32’hFFFFFFFF;
        end else
        begin
          xgmii <= {1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07, 1’b1, 8’h07,
          1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07,1’b1, 8’h07};
        end

  if(valid)
    begin
newcrc[0]    = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[37] ^ xgmiirev[34] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[16] ^ xgmiirev[12] ^
xgmiirev[10] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[0] ^ c[0] ^ c[2] ^ c[5] ^
c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[18] ^ c[21] ^ c[22] ^ c[23] ^ c[26] ^ c[28] ^
c[29] ^ c[31];
newcrc[1]    = xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^
gmiirev[58] ^ xgmiirev[56] ^ xgmiirev[53] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[38] ^
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[24] ^ xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[7] ^ xgmiirev[6] ^
xgmiirev[1] ^ xgmiirev[0] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^ c[6] ^ c[12] ^ c[14] ^
```

```
c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[24] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^
c[31];
newcrc[2]    = xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[55] ^
xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[44] ^ xgmiirev[39] ^
xgmiirev[38] ^ xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[32] ^
xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[18] ^
xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^ xgmiirev[9] ^
xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[2] ^ xgmiirev[1] ^
xgmiirev[0] ^ c[0] ^ c[3] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[12] ^ c[19] ^ c[20] ^
c[21] ^ c[23] ^ c[25] ^ c[26] ^ c[27];
newcrc[3]    = xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[56] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[45] ^ xgmiirev[40] ^
xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[33] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[27] ^ xgmiirev[25] ^ xgmiirev[19] ^
xgmiirev[18] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[14] ^ xgmiirev[10] ^
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[3] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[0] ^ c[1] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[8] ^ c[13] ^ c[20] ^
c[21] ^ c[22] ^ c[24] ^ c[26] ^ c[27] ^ c[28];
newcrc[4]    = xgmiirev[63] ^ xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^
xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[45] ^
xgmiirev[44] ^ xgmiirev[41] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[38] ^
xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[25] ^
xgmiirev[24] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[15] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[1] ^ c[6] ^ c[7] ^ c[8] ^ c[9] ^
c[12] ^ c[13] ^ c[14] ^ c[15] ^ c[16] ^ c[18] ^ c[25] ^ c[26] ^ c[27] ^ c[31];
newcrc[5]    = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[55] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[49] ^
xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^ xgmiirev[40] ^
xgmiirev[39] ^ xgmiirev[37] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[24] ^
xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[13] ^ xgmiirev[10] ^
xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[3] ^
xgmiirev[1] ^ xgmiirev[0] ^ c[5] ^ c[7] ^ c[8] ^ c[9] ^ c[10] ^ c[12] ^ c[14] ^
c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[23] ^ c[27] ^ c[29] ^ c[31];
newcrc[6]    = xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[56] ^ xgmiirev[55] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[47] ^
xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[41] ^ xgmiirev[40] ^
xgmiirev[38] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[25] ^ xgmiirev[22] ^
xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[14] ^ xgmiirev[11] ^ xgmiirev[8] ^
xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[6] ^ c[8] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[15] ^ c[18] ^
c[19] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[28] ^ c[30];
newcrc[7]    = xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[56] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[47] ^
xgmiirev[46] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[39] ^ xgmiirev[37] ^ xgmiirev[34] ^ xgmiirev[32] ^ xgmiirev[29] ^
xgmiirev[28] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^
xgmiirev[21] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[10] ^ xgmiirev[8] ^
```

```
xgmiirev[7] ^ xgmiirev[5] ^ xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[0] ^
c[2] ^ c[5] ^ c[7] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[15] ^ c[18] ^
c[19] ^ c[20] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[28];
newcrc[8]    = xgmiirev[63] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[46] ^
xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[40] ^ xgmiirev[38] ^
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[32] ^
xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[17] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[10] ^ xgmiirev[8] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^
c[6] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^
c[25] ^ c[27] ^ c[28] ^ c[31];
newcrc[9]    = xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[55] ^
xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[47] ^ xgmiirev[46] ^
xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^ xgmiirev[39] ^ xgmiirev[38] ^
xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[32] ^
xgmiirev[29] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[18] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[5] ^ xgmiirev[4] ^
xgmiirev[2] ^ xgmiirev[1] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[7] ^
c[9] ^ c[11] ^ c[12] ^ c[14] ^ c[15] ^ c[19] ^ c[20] ^ c[21] ^ c[23] ^ c[26] ^
c[28] ^ c[29];
newcrc[10]  = xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^
xgmiirev[58] ^ xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[50] ^
xgmiirev[42] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[36] ^ xgmiirev[35] ^
xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[19] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^
xgmiirev[9] ^ xgmiirev[5] ^ xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[0] ^
c[1] ^ c[3] ^ c[4] ^ c[7] ^ c[8] ^ c[10] ^ c[18] ^ c[20] ^ c[23] ^ c[24] ^
c[26] ^ c[27] ^ c[28] ^ c[30] ^ c[31];
newcrc[11]  = xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[56] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^
xgmiirev[40] ^ xgmiirev[36] ^ xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[20] ^
xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[14] ^ xgmiirev[12] ^
xgmiirev[9] ^ xgmiirev[4] ^ xgmiirev[3] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[1] ^
c[4] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[18] ^ c[19] ^
c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[26] ^ c[27];
newcrc[12]  = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[42] ^
xgmiirev[41] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[24] ^
xgmiirev[21] ^ xgmiirev[18] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^
xgmiirev[2] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[9] ^ c[10] ^ c[14] ^ c[15] ^ c[17]
^ c[18] ^ c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[24] ^ c[25] ^ c[27] ^ c[29] ^
c[31];
newcrc[13]  = xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[57] ^
```

```
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[43] ^ xgmiirev[42] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[25] ^ xgmiirev[22] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^
xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[3] ^
xgmiirev[2] ^ xgmiirev[1] ^ c[0] ^ c[10] ^ c[11] ^ c[15] ^ c[16] ^ c[18] ^
c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[25] ^ c[26] ^ c[28] ^ c[30];
newcrc[14]  = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[58] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^
xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[44] ^ xgmiirev[43] ^
xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[26] ^ xgmiirev[23] ^
xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[14] ^
xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[2] ^ c[0] ^ c[1] ^ c[11] ^ c[12] ^ c[16] ^ c[17] ^ c[19]
^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^ c[29] ^ c[31];
newcrc[15]  = xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[34] ^
xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[24] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[12] ^
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[5] ^ xgmiirev[4] ^
xgmiirev[3] ^ c[1] ^ c[2] ^ c[12] ^ c[13] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^
c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[30];
newcrc[16]  = xgmiirev[57] ^ xgmiirev[56] ^ xgmiirev[51] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[37] ^ xgmiirev[35] ^
xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[26] ^ xgmiirev[24] ^
xgmiirev[22] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[8] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[0] ^ c[0] ^
c[3] ^ c[5] ^ c[12] ^ c[14] ^ c[15] ^ c[16] ^ c[19] ^ c[24] ^ c[25];
newcrc[17]  = xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[52] ^ xgmiirev[49] ^
xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[38] ^ xgmiirev[36] ^
xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[25] ^
xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[14] ^
xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[1] ^ c[1] ^
c[4] ^ c[6] ^ c[13] ^ c[15] ^ c[16] ^ c[17] ^ c[20] ^ c[25] ^ c[26];
newcrc[18]  = xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[53] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[46] ^ xgmiirev[39] ^ xgmiirev[37] ^
xgmiirev[34] ^ xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[26] ^
xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[15] ^
xgmiirev[14] ^ xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[2] ^ c[0] ^
c[2] ^ c[5] ^ c[7] ^ c[14] ^ c[16] ^ c[17] ^ c[18] ^ c[21] ^ c[26] ^ c[27];
newcrc[19]  = xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[54] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[40] ^ xgmiirev[38] ^
xgmiirev[35] ^ xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[27] ^
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[20] ^ xgmiirev[16] ^
xgmiirev[15] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[3] ^ c[0] ^
c[1] ^ c[3] ^ c[6] ^ c[8] ^ c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[22] ^ c[27] ^
c[28];
```

```
newcrc[20]  = xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[55] ^ xgmiirev[52] ^
xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[41] ^ xgmiirev[39] ^
xgmiirev[36] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[23] ^ xgmiirev[21] ^ xgmiirev[17] ^
xgmiirev[16] ^ xgmiirev[12] ^ xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[4] ^ c[1] ^
c[2] ^ c[4] ^ c[7] ^ c[9] ^ c[16] ^ c[18] ^ c[19] ^ c[20] ^ c[23] ^ c[28] ^
c[29];
newcrc[21]  = xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[56] ^ xgmiirev[53] ^
xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[42] ^ xgmiirev[40] ^
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^
xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[18] ^
xgmiirev[17] ^ xgmiirev[13] ^ xgmiirev[10] ^ xgmiirev[9] ^ xgmiirev[5] ^ c[2] ^
c[3] ^ c[5] ^ c[8] ^ c[10] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^ c[24] ^ c[29] ^
c[30];
newcrc[22]  = xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[48] ^ xgmiirev[47] ^
xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^ xgmiirev[38] ^
xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^
xgmiirev[29] ^ xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[23] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[12] ^
xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[0] ^ c[2] ^ c[3] ^ c[4] ^ c[5] ^ c[6] ^
c[9] ^ c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[20] ^ c[23] ^ c[25] ^ c[26] ^
c[28] ^ c[29] ^ c[30];
newcrc[23]  = xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[56] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^
xgmiirev[46] ^ xgmiirev[42] ^ xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[36] ^
xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[16] ^
xgmiirev[15] ^ xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[1] ^
xgmiirev[0] ^ c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[7] ^ c[10] ^ c[14] ^ c[15] ^ c[17]
^ c[18] ^ c[22] ^ c[23] ^ c[24] ^ c[27] ^ c[28] ^ c[30];
newcrc[24]  = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[43] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[37] ^
xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[17] ^
xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[0] ^ c[3] ^ c[4] ^ c[5] ^ c[7] ^ c[8] ^ c[11] ^ c[15] ^ c[16] ^
c[18] ^ c[19] ^ c[23] ^ c[24] ^ c[25] ^ c[28] ^ c[29] ^ c[31];
newcrc[25]  = xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[58] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[48] ^
xgmiirev[44] ^ xgmiirev[41] ^ xgmiirev[40] ^ xgmiirev[38] ^ xgmiirev[37] ^
xgmiirev[36] ^ xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[22] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[17] ^
xgmiirev[15] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[3] ^ xgmiirev[2] ^ c[1] ^
c[4] ^ c[5] ^ c[6] ^ c[8] ^ c[9] ^ c[12] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^
c[24] ^ c[25] ^ c[26] ^ c[29] ^ c[30];
newcrc[26]  = xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[59] ^
```

```
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[49] ^
xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[26] ^
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[20] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[10] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[0] ^ c[6] ^ c[7] ^ c[9] ^ c[10] ^ c[12] ^ c[15] ^ c[16]
^ c[17] ^ c[20] ^ c[22] ^ c[23] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[30];
newcrc[27]  = xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^
xgmiirev[58] ^ xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[53] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^
xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[11] ^ xgmiirev[7] ^ xgmiirev[5] ^
xgmiirev[4] ^ xgmiirev[1] ^ c[0] ^ c[7] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[16]
^ c[17] ^ c[18] ^ c[21] ^ c[23] ^ c[24] ^ c[26] ^ c[28] ^ c[29] ^ c[30] ^
c[31];
newcrc[28]  = xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[59] ^
xgmiirev[57] ^ xgmiirev[56] ^ xgmiirev[54] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^
xgmiirev[40] ^ xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[12] ^ xgmiirev[8] ^ xgmiirev[6] ^ xgmiirev[5] ^
xgmiirev[2] ^ c[1] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[14] ^ c[17] ^ c[18] ^
c[19] ^ c[22] ^ c[24] ^ c[25] ^ c[27] ^ c[29] ^ c[30] ^ c[31];
newcrc[29]  = xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[21] ^
xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[3] ^ c[2] ^
c[9] ^ c[10] ^ c[12] ^ c[13] ^ c[15] ^ c[18] ^ c[19] ^ c[20] ^ c[23] ^ c[25] ^
c[26] ^ c[28] ^ c[30] ^ c[31];
newcrc[30]  = xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[58] ^
xgmiirev[56] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[48] ^
xgmiirev[46] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[35] ^
xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[14] ^
xgmiirev[10] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[4] ^ c[0] ^ c[3] ^ c[10] ^
c[11] ^ c[13] ^ c[14] ^ c[16] ^ c[19] ^ c[20] ^ c[21] ^ c[24] ^ c[26] ^ c[27] ^
 c[29] ^ c[31];
newcrc[31]  = xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[49] ^ xgmiirev[47] ^
xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[36] ^ xgmiirev[33] ^
xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[15] ^ xgmiirev[11] ^
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[5] ^ c[1] ^ c[4] ^ c[11] ^ c[12] ^ c[14] ^
c[15] ^ c[17] ^ c[20] ^ c[21] ^ c[22] ^ c[25] ^ c[27] ^ c[28] ^ c[30];
  end
```

```
end
endmodule
```

mac_rx.sv (adapter from previous project)

```
module mac_rx(
  input logic clk,
  input logic [71:0] xgmiidata,// input data
  input logic reset,
  input logic [31:0]c,//inputcrc
  output logic [63:0] data,//output data
  output logic [2:0] empty,
  output logic sop,
  output logic eop,
  output logic valid,
  output logic ready,
  output logic [31:0] newcrc,//outputcrc
  output logic checksum_err,
  output logic [31:0] cc); ///cc is for debugging, inverted
  logic [63:0] xgmii;
  logic [63:0] xgmiibuf;
  logic [31:0] newcrc8;
  logic [63:0]xgmiirev;
  logic [63:0]swapxgmiirev;
  logic [7:0] xgmiicontrol;
  logic valid_flag;
  logic valid_flag_buff;
  logic sop_flag;
  logic sop_flag_buff;
  logic sop_flag_buff2;
  logic eop_flag=0;
  logic eop_flag_buff =0;
  logic crc_flag;
  logic checksum_buff =0;
  logic checksum_flag=0;
  logic [2:0] empty_flag;
  logic [63:0]swapxgmiidata;
  logic[7:0] xgmiirev8;
  logic[15:0] xgmiirev16;
  logic[23:0] xgmiirev24;
  logic[31:0] xgmiirev32;
  logic[39:0] xgmiirev40;
  logic[47:0] xgmiirev48;
  logic[55:0] xgmiirev56;
  assign ready = 1;
assign xgmiicontrol = {xgmiidata[44], xgmiidata[53],xgmiidata[62],
xgmiidata[71],xgmiidata[8], xgmiidata[17], xgmiidata[26],
xgmiidata[35]};
```

```
assign xgmiirev8=swapxgmiirev[63:56];
assign xgmiirev16=swapxgmiirev[63:48];
assign xgmiirev24=swapxgmiirev[63:40];
assign xgmiirev32=swapxgmiirev[63:32];

assign xgmiirev40=xgmiirev[63:24];
assign xgmiirev48=xgmiirev[63:16];
assign xgmiirev56=xgmiirev[63:8];

assign swapxgmiidata[7:0] = xgmiidata[43:36];
assign swapxgmiidata[15:8] =  xgmiidata[52:45];
assign swapxgmiidata[23:16] = xgmiidata[61:54];
assign swapxgmiidata[31:24] = xgmiidata[70:63];
assign swapxgmiidata[39:32] = xgmiidata[7:0];
assign swapxgmiidata[47:40] = xgmiidata[16:9];
assign swapxgmiidata[55:48] = xgmiidata[25:18];
assign swapxgmiidata[63:56] = xgmiidata[34:27];

    integer i;
    integer g;
    integer k;

    always@* begin
        for (i=0;i<8;i++)begin
          for(g=0;g<8;g++)begin
             xgmiirev[g+i*8]=swapxgmiidata[(7-g)+i*8];
   swapxgmiirev[g+i*8]=swapxgmiidata[(7-g)+i*8];
          end
        end


        for (k=0;k<32;k++)begin
           cc[k] = ~newcrc[31-k];
        end
     end

    always @(posedge clk) begin

      if (reset == 1)
        begin
          xgmii[63:0]<= 64'b0;
          sop_flag <= 0;
          sop_flag_buff <= 0;
          eop <= 0;
      sop <=0;
          valid_flag <= 0;
          crc_flag <=0;
          valid <= 0;
          newcrc<=32'hFFFFFFFF;
          valid_flag_buff <= 0;
          checksum_err <=0;
        empty <=3'b0;
      end
      else begin
        empty <= empty_flag;
        eop_flag <= eop_flag_buff;
        ///detect sop///
        if(xgmiicontrol == 1) begin
          sop_flag<=1;
          crc_flag <= 1;
   end
     if(sop_flag) begin
       sop_flag_buff<= 1;
       sop_flag<=0;
     end
     if(sop_flag_buff) begin
       sop_flag_buff<= 0;
       sop_flag_buff2<=1;
     end
    if(sop_flag_buff2)begin
      sop <=1;
      valid <= 1;
      sop_flag_buff2 <=0;
    end
    else begin
  sop <= 0;
end
/////////////////////

///when in middle of packet///
data[63:0] <= xgmiibuf[63:0];
xgmiibuf[63:0]<=xgmii[63:0];

if(crc_flag)begin
newcrc[0]   <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[37] ^ xgmiirev[34] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[16] ^ xgmiirev[12] ^
xgmiirev[10] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[0] ^ c[0] ^ c[2] ^ c[5] ^
c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[18] ^ c[21] ^ c[22] ^ c[23] ^ c[26] ^ c[28] ^
c[29] ^ c[31];
newcrc[1]   <= xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^
xgmiirev[58] ^ xgmiirev[56] ^ xgmiirev[53] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[38] ^
```

```
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[24] ^ xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[7] ^ xgmiirev[6] ^
xgmiirev[1] ^ xgmiirev[0] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^ c[6] ^ c[12] ^ c[14] ^
c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[24] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^
c[31];
newcrc[2]   <= xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[55] ^
xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[44] ^ xgmiirev[39] ^
xgmiirev[38] ^ xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[32] ^
xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[18] ^
xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^ xgmiirev[9] ^
xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[2] ^ xgmiirev[1] ^
xgmiirev[0] ^ c[0] ^ c[3] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[12] ^ c[19] ^ c[20] ^
c[21] ^ c[23] ^ c[25] ^ c[26] ^ c[27];
newcrc[3]   <= xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[56] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[45] ^ xgmiirev[40] ^
xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[33] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[27] ^ xgmiirev[25] ^ xgmiirev[19] ^
xgmiirev[18] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[14] ^ xgmiirev[10] ^
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[3] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[0] ^ c[1] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[8] ^ c[13] ^ c[20] ^
c[21] ^ c[22] ^ c[24] ^ c[26] ^ c[27] ^ c[28];
newcrc[4]   <= xgmiirev[63] ^ xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^
xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[45] ^
xgmiirev[44] ^ xgmiirev[41] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[38] ^
xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[25] ^
xgmiirev[24] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[15] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[1] ^ c[6] ^ c[7] ^ c[8] ^ c[9] ^
c[12] ^ c[13] ^ c[14] ^ c[15] ^ c[16] ^ c[18] ^ c[25] ^ c[26] ^ c[27] ^ c[31];
newcrc[5]   <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[55] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[49] ^
xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^ xgmiirev[40] ^
xgmiirev[39] ^ xgmiirev[37] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[24] ^
xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[13] ^ xgmiirev[10] ^
xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[3] ^
xgmiirev[1] ^ xgmiirev[0] ^ c[5] ^ c[7] ^ c[8] ^ c[9] ^ c[10] ^ c[12] ^ c[14] ^
c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[23] ^ c[27] ^ c[29] ^ c[31];
newcrc[6]   <= xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[56] ^ xgmiirev[55] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[47] ^
xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[41] ^ xgmiirev[40] ^
xgmiirev[38] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[25] ^ xgmiirev[22] ^
xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[14] ^ xgmiirev[11] ^ xgmiirev[8] ^
xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[6] ^ c[8] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[15] ^ c[18] ^
c[19] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[28] ^ c[30];
newcrc[7]   <= xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[56] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[47] ^
```

```
xgmiirev[46] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[39] ^ xgmiirev[37] ^ xgmiirev[34] ^ xgmiirev[32] ^ xgmiirev[29] ^
xgmiirev[28] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^
xgmiirev[21] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[10] ^ xgmiirev[8] ^
xgmiirev[7] ^ xgmiirev[5] ^ xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[0] ^
c[2] ^ c[5] ^ c[7] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[15] ^ c[18] ^
c[19] ^ c[20] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[28];
newcrc[8]   <= xgmiirev[63] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[46] ^
xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[40] ^ xgmiirev[38] ^
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[32] ^
xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[17] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[10] ^ xgmiirev[8] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^
c[6] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^
c[25] ^ c[27] ^ c[28] ^ c[31];
newcrc[9]   <= xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[55] ^
xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[47] ^ xgmiirev[46] ^
xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^ xgmiirev[39] ^ xgmiirev[38] ^
xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[32] ^
xgmiirev[29] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[18] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[5] ^ xgmiirev[4] ^
xgmiirev[2] ^ xgmiirev[1] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[7] ^
c[9] ^ c[11] ^ c[12] ^ c[14] ^ c[15] ^ c[19] ^ c[20] ^ c[21] ^ c[23] ^ c[26] ^
c[28] ^ c[29];
newcrc[10] <= xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^
xgmiirev[58] ^ xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[50] ^
xgmiirev[42] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[36] ^ xgmiirev[35] ^
xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[19] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^
xgmiirev[9] ^ xgmiirev[5] ^ xgmiirev[3] ^ xgmiirev[2] ^ xgmiirev[0] ^ c[0] ^
c[1] ^ c[3] ^ c[4] ^ c[7] ^ c[8] ^ c[10] ^ c[18] ^ c[20] ^ c[23] ^ c[24] ^
c[26] ^ c[27] ^ c[28] ^ c[30] ^ c[31];
newcrc[11] <= xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[56] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^
xgmiirev[40] ^ xgmiirev[36] ^ xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[20] ^
xgmiirev[17] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[14] ^ xgmiirev[12] ^
xgmiirev[9] ^ xgmiirev[4] ^ xgmiirev[3] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[1] ^
c[4] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[18] ^ c[19] ^
c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[26] ^ c[27];
newcrc[12] <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[42] ^
xgmiirev[41] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[24] ^
xgmiirev[21] ^ xgmiirev[18] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[4] ^
```

```
xgmiirev[2] ^ xgmiirev[1] ^ xgmiirev[0] ^ c[9] ^ c[10] ^ c[14] ^ c[15] ^ c[17]
^ c[18] ^ c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[24] ^ c[25] ^ c[27] ^ c[29] ^
c[31];
newcrc[13] <= xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[58] ^ xgmiirev[57] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[43] ^ xgmiirev[42] ^
xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[25] ^ xgmiirev[22] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[13] ^
xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[3] ^
xgmiirev[2] ^ xgmiirev[1] ^ c[0] ^ c[10] ^ c[11] ^ c[15] ^ c[16] ^ c[18] ^
c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[25] ^ c[26] ^ c[28] ^ c[30];
newcrc[14] <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[58] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^
xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[44] ^ xgmiirev[43] ^
xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[26] ^ xgmiirev[23] ^
xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[15] ^ xgmiirev[14] ^
xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[2] ^ c[0] ^ c[1] ^ c[11] ^ c[12] ^ c[16] ^ c[17] ^ c[19]
^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^ c[29] ^ c[31];
newcrc[15] <= xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[34] ^
xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[24] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[15] ^ xgmiirev[12] ^
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[5] ^ xgmiirev[4] ^
xgmiirev[3] ^ c[1] ^ c[2] ^ c[12] ^ c[13] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^
c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[30];
newcrc[16] <= xgmiirev[57] ^ xgmiirev[56] ^ xgmiirev[51] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[37] ^ xgmiirev[35] ^
xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[26] ^ xgmiirev[24] ^
xgmiirev[22] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[13] ^
xgmiirev[12] ^ xgmiirev[8] ^ xgmiirev[5] ^ xgmiirev[4] ^ xgmiirev[0] ^ c[0] ^
c[3] ^ c[5] ^ c[12] ^ c[14] ^ c[15] ^ c[16] ^ c[19] ^ c[24] ^ c[25];
newcrc[17] <= xgmiirev[58] ^ xgmiirev[57] ^ xgmiirev[52] ^ xgmiirev[49] ^
xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[38] ^ xgmiirev[36] ^
xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[27] ^ xgmiirev[25] ^
xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[14] ^
xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[5] ^ xgmiirev[1] ^ c[1] ^
c[4] ^ c[6] ^ c[13] ^ c[15] ^ c[16] ^ c[17] ^ c[20] ^ c[25] ^ c[26];
newcrc[18] <= xgmiirev[59] ^ xgmiirev[58] ^ xgmiirev[53] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[46] ^ xgmiirev[39] ^ xgmiirev[37] ^
xgmiirev[34] ^ xgmiirev[32] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[26] ^
xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[15] ^
xgmiirev[14] ^ xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[2] ^ c[0] ^
c[2] ^ c[5] ^ c[7] ^ c[14] ^ c[16] ^ c[17] ^ c[18] ^ c[21] ^ c[26] ^ c[27];
newcrc[19] <= xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[54] ^ xgmiirev[51] ^
xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^ xgmiirev[40] ^ xgmiirev[38] ^
xgmiirev[35] ^ xgmiirev[33] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[27] ^
```

```
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[20] ^ xgmiirev[16] ^
xgmiirev[15] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[3] ^ c[0] ^
c[1] ^ c[3] ^ c[6] ^ c[8] ^ c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[22] ^ c[27] ^
c[28];
newcrc[20] <= xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[55] ^ xgmiirev[52] ^
xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^ xgmiirev[41] ^ xgmiirev[39] ^
xgmiirev[36] ^ xgmiirev[34] ^ xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[28] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[23] ^ xgmiirev[21] ^ xgmiirev[17] ^
xgmiirev[16] ^ xgmiirev[12] ^ xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[4] ^ c[1] ^
c[2] ^ c[4] ^ c[7] ^ c[9] ^ c[16] ^ c[18] ^ c[19] ^ c[20] ^ c[23] ^ c[28] ^ c[29];
newcrc[21] <= xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[56] ^ xgmiirev[53] ^
xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[42] ^ xgmiirev[40] ^
xgmiirev[37] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^
xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[18] ^
xgmiirev[17] ^ xgmiirev[13] ^ xgmiirev[10] ^ xgmiirev[9] ^ xgmiirev[5] ^ c[2] ^
c[3] ^ c[5] ^ c[8] ^ c[10] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^ c[24] ^ c[29] ^
c[30];
newcrc[22] <= xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[48] ^ xgmiirev[47] ^
xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^ xgmiirev[38] ^
xgmiirev[37] ^ xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^
xgmiirev[29] ^ xgmiirev[27] ^ xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[23] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[12] ^
xgmiirev[11] ^ xgmiirev[9] ^ xgmiirev[0] ^ c[2] ^ c[3] ^ c[4] ^ c[5] ^ c[6] ^
 c[9] ^ c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[20] ^ c[23] ^ c[25] ^ c[26] ^
 c[28] ^ c[29] ^ c[30];
newcrc[23] <= xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[56] ^
xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[50] ^ xgmiirev[49] ^ xgmiirev[47] ^
xgmiirev[46] ^ xgmiirev[42] ^ xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[36] ^
xgmiirev[35] ^ xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[17] ^ xgmiirev[16] ^
xgmiirev[15] ^ xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[6] ^ xgmiirev[1] ^
xgmiirev[0] ^ c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[7] ^ c[10] ^ c[14] ^ c[15] ^ c[17]
^ c[18] ^ c[22] ^ c[23] ^ c[24] ^ c[27] ^ c[28] ^ c[30];
newcrc[24] <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[51] ^ xgmiirev[50] ^ xgmiirev[48] ^
xgmiirev[47] ^ xgmiirev[43] ^ xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[37] ^
xgmiirev[36] ^ xgmiirev[35] ^ xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[28] ^
xgmiirev[27] ^ xgmiirev[21] ^ xgmiirev[20] ^ xgmiirev[18] ^ xgmiirev[17] ^
xgmiirev[16] ^ xgmiirev[14] ^ xgmiirev[10] ^ xgmiirev[7] ^ xgmiirev[2] ^
xgmiirev[1] ^ c[0] ^ c[3] ^ c[4] ^ c[5] ^ c[7] ^ c[8] ^ c[11] ^ c[15] ^ c[16] ^
c[18] ^ c[19] ^ c[23] ^ c[24] ^ c[25] ^ c[28] ^ c[29] ^ c[31];
newcrc[25] <= xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[58] ^ xgmiirev[57] ^
xgmiirev[56] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[49] ^ xgmiirev[48] ^
xgmiirev[44] ^ xgmiirev[41] ^ xgmiirev[40] ^ xgmiirev[38] ^ xgmiirev[37] ^
xgmiirev[36] ^ xgmiirev[33] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^
xgmiirev[22] ^ xgmiirev[21] ^ xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[17] ^
xgmiirev[15] ^ xgmiirev[11] ^ xgmiirev[8] ^ xgmiirev[3] ^ xgmiirev[2] ^ c[1] ^
```

```
c[4] ^ c[5] ^ c[6] ^ c[8] ^ c[9] ^ c[12] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^
c[24] ^ c[25] ^ c[26] ^ c[29] ^ c[30];
newcrc[26] <= xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^ xgmiirev[59] ^
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[54] ^ xgmiirev[52] ^ xgmiirev[49] ^
xgmiirev[48] ^ xgmiirev[47] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[39] ^ xgmiirev[38] ^ xgmiirev[31] ^ xgmiirev[28] ^ xgmiirev[26] ^
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[20] ^
xgmiirev[19] ^ xgmiirev[18] ^ xgmiirev[10] ^ xgmiirev[6] ^ xgmiirev[4] ^
xgmiirev[3] ^ xgmiirev[0] ^ c[6] ^ c[7] ^ c[9] ^ c[10] ^ c[12] ^ c[15] ^ c[16]
^ c[17] ^ c[20] ^ c[22] ^ c[23] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[30];
newcrc[27] <= xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[60] ^
xgmiirev[58] ^ xgmiirev[56] ^ xgmiirev[55] ^ xgmiirev[53] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[48] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^
xgmiirev[40] ^ xgmiirev[39] ^ xgmiirev[32] ^ xgmiirev[29] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[19] ^ xgmiirev[11] ^ xgmiirev[7] ^ xgmiirev[5] ^
xgmiirev[4] ^ xgmiirev[1] ^ c[0] ^ c[7] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[16]
^ c[17] ^ c[18] ^ c[21] ^ c[23] ^ c[24] ^ c[26] ^ c[28] ^ c[29] ^ c[30] ^
c[31];
newcrc[28] <= xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[61] ^ xgmiirev[59] ^
xgmiirev[57] ^ xgmiirev[56] ^ xgmiirev[54] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[49] ^ xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[41] ^
xgmiirev[40] ^ xgmiirev[33] ^ xgmiirev[30] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[22] ^ xgmiirev[21] ^
xgmiirev[20] ^ xgmiirev[12] ^ xgmiirev[8] ^ xgmiirev[6] ^ xgmiirev[5] ^
xgmiirev[2] ^ c[1] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[14] ^ c[17] ^ c[18] ^
c[19] ^ c[22] ^ c[24] ^ c[25] ^ c[27] ^ c[29] ^ c[30] ^ c[31];
newcrc[29] <= xgmiirev[63] ^ xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[58] ^
xgmiirev[57] ^ xgmiirev[55] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[50] ^
xgmiirev[47] ^ xgmiirev[45] ^ xgmiirev[44] ^ xgmiirev[42] ^ xgmiirev[41] ^
xgmiirev[34] ^ xgmiirev[31] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[25] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[21] ^
xgmiirev[13] ^ xgmiirev[9] ^ xgmiirev[7] ^ xgmiirev[6] ^ xgmiirev[3] ^ c[2] ^
c[9] ^ c[10] ^ c[12] ^ c[13] ^ c[15] ^ c[18] ^ c[19] ^ c[20] ^ c[23] ^ c[25] ^
c[26] ^ c[28] ^ c[30] ^ c[31];
newcrc[30] <= xgmiirev[63] ^ xgmiirev[61] ^ xgmiirev[59] ^ xgmiirev[58] ^
xgmiirev[56] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[51] ^ xgmiirev[48] ^
xgmiirev[46] ^ xgmiirev[45] ^ xgmiirev[43] ^ xgmiirev[42] ^ xgmiirev[35] ^
xgmiirev[32] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[26] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[22] ^ xgmiirev[14] ^
xgmiirev[10] ^ xgmiirev[8] ^ xgmiirev[7] ^ xgmiirev[4] ^ c[0] ^ c[3] ^ c[10] ^
c[11] ^ c[13] ^ c[14] ^ c[16] ^ c[19] ^ c[20] ^ c[21] ^ c[24] ^ c[26] ^ c[27] ^
c[29] ^ c[31];
newcrc[31] <= xgmiirev[62] ^ xgmiirev[60] ^ xgmiirev[59] ^ xgmiirev[57] ^
xgmiirev[54] ^ xgmiirev[53] ^ xgmiirev[52] ^ xgmiirev[49] ^ xgmiirev[47] ^
xgmiirev[46] ^ xgmiirev[44] ^ xgmiirev[43] ^ xgmiirev[36] ^ xgmiirev[33] ^
xgmiirev[31] ^ xgmiirev[30] ^ xgmiirev[29] ^ xgmiirev[28] ^ xgmiirev[27] ^
xgmiirev[25] ^ xgmiirev[24] ^ xgmiirev[23] ^ xgmiirev[15] ^ xgmiirev[11] ^
```

```
xgmiirev[9] ^ xgmiirev[8] ^ xgmiirev[5] ^ c[1] ^ c[4] ^ c[11] ^ c[12] ^ c[14] ^
c[15] ^ c[17] ^ c[20] ^ c[21] ^ c[22] ^ c[25] ^ c[27] ^ c[28] ^ c[30];
end
///
if(xgmiicontrol==8’h00)begin
  xgmii[7:0] <= xgmiidata[43:36];
  xgmii[15:8] <=  xgmiidata[52:45];
  xgmii[23:16] <= xgmiidata[61:54];
  xgmii[31:24] <= xgmiidata[70:63];
  xgmii[39:32] <= xgmiidata[7:0];
  xgmii[47:40] <= xgmiidata[16:9];
  xgmii[55:48] <= xgmiidata[25:18];
  xgmii[63:56] <= xgmiidata[34:27];
  xgmiibuf[63:0]<=xgmii[63:0];
end
  ///end of packet detect through terminating FD
  //Buff data is eop
  else if (xgmiidata[34:27]==8’hFD) begin
  xgmii[63:0] <= 0;
  eop_flag <= 1;
  empty<=4;
  empty_flag<=4;
  checksum_flag <=1;
  newcrc8<=c;
  crc_flag<=0;
end
else if (xgmiidata[25:18]==8’hFD) begin
  xgmii[63:56]<=xgmiidata[34:27];
  xgmii[55:0]<=0;
  empty<=3;
  empty_flag<=3;
  eop_flag <= 1;


newcrc8[0]  <= xgmiirev8[6] ^ xgmiirev8[0] ^ c[24] ^ c[30]; newcrc8[1]  <=
xgmiirev8[7] ^ xgmiirev8[6] ^ xgmiirev8[1] ^ xgmiirev8[0] ^ c[24] ^ c[25] ^
c[30] ^ c[31]; newcrc8[2]  <= xgmiirev8[7] ^ xgmiirev8[6] ^ xgmiirev8[2] ^
xgmiirev8[1] ^ xgmiirev8[0] ^ c[24] ^ c[25] ^ c[26] ^ c[30] ^ c[31]; newcrc8[3]
<= xgmiirev8[7] ^ xgmiirev8[3] ^ xgmiirev8[2] ^ xgmiirev8[1] ^ c[25] ^ c[26] ^
c[27] ^ c[31]; newcrc8[4]  <= xgmiirev8[6] ^ xgmiirev8[4] ^ xgmiirev8[3] ^
xgmiirev8[2] ^ xgmiirev8[0] ^ c[24] ^ c[26] ^ c[27] ^ c[28] ^ c[30]; newcrc8[5]
<= xgmiirev8[7] ^ xgmiirev8[6] ^ xgmiirev8[5] ^ xgmiirev8[4] ^ xgmiirev8[3] ^
xgmiirev8[1] ^ xgmiirev8[0] ^ c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[30] ^
c[31]; newcrc8[6]  <= xgmiirev8[7] ^ xgmiirev8[6] ^ xgmiirev8[5] ^ xgmiirev8[4]
^ xgmiirev8[2] ^ xgmiirev8[1] ^ c[25] ^ c[26] ^ c[28] ^ c[29] ^ c[30] ^ c[31];
newcrc8[7]  <= xgmiirev8[7] ^ xgmiirev8[5] ^ xgmiirev8[3] ^ xgmiirev8[2] ^
xgmiirev8[0] ^ c[24] ^ c[26] ^ c[27] ^ c[29] ^ c[31]; newcrc8[8]  <=
xgmiirev8[4] ^ xgmiirev8[3] ^ xgmiirev8[1] ^ xgmiirev8[0] ^ c[0] ^ c[24] ^ c[25]
```

```
^ c[27] ^ c[28]; newcrc8[9]  <= xgmiirev8[5] ^ xgmiirev8[4] ^ xgmiirev8[2] ^
xgmiirev8[1] ^ c[1] ^ c[25] ^ c[26] ^ c[28] ^ c[29]; newcrc8[10] <= xgmiirev8[5]
^ xgmiirev8[3] ^ xgmiirev8[2] ^ xgmiirev8[0] ^ c[2] ^ c[24] ^ c[26] ^ c[27] ^
c[29]; newcrc8[11] <= xgmiirev8[4] ^ xgmiirev8[3] ^ xgmiirev8[1] ^ xgmiirev8[0]
^ c[3] ^ c[24] ^ c[25] ^ c[27] ^ c[28]; newcrc8[12] <= xgmiirev8[6] ^
xgmiirev8[5] ^ xgmiirev8[4] ^ xgmiirev8[2] ^ xgmiirev8[1] ^ xgmiirev8[0] ^ c[4]
^ c[24] ^ c[25] ^ c[26] ^ c[28] ^ c[29] ^ c[30]; newcrc8[13] <= xgmiirev8[7] ^
xgmiirev8[6] ^ xgmiirev8[5] ^ xgmiirev8[3] ^ xgmiirev8[2] ^ xgmiirev8[1] ^ c[5]
^ c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[30] ^ c[31]; newcrc8[14] <= xgmiirev8[7] ^
xgmiirev8[6] ^ xgmiirev8[4] ^ xgmiirev8[3] ^ xgmiirev8[2] ^ c[6] ^ c[26] ^ c[27]
^ c[28] ^ c[30] ^ c[31]; newcrc8[15] <= xgmiirev8[7] ^ xgmiirev8[5] ^
xgmiirev8[4] ^ xgmiirev8[3] ^ c[7] ^ c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[16]
<= xgmiirev8[5] ^ xgmiirev8[4] ^ xgmiirev8[0] ^ c[8] ^ c[24] ^ c[28] ^ c[29];
newcrc8[17] <= xgmiirev8[6] ^ xgmiirev8[5] ^ xgmiirev8[1] ^ c[9] ^ c[25] ^ c[29]
^ c[30]; newcrc8[18] <= xgmiirev8[7] ^ xgmiirev8[6] ^ xgmiirev8[2] ^ c[10] ^
c[26] ^ c[30] ^ c[31]; newcrc8[19] <= xgmiirev8[7] ^ xgmiirev8[3] ^ c[11] ^
c[27] ^ c[31]; newcrc8[20] <= xgmiirev8[4] ^ c[12] ^ c[28]; newcrc8[21] <=
xgmiirev8[5] ^ c[13] ^ c[29]; newcrc8[22] <= xgmiirev8[0] ^ c[14] ^ c[24];
newcrc8[23] <= xgmiirev8[6] ^ xgmiirev8[1] ^ xgmiirev8[0] ^ c[15] ^ c[24] ^
c[25] ^ c[30]; newcrc8[24] <= xgmiirev8[7] ^ xgmiirev8[2] ^ xgmiirev8[1] ^ c[16]
^ c[25] ^ c[26] ^ c[31]; newcrc8[25] <= xgmiirev8[3] ^ xgmiirev8[2] ^ c[17] ^
c[26] ^ c[27]; newcrc8[26] <= xgmiirev8[6] ^ xgmiirev8[4] ^ xgmiirev8[3] ^
xgmiirev8[0] ^ c[18] ^ c[24] ^ c[27] ^ c[28] ^ c[30]; newcrc8[27] <=
xgmiirev8[7] ^ xgmiirev8[5] ^ xgmiirev8[4] ^ xgmiirev8[1] ^ c[19] ^ c[25] ^
c[28] ^ c[29] ^ c[31]; newcrc8[28] <= xgmiirev8[6] ^ xgmiirev8[5] ^ xgmiirev8[2]
^ c[20] ^ c[26] ^ c[29] ^ c[30]; newcrc8[29] <= xgmiirev8[7] ^ xgmiirev8[6] ^
xgmiirev8[3] ^ c[21] ^ c[27] ^ c[30] ^ c[31]; newcrc8[30] <= xgmiirev8[7] ^
xgmiirev8[4] ^ c[22] ^ c[28] ^ c[31]; newcrc8[31] <= xgmiirev8[5] ^ c[23] ^
c[29]; checksum_flag<=1; crc_flag<=0; end  else if (xgmiidata[16:9]==8'hFD)
begin  xgmii[55:48] <= xgmiidata[25:18]; xgmii[63:56] <= xgmiidata[34:27];
xgmii[47:0]<=0; empty<=2; empty_flag <= 2; eop_flag <= 1;   newcrc8[0]  <=
xgmiirev16[12] ^ xgmiirev16[10] ^ xgmiirev16[9] ^ xgmiirev16[6] ^ xgmiirev16[0]
^ c[16] ^ c[22] ^ c[25] ^ c[26] ^ c[28]; newcrc8[1]  <= xgmiirev16[13] ^
xgmiirev16[12] ^ xgmiirev16[11] ^ xgmiirev16[9] ^ xgmiirev16[7] ^ xgmiirev16[6]
^ xgmiirev16[1] ^ xgmiirev16[0] ^ c[16] ^ c[17] ^ c[22] ^ c[23] ^ c[25] ^ c[27]
^ c[28] ^ c[29]; newcrc8[2]  <= xgmiirev16[14] ^ xgmiirev16[13] ^ xgmiirev16[9]
^ xgmiirev16[8] ^ xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[2] ^ xgmiirev16[1]
^ xgmiirev16[0] ^ c[16] ^ c[17] ^ c[18] ^ c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[29]
^ c[30]; newcrc8[3]  <= xgmiirev16[15] ^ xgmiirev16[14] ^ xgmiirev16[10] ^
xgmiirev16[9] ^ xgmiirev16[8] ^ xgmiirev16[7] ^ xgmiirev16[3] ^ xgmiirev16[2] ^
xgmiirev16[1] ^ c[17] ^ c[18] ^ c[19] ^ c[23] ^ c[24] ^ c[25] ^ c[26] ^ c[30] ^
c[31]; newcrc8[4]  <= xgmiirev16[15] ^ xgmiirev16[12] ^ xgmiirev16[11] ^
xgmiirev16[8] ^ xgmiirev16[6] ^ xgmiirev16[4] ^ xgmiirev16[3] ^ xgmiirev16[2] ^
xgmiirev16[0] ^ c[16] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^ c[24] ^ c[27] ^ c[28] ^
c[31]; newcrc8[5]  <= xgmiirev16[13] ^ xgmiirev16[10] ^ xgmiirev16[7] ^
xgmiirev16[6] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[3] ^ xgmiirev16[1] ^
xgmiirev16[0] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[26] ^
c[29]; newcrc8[6]  <= xgmiirev16[14] ^ xgmiirev16[11] ^ xgmiirev16[8] ^
```

```
xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[2] ^
xgmiirev16[1] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[27] ^
c[30]; newcrc8[7]  <= xgmiirev16[15] ^ xgmiirev16[10] ^ xgmiirev16[8] ^
xgmiirev16[7] ^ xgmiirev16[5] ^ xgmiirev16[3] ^ xgmiirev16[2] ^ xgmiirev16[0] ^
c[16] ^ c[18] ^ c[19] ^ c[21] ^ c[23] ^ c[24] ^ c[26] ^ c[31]; newcrc8[8]  <=
xgmiirev16[12] ^ xgmiirev16[11] ^ xgmiirev16[10] ^ xgmiirev16[8] ^ xgmiirev16[4]
^ xgmiirev16[3] ^ xgmiirev16[1] ^ xgmiirev16[0] ^ c[16] ^ c[17] ^ c[19] ^ c[20]
^ c[24] ^ c[26] ^ c[27] ^ c[28]; newcrc8[9]  <= xgmiirev16[13] ^ xgmiirev16[12]
^ xgmiirev16[11] ^ xgmiirev16[9] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[2]
^ xgmiirev16[1] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^ c[25] ^ c[27] ^ c[28] ^ c[29];
newcrc8[10] <= xgmiirev16[14] ^ xgmiirev16[13] ^ xgmiirev16[9] ^ xgmiirev16[5] ^
xgmiirev16[3] ^ xgmiirev16[2] ^ xgmiirev16[0] ^ c[16] ^ c[18] ^ c[19] ^ c[21] ^
c[25] ^ c[29] ^ c[30]; newcrc8[11] <= xgmiirev16[15] ^ xgmiirev16[14] ^
xgmiirev16[12] ^ xgmiirev16[9] ^ xgmiirev16[4] ^ xgmiirev16[3] ^ xgmiirev16[1] ^
xgmiirev16[0] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[25] ^ c[28] ^ c[30] ^ c[31];
newcrc8[12] <= xgmiirev16[15] ^ xgmiirev16[13] ^ xgmiirev16[12] ^ xgmiirev16[9]
^ xgmiirev16[6] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[2] ^ xgmiirev16[1]
^ xgmiirev16[0] ^ c[16] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^ c[22] ^ c[25] ^ c[28]
^ c[29] ^ c[31]; newcrc8[13] <= xgmiirev16[14] ^ xgmiirev16[13] ^ xgmiirev16[10]
^ xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[5] ^ xgmiirev16[3] ^ xgmiirev16[2]
^ xgmiirev16[1] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[23] ^ c[26] ^ c[29]
^ c[30]; newcrc8[14] <= xgmiirev16[15] ^ xgmiirev16[14] ^ xgmiirev16[11] ^
xgmiirev16[8] ^ xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[4] ^ xgmiirev16[3] ^
xgmiirev16[2] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[27] ^ c[30] ^
c[31]; newcrc8[15] <= xgmiirev16[15] ^ xgmiirev16[12] ^ xgmiirev16[9] ^
xgmiirev16[8] ^ xgmiirev16[7] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[3] ^
c[19] ^ c[20] ^ c[21] ^ c[23] ^ c[24] ^ c[25] ^ c[28] ^ c[31]; newcrc8[16] <=
xgmiirev16[13] ^ xgmiirev16[12] ^ xgmiirev16[8] ^ xgmiirev16[5] ^ xgmiirev16[4]
^ xgmiirev16[0] ^ c[0] ^ c[16] ^ c[20] ^ c[21] ^ c[24] ^ c[28] ^ c[29];
newcrc8[17] <= xgmiirev16[14] ^ xgmiirev16[13] ^ xgmiirev16[9] ^ xgmiirev16[6] ^
xgmiirev16[5] ^ xgmiirev16[1] ^ c[1] ^ c[17] ^ c[21] ^ c[22] ^ c[25] ^ c[29] ^
c[30]; newcrc8[18] <= xgmiirev16[15] ^ xgmiirev16[14] ^ xgmiirev16[10] ^
xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[2] ^ c[2] ^ c[18] ^ c[22] ^ c[23] ^
c[26] ^ c[30] ^ c[31]; newcrc8[19] <= xgmiirev16[15] ^ xgmiirev16[11] ^
xgmiirev16[8] ^ xgmiirev16[7] ^ xgmiirev16[3] ^ c[3] ^ c[19] ^ c[23] ^ c[24] ^
c[27] ^ c[31]; newcrc8[20] <= xgmiirev16[12] ^ xgmiirev16[9] ^ xgmiirev16[8] ^
xgmiirev16[4] ^ c[4] ^ c[20] ^ c[24] ^ c[25] ^ c[28]; newcrc8[21] <=
xgmiirev16[13] ^ xgmiirev16[10] ^ xgmiirev16[9] ^ xgmiirev16[5] ^ c[5] ^ c[21] ^
c[25] ^ c[26] ^ c[29]; newcrc8[22] <= xgmiirev16[14] ^ xgmiirev16[12] ^
xgmiirev16[11] ^ xgmiirev16[9] ^ xgmiirev16[0] ^ c[6] ^ c[16] ^ c[25] ^ c[27] ^
c[28] ^ c[30]; newcrc8[23] <= xgmiirev16[15] ^ xgmiirev16[13] ^ xgmiirev16[9] ^
xgmiirev16[6] ^ xgmiirev16[1] ^ xgmiirev16[0] ^ c[7] ^ c[16] ^ c[17] ^ c[22] ^
c[25] ^ c[29] ^ c[31]; newcrc8[24] <= xgmiirev16[14] ^ xgmiirev16[10] ^
xgmiirev16[7] ^ xgmiirev16[2] ^ xgmiirev16[1] ^ c[8] ^ c[17] ^ c[18] ^ c[23] ^
c[26] ^ c[30]; newcrc8[25] <= xgmiirev16[15] ^ xgmiirev16[11] ^ xgmiirev16[8] ^
xgmiirev16[3] ^ xgmiirev16[2] ^ c[9] ^ c[18] ^ c[19] ^ c[24] ^ c[27] ^ c[31];
newcrc8[26] <= xgmiirev16[10] ^ xgmiirev16[6] ^ xgmiirev16[4] ^ xgmiirev16[3] ^
xgmiirev16[0] ^ c[10] ^ c[16] ^ c[19] ^ c[20] ^ c[22] ^ c[26]; newcrc8[27] <=
xgmiirev16[11] ^ xgmiirev16[7] ^ xgmiirev16[5] ^ xgmiirev16[4] ^ xgmiirev16[1] ^
c[11] ^ c[17] ^ c[20] ^ c[21] ^ c[23] ^ c[27]; newcrc8[28] <= xgmiirev16[12] ^
xgmiirev16[8] ^ xgmiirev16[6] ^ xgmiirev16[5] ^ xgmiirev16[2] ^ c[12] ^ c[18] ^
c[21] ^ c[22] ^ c[24] ^ c[28]; newcrc8[29] <= xgmiirev16[13] ^ xgmiirev16[9] ^
xgmiirev16[7] ^ xgmiirev16[6] ^ xgmiirev16[3] ^ c[13] ^ c[19] ^ c[22] ^ c[23] ^
c[25] ^ c[29]; newcrc8[30] <= xgmiirev16[14] ^ xgmiirev16[10] ^ xgmiirev16[8] ^
xgmiirev16[7] ^ xgmiirev16[4] ^ c[14] ^ c[20] ^ c[23] ^ c[24] ^ c[26] ^ c[30];
newcrc8[31] <= xgmiirev16[15] ^ xgmiirev16[11] ^ xgmiirev16[9] ^ xgmiirev16[8] ^
xgmiirev16[5] ^ c[15] ^ c[21] ^ c[24] ^ c[25] ^ c[27] ^ c[31]; checksum_flag<=1;
crc_flag<=0; end else if (xgmiidata[7:0]==8’hFD) begin  xgmii[47:40] <=
xgmiidata[16:9]; xgmii[55:48] <= xgmiidata[25:18]; xgmii[63:56] <=
xgmiidata[34:27]; xgmii[39:0]<=0; empty<=1; empty_flag<=1; eop_flag <= 1;
newcrc8[0]  <= xgmiirev24[16] ^ xgmiirev24[12] ^ xgmiirev24[10] ^ xgmiirev24[9]
^ xgmiirev24[6] ^ xgmiirev24[0] ^ c[8] ^ c[14] ^ c[17] ^ c[18] ^ c[20] ^ c[24];
newcrc8[1]  <= xgmiirev24[17] ^ xgmiirev24[16] ^ xgmiirev24[13] ^ xgmiirev24[12]
^ xgmiirev24[11] ^ xgmiirev24[9] ^ xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[1]
^ xgmiirev24[0] ^ c[8] ^ c[9] ^ c[14] ^ c[15] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^
c[24] ^ c[25]; newcrc8[2]  <= xgmiirev24[18] ^ xgmiirev24[17] ^ xgmiirev24[16] ^
xgmiirev24[14] ^ xgmiirev24[13] ^ xgmiirev24[9] ^ xgmiirev24[8] ^ xgmiirev24[7]
^ xgmiirev24[6] ^ xgmiirev24[2] ^ xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^
c[10] ^ c[14] ^ c[15] ^ c[16] ^ c[17] ^ c[21] ^ c[22] ^ c[24] ^ c[25] ^ c[26];
newcrc8[3]  <= xgmiirev24[19] ^ xgmiirev24[18] ^ xgmiirev24[17] ^ xgmiirev24[15]
^ xgmiirev24[14] ^ xgmiirev24[10] ^ xgmiirev24[9] ^ xgmiirev24[8] ^
xgmiirev24[7] ^ xgmiirev24[3] ^ xgmiirev24[2] ^ xgmiirev24[1] ^ c[9] ^ c[10] ^
c[11] ^ c[15] ^ c[16] ^ c[17] ^ c[18] ^ c[22] ^ c[23] ^ c[25] ^ c[26] ^ c[27];
newcrc8[4]  <= xgmiirev24[20] ^ xgmiirev24[19] ^ xgmiirev24[18] ^ xgmiirev24[15]
^ xgmiirev24[12] ^ xgmiirev24[11] ^ xgmiirev24[8] ^ xgmiirev24[6] ^
xgmiirev24[4] ^ xgmiirev24[3] ^ xgmiirev24[2] ^ xgmiirev24[0] ^ c[8] ^ c[10] ^
c[11] ^ c[12] ^ c[14] ^ c[16] ^ c[19] ^ c[20] ^ c[23] ^ c[26] ^ c[27] ^ c[28];
newcrc8[5]  <= xgmiirev24[21] ^ xgmiirev24[20] ^ xgmiirev24[19] ^ xgmiirev24[13]
^ xgmiirev24[10] ^ xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[5] ^ xgmiirev24[4]
^ xgmiirev24[3] ^ xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^
c[13] ^ c[14] ^ c[15] ^ c[18] ^ c[21] ^ c[27] ^ c[28] ^ c[29]; newcrc8[6]  <=
xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[20] ^ xgmiirev24[14] ^
xgmiirev24[11] ^ xgmiirev24[8] ^ xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[5] ^
xgmiirev24[4] ^ xgmiirev24[2] ^ xgmiirev24[1] ^ c[9] ^ c[10] ^ c[12] ^ c[13] ^
c[14] ^ c[15] ^ c[16] ^ c[19] ^ c[22] ^ c[28] ^ c[29] ^ c[30]; newcrc8[7]  <=
xgmiirev24[23] ^ xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[16] ^
xgmiirev24[15] ^ xgmiirev24[10] ^ xgmiirev24[8] ^ xgmiirev24[7] ^ xgmiirev24[5]
^ xgmiirev24[3] ^ xgmiirev24[2] ^ xgmiirev24[0] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^
c[15] ^ c[16] ^ c[18] ^ c[23] ^ c[24] ^ c[29] ^ c[30] ^ c[31]; newcrc8[8]  <=
xgmiirev24[23] ^ xgmiirev24[22] ^ xgmiirev24[17] ^ xgmiirev24[12] ^
xgmiirev24[11] ^ xgmiirev24[10] ^ xgmiirev24[8] ^ xgmiirev24[4] ^ xgmiirev24[3]
^ xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[16] ^ c[18] ^
c[19] ^ c[20] ^ c[25] ^ c[30] ^ c[31]; newcrc8[9]  <= xgmiirev24[23] ^
xgmiirev24[18] ^ xgmiirev24[13] ^ xgmiirev24[12] ^ xgmiirev24[11] ^
xgmiirev24[9] ^ xgmiirev24[5] ^ xgmiirev24[4] ^ xgmiirev24[2] ^ xgmiirev24[1] ^
c[9] ^ c[10] ^ c[12] ^ c[13] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^ c[26] ^ c[31];
```

```
newcrc8[10] <= xgmiirev24[19] ^ xgmiirev24[16] ^ xgmiirev24[14] ^ xgmiirev24[13]
^ xgmiirev24[9] ^ xgmiirev24[5] ^ xgmiirev24[3] ^ xgmiirev24[2] ^ xgmiirev24[0]
^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[17] ^ c[21] ^ c[22] ^ c[24] ^ c[27];
newcrc8[11] <= xgmiirev24[20] ^ xgmiirev24[17] ^ xgmiirev24[16] ^ xgmiirev24[15]
^ xgmiirev24[14] ^ xgmiirev24[12] ^ xgmiirev24[9] ^ xgmiirev24[4] ^
xgmiirev24[3] ^ xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^
c[17] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[28]; newcrc8[12] <=
xgmiirev24[21] ^ xgmiirev24[18] ^ xgmiirev24[17] ^ xgmiirev24[15] ^
xgmiirev24[13] ^ xgmiirev24[12] ^ xgmiirev24[9] ^ xgmiirev24[6] ^ xgmiirev24[5]
^ xgmiirev24[4] ^ xgmiirev24[2] ^ xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^
c[10] ^ c[12] ^ c[13] ^ c[14] ^ c[17] ^ c[20] ^ c[21] ^ c[23] ^ c[25] ^ c[26] ^
c[29]; newcrc8[13] <= xgmiirev24[22] ^ xgmiirev24[19] ^ xgmiirev24[18] ^
xgmiirev24[16] ^ xgmiirev24[14] ^ xgmiirev24[13] ^ xgmiirev24[10] ^
xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[5] ^ xgmiirev24[3] ^ xgmiirev24[2] ^
xgmiirev24[1] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[15] ^ c[18] ^ c[21] ^
c[22] ^ c[24] ^ c[26] ^ c[27] ^ c[30]; newcrc8[14] <= xgmiirev24[23] ^
xgmiirev24[20] ^ xgmiirev24[19] ^ xgmiirev24[17] ^ xgmiirev24[15] ^
xgmiirev24[14] ^ xgmiirev24[11] ^ xgmiirev24[8] ^ xgmiirev24[7] ^ xgmiirev24[6]
^ xgmiirev24[4] ^ xgmiirev24[3] ^ xgmiirev24[2] ^ c[10] ^ c[11] ^ c[12] ^ c[14]
^ c[15] ^ c[16] ^ c[19] ^ c[22] ^ c[23] ^ c[25] ^ c[27] ^ c[28] ^ c[31];
newcrc8[15] <= xgmiirev24[21] ^ xgmiirev24[20] ^ xgmiirev24[18] ^ xgmiirev24[16]
^ xgmiirev24[15] ^ xgmiirev24[12] ^ xgmiirev24[9] ^ xgmiirev24[8] ^
xgmiirev24[7] ^ xgmiirev24[5] ^ xgmiirev24[4] ^ xgmiirev24[3] ^ c[11] ^ c[12] ^
c[13] ^ c[15] ^ c[16] ^ c[17] ^ c[20] ^ c[23] ^ c[24] ^ c[26] ^ c[28] ^ c[29];
newcrc8[16] <= xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[19] ^ xgmiirev24[17]
^ xgmiirev24[13] ^ xgmiirev24[12] ^ xgmiirev24[8] ^ xgmiirev24[5] ^
xgmiirev24[4] ^ xgmiirev24[0] ^ c[8] ^ c[12] ^ c[13] ^ c[16] ^ c[20] ^ c[21] ^
c[25] ^ c[27] ^ c[29] ^ c[30]; newcrc8[17] <= xgmiirev24[23] ^ xgmiirev24[22] ^
xgmiirev24[20] ^ xgmiirev24[18] ^ xgmiirev24[14] ^ xgmiirev24[13] ^
xgmiirev24[9] ^ xgmiirev24[6] ^ xgmiirev24[5] ^ xgmiirev24[1] ^ c[9] ^ c[13] ^
c[14] ^ c[17] ^ c[21] ^ c[22] ^ c[26] ^ c[28] ^ c[30] ^ c[31]; newcrc8[18] <=
xgmiirev24[23] ^ xgmiirev24[21] ^ xgmiirev24[19] ^ xgmiirev24[15] ^
xgmiirev24[14] ^ xgmiirev24[10] ^ xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[2]
^ c[10] ^ c[14] ^ c[15] ^ c[18] ^ c[22] ^ c[23] ^ c[27] ^ c[29] ^ c[31];
newcrc8[19] <= xgmiirev24[22] ^ xgmiirev24[20] ^ xgmiirev24[16] ^ xgmiirev24[15]
^ xgmiirev24[11] ^ xgmiirev24[8] ^ xgmiirev24[7] ^ xgmiirev24[3] ^ c[11] ^ c[15]
^ c[16] ^ c[19] ^ c[23] ^ c[24] ^ c[28] ^ c[30]; newcrc8[20] <= xgmiirev24[23] ^
xgmiirev24[21] ^ xgmiirev24[17] ^ xgmiirev24[16] ^ xgmiirev24[12] ^
xgmiirev24[9] ^ xgmiirev24[8] ^ xgmiirev24[4] ^ c[12] ^ c[16] ^ c[17] ^ c[20] ^
c[24] ^ c[25] ^ c[29] ^ c[31]; newcrc8[21] <= xgmiirev24[22] ^ xgmiirev24[18] ^
xgmiirev24[17] ^ xgmiirev24[13] ^ xgmiirev24[10] ^ xgmiirev24[9] ^ xgmiirev24[5]
^ c[13] ^ c[17] ^ c[18] ^ c[21] ^ c[25] ^ c[26] ^ c[30]; newcrc8[22] <=
xgmiirev24[23] ^ xgmiirev24[19] ^ xgmiirev24[18] ^ xgmiirev24[16] ^
xgmiirev24[14] ^ xgmiirev24[12] ^ xgmiirev24[11] ^ xgmiirev24[9] ^ xgmiirev24[0]
^ c[8] ^ c[17] ^ c[19] ^ c[20] ^ c[22] ^ c[24] ^ c[26] ^ c[27] ^ c[31];
newcrc8[23] <= xgmiirev24[20] ^ xgmiirev24[19] ^ xgmiirev24[17] ^ xgmiirev24[16]
^ xgmiirev24[15] ^ xgmiirev24[13] ^ xgmiirev24[9] ^ xgmiirev24[6] ^
xgmiirev24[1] ^ xgmiirev24[0] ^ c[8] ^ c[9] ^ c[14] ^ c[17] ^ c[21] ^ c[23] ^
```

```
c[24] ^ c[25] ^ c[27] ^ c[28]; newcrc8[24] <= xgmiirev24[21] ^ xgmiirev24[20] ^
xgmiirev24[18] ^ xgmiirev24[17] ^ xgmiirev24[16] ^ xgmiirev24[14] ^
xgmiirev24[10] ^ xgmiirev24[7] ^ xgmiirev24[2] ^ xgmiirev24[1] ^ c[0] ^ c[9] ^
c[10] ^ c[15] ^ c[18] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[28] ^ c[29];
newcrc8[25] <= xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[19] ^ xgmiirev24[18]
^ xgmiirev24[17] ^ xgmiirev24[15] ^ xgmiirev24[11] ^ xgmiirev24[8] ^
xgmiirev24[3] ^ xgmiirev24[2] ^ c[1] ^ c[10] ^ c[11] ^ c[16] ^ c[19] ^ c[23] ^
c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[30]; newcrc8[26] <= xgmiirev24[23] ^
xgmiirev24[22] ^ xgmiirev24[20] ^ xgmiirev24[19] ^ xgmiirev24[18] ^
xgmiirev24[10] ^ xgmiirev24[6] ^ xgmiirev24[4] ^ xgmiirev24[3] ^ xgmiirev24[0] ^
c[2] ^ c[8] ^ c[11] ^ c[12] ^ c[14] ^ c[18] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^
c[31]; newcrc8[27] <= xgmiirev24[23] ^ xgmiirev24[21] ^ xgmiirev24[20] ^
xgmiirev24[19] ^ xgmiirev24[11] ^ xgmiirev24[7] ^ xgmiirev24[5] ^ xgmiirev24[4]
^ xgmiirev24[1] ^ c[3] ^ c[9] ^ c[12] ^ c[13] ^ c[15] ^ c[19] ^ c[27] ^ c[28] ^
c[29] ^ c[31]; newcrc8[28] <= xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[20] ^
xgmiirev24[12] ^ xgmiirev24[8] ^ xgmiirev24[6] ^ xgmiirev24[5] ^ xgmiirev24[2] ^
c[4] ^ c[10] ^ c[13] ^ c[14] ^ c[16] ^ c[20] ^ c[28] ^ c[29] ^ c[30];
newcrc8[29] <= xgmiirev24[23] ^ xgmiirev24[22] ^ xgmiirev24[21] ^ xgmiirev24[13]
^ xgmiirev24[9] ^ xgmiirev24[7] ^ xgmiirev24[6] ^ xgmiirev24[3] ^ c[5] ^ c[11] ^
c[14] ^ c[15] ^ c[17] ^ c[21] ^ c[29] ^ c[30] ^ c[31]; newcrc8[30] <=
xgmiirev24[23] ^ xgmiirev24[22] ^ xgmiirev24[14] ^ xgmiirev24[10] ^
xgmiirev24[8] ^ xgmiirev24[7] ^ xgmiirev24[4] ^ c[6] ^ c[12] ^ c[15] ^ c[16] ^
c[18] ^ c[22] ^ c[30] ^ c[31]; newcrc8[31] <= xgmiirev24[23] ^ xgmiirev24[15] ^
xgmiirev24[11] ^ xgmiirev24[9] ^ xgmiirev24[8] ^ xgmiirev24[5] ^ c[7] ^ c[13] ^
c[16] ^ c[17] ^ c[19] ^ c[23] ^ c[31]; checksum_flag<=1; crc_flag<=0; end else
if (xgmiidata[70:63]==8’hFD) begin  xgmii[39:32] <= xgmiidata[7:0]; xgmii[47:40]
<= xgmiidata[16:9]; xgmii[55:48] <= xgmiidata[25:18]; xgmii[63:56] <=
xgmiidata[34:27]; xgmii[32:0]<=0; empty<=0; empty_flag<=0; eop_flag <= 1;
newcrc8[0]  <= xgmiirev32[31] ^ xgmiirev32[30] ^ xgmiirev32[29] ^ xgmiirev32[28]
^ xgmiirev32[26] ^ xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[16] ^
xgmiirev32[12] ^ xgmiirev32[10] ^ xgmiirev32[9] ^ xgmiirev32[6] ^ xgmiirev32[0]
^ c[0] ^ c[6] ^ c[9] ^ c[10] ^ c[12] ^ c[16] ^ c[24] ^ c[25] ^ c[26] ^ c[28] ^
c[29] ^ c[30] ^ c[31]; newcrc8[1]  <= xgmiirev32[28] ^ xgmiirev32[27] ^
xgmiirev32[24] ^ xgmiirev32[17] ^ xgmiirev32[16] ^ xgmiirev32[13] ^
xgmiirev32[12] ^ xgmiirev32[11] ^ xgmiirev32[9] ^ xgmiirev32[7] ^ xgmiirev32[6]
^ xgmiirev32[1] ^ xgmiirev32[0] ^ c[0] ^ c[1] ^ c[6] ^ c[7] ^ c[9] ^ c[11] ^
c[12] ^ c[13] ^ c[16] ^ c[17] ^ c[24] ^ c[27] ^ c[28]; newcrc8[2]  <=
xgmiirev32[31] ^ xgmiirev32[30] ^ xgmiirev32[26] ^ xgmiirev32[24] ^
xgmiirev32[18] ^ xgmiirev32[17] ^ xgmiirev32[16] ^ xgmiirev32[14] ^
xgmiirev32[13] ^ xgmiirev32[9] ^ xgmiirev32[8] ^ xgmiirev32[7] ^ xgmiirev32[6] ^
xgmiirev32[2] ^ xgmiirev32[1] ^ xgmiirev32[0] ^ c[0] ^ c[1] ^ c[2] ^ c[6] ^ c[7]
^ c[8] ^ c[9] ^ c[13] ^ c[14] ^ c[16] ^ c[17] ^ c[18] ^ c[24] ^ c[26] ^ c[30] ^
c[31]; newcrc8[3]  <= xgmiirev32[31] ^ xgmiirev32[27] ^ xgmiirev32[25] ^
xgmiirev32[19] ^ xgmiirev32[18] ^ xgmiirev32[17] ^ xgmiirev32[15] ^
xgmiirev32[14] ^ xgmiirev32[10] ^ xgmiirev32[9] ^ xgmiirev32[8] ^ xgmiirev32[7]
^ xgmiirev32[3] ^ xgmiirev32[2] ^ xgmiirev32[1] ^ c[1] ^ c[2] ^ c[3] ^ c[7] ^
c[8] ^ c[9] ^ c[10] ^ c[14] ^ c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[25] ^ c[27] ^
c[31]; newcrc8[4]  <= xgmiirev32[31] ^ xgmiirev32[30] ^ xgmiirev32[29] ^
```

```
xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[20] ^ xgmiirev32[19] ^
xgmiirev32[18] ^ xgmiirev32[15] ^ xgmiirev32[12] ^ xgmiirev32[11] ^
xgmiirev32[8] ^ xgmiirev32[6] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ xgmiirev32[2] ^
xgmiirev32[0] ^ c[0] ^ c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[8] ^ c[11] ^ c[12] ^ c[15]
^ c[18] ^ c[19] ^ c[20] ^ c[24] ^ c[25] ^ c[29] ^ c[30] ^ c[31]; newcrc8[5]  <=
xgmiirev32[29] ^ xgmiirev32[28] ^ xgmiirev32[24] ^ xgmiirev32[21] ^
xgmiirev32[20] ^ xgmiirev32[19] ^ xgmiirev32[13] ^ xgmiirev32[10] ^
xgmiirev32[7] ^ xgmiirev32[6] ^ xgmiirev32[5] ^ xgmiirev32[4] ^ xgmiirev32[3] ^
xgmiirev32[1] ^ xgmiirev32[0] ^ c[0] ^ c[1] ^ c[3] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^
c[10] ^ c[13] ^ c[19] ^ c[20] ^ c[21] ^ c[24] ^ c[28] ^ c[29]; newcrc8[6]  <=
xgmiirev32[30] ^ xgmiirev32[29] ^ xgmiirev32[25] ^ xgmiirev32[22] ^
xgmiirev32[21] ^ xgmiirev32[20] ^ xgmiirev32[14] ^ xgmiirev32[11] ^
xgmiirev32[8] ^ xgmiirev32[7] ^ xgmiirev32[6] ^ xgmiirev32[5] ^ xgmiirev32[4] ^
xgmiirev32[2] ^ xgmiirev32[1] ^ c[1] ^ c[2] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[8] ^
c[11] ^ c[14] ^ c[20] ^ c[21] ^ c[22] ^ c[25] ^ c[29] ^ c[30]; newcrc8[7]  <=
xgmiirev32[29] ^ xgmiirev32[28] ^ xgmiirev32[25] ^ xgmiirev32[24] ^
xgmiirev32[23] ^ xgmiirev32[22] ^ xgmiirev32[21] ^ xgmiirev32[16] ^
xgmiirev32[15] ^ xgmiirev32[10] ^ xgmiirev32[8] ^ xgmiirev32[7] ^ xgmiirev32[5]
^ xgmiirev32[3] ^ xgmiirev32[2] ^ xgmiirev32[0] ^ c[0] ^ c[2] ^ c[3] ^ c[5] ^
c[7] ^ c[8] ^ c[10] ^ c[15] ^ c[16] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[25] ^
c[28] ^ c[29]; newcrc8[8]  <= xgmiirev32[31] ^ xgmiirev32[28] ^ xgmiirev32[23] ^
xgmiirev32[22] ^ xgmiirev32[17] ^ xgmiirev32[12] ^ xgmiirev32[11] ^
xgmiirev32[10] ^ xgmiirev32[8] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ xgmiirev32[1] ^
xgmiirev32[0] ^ c[0] ^ c[1] ^ c[3] ^ c[4] ^ c[8] ^ c[10] ^ c[11] ^ c[12] ^ c[17]
^ c[22] ^ c[23] ^ c[28] ^ c[31]; newcrc8[9]  <= xgmiirev32[29] ^ xgmiirev32[24]
^ xgmiirev32[23] ^ xgmiirev32[18] ^ xgmiirev32[13] ^ xgmiirev32[12] ^
xgmiirev32[11] ^ xgmiirev32[9] ^ xgmiirev32[5] ^ xgmiirev32[4] ^ xgmiirev32[2] ^
xgmiirev32[1] ^ c[1] ^ c[2] ^ c[4] ^ c[5] ^ c[9] ^ c[11] ^ c[12] ^ c[13] ^ c[18]
^ c[23] ^ c[24] ^ c[29]; newcrc8[10] <= xgmiirev32[31] ^ xgmiirev32[29] ^
xgmiirev32[28] ^ xgmiirev32[26] ^ xgmiirev32[19] ^ xgmiirev32[16] ^
xgmiirev32[14] ^ xgmiirev32[13] ^ xgmiirev32[9] ^ xgmiirev32[5] ^ xgmiirev32[3]
^ xgmiirev32[2] ^ xgmiirev32[0] ^ c[0] ^ c[2] ^ c[3] ^ c[5] ^ c[9] ^ c[13] ^
c[14] ^ c[16] ^ c[19] ^ c[26] ^ c[28] ^ c[29] ^ c[31]; newcrc8[11] <=
xgmiirev32[31] ^ xgmiirev32[28] ^ xgmiirev32[27] ^ xgmiirev32[26] ^
xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[20] ^ xgmiirev32[17] ^
xgmiirev32[16] ^ xgmiirev32[15] ^ xgmiirev32[14] ^ xgmiirev32[12] ^
xgmiirev32[9] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ xgmiirev32[1] ^ xgmiirev32[0] ^
c[0] ^ c[1] ^ c[3] ^ c[4] ^ c[9] ^ c[12] ^ c[14] ^ c[15] ^ c[16] ^ c[17] ^ c[20]
^ c[24] ^ c[25] ^ c[26] ^ c[27] ^ c[28] ^ c[31]; newcrc8[12] <= xgmiirev32[31] ^
xgmiirev32[30] ^ xgmiirev32[27] ^ xgmiirev32[24] ^ xgmiirev32[21] ^
xgmiirev32[18] ^ xgmiirev32[17] ^ xgmiirev32[15] ^ xgmiirev32[13] ^
xgmiirev32[12] ^ xgmiirev32[9] ^ xgmiirev32[6] ^ xgmiirev32[5] ^ xgmiirev32[4] ^
xgmiirev32[2] ^ xgmiirev32[1] ^ xgmiirev32[0] ^ c[0] ^ c[1] ^ c[2] ^ c[4] ^ c[5]
^ c[6] ^ c[9] ^ c[12] ^ c[13] ^ c[15] ^ c[17] ^ c[18] ^ c[21] ^ c[24] ^ c[27] ^
c[30] ^ c[31]; newcrc8[13] <= xgmiirev32[31] ^ xgmiirev32[28] ^ xgmiirev32[25] ^
xgmiirev32[22] ^ xgmiirev32[19] ^ xgmiirev32[18] ^ xgmiirev32[16] ^
xgmiirev32[14] ^ xgmiirev32[13] ^ xgmiirev32[10] ^ xgmiirev32[7] ^ xgmiirev32[6]
^ xgmiirev32[5] ^ xgmiirev32[3] ^ xgmiirev32[2] ^ xgmiirev32[1] ^ c[1] ^ c[2] ^
```

```
c[3] ^ c[5] ^ c[6] ^ c[7] ^ c[10] ^ c[13] ^ c[14] ^ c[16] ^ c[18] ^ c[19] ^
c[22] ^ c[25] ^ c[28] ^ c[31]; newcrc8[14] <= xgmiirev32[29] ^ xgmiirev32[26] ^
xgmiirev32[23] ^ xgmiirev32[20] ^ xgmiirev32[19] ^ xgmiirev32[17] ^
xgmiirev32[15] ^ xgmiirev32[14] ^ xgmiirev32[11] ^ xgmiirev32[8] ^ xgmiirev32[7]
^ xgmiirev32[6] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ xgmiirev32[2] ^ c[2] ^ c[3] ^
c[4] ^ c[6] ^ c[7] ^ c[8] ^ c[11] ^ c[14] ^ c[15] ^ c[17] ^ c[19] ^ c[20] ^
c[23] ^ c[26] ^ c[29]; newcrc8[15] <= xgmiirev32[30] ^ xgmiirev32[27] ^
xgmiirev32[24] ^ xgmiirev32[21] ^ xgmiirev32[20] ^ xgmiirev32[18] ^
xgmiirev32[16] ^ xgmiirev32[15] ^ xgmiirev32[12] ^ xgmiirev32[9] ^ xgmiirev32[8]
^ xgmiirev32[7] ^ xgmiirev32[5] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ c[3] ^ c[4] ^
c[5] ^ c[7] ^ c[8] ^ c[9] ^ c[12] ^ c[15] ^ c[16] ^ c[18] ^ c[20] ^ c[21] ^
c[24] ^ c[27] ^ c[30]; newcrc8[16] <= xgmiirev32[30] ^ xgmiirev32[29] ^
xgmiirev32[26] ^ xgmiirev32[24] ^ xgmiirev32[22] ^ xgmiirev32[21] ^
xgmiirev32[19] ^ xgmiirev32[17] ^ xgmiirev32[13] ^ xgmiirev32[12] ^
xgmiirev32[8] ^ xgmiirev32[5] ^ xgmiirev32[4] ^ xgmiirev32[0] ^ c[0] ^ c[4] ^
c[5] ^ c[8] ^ c[12] ^ c[13] ^ c[17] ^ c[19] ^ c[21] ^ c[22] ^ c[24] ^ c[26] ^
c[29] ^ c[30]; newcrc8[17] <= xgmiirev32[31] ^ xgmiirev32[30] ^ xgmiirev32[27] ^
xgmiirev32[25] ^ xgmiirev32[23] ^ xgmiirev32[22] ^ xgmiirev32[20] ^
xgmiirev32[18] ^ xgmiirev32[14] ^ xgmiirev32[13] ^ xgmiirev32[9] ^ xgmiirev32[6]
^ xgmiirev32[5] ^ xgmiirev32[1] ^ c[1] ^ c[5] ^ c[6] ^ c[9] ^ c[13] ^ c[14] ^
c[18] ^ c[20] ^ c[22] ^ c[23] ^ c[25] ^ c[27] ^ c[30] ^ c[31]; newcrc8[18] <=
xgmiirev32[31] ^ xgmiirev32[28] ^ xgmiirev32[26] ^ xgmiirev32[24] ^
xgmiirev32[23] ^ xgmiirev32[21] ^ xgmiirev32[19] ^ xgmiirev32[15] ^
xgmiirev32[14] ^ xgmiirev32[10] ^ xgmiirev32[7] ^ xgmiirev32[6] ^ xgmiirev32[2]
^ c[2] ^ c[6] ^ c[7] ^ c[10] ^ c[14] ^ c[15] ^ c[19] ^ c[21] ^ c[23] ^ c[24] ^
c[26] ^ c[28] ^ c[31]; newcrc8[19] <= xgmiirev32[29] ^ xgmiirev32[27] ^
xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[22] ^ xgmiirev32[20] ^
xgmiirev32[16] ^ xgmiirev32[15] ^ xgmiirev32[11] ^ xgmiirev32[8] ^ xgmiirev32[7]
^ xgmiirev32[3] ^ c[3] ^ c[7] ^ c[8] ^ c[11] ^ c[15] ^ c[16] ^ c[20] ^ c[22] ^
c[24] ^ c[25] ^ c[27] ^ c[29]; newcrc8[20] <= xgmiirev32[30] ^ xgmiirev32[28] ^
xgmiirev32[26] ^ xgmiirev32[25] ^ xgmiirev32[23] ^ xgmiirev32[21] ^
xgmiirev32[17] ^ xgmiirev32[16] ^ xgmiirev32[12] ^ xgmiirev32[9] ^ xgmiirev32[8]
^ xgmiirev32[4] ^ c[4] ^ c[8] ^ c[9] ^ c[12] ^ c[16] ^ c[17] ^ c[21] ^ c[23] ^
c[25] ^ c[26] ^ c[28] ^ c[30]; newcrc8[21] <= xgmiirev32[31] ^ xgmiirev32[29] ^
xgmiirev32[27] ^ xgmiirev32[26] ^ xgmiirev32[24] ^ xgmiirev32[22] ^
xgmiirev32[18] ^ xgmiirev32[17] ^ xgmiirev32[13] ^ xgmiirev32[10] ^
xgmiirev32[9] ^ xgmiirev32[5] ^ c[5] ^ c[9] ^ c[10] ^ c[13] ^ c[17] ^ c[18] ^
c[22] ^ c[24] ^ c[26] ^ c[27] ^ c[29] ^ c[31]; newcrc8[22] <= xgmiirev32[31] ^
xgmiirev32[29] ^ xgmiirev32[27] ^ xgmiirev32[26] ^ xgmiirev32[24] ^
xgmiirev32[23] ^ xgmiirev32[19] ^ xgmiirev32[18] ^ xgmiirev32[16] ^
xgmiirev32[14] ^ xgmiirev32[12] ^ xgmiirev32[11] ^ xgmiirev32[9] ^ xgmiirev32[0]
^ c[0] ^ c[9] ^ c[11] ^ c[12] ^ c[14] ^ c[16] ^ c[18] ^ c[19] ^ c[23] ^ c[24] ^
c[26] ^ c[27] ^ c[29] ^ c[31]; newcrc8[23] <= xgmiirev32[31] ^ xgmiirev32[29] ^
xgmiirev32[27] ^ xgmiirev32[26] ^ xgmiirev32[20] ^ xgmiirev32[19] ^
xgmiirev32[17] ^ xgmiirev32[16] ^ xgmiirev32[15] ^ xgmiirev32[13] ^
xgmiirev32[9] ^ xgmiirev32[6] ^ xgmiirev32[1] ^ xgmiirev32[0] ^ c[0] ^ c[1] ^
c[6] ^ c[9] ^ c[13] ^ c[15] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[26] ^ c[27] ^
c[29] ^ c[31]; newcrc8[24] <= xgmiirev32[30] ^ xgmiirev32[28] ^ xgmiirev32[27] ^
```

```
xgmiirev32[21] ^ xgmiirev32[20] ^ xgmiirev32[18] ^ xgmiirev32[17] ^
xgmiirev32[16] ^ xgmiirev32[14] ^ xgmiirev32[10] ^ xgmiirev32[7] ^ xgmiirev32[2]
^ xgmiirev32[1] ^ c[1] ^ c[2] ^ c[7] ^ c[10] ^ c[14] ^ c[16] ^ c[17] ^ c[18] ^
c[20] ^ c[21] ^ c[27] ^ c[28] ^ c[30]; newcrc8[25] <= xgmiirev32[31] ^
xgmiirev32[29] ^ xgmiirev32[28] ^ xgmiirev32[22] ^ xgmiirev32[21] ^
xgmiirev32[19] ^ xgmiirev32[18] ^ xgmiirev32[17] ^ xgmiirev32[15] ^
xgmiirev32[11] ^ xgmiirev32[8] ^ xgmiirev32[3] ^ xgmiirev32[2] ^ c[2] ^ c[3] ^
c[8] ^ c[11] ^ c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[28] ^ c[29] ^
c[31]; newcrc8[26] <= xgmiirev32[31] ^ xgmiirev32[28] ^ xgmiirev32[26] ^
xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[23] ^ xgmiirev32[22] ^
xgmiirev32[20] ^ xgmiirev32[19] ^ xgmiirev32[18] ^ xgmiirev32[10] ^
xgmiirev32[6] ^ xgmiirev32[4] ^ xgmiirev32[3] ^ xgmiirev32[0] ^ c[0] ^ c[3] ^
c[4] ^ c[6] ^ c[10] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[25] ^
c[26] ^ c[28] ^ c[31]; newcrc8[27] <= xgmiirev32[29] ^ xgmiirev32[27] ^
xgmiirev32[26] ^ xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[23] ^
xgmiirev32[21] ^ xgmiirev32[20] ^ xgmiirev32[19] ^ xgmiirev32[11] ^
xgmiirev32[7] ^ xgmiirev32[5] ^ xgmiirev32[4] ^ xgmiirev32[1] ^ c[1] ^ c[4] ^
c[5] ^ c[7] ^ c[11] ^ c[19] ^ c[20] ^ c[21] ^ c[23] ^ c[24] ^ c[25] ^ c[26] ^
c[27] ^ c[29]; newcrc8[28] <= xgmiirev32[30] ^ xgmiirev32[28] ^ xgmiirev32[27] ^
xgmiirev32[26] ^ xgmiirev32[25] ^ xgmiirev32[24] ^ xgmiirev32[22] ^
xgmiirev32[21] ^ xgmiirev32[20] ^ xgmiirev32[12] ^ xgmiirev32[8] ^ xgmiirev32[6]
^ xgmiirev32[5] ^ xgmiirev32[2] ^ c[2] ^ c[5] ^ c[6] ^ c[8] ^ c[12] ^ c[20] ^
c[21] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[27] ^ c[28] ^ c[30]; newcrc8[29] <=
xgmiirev32[31] ^ xgmiirev32[29] ^ xgmiirev32[28] ^ xgmiirev32[27] ^
xgmiirev32[26] ^ xgmiirev32[25] ^ xgmiirev32[23] ^ xgmiirev32[22] ^
xgmiirev32[21] ^ xgmiirev32[13] ^ xgmiirev32[9] ^ xgmiirev32[7] ^ xgmiirev32[6]
^ xgmiirev32[3] ^ c[3] ^ c[6] ^ c[7] ^ c[9] ^ c[13] ^ c[21] ^ c[22] ^ c[23] ^
c[25] ^ c[26] ^ c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[30] <= xgmiirev32[30] ^
xgmiirev32[29] ^ xgmiirev32[28] ^ xgmiirev32[27] ^ xgmiirev32[26] ^
xgmiirev32[24] ^ xgmiirev32[23] ^ xgmiirev32[22] ^ xgmiirev32[14] ^
xgmiirev32[10] ^ xgmiirev32[8] ^ xgmiirev32[7] ^ xgmiirev32[4] ^ c[4] ^ c[7] ^
c[8] ^ c[10] ^ c[14] ^ c[22] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^ c[28] ^ c[29] ^
c[30]; newcrc8[31] <= xgmiirev32[31] ^ xgmiirev32[30] ^ xgmiirev32[29] ^
xgmiirev32[28] ^ xgmiirev32[27] ^ xgmiirev32[25] ^ xgmiirev32[24] ^
xgmiirev32[23] ^ xgmiirev32[15] ^ xgmiirev32[11] ^ xgmiirev32[9] ^ xgmiirev32[8]
^ xgmiirev32[5] ^ c[5] ^ c[8] ^ c[9] ^ c[11] ^ c[15] ^ c[23] ^ c[24] ^ c[25] ^
c[27] ^ c[28] ^ c[29] ^ c[30] ^ c[31]; checksum_flag<=1; crc_flag<=0; end //
//input data is eop else if (xgmiidata[61:54]==8'hFD) begin  xgmii[31:24] <=
xgmiidata[70:63]; xgmii[39:32] <= xgmiidata[7:0]; xgmii[47:40] <=
xgmiidata[16:9]; xgmii[55:48] <= xgmiidata[25:18]; xgmii[63:56] <=
xgmiidata[34:27]; xgmii[24:0]<=0; empty_flag<=7; eop_flag_buff <= 1; newcrc8[0]
<= xgmiirev40[37] ^ xgmiirev40[34] ^ xgmiirev40[32] ^ xgmiirev40[31] ^
xgmiirev40[30] ^ xgmiirev40[29] ^ xgmiirev40[28] ^ xgmiirev40[26] ^
xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[16] ^ xgmiirev40[12] ^
xgmiirev40[10] ^ xgmiirev40[9] ^ xgmiirev40[6] ^ xgmiirev40[0] ^ c[1] ^ c[2] ^
c[4] ^ c[8] ^ c[16] ^ c[17] ^ c[18] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^
c[26] ^ c[29]; newcrc8[1]  <= xgmiirev40[38] ^ xgmiirev40[37] ^ xgmiirev40[35] ^
xgmiirev40[34] ^ xgmiirev40[33] ^ xgmiirev40[28] ^ xgmiirev40[27] ^
```

```
xgmiirev40[24] ^ xgmiirev40[17] ^ xgmiirev40[16] ^ xgmiirev40[13] ^
xgmiirev40[12] ^ xgmiirev40[11] ^ xgmiirev40[9] ^ xgmiirev40[7] ^ xgmiirev40[6]
^ xgmiirev40[1] ^ xgmiirev40[0] ^ c[1] ^ c[3] ^ c[4] ^ c[5] ^ c[8] ^ c[9] ^
c[16] ^ c[19] ^ c[20] ^ c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[30]; newcrc8[2]  <=
xgmiirev40[39] ^ xgmiirev40[38] ^ xgmiirev40[37] ^ xgmiirev40[36] ^
xgmiirev40[35] ^ xgmiirev40[32] ^ xgmiirev40[31] ^ xgmiirev40[30] ^
xgmiirev40[26] ^ xgmiirev40[24] ^ xgmiirev40[18] ^ xgmiirev40[17] ^
xgmiirev40[16] ^ xgmiirev40[14] ^ xgmiirev40[13] ^ xgmiirev40[9] ^ xgmiirev40[8]
^ xgmiirev40[7] ^ xgmiirev40[6] ^ xgmiirev40[2] ^ xgmiirev40[1] ^ xgmiirev40[0]
^ c[0] ^ c[1] ^ c[5] ^ c[6] ^ c[8] ^ c[9] ^ c[10] ^ c[16] ^ c[18] ^ c[22] ^
c[23] ^ c[24] ^ c[27] ^ c[28] ^ c[29] ^ c[30] ^ c[31]; newcrc8[3]  <=
xgmiirev40[39] ^ xgmiirev40[38] ^ xgmiirev40[37] ^ xgmiirev40[36] ^
xgmiirev40[33] ^ xgmiirev40[32] ^ xgmiirev40[31] ^ xgmiirev40[27] ^
xgmiirev40[25] ^ xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[17] ^
xgmiirev40[15] ^ xgmiirev40[14] ^ xgmiirev40[10] ^ xgmiirev40[9] ^ xgmiirev40[8]
^ xgmiirev40[7] ^ xgmiirev40[3] ^ xgmiirev40[2] ^ xgmiirev40[1] ^ c[0] ^ c[1] ^
c[2] ^ c[6] ^ c[7] ^ c[9] ^ c[10] ^ c[11] ^ c[17] ^ c[19] ^ c[23] ^ c[24] ^
c[25] ^ c[28] ^ c[29] ^ c[30] ^ c[31]; newcrc8[4]  <= xgmiirev40[39] ^
xgmiirev40[38] ^ xgmiirev40[33] ^ xgmiirev40[31] ^ xgmiirev40[30] ^
xgmiirev40[29] ^ xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[20] ^
xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[15] ^ xgmiirev40[12] ^
xgmiirev40[11] ^ xgmiirev40[8] ^ xgmiirev40[6] ^ xgmiirev40[4] ^ xgmiirev40[3] ^
xgmiirev40[2] ^ xgmiirev40[0] ^ c[0] ^ c[3] ^ c[4] ^ c[7] ^ c[10] ^ c[11] ^
c[12] ^ c[16] ^ c[17] ^ c[21] ^ c[22] ^ c[23] ^ c[25] ^ c[30] ^ c[31];
newcrc8[5]  <= xgmiirev40[39] ^ xgmiirev40[37] ^ xgmiirev40[29] ^ xgmiirev40[28]
^ xgmiirev40[24] ^ xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[19] ^
xgmiirev40[13] ^ xgmiirev40[10] ^ xgmiirev40[7] ^ xgmiirev40[6] ^ xgmiirev40[5]
^ xgmiirev40[4] ^ xgmiirev40[3] ^ xgmiirev40[1] ^ xgmiirev40[0] ^ c[2] ^ c[5] ^
c[11] ^ c[12] ^ c[13] ^ c[16] ^ c[20] ^ c[21] ^ c[29] ^ c[31]; newcrc8[6]  <=
xgmiirev40[38] ^ xgmiirev40[30] ^ xgmiirev40[29] ^ xgmiirev40[25] ^
xgmiirev40[22] ^ xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[14] ^
xgmiirev40[11] ^ xgmiirev40[8] ^ xgmiirev40[7] ^ xgmiirev40[6] ^ xgmiirev40[5] ^
xgmiirev40[4] ^ xgmiirev40[2] ^ xgmiirev40[1] ^ c[0] ^ c[3] ^ c[6] ^ c[12] ^
c[13] ^ c[14] ^ c[17] ^ c[21] ^ c[22] ^ c[30]; newcrc8[7]  <= xgmiirev40[39] ^
xgmiirev40[37] ^ xgmiirev40[34] ^ xgmiirev40[32] ^ xgmiirev40[29] ^
xgmiirev40[28] ^ xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[23] ^
xgmiirev40[22] ^ xgmiirev40[21] ^ xgmiirev40[16] ^ xgmiirev40[15] ^
xgmiirev40[10] ^ xgmiirev40[8] ^ xgmiirev40[7] ^ xgmiirev40[5] ^ xgmiirev40[3] ^
xgmiirev40[2] ^ xgmiirev40[0] ^ c[0] ^ c[2] ^ c[7] ^ c[8] ^ c[13] ^ c[14] ^
c[15] ^ c[16] ^ c[17] ^ c[20] ^ c[21] ^ c[24] ^ c[26] ^ c[29] ^ c[31];
newcrc8[8]  <= xgmiirev40[38] ^ xgmiirev40[37] ^ xgmiirev40[35] ^ xgmiirev40[34]
^ xgmiirev40[33] ^ xgmiirev40[32] ^ xgmiirev40[31] ^ xgmiirev40[28] ^
xgmiirev40[23] ^ xgmiirev40[22] ^ xgmiirev40[17] ^ xgmiirev40[12] ^
xgmiirev40[11] ^ xgmiirev40[10] ^ xgmiirev40[8] ^ xgmiirev40[4] ^ xgmiirev40[3]
^ xgmiirev40[1] ^ xgmiirev40[0] ^ c[0] ^ c[2] ^ c[3] ^ c[4] ^ c[9] ^ c[14] ^
c[15] ^ c[20] ^ c[23] ^ c[24] ^ c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[30];
newcrc8[9]  <= xgmiirev40[39] ^ xgmiirev40[38] ^ xgmiirev40[36] ^ xgmiirev40[35]
^ xgmiirev40[34] ^ xgmiirev40[33] ^ xgmiirev40[32] ^ xgmiirev40[29] ^
```

```
xgmiirev40[24] ^ xgmiirev40[23] ^ xgmiirev40[18] ^ xgmiirev40[13] ^
xgmiirev40[12] ^ xgmiirev40[11] ^ xgmiirev40[9] ^ xgmiirev40[5] ^ xgmiirev40[4]
^ xgmiirev40[2] ^ xgmiirev40[1] ^ c[1] ^ c[3] ^ c[4] ^ c[5] ^ c[10] ^ c[15] ^
c[16] ^ c[21] ^ c[24] ^ c[25] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^ c[31];
newcrc8[10] <= xgmiirev40[39] ^ xgmiirev40[36] ^ xgmiirev40[35] ^ xgmiirev40[33]
^ xgmiirev40[32] ^ xgmiirev40[31] ^ xgmiirev40[29] ^ xgmiirev40[28] ^
xgmiirev40[26] ^ xgmiirev40[19] ^ xgmiirev40[16] ^ xgmiirev40[14] ^
xgmiirev40[13] ^ xgmiirev40[9] ^ xgmiirev40[5] ^ xgmiirev40[3] ^ xgmiirev40[2] ^
xgmiirev40[0] ^ c[1] ^ c[5] ^ c[6] ^ c[8] ^ c[11] ^ c[18] ^ c[20] ^ c[21] ^
c[23] ^ c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[31]; newcrc8[11] <= xgmiirev40[36] ^
xgmiirev40[33] ^ xgmiirev40[31] ^ xgmiirev40[28] ^ xgmiirev40[27] ^
xgmiirev40[26] ^ xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[20] ^
xgmiirev40[17] ^ xgmiirev40[16] ^ xgmiirev40[15] ^ xgmiirev40[14] ^
xgmiirev40[12] ^ xgmiirev40[9] ^ xgmiirev40[4] ^ xgmiirev40[3] ^ xgmiirev40[1] ^
xgmiirev40[0] ^ c[1] ^ c[4] ^ c[6] ^ c[7] ^ c[8] ^ c[9] ^ c[12] ^ c[16] ^ c[17]
^ c[18] ^ c[19] ^ c[20] ^ c[23] ^ c[25] ^ c[28]; newcrc8[12] <= xgmiirev40[31] ^
xgmiirev40[30] ^ xgmiirev40[27] ^ xgmiirev40[24] ^ xgmiirev40[21] ^
xgmiirev40[18] ^ xgmiirev40[17] ^ xgmiirev40[15] ^ xgmiirev40[13] ^
xgmiirev40[12] ^ xgmiirev40[9] ^ xgmiirev40[6] ^ xgmiirev40[5] ^ xgmiirev40[4] ^
xgmiirev40[2] ^ xgmiirev40[1] ^ xgmiirev40[0] ^ c[1] ^ c[4] ^ c[5] ^ c[7] ^ c[9]
^ c[10] ^ c[13] ^ c[16] ^ c[19] ^ c[22] ^ c[23]; newcrc8[13] <= xgmiirev40[32] ^
xgmiirev40[31] ^ xgmiirev40[28] ^ xgmiirev40[25] ^ xgmiirev40[22] ^
xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[16] ^ xgmiirev40[14] ^
xgmiirev40[13] ^ xgmiirev40[10] ^ xgmiirev40[7] ^ xgmiirev40[6] ^ xgmiirev40[5]
^ xgmiirev40[3] ^ xgmiirev40[2] ^ xgmiirev40[1] ^ c[2] ^ c[5] ^ c[6] ^ c[8] ^
c[10] ^ c[11] ^ c[14] ^ c[17] ^ c[20] ^ c[23] ^ c[24]; newcrc8[14] <=
xgmiirev40[33] ^ xgmiirev40[32] ^ xgmiirev40[29] ^ xgmiirev40[26] ^
xgmiirev40[23] ^ xgmiirev40[20] ^ xgmiirev40[19] ^ xgmiirev40[17] ^
xgmiirev40[15] ^ xgmiirev40[14] ^ xgmiirev40[11] ^ xgmiirev40[8] ^ xgmiirev40[7]
^ xgmiirev40[6] ^ xgmiirev40[4] ^ xgmiirev40[3] ^ xgmiirev40[2] ^ c[0] ^ c[3] ^
c[6] ^ c[7] ^ c[9] ^ c[11] ^ c[12] ^ c[15] ^ c[18] ^ c[21] ^ c[24] ^ c[25];
newcrc8[15] <= xgmiirev40[34] ^ xgmiirev40[33] ^ xgmiirev40[30] ^ xgmiirev40[27]
^ xgmiirev40[24] ^ xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[18] ^
xgmiirev40[16] ^ xgmiirev40[15] ^ xgmiirev40[12] ^ xgmiirev40[9] ^ xgmiirev40[8]
^ xgmiirev40[7] ^ xgmiirev40[5] ^ xgmiirev40[4] ^ xgmiirev40[3] ^ c[0] ^ c[1] ^
c[4] ^ c[7] ^ c[8] ^ c[10] ^ c[12] ^ c[13] ^ c[16] ^ c[19] ^ c[22] ^ c[25] ^
c[26]; newcrc8[16] <= xgmiirev40[37] ^ xgmiirev40[35] ^ xgmiirev40[32] ^
xgmiirev40[30] ^ xgmiirev40[29] ^ xgmiirev40[26] ^ xgmiirev40[24] ^
xgmiirev40[22] ^ xgmiirev40[21] ^ xgmiirev40[19] ^ xgmiirev40[17] ^
xgmiirev40[13] ^ xgmiirev40[12] ^ xgmiirev40[8] ^ xgmiirev40[5] ^ xgmiirev40[4]
^ xgmiirev40[0] ^ c[0] ^ c[4] ^ c[5] ^ c[9] ^ c[11] ^ c[13] ^ c[14] ^ c[16] ^
c[18] ^ c[21] ^ c[22] ^ c[24] ^ c[27] ^ c[29]; newcrc8[17] <= xgmiirev40[38] ^
xgmiirev40[36] ^ xgmiirev40[33] ^ xgmiirev40[31] ^ xgmiirev40[30] ^
xgmiirev40[27] ^ xgmiirev40[25] ^ xgmiirev40[23] ^ xgmiirev40[22] ^
xgmiirev40[20] ^ xgmiirev40[18] ^ xgmiirev40[14] ^ xgmiirev40[13] ^
xgmiirev40[9] ^ xgmiirev40[6] ^ xgmiirev40[5] ^ xgmiirev40[1] ^ c[1] ^ c[5] ^
c[6] ^ c[10] ^ c[12] ^ c[14] ^ c[15] ^ c[17] ^ c[19] ^ c[22] ^ c[23] ^ c[25] ^
c[28] ^ c[30]; newcrc8[18] <= xgmiirev40[39] ^ xgmiirev40[37] ^ xgmiirev40[34] ^
```

```
xgmiirev40[32] ^ xgmiirev40[31] ^ xgmiirev40[28] ^ xgmiirev40[26] ^
xgmiirev40[24] ^ xgmiirev40[23] ^ xgmiirev40[21] ^ xgmiirev40[19] ^
xgmiirev40[15] ^ xgmiirev40[14] ^ xgmiirev40[10] ^ xgmiirev40[7] ^ xgmiirev40[6]
^ xgmiirev40[2] ^ c[2] ^ c[6] ^ c[7] ^ c[11] ^ c[13] ^ c[15] ^ c[16] ^ c[18] ^
c[20] ^ c[23] ^ c[24] ^ c[26] ^ c[29] ^ c[31]; newcrc8[19] <= xgmiirev40[38] ^
xgmiirev40[35] ^ xgmiirev40[33] ^ xgmiirev40[32] ^ xgmiirev40[29] ^
xgmiirev40[27] ^ xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[22] ^
xgmiirev40[20] ^ xgmiirev40[16] ^ xgmiirev40[15] ^ xgmiirev40[11] ^
xgmiirev40[8] ^ xgmiirev40[7] ^ xgmiirev40[3] ^ c[0] ^ c[3] ^ c[7] ^ c[8] ^
c[12] ^ c[14] ^ c[16] ^ c[17] ^ c[19] ^ c[21] ^ c[24] ^ c[25] ^ c[27] ^ c[30];
newcrc8[20] <= xgmiirev40[39] ^ xgmiirev40[36] ^ xgmiirev40[34] ^ xgmiirev40[33]
^ xgmiirev40[30] ^ xgmiirev40[28] ^ xgmiirev40[26] ^ xgmiirev40[25] ^
xgmiirev40[23] ^ xgmiirev40[21] ^ xgmiirev40[17] ^ xgmiirev40[16] ^
xgmiirev40[12] ^ xgmiirev40[9] ^ xgmiirev40[8] ^ xgmiirev40[4] ^ c[0] ^ c[1] ^
c[4] ^ c[8] ^ c[9] ^ c[13] ^ c[15] ^ c[17] ^ c[18] ^ c[20] ^ c[22] ^ c[25] ^
c[26] ^ c[28] ^ c[31]; newcrc8[21] <= xgmiirev40[37] ^ xgmiirev40[35] ^
xgmiirev40[34] ^ xgmiirev40[31] ^ xgmiirev40[29] ^ xgmiirev40[27] ^
xgmiirev40[26] ^ xgmiirev40[24] ^ xgmiirev40[22] ^ xgmiirev40[18] ^
xgmiirev40[17] ^ xgmiirev40[13] ^ xgmiirev40[10] ^ xgmiirev40[9] ^ xgmiirev40[5]
^ c[1] ^ c[2] ^ c[5] ^ c[9] ^ c[10] ^ c[14] ^ c[16] ^ c[18] ^ c[19] ^ c[21] ^
c[23] ^ c[26] ^ c[27] ^ c[29]; newcrc8[22] <= xgmiirev40[38] ^ xgmiirev40[37] ^
xgmiirev40[36] ^ xgmiirev40[35] ^ xgmiirev40[34] ^ xgmiirev40[31] ^
xgmiirev40[29] ^ xgmiirev40[27] ^ xgmiirev40[26] ^ xgmiirev40[24] ^
xgmiirev40[23] ^ xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[16] ^
xgmiirev40[14] ^ xgmiirev40[12] ^ xgmiirev40[11] ^ xgmiirev40[9] ^ xgmiirev40[0]
^ c[1] ^ c[3] ^ c[4] ^ c[6] ^ c[8] ^ c[10] ^ c[11] ^ c[15] ^ c[16] ^ c[18] ^
c[19] ^ c[21] ^ c[23] ^ c[26] ^ c[27] ^ c[28] ^ c[29] ^ c[30]; newcrc8[23] <=
xgmiirev40[39] ^ xgmiirev40[38] ^ xgmiirev40[36] ^ xgmiirev40[35] ^
xgmiirev40[34] ^ xgmiirev40[31] ^ xgmiirev40[29] ^ xgmiirev40[27] ^
xgmiirev40[26] ^ xgmiirev40[20] ^ xgmiirev40[19] ^ xgmiirev40[17] ^
xgmiirev40[16] ^ xgmiirev40[15] ^ xgmiirev40[13] ^ xgmiirev40[9] ^ xgmiirev40[6]
^ xgmiirev40[1] ^ xgmiirev40[0] ^ c[1] ^ c[5] ^ c[7] ^ c[8] ^ c[9] ^ c[11] ^
c[12] ^ c[18] ^ c[19] ^ c[21] ^ c[23] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^ c[31];
newcrc8[24] <= xgmiirev40[39] ^ xgmiirev40[37] ^ xgmiirev40[36] ^ xgmiirev40[35]
^ xgmiirev40[32] ^ xgmiirev40[30] ^ xgmiirev40[28] ^ xgmiirev40[27] ^
xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[18] ^ xgmiirev40[17] ^
xgmiirev40[16] ^ xgmiirev40[14] ^ xgmiirev40[10] ^ xgmiirev40[7] ^ xgmiirev40[2]
^ xgmiirev40[1] ^ c[2] ^ c[6] ^ c[8] ^ c[9] ^ c[10] ^ c[12] ^ c[13] ^ c[19] ^
c[20] ^ c[22] ^ c[24] ^ c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[25] <=
xgmiirev40[38] ^ xgmiirev40[37] ^ xgmiirev40[36] ^ xgmiirev40[33] ^
xgmiirev40[31] ^ xgmiirev40[29] ^ xgmiirev40[28] ^ xgmiirev40[22] ^
xgmiirev40[21] ^ xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[17] ^
xgmiirev40[15] ^ xgmiirev40[11] ^ xgmiirev40[8] ^ xgmiirev40[3] ^ xgmiirev40[2]
^ c[0] ^ c[3] ^ c[7] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[20] ^ c[21] ^
c[23] ^ c[25] ^ c[28] ^ c[29] ^ c[30]; newcrc8[26] <= xgmiirev40[39] ^
xgmiirev40[38] ^ xgmiirev40[31] ^ xgmiirev40[28] ^ xgmiirev40[26] ^
xgmiirev40[25] ^ xgmiirev40[24] ^ xgmiirev40[23] ^ xgmiirev40[22] ^
xgmiirev40[20] ^ xgmiirev40[19] ^ xgmiirev40[18] ^ xgmiirev40[10] ^
```

```
xgmiirev40[6] ^ xgmiirev40[4] ^ xgmiirev40[3] ^ xgmiirev40[0] ^ c[2] ^ c[10] ^
c[11] ^ c[12] ^ c[14] ^ c[15] ^ c[16] ^ c[17] ^ c[18] ^ c[20] ^ c[23] ^ c[30] ^
c[31]; newcrc8[27] <= xgmiirev40[39] ^ xgmiirev40[32] ^ xgmiirev40[29] ^
xgmiirev40[27] ^ xgmiirev40[26] ^ xgmiirev40[25] ^ xgmiirev40[24] ^
xgmiirev40[23] ^ xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[19] ^
xgmiirev40[11] ^ xgmiirev40[7] ^ xgmiirev40[5] ^ xgmiirev40[4] ^ xgmiirev40[1] ^
c[3] ^ c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^
c[24] ^ c[31]; newcrc8[28] <= xgmiirev40[33] ^ xgmiirev40[30] ^ xgmiirev40[28] ^
xgmiirev40[27] ^ xgmiirev40[26] ^ xgmiirev40[25] ^ xgmiirev40[24] ^
xgmiirev40[22] ^ xgmiirev40[21] ^ xgmiirev40[20] ^ xgmiirev40[12] ^
xgmiirev40[8] ^ xgmiirev40[6] ^ xgmiirev40[5] ^ xgmiirev40[2] ^ c[0] ^ c[4] ^
c[12] ^ c[13] ^ c[14] ^ c[16] ^ c[17] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^ c[25];
newcrc8[29] <= xgmiirev40[34] ^ xgmiirev40[31] ^ xgmiirev40[29] ^ xgmiirev40[28]
^ xgmiirev40[27] ^ xgmiirev40[26] ^ xgmiirev40[25] ^ xgmiirev40[23] ^
xgmiirev40[22] ^ xgmiirev40[21] ^ xgmiirev40[13] ^ xgmiirev40[9] ^ xgmiirev40[7]
^ xgmiirev40[6] ^ xgmiirev40[3] ^ c[1] ^ c[5] ^ c[13] ^ c[14] ^ c[15] ^ c[17] ^
c[18] ^ c[19] ^ c[20] ^ c[21] ^ c[23] ^ c[26]; newcrc8[30] <= xgmiirev40[35] ^
xgmiirev40[32] ^ xgmiirev40[30] ^ xgmiirev40[29] ^ xgmiirev40[28] ^
xgmiirev40[27] ^ xgmiirev40[26] ^ xgmiirev40[24] ^ xgmiirev40[23] ^
xgmiirev40[22] ^ xgmiirev40[14] ^ xgmiirev40[10] ^ xgmiirev40[8] ^ xgmiirev40[7]
^ xgmiirev40[4] ^ c[0] ^ c[2] ^ c[6] ^ c[14] ^ c[15] ^ c[16] ^ c[18] ^ c[19] ^
c[20] ^ c[21] ^ c[22] ^ c[24] ^ c[27]; newcrc8[31] <= xgmiirev40[36] ^
xgmiirev40[33] ^ xgmiirev40[31] ^ xgmiirev40[30] ^ xgmiirev40[29] ^
xgmiirev40[28] ^ xgmiirev40[27] ^ xgmiirev40[25] ^ xgmiirev40[24] ^
xgmiirev40[23] ^ xgmiirev40[15] ^ xgmiirev40[11] ^ xgmiirev40[9] ^ xgmiirev40[8]
^ xgmiirev40[5] ^ c[0] ^ c[1] ^ c[3] ^ c[7] ^ c[15] ^ c[16] ^ c[17] ^ c[19] ^
c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[25] ^ c[28];

checksum_buff<=1; end else if (xgmiidata[52:45]==8'hFD) begin  xgmii[23:16] <=
xgmiidata[61:54]; xgmii[31:24] <= xgmiidata[70:63]; xgmii[39:32] <=
xgmiidata[7:0]; xgmii[47:40] <= xgmiidata[16:9]; xgmii[55:48] <=
xgmiidata[25:18]; xgmii[63:56] <= xgmiidata[34:27]; xgmii[16:0]<=0;
empty_flag<=6; eop_flag_buff <= 1; newcrc8[0]  <= xgmiirev48[47] ^
xgmiirev48[45] ^ xgmiirev48[44] ^ xgmiirev48[37] ^ xgmiirev48[34] ^
xgmiirev48[32] ^ xgmiirev48[31] ^ xgmiirev48[30] ^ xgmiirev48[29] ^
xgmiirev48[28] ^ xgmiirev48[26] ^ xgmiirev48[25] ^ xgmiirev48[24] ^
xgmiirev48[16] ^ xgmiirev48[12] ^ xgmiirev48[10] ^ xgmiirev48[9] ^ xgmiirev48[6]
^ xgmiirev48[0] ^ c[0] ^ c[8] ^ c[9] ^ c[10] ^ c[12] ^ c[13] ^ c[14] ^ c[15] ^
c[16] ^ c[18] ^ c[21] ^ c[28] ^ c[29] ^ c[31]; newcrc8[1]  <= xgmiirev48[47] ^
xgmiirev48[46] ^ xgmiirev48[44] ^ xgmiirev48[38] ^ xgmiirev48[37] ^
xgmiirev48[35] ^ xgmiirev48[34] ^ xgmiirev48[33] ^ xgmiirev48[28] ^
xgmiirev48[27] ^ xgmiirev48[24] ^ xgmiirev48[17] ^ xgmiirev48[16] ^
xgmiirev48[13] ^ xgmiirev48[12] ^ xgmiirev48[11] ^ xgmiirev48[9] ^ xgmiirev48[7]
^ xgmiirev48[6] ^ xgmiirev48[1] ^ xgmiirev48[0] ^ c[0] ^ c[1] ^ c[8] ^ c[11] ^
c[12] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[28] ^ c[30] ^ c[31];
newcrc8[2]  <= xgmiirev48[44] ^ xgmiirev48[39] ^ xgmiirev48[38] ^ xgmiirev48[37]
^ xgmiirev48[36] ^ xgmiirev48[35] ^ xgmiirev48[32] ^ xgmiirev48[31] ^
xgmiirev48[30] ^ xgmiirev48[26] ^ xgmiirev48[24] ^ xgmiirev48[18] ^
```

```
xgmiirev48[17] ^ xgmiirev48[16] ^ xgmiirev48[14] ^ xgmiirev48[13] ^
xgmiirev48[9] ^ xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[6] ^ xgmiirev48[2] ^
xgmiirev48[1] ^ xgmiirev48[0] ^ c[0] ^ c[1] ^ c[2] ^ c[8] ^ c[10] ^ c[14] ^
c[15] ^ c[16] ^ c[19] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[28]; newcrc8[3]  <=
xgmiirev48[45] ^ xgmiirev48[40] ^ xgmiirev48[39] ^ xgmiirev48[38] ^
xgmiirev48[37] ^ xgmiirev48[36] ^ xgmiirev48[33] ^ xgmiirev48[32] ^
xgmiirev48[31] ^ xgmiirev48[27] ^ xgmiirev48[25] ^ xgmiirev48[19] ^
xgmiirev48[18] ^ xgmiirev48[17] ^ xgmiirev48[15] ^ xgmiirev48[14] ^
xgmiirev48[10] ^ xgmiirev48[9] ^ xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[3] ^
xgmiirev48[2] ^ xgmiirev48[1] ^ c[1] ^ c[2] ^ c[3] ^ c[9] ^ c[11] ^ c[15] ^
c[16] ^ c[17] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[29]; newcrc8[4]  <=
xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[45] ^ xgmiirev48[44] ^
xgmiirev48[41] ^ xgmiirev48[40] ^ xgmiirev48[39] ^ xgmiirev48[38] ^
xgmiirev48[33] ^ xgmiirev48[31] ^ xgmiirev48[30] ^ xgmiirev48[29] ^
xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[20] ^ xgmiirev48[19] ^
xgmiirev48[18] ^ xgmiirev48[15] ^ xgmiirev48[12] ^ xgmiirev48[11] ^
xgmiirev48[8] ^ xgmiirev48[6] ^ xgmiirev48[4] ^ xgmiirev48[3] ^ xgmiirev48[2] ^
xgmiirev48[0] ^ c[2] ^ c[3] ^ c[4] ^ c[8] ^ c[9] ^ c[13] ^ c[14] ^ c[15] ^ c[17]
^ c[22] ^ c[23] ^ c[24] ^ c[25] ^ c[28] ^ c[29] ^ c[30] ^ c[31]; newcrc8[5]  <=
xgmiirev48[46] ^ xgmiirev48[44] ^ xgmiirev48[42] ^ xgmiirev48[41] ^
xgmiirev48[40] ^ xgmiirev48[39] ^ xgmiirev48[37] ^ xgmiirev48[29] ^
xgmiirev48[28] ^ xgmiirev48[24] ^ xgmiirev48[21] ^ xgmiirev48[20] ^
xgmiirev48[19] ^ xgmiirev48[13] ^ xgmiirev48[10] ^ xgmiirev48[7] ^ xgmiirev48[6]
^ xgmiirev48[5] ^ xgmiirev48[4] ^ xgmiirev48[3] ^ xgmiirev48[1] ^ xgmiirev48[0]
^ c[3] ^ c[4] ^ c[5] ^ c[8] ^ c[12] ^ c[13] ^ c[21] ^ c[23] ^ c[24] ^ c[25] ^
c[26] ^ c[28] ^ c[30]; newcrc8[6]  <= xgmiirev48[47] ^ xgmiirev48[45] ^
xgmiirev48[43] ^ xgmiirev48[42] ^ xgmiirev48[41] ^ xgmiirev48[40] ^
xgmiirev48[38] ^ xgmiirev48[30] ^ xgmiirev48[29] ^ xgmiirev48[25] ^
xgmiirev48[22] ^ xgmiirev48[21] ^ xgmiirev48[20] ^ xgmiirev48[14] ^
xgmiirev48[11] ^ xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[6] ^ xgmiirev48[5] ^
xgmiirev48[4] ^ xgmiirev48[2] ^ xgmiirev48[1] ^ c[4] ^ c[5] ^ c[6] ^ c[9] ^
c[13] ^ c[14] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[31];
newcrc8[7]  <= xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[45] ^ xgmiirev48[43]
^ xgmiirev48[42] ^ xgmiirev48[41] ^ xgmiirev48[39] ^ xgmiirev48[37] ^
xgmiirev48[34] ^ xgmiirev48[32] ^ xgmiirev48[29] ^ xgmiirev48[28] ^
xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[23] ^ xgmiirev48[22] ^
xgmiirev48[21] ^ xgmiirev48[16] ^ xgmiirev48[15] ^ xgmiirev48[10] ^
xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[5] ^ xgmiirev48[3] ^ xgmiirev48[2] ^
xgmiirev48[0] ^ c[0] ^ c[5] ^ c[6] ^ c[7] ^ c[8] ^ c[9] ^ c[12] ^ c[13] ^ c[16]
^ c[18] ^ c[21] ^ c[23] ^ c[25] ^ c[26] ^ c[27] ^ c[29] ^ c[30] ^ c[31];
newcrc8[8]  <= xgmiirev48[46] ^ xgmiirev48[45] ^ xgmiirev48[43] ^ xgmiirev48[42]
^ xgmiirev48[40] ^ xgmiirev48[38] ^ xgmiirev48[37] ^ xgmiirev48[35] ^
xgmiirev48[34] ^ xgmiirev48[33] ^ xgmiirev48[32] ^ xgmiirev48[31] ^
xgmiirev48[28] ^ xgmiirev48[23] ^ xgmiirev48[22] ^ xgmiirev48[17] ^
xgmiirev48[12] ^ xgmiirev48[11] ^ xgmiirev48[10] ^ xgmiirev48[8] ^ xgmiirev48[4]
^ xgmiirev48[3] ^ xgmiirev48[1] ^ xgmiirev48[0] ^ c[1] ^ c[6] ^ c[7] ^ c[12] ^
c[15] ^ c[16] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[24] ^ c[26] ^ c[27] ^
c[29] ^ c[30]; newcrc8[9]  <= xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[44] ^
```

```
xgmiirev48[43] ^ xgmiirev48[41] ^ xgmiirev48[39] ^ xgmiirev48[38] ^
xgmiirev48[36] ^ xgmiirev48[35] ^ xgmiirev48[34] ^ xgmiirev48[33] ^
xgmiirev48[32] ^ xgmiirev48[29] ^ xgmiirev48[24] ^ xgmiirev48[23] ^
xgmiirev48[18] ^ xgmiirev48[13] ^ xgmiirev48[12] ^ xgmiirev48[11] ^
xgmiirev48[9] ^ xgmiirev48[5] ^ xgmiirev48[4] ^ xgmiirev48[2] ^ xgmiirev48[1] ^
c[2] ^ c[7] ^ c[8] ^ c[13] ^ c[16] ^ c[17] ^ c[18] ^ c[19] ^ c[20] ^ c[22] ^
c[23] ^ c[25] ^ c[27] ^ c[28] ^ c[30] ^ c[31]; newcrc8[10] <= xgmiirev48[42] ^
xgmiirev48[40] ^ xgmiirev48[39] ^ xgmiirev48[36] ^ xgmiirev48[35] ^
xgmiirev48[33] ^ xgmiirev48[32] ^ xgmiirev48[31] ^ xgmiirev48[29] ^
xgmiirev48[28] ^ xgmiirev48[26] ^ xgmiirev48[19] ^ xgmiirev48[16] ^
xgmiirev48[14] ^ xgmiirev48[13] ^ xgmiirev48[9] ^ xgmiirev48[5] ^ xgmiirev48[3]
^ xgmiirev48[2] ^ xgmiirev48[0] ^ c[0] ^ c[3] ^ c[10] ^ c[12] ^ c[13] ^ c[15] ^
c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[23] ^ c[24] ^ c[26]; newcrc8[11] <=
xgmiirev48[47] ^ xgmiirev48[45] ^ xgmiirev48[44] ^ xgmiirev48[43] ^
xgmiirev48[41] ^ xgmiirev48[40] ^ xgmiirev48[36] ^ xgmiirev48[33] ^
xgmiirev48[31] ^ xgmiirev48[28] ^ xgmiirev48[27] ^ xgmiirev48[26] ^
xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[20] ^ xgmiirev48[17] ^
xgmiirev48[16] ^ xgmiirev48[15] ^ xgmiirev48[14] ^ xgmiirev48[12] ^
xgmiirev48[9] ^ xgmiirev48[4] ^ xgmiirev48[3] ^ xgmiirev48[1] ^ xgmiirev48[0] ^
c[0] ^ c[1] ^ c[4] ^ c[8] ^ c[9] ^ c[10] ^ c[11] ^ c[12] ^ c[15] ^ c[17] ^ c[20]
^ c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[12] <= xgmiirev48[47] ^
xgmiirev48[46] ^ xgmiirev48[42] ^ xgmiirev48[41] ^ xgmiirev48[31] ^
xgmiirev48[30] ^ xgmiirev48[27] ^ xgmiirev48[24] ^ xgmiirev48[21] ^
xgmiirev48[18] ^ xgmiirev48[17] ^ xgmiirev48[15] ^ xgmiirev48[13] ^
xgmiirev48[12] ^ xgmiirev48[9] ^ xgmiirev48[6] ^ xgmiirev48[5] ^ xgmiirev48[4] ^
xgmiirev48[2] ^ xgmiirev48[1] ^ xgmiirev48[0] ^ c[1] ^ c[2] ^ c[5] ^ c[8] ^
c[11] ^ c[14] ^ c[15] ^ c[25] ^ c[26] ^ c[30] ^ c[31]; newcrc8[13] <=
xgmiirev48[47] ^ xgmiirev48[43] ^ xgmiirev48[42] ^ xgmiirev48[32] ^
xgmiirev48[31] ^ xgmiirev48[28] ^ xgmiirev48[25] ^ xgmiirev48[22] ^
xgmiirev48[19] ^ xgmiirev48[18] ^ xgmiirev48[16] ^ xgmiirev48[14] ^
xgmiirev48[13] ^ xgmiirev48[10] ^ xgmiirev48[7] ^ xgmiirev48[6] ^ xgmiirev48[5]
^ xgmiirev48[3] ^ xgmiirev48[2] ^ xgmiirev48[1] ^ c[0] ^ c[2] ^ c[3] ^ c[6] ^
c[9] ^ c[12] ^ c[15] ^ c[16] ^ c[26] ^ c[27] ^ c[31]; newcrc8[14] <=
xgmiirev48[44] ^ xgmiirev48[43] ^ xgmiirev48[33] ^ xgmiirev48[32] ^
xgmiirev48[29] ^ xgmiirev48[26] ^ xgmiirev48[23] ^ xgmiirev48[20] ^
xgmiirev48[19] ^ xgmiirev48[17] ^ xgmiirev48[15] ^ xgmiirev48[14] ^
xgmiirev48[11] ^ xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[6] ^ xgmiirev48[4] ^
xgmiirev48[3] ^ xgmiirev48[2] ^ c[1] ^ c[3] ^ c[4] ^ c[7] ^ c[10] ^ c[13] ^
c[16] ^ c[17] ^ c[27] ^ c[28]; newcrc8[15] <= xgmiirev48[45] ^ xgmiirev48[44] ^
xgmiirev48[34] ^ xgmiirev48[33] ^ xgmiirev48[30] ^ xgmiirev48[27] ^
xgmiirev48[24] ^ xgmiirev48[21] ^ xgmiirev48[20] ^ xgmiirev48[18] ^
xgmiirev48[16] ^ xgmiirev48[15] ^ xgmiirev48[12] ^ xgmiirev48[9] ^ xgmiirev48[8]
^ xgmiirev48[7] ^ xgmiirev48[5] ^ xgmiirev48[4] ^ xgmiirev48[3] ^ c[0] ^ c[2] ^
c[4] ^ c[5] ^ c[8] ^ c[11] ^ c[14] ^ c[17] ^ c[18] ^ c[28] ^ c[29]; newcrc8[16]
<= xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[44] ^ xgmiirev48[37] ^
xgmiirev48[35] ^ xgmiirev48[32] ^ xgmiirev48[30] ^ xgmiirev48[29] ^
xgmiirev48[26] ^ xgmiirev48[24] ^ xgmiirev48[22] ^ xgmiirev48[21] ^
xgmiirev48[19] ^ xgmiirev48[17] ^ xgmiirev48[13] ^ xgmiirev48[12] ^
```

```
xgmiirev48[8] ^ xgmiirev48[5] ^ xgmiirev48[4] ^ xgmiirev48[0] ^ c[1] ^ c[3] ^
c[5] ^ c[6] ^ c[8] ^ c[10] ^ c[13] ^ c[14] ^ c[16] ^ c[19] ^ c[21] ^ c[28] ^
c[30] ^ c[31]; newcrc8[17] <= xgmiirev48[47] ^ xgmiirev48[45] ^ xgmiirev48[38] ^
xgmiirev48[36] ^ xgmiirev48[33] ^ xgmiirev48[31] ^ xgmiirev48[30] ^
xgmiirev48[27] ^ xgmiirev48[25] ^ xgmiirev48[23] ^ xgmiirev48[22] ^
xgmiirev48[20] ^ xgmiirev48[18] ^ xgmiirev48[14] ^ xgmiirev48[13] ^
xgmiirev48[9] ^ xgmiirev48[6] ^ xgmiirev48[5] ^ xgmiirev48[1] ^ c[2] ^ c[4] ^
c[6] ^ c[7] ^ c[9] ^ c[11] ^ c[14] ^ c[15] ^ c[17] ^ c[20] ^ c[22] ^ c[29] ^
c[31]; newcrc8[18] <= xgmiirev48[46] ^ xgmiirev48[39] ^ xgmiirev48[37] ^
xgmiirev48[34] ^ xgmiirev48[32] ^ xgmiirev48[31] ^ xgmiirev48[28] ^
xgmiirev48[26] ^ xgmiirev48[24] ^ xgmiirev48[23] ^ xgmiirev48[21] ^
xgmiirev48[19] ^ xgmiirev48[15] ^ xgmiirev48[14] ^ xgmiirev48[10] ^
xgmiirev48[7] ^ xgmiirev48[6] ^ xgmiirev48[2] ^ c[3] ^ c[5] ^ c[7] ^ c[8] ^
c[10] ^ c[12] ^ c[15] ^ c[16] ^ c[18] ^ c[21] ^ c[23] ^ c[30]; newcrc8[19] <=
xgmiirev48[47] ^ xgmiirev48[40] ^ xgmiirev48[38] ^ xgmiirev48[35] ^
xgmiirev48[33] ^ xgmiirev48[32] ^ xgmiirev48[29] ^ xgmiirev48[27] ^
xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[22] ^ xgmiirev48[20] ^
xgmiirev48[16] ^ xgmiirev48[15] ^ xgmiirev48[11] ^ xgmiirev48[8] ^ xgmiirev48[7]
^ xgmiirev48[3] ^ c[0] ^ c[4] ^ c[6] ^ c[8] ^ c[9] ^ c[11] ^ c[13] ^ c[16] ^
c[17] ^ c[19] ^ c[22] ^ c[24] ^ c[31]; newcrc8[20] <= xgmiirev48[41] ^
xgmiirev48[39] ^ xgmiirev48[36] ^ xgmiirev48[34] ^ xgmiirev48[33] ^
xgmiirev48[30] ^ xgmiirev48[28] ^ xgmiirev48[26] ^ xgmiirev48[25] ^
xgmiirev48[23] ^ xgmiirev48[21] ^ xgmiirev48[17] ^ xgmiirev48[16] ^
xgmiirev48[12] ^ xgmiirev48[9] ^ xgmiirev48[8] ^ xgmiirev48[4] ^ c[0] ^ c[1] ^
c[5] ^ c[7] ^ c[9] ^ c[10] ^ c[12] ^ c[14] ^ c[17] ^ c[18] ^ c[20] ^ c[23] ^
c[25]; newcrc8[21] <= xgmiirev48[42] ^ xgmiirev48[40] ^ xgmiirev48[37] ^
xgmiirev48[35] ^ xgmiirev48[34] ^ xgmiirev48[31] ^ xgmiirev48[29] ^
xgmiirev48[27] ^ xgmiirev48[26] ^ xgmiirev48[24] ^ xgmiirev48[22] ^
xgmiirev48[18] ^ xgmiirev48[17] ^ xgmiirev48[13] ^ xgmiirev48[10] ^
xgmiirev48[9] ^ xgmiirev48[5] ^ c[1] ^ c[2] ^ c[6] ^ c[8] ^ c[10] ^ c[11] ^
c[13] ^ c[15] ^ c[18] ^ c[19] ^ c[21] ^ c[24] ^ c[26]; newcrc8[22] <=
xgmiirev48[47] ^ xgmiirev48[45] ^ xgmiirev48[44] ^ xgmiirev48[43] ^
xgmiirev48[41] ^ xgmiirev48[38] ^ xgmiirev48[37] ^ xgmiirev48[36] ^
xgmiirev48[35] ^ xgmiirev48[34] ^ xgmiirev48[31] ^ xgmiirev48[29] ^
xgmiirev48[27] ^ xgmiirev48[26] ^ xgmiirev48[24] ^ xgmiirev48[23] ^
xgmiirev48[19] ^ xgmiirev48[18] ^ xgmiirev48[16] ^ xgmiirev48[14] ^
xgmiirev48[12] ^ xgmiirev48[11] ^ xgmiirev48[9] ^ xgmiirev48[0] ^ c[0] ^ c[2] ^
c[3] ^ c[7] ^ c[8] ^ c[10] ^ c[11] ^ c[13] ^ c[15] ^ c[18] ^ c[19] ^ c[20] ^
c[21] ^ c[22] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[23] <=
xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[42] ^ xgmiirev48[39] ^
xgmiirev48[38] ^ xgmiirev48[36] ^ xgmiirev48[35] ^ xgmiirev48[34] ^
xgmiirev48[31] ^ xgmiirev48[29] ^ xgmiirev48[27] ^ xgmiirev48[26] ^
xgmiirev48[20] ^ xgmiirev48[19] ^ xgmiirev48[17] ^ xgmiirev48[16] ^
xgmiirev48[15] ^ xgmiirev48[13] ^ xgmiirev48[9] ^ xgmiirev48[6] ^ xgmiirev48[1]
^ xgmiirev48[0] ^ c[0] ^ c[1] ^ c[3] ^ c[4] ^ c[10] ^ c[11] ^ c[13] ^ c[15] ^
c[18] ^ c[19] ^ c[20] ^ c[22] ^ c[23] ^ c[26] ^ c[30] ^ c[31]; newcrc8[24] <=
xgmiirev48[47] ^ xgmiirev48[43] ^ xgmiirev48[40] ^ xgmiirev48[39] ^
xgmiirev48[37] ^ xgmiirev48[36] ^ xgmiirev48[35] ^ xgmiirev48[32] ^
```

```
xgmiirev48[30] ^ xgmiirev48[28] ^ xgmiirev48[27] ^ xgmiirev48[21] ^
xgmiirev48[20] ^ xgmiirev48[18] ^ xgmiirev48[17] ^ xgmiirev48[16] ^
xgmiirev48[14] ^ xgmiirev48[10] ^ xgmiirev48[7] ^ xgmiirev48[2] ^ xgmiirev48[1]
^ c[0] ^ c[1] ^ c[2] ^ c[4] ^ c[5] ^ c[11] ^ c[12] ^ c[14] ^ c[16] ^ c[19] ^
c[20] ^ c[21] ^ c[23] ^ c[24] ^ c[27] ^ c[31]; newcrc8[25] <= xgmiirev48[44] ^
xgmiirev48[41] ^ xgmiirev48[40] ^ xgmiirev48[38] ^ xgmiirev48[37] ^
xgmiirev48[36] ^ xgmiirev48[33] ^ xgmiirev48[31] ^ xgmiirev48[29] ^
xgmiirev48[28] ^ xgmiirev48[22] ^ xgmiirev48[21] ^ xgmiirev48[19] ^
xgmiirev48[18] ^ xgmiirev48[17] ^ xgmiirev48[15] ^ xgmiirev48[11] ^
xgmiirev48[8] ^ xgmiirev48[3] ^ xgmiirev48[2] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^ c[6]
^ c[12] ^ c[13] ^ c[15] ^ c[17] ^ c[20] ^ c[21] ^ c[22] ^ c[24] ^ c[25] ^ c[28];
newcrc8[26] <= xgmiirev48[47] ^ xgmiirev48[44] ^ xgmiirev48[42] ^ xgmiirev48[41]
^ xgmiirev48[39] ^ xgmiirev48[38] ^ xgmiirev48[31] ^ xgmiirev48[28] ^
xgmiirev48[26] ^ xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[23] ^
xgmiirev48[22] ^ xgmiirev48[20] ^ xgmiirev48[19] ^ xgmiirev48[18] ^
xgmiirev48[10] ^ xgmiirev48[6] ^ xgmiirev48[4] ^ xgmiirev48[3] ^ xgmiirev48[0] ^
c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[7] ^ c[8] ^ c[9] ^ c[10] ^ c[12] ^ c[15] ^ c[22] ^
c[23] ^ c[25] ^ c[26] ^ c[28] ^ c[31]; newcrc8[27] <= xgmiirev48[45] ^
xgmiirev48[43] ^ xgmiirev48[42] ^ xgmiirev48[40] ^ xgmiirev48[39] ^
xgmiirev48[32] ^ xgmiirev48[29] ^ xgmiirev48[27] ^ xgmiirev48[26] ^
xgmiirev48[25] ^ xgmiirev48[24] ^ xgmiirev48[23] ^ xgmiirev48[21] ^
xgmiirev48[20] ^ xgmiirev48[19] ^ xgmiirev48[11] ^ xgmiirev48[7] ^ xgmiirev48[5]
^ xgmiirev48[4] ^ xgmiirev48[1] ^ c[3] ^ c[4] ^ c[5] ^ c[7] ^ c[8] ^ c[9] ^
c[10] ^ c[11] ^ c[13] ^ c[16] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^ c[29];
newcrc8[28] <= xgmiirev48[46] ^ xgmiirev48[44] ^ xgmiirev48[43] ^ xgmiirev48[41]
^ xgmiirev48[40] ^ xgmiirev48[33] ^ xgmiirev48[30] ^ xgmiirev48[28] ^
xgmiirev48[27] ^ xgmiirev48[26] ^ xgmiirev48[25] ^ xgmiirev48[24] ^
xgmiirev48[22] ^ xgmiirev48[21] ^ xgmiirev48[20] ^ xgmiirev48[12] ^
xgmiirev48[8] ^ xgmiirev48[6] ^ xgmiirev48[5] ^ xgmiirev48[2] ^ c[4] ^ c[5] ^
c[6] ^ c[8] ^ c[9] ^ c[10] ^ c[11] ^ c[12] ^ c[14] ^ c[17] ^ c[24] ^ c[25] ^
c[27] ^ c[28] ^ c[30]; newcrc8[29] <= xgmiirev48[47] ^ xgmiirev48[45] ^
xgmiirev48[44] ^ xgmiirev48[42] ^ xgmiirev48[41] ^ xgmiirev48[34] ^
xgmiirev48[31] ^ xgmiirev48[29] ^ xgmiirev48[28] ^ xgmiirev48[27] ^
xgmiirev48[26] ^ xgmiirev48[25] ^ xgmiirev48[23] ^ xgmiirev48[22] ^
xgmiirev48[21] ^ xgmiirev48[13] ^ xgmiirev48[9] ^ xgmiirev48[7] ^ xgmiirev48[6]
^ xgmiirev48[3] ^ c[5] ^ c[6] ^ c[7] ^ c[9] ^ c[10] ^ c[11] ^ c[12] ^ c[13] ^
c[15] ^ c[18] ^ c[25] ^ c[26] ^ c[28] ^ c[29] ^ c[31]; newcrc8[30] <=
xgmiirev48[46] ^ xgmiirev48[45] ^ xgmiirev48[43] ^ xgmiirev48[42] ^
xgmiirev48[35] ^ xgmiirev48[32] ^ xgmiirev48[30] ^ xgmiirev48[29] ^
xgmiirev48[28] ^ xgmiirev48[27] ^ xgmiirev48[26] ^ xgmiirev48[24] ^
xgmiirev48[23] ^ xgmiirev48[22] ^ xgmiirev48[14] ^ xgmiirev48[10] ^
xgmiirev48[8] ^ xgmiirev48[7] ^ xgmiirev48[4] ^ c[6] ^ c[7] ^ c[8] ^ c[10] ^
c[11] ^ c[12] ^ c[13] ^ c[14] ^ c[16] ^ c[19] ^ c[26] ^ c[27] ^ c[29] ^ c[30];
newcrc8[31] <= xgmiirev48[47] ^ xgmiirev48[46] ^ xgmiirev48[44] ^ xgmiirev48[43]
^ xgmiirev48[36] ^ xgmiirev48[33] ^ xgmiirev48[31] ^ xgmiirev48[30] ^
xgmiirev48[29] ^ xgmiirev48[28] ^ xgmiirev48[27] ^ xgmiirev48[25] ^
xgmiirev48[24] ^ xgmiirev48[23] ^ xgmiirev48[15] ^ xgmiirev48[11] ^
xgmiirev48[9] ^ xgmiirev48[8] ^ xgmiirev48[5] ^ c[7] ^ c[8] ^ c[9] ^ c[11] ^
```

```
c[12] ^ c[13] ^ c[14] ^ c[15] ^ c[17] ^ c[20] ^ c[27] ^ c[28] ^ c[30] ^ c[31];

checksum_buff<=1; end else if (xgmiidata[43:36]==8’hFD) begin  xgmii[15:8] <=
xgmiidata[52:45];  xgmii[23:16] <= xgmiidata[61:54]; xgmii[31:24] <=
xgmiidata[70:63]; xgmii[39:32] <= xgmiidata[7:0]; xgmii[47:40] <=
xgmiidata[16:9]; xgmii[55:48] <= xgmiidata[25:18]; xgmii[63:56] <=
xgmiidata[34:27]; xgmii[7:0]<=0; empty_flag<=5; eop_flag_buff <= 1; newcrc8[0]
<= xgmiirev56[55] ^ xgmiirev56[54] ^ xgmiirev56[53] ^ xgmiirev56[50] ^
xgmiirev56[48] ^ xgmiirev56[47] ^ xgmiirev56[45] ^ xgmiirev56[44] ^
xgmiirev56[37] ^ xgmiirev56[34] ^ xgmiirev56[32] ^ xgmiirev56[31] ^
xgmiirev56[30] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[26] ^
xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[16] ^ xgmiirev56[12] ^
xgmiirev56[10] ^ xgmiirev56[9] ^ xgmiirev56[6] ^ xgmiirev56[0] ^ c[0] ^ c[1] ^
c[2] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[8] ^ c[10] ^ c[13] ^ c[20] ^ c[21] ^ c[23]
^ c[24] ^ c[26] ^ c[29] ^ c[30] ^ c[31]; newcrc8[1]  <= xgmiirev56[53] ^
xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[49] ^ xgmiirev56[47] ^
xgmiirev56[46] ^ xgmiirev56[44] ^ xgmiirev56[38] ^ xgmiirev56[37] ^
xgmiirev56[35] ^ xgmiirev56[34] ^ xgmiirev56[33] ^ xgmiirev56[28] ^
xgmiirev56[27] ^ xgmiirev56[24] ^ xgmiirev56[17] ^ xgmiirev56[16] ^
xgmiirev56[13] ^ xgmiirev56[12] ^ xgmiirev56[11] ^ xgmiirev56[9] ^ xgmiirev56[7]
^ xgmiirev56[6] ^ xgmiirev56[1] ^ xgmiirev56[0] ^ c[0] ^ c[3] ^ c[4] ^ c[9] ^
c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[20] ^ c[22] ^ c[23] ^ c[25] ^ c[26] ^ c[27] ^
c[29]; newcrc8[2]  <= xgmiirev56[55] ^ xgmiirev56[53] ^ xgmiirev56[52] ^
xgmiirev56[51] ^ xgmiirev56[44] ^ xgmiirev56[39] ^ xgmiirev56[38] ^
xgmiirev56[37] ^ xgmiirev56[36] ^ xgmiirev56[35] ^ xgmiirev56[32] ^
xgmiirev56[31] ^ xgmiirev56[30] ^ xgmiirev56[26] ^ xgmiirev56[24] ^
xgmiirev56[18] ^ xgmiirev56[17] ^ xgmiirev56[16] ^ xgmiirev56[14] ^
xgmiirev56[13] ^ xgmiirev56[9] ^ xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[6] ^
xgmiirev56[2] ^ xgmiirev56[1] ^ xgmiirev56[0] ^ c[0] ^ c[2] ^ c[6] ^ c[7] ^ c[8]
^ c[11] ^ c[12] ^ c[13] ^ c[14] ^ c[15] ^ c[20] ^ c[27] ^ c[28] ^ c[29] ^ c[31];
newcrc8[3]  <= xgmiirev56[54] ^ xgmiirev56[53] ^ xgmiirev56[52] ^ xgmiirev56[45]
^ xgmiirev56[40] ^ xgmiirev56[39] ^ xgmiirev56[38] ^ xgmiirev56[37] ^
xgmiirev56[36] ^ xgmiirev56[33] ^ xgmiirev56[32] ^ xgmiirev56[31] ^
xgmiirev56[27] ^ xgmiirev56[25] ^ xgmiirev56[19] ^ xgmiirev56[18] ^
xgmiirev56[17] ^ xgmiirev56[15] ^ xgmiirev56[14] ^ xgmiirev56[10] ^
xgmiirev56[9] ^ xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[3] ^ xgmiirev56[2] ^
xgmiirev56[1] ^ c[1] ^ c[3] ^ c[7] ^ c[8] ^ c[9] ^ c[12] ^ c[13] ^ c[14] ^ c[15]
^ c[16] ^ c[21] ^ c[28] ^ c[29] ^ c[30]; newcrc8[4]  <= xgmiirev56[50] ^
xgmiirev56[48] ^ xgmiirev56[47] ^ xgmiirev56[46] ^ xgmiirev56[45] ^
xgmiirev56[44] ^ xgmiirev56[41] ^ xgmiirev56[40] ^ xgmiirev56[39] ^
xgmiirev56[38] ^ xgmiirev56[33] ^ xgmiirev56[31] ^ xgmiirev56[30] ^
xgmiirev56[29] ^ xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[20] ^
xgmiirev56[19] ^ xgmiirev56[18] ^ xgmiirev56[15] ^ xgmiirev56[12] ^
xgmiirev56[11] ^ xgmiirev56[8] ^ xgmiirev56[6] ^ xgmiirev56[4] ^ xgmiirev56[3] ^
xgmiirev56[2] ^ xgmiirev56[0] ^ c[0] ^ c[1] ^ c[5] ^ c[6] ^ c[7] ^ c[9] ^ c[14]
^ c[15] ^ c[16] ^ c[17] ^ c[20] ^ c[21] ^ c[22] ^ c[23] ^ c[24] ^ c[26];
newcrc8[5]  <= xgmiirev56[55] ^ xgmiirev56[54] ^ xgmiirev56[53] ^ xgmiirev56[51]
^ xgmiirev56[50] ^ xgmiirev56[49] ^ xgmiirev56[46] ^ xgmiirev56[44] ^
```

```
xgmiirev56[42] ^ xgmiirev56[41] ^ xgmiirev56[40] ^ xgmiirev56[39] ^
xgmiirev56[37] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[24] ^
xgmiirev56[21] ^ xgmiirev56[20] ^ xgmiirev56[19] ^ xgmiirev56[13] ^
xgmiirev56[10] ^ xgmiirev56[7] ^ xgmiirev56[6] ^ xgmiirev56[5] ^ xgmiirev56[4] ^
xgmiirev56[3] ^ xgmiirev56[1] ^ xgmiirev56[0] ^ c[0] ^ c[4] ^ c[5] ^ c[13] ^
c[15] ^ c[16] ^ c[17] ^ c[18] ^ c[20] ^ c[22] ^ c[25] ^ c[26] ^ c[27] ^ c[29] ^
c[30] ^ c[31]; newcrc8[6]  <= xgmiirev56[55] ^ xgmiirev56[54] ^ xgmiirev56[52] ^
xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[47] ^ xgmiirev56[45] ^
xgmiirev56[43] ^ xgmiirev56[42] ^ xgmiirev56[41] ^ xgmiirev56[40] ^
xgmiirev56[38] ^ xgmiirev56[30] ^ xgmiirev56[29] ^ xgmiirev56[25] ^
xgmiirev56[22] ^ xgmiirev56[21] ^ xgmiirev56[20] ^ xgmiirev56[14] ^
xgmiirev56[11] ^ xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[6] ^ xgmiirev56[5] ^
xgmiirev56[4] ^ xgmiirev56[2] ^ xgmiirev56[1] ^ c[1] ^ c[5] ^ c[6] ^ c[14] ^
c[16] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[23] ^ c[26] ^ c[27] ^ c[28] ^ c[30] ^
c[31]; newcrc8[7]  <= xgmiirev56[54] ^ xgmiirev56[52] ^ xgmiirev56[51] ^
xgmiirev56[50] ^ xgmiirev56[47] ^ xgmiirev56[46] ^ xgmiirev56[45] ^
xgmiirev56[43] ^ xgmiirev56[42] ^ xgmiirev56[41] ^ xgmiirev56[39] ^
xgmiirev56[37] ^ xgmiirev56[34] ^ xgmiirev56[32] ^ xgmiirev56[29] ^
xgmiirev56[28] ^ xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[23] ^
xgmiirev56[22] ^ xgmiirev56[21] ^ xgmiirev56[16] ^ xgmiirev56[15] ^
xgmiirev56[10] ^ xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[5] ^ xgmiirev56[3] ^
xgmiirev56[2] ^ xgmiirev56[0] ^ c[0] ^ c[1] ^ c[4] ^ c[5] ^ c[8] ^ c[10] ^ c[13]
^ c[15] ^ c[17] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[23] ^ c[26] ^ c[27] ^ c[28]
^ c[30]; newcrc8[8]  <= xgmiirev56[54] ^ xgmiirev56[52] ^ xgmiirev56[51] ^
xgmiirev56[50] ^ xgmiirev56[46] ^ xgmiirev56[45] ^ xgmiirev56[43] ^
xgmiirev56[42] ^ xgmiirev56[40] ^ xgmiirev56[38] ^ xgmiirev56[37] ^
xgmiirev56[35] ^ xgmiirev56[34] ^ xgmiirev56[33] ^ xgmiirev56[32] ^
xgmiirev56[31] ^ xgmiirev56[28] ^ xgmiirev56[23] ^ xgmiirev56[22] ^
xgmiirev56[17] ^ xgmiirev56[12] ^ xgmiirev56[11] ^ xgmiirev56[10] ^
xgmiirev56[8] ^ xgmiirev56[4] ^ xgmiirev56[3] ^ xgmiirev56[1] ^ xgmiirev56[0] ^
c[4] ^ c[7] ^ c[8] ^ c[9] ^ c[10] ^ c[11] ^ c[13] ^ c[14] ^ c[16] ^ c[18] ^
c[19] ^ c[21] ^ c[22] ^ c[26] ^ c[27] ^ c[28] ^ c[30]; newcrc8[9]  <=
xgmiirev56[55] ^ xgmiirev56[53] ^ xgmiirev56[52] ^ xgmiirev56[51] ^
xgmiirev56[47] ^ xgmiirev56[46] ^ xgmiirev56[44] ^ xgmiirev56[43] ^
xgmiirev56[41] ^ xgmiirev56[39] ^ xgmiirev56[38] ^ xgmiirev56[36] ^
xgmiirev56[35] ^ xgmiirev56[34] ^ xgmiirev56[33] ^ xgmiirev56[32] ^
xgmiirev56[29] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[18] ^
xgmiirev56[13] ^ xgmiirev56[12] ^ xgmiirev56[11] ^ xgmiirev56[9] ^ xgmiirev56[5]
^ xgmiirev56[4] ^ xgmiirev56[2] ^ xgmiirev56[1] ^ c[0] ^ c[5] ^ c[8] ^ c[9] ^
c[10] ^ c[11] ^ c[12] ^ c[14] ^ c[15] ^ c[17] ^ c[19] ^ c[20] ^ c[22] ^ c[23] ^
c[27] ^ c[28] ^ c[29] ^ c[31]; newcrc8[10] <= xgmiirev56[55] ^ xgmiirev56[52] ^
xgmiirev56[50] ^ xgmiirev56[42] ^ xgmiirev56[40] ^ xgmiirev56[39] ^
xgmiirev56[36] ^ xgmiirev56[35] ^ xgmiirev56[33] ^ xgmiirev56[32] ^
xgmiirev56[31] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[26] ^
xgmiirev56[19] ^ xgmiirev56[16] ^ xgmiirev56[14] ^ xgmiirev56[13] ^
xgmiirev56[9] ^ xgmiirev56[5] ^ xgmiirev56[3] ^ xgmiirev56[2] ^ xgmiirev56[0] ^
c[2] ^ c[4] ^ c[5] ^ c[7] ^ c[8] ^ c[9] ^ c[11] ^ c[12] ^ c[15] ^ c[16] ^ c[18]
^ c[26] ^ c[28] ^ c[31]; newcrc8[11] <= xgmiirev56[55] ^ xgmiirev56[54] ^
```

```
xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[48] ^ xgmiirev56[47] ^
xgmiirev56[45] ^ xgmiirev56[44] ^ xgmiirev56[43] ^ xgmiirev56[41] ^
xgmiirev56[40] ^ xgmiirev56[36] ^ xgmiirev56[33] ^ xgmiirev56[31] ^
xgmiirev56[28] ^ xgmiirev56[27] ^ xgmiirev56[26] ^ xgmiirev56[25] ^
xgmiirev56[24] ^ xgmiirev56[20] ^ xgmiirev56[17] ^ xgmiirev56[16] ^
xgmiirev56[15] ^ xgmiirev56[14] ^ xgmiirev56[12] ^ xgmiirev56[9] ^ xgmiirev56[4]
^ xgmiirev56[3] ^ xgmiirev56[1] ^ xgmiirev56[0] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^
c[4] ^ c[7] ^ c[9] ^ c[12] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^ c[23] ^
c[24] ^ c[26] ^ c[27] ^ c[30] ^ c[31]; newcrc8[12] <= xgmiirev56[54] ^
xgmiirev56[53] ^ xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[50] ^
xgmiirev56[49] ^ xgmiirev56[47] ^ xgmiirev56[46] ^ xgmiirev56[42] ^
xgmiirev56[41] ^ xgmiirev56[31] ^ xgmiirev56[30] ^ xgmiirev56[27] ^
xgmiirev56[24] ^ xgmiirev56[21] ^ xgmiirev56[18] ^ xgmiirev56[17] ^
xgmiirev56[15] ^ xgmiirev56[13] ^ xgmiirev56[12] ^ xgmiirev56[9] ^ xgmiirev56[6]
^ xgmiirev56[5] ^ xgmiirev56[4] ^ xgmiirev56[2] ^ xgmiirev56[1] ^ xgmiirev56[0]
^ c[0] ^ c[3] ^ c[6] ^ c[7] ^ c[17] ^ c[18] ^ c[22] ^ c[23] ^ c[25] ^ c[26] ^
c[27] ^ c[28] ^ c[29] ^ c[30]; newcrc8[13] <= xgmiirev56[55] ^ xgmiirev56[54] ^
xgmiirev56[53] ^ xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[50] ^
xgmiirev56[48] ^ xgmiirev56[47] ^ xgmiirev56[43] ^ xgmiirev56[42] ^
xgmiirev56[32] ^ xgmiirev56[31] ^ xgmiirev56[28] ^ xgmiirev56[25] ^
xgmiirev56[22] ^ xgmiirev56[19] ^ xgmiirev56[18] ^ xgmiirev56[16] ^
xgmiirev56[14] ^ xgmiirev56[13] ^ xgmiirev56[10] ^ xgmiirev56[7] ^ xgmiirev56[6]
^ xgmiirev56[5] ^ xgmiirev56[3] ^ xgmiirev56[2] ^ xgmiirev56[1] ^ c[1] ^ c[4] ^
c[7] ^ c[8] ^ c[18] ^ c[19] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^ c[28] ^ c[29] ^
c[30] ^ c[31]; newcrc8[14] <= xgmiirev56[55] ^ xgmiirev56[54] ^ xgmiirev56[53] ^
xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[49] ^ xgmiirev56[48] ^
xgmiirev56[44] ^ xgmiirev56[43] ^ xgmiirev56[33] ^ xgmiirev56[32] ^
xgmiirev56[29] ^ xgmiirev56[26] ^ xgmiirev56[23] ^ xgmiirev56[20] ^
xgmiirev56[19] ^ xgmiirev56[17] ^ xgmiirev56[15] ^ xgmiirev56[14] ^
xgmiirev56[11] ^ xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[6] ^ xgmiirev56[4] ^
xgmiirev56[3] ^ xgmiirev56[2] ^ c[2] ^ c[5] ^ c[8] ^ c[9] ^ c[19] ^ c[20] ^
c[24] ^ c[25] ^ c[27] ^ c[28] ^ c[29] ^ c[30] ^ c[31]; newcrc8[15] <=
xgmiirev56[55] ^ xgmiirev56[54] ^ xgmiirev56[53] ^ xgmiirev56[52] ^
xgmiirev56[50] ^ xgmiirev56[49] ^ xgmiirev56[45] ^ xgmiirev56[44] ^
xgmiirev56[34] ^ xgmiirev56[33] ^ xgmiirev56[30] ^ xgmiirev56[27] ^
xgmiirev56[24] ^ xgmiirev56[21] ^ xgmiirev56[20] ^ xgmiirev56[18] ^
xgmiirev56[16] ^ xgmiirev56[15] ^ xgmiirev56[12] ^ xgmiirev56[9] ^ xgmiirev56[8]
^ xgmiirev56[7] ^ xgmiirev56[5] ^ xgmiirev56[4] ^ xgmiirev56[3] ^ c[0] ^ c[3] ^
c[6] ^ c[9] ^ c[10] ^ c[20] ^ c[21] ^ c[25] ^ c[26] ^ c[28] ^ c[29] ^ c[30] ^
c[31]; newcrc8[16] <= xgmiirev56[51] ^ xgmiirev56[48] ^ xgmiirev56[47] ^
xgmiirev56[46] ^ xgmiirev56[44] ^ xgmiirev56[37] ^ xgmiirev56[35] ^
xgmiirev56[32] ^ xgmiirev56[30] ^ xgmiirev56[29] ^ xgmiirev56[26] ^
xgmiirev56[24] ^ xgmiirev56[22] ^ xgmiirev56[21] ^ xgmiirev56[19] ^
xgmiirev56[17] ^ xgmiirev56[13] ^ xgmiirev56[12] ^ xgmiirev56[8] ^ xgmiirev56[5]
^ xgmiirev56[4] ^ xgmiirev56[0] ^ c[0] ^ c[2] ^ c[5] ^ c[6] ^ c[8] ^ c[11] ^
c[13] ^ c[20] ^ c[22] ^ c[23] ^ c[24] ^ c[27]; newcrc8[17] <= xgmiirev56[52] ^
xgmiirev56[49] ^ xgmiirev56[48] ^ xgmiirev56[47] ^ xgmiirev56[45] ^
xgmiirev56[38] ^ xgmiirev56[36] ^ xgmiirev56[33] ^ xgmiirev56[31] ^
```

```
xgmiirev56[30] ^ xgmiirev56[27] ^ xgmiirev56[25] ^ xgmiirev56[23] ^
xgmiirev56[22] ^ xgmiirev56[20] ^ xgmiirev56[18] ^ xgmiirev56[14] ^
xgmiirev56[13] ^ xgmiirev56[9] ^ xgmiirev56[6] ^ xgmiirev56[5] ^ xgmiirev56[1] ^
c[1] ^ c[3] ^ c[6] ^ c[7] ^ c[9] ^ c[12] ^ c[14] ^ c[21] ^ c[23] ^ c[24] ^ c[25]
^ c[28]; newcrc8[18] <= xgmiirev56[53] ^ xgmiirev56[50] ^ xgmiirev56[49] ^
xgmiirev56[48] ^ xgmiirev56[46] ^ xgmiirev56[39] ^ xgmiirev56[37] ^
xgmiirev56[34] ^ xgmiirev56[32] ^ xgmiirev56[31] ^ xgmiirev56[28] ^
xgmiirev56[26] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[21] ^
xgmiirev56[19] ^ xgmiirev56[15] ^ xgmiirev56[14] ^ xgmiirev56[10] ^
xgmiirev56[7] ^ xgmiirev56[6] ^ xgmiirev56[2] ^ c[0] ^ c[2] ^ c[4] ^ c[7] ^ c[8]
^ c[10] ^ c[13] ^ c[15] ^ c[22] ^ c[24] ^ c[25] ^ c[26] ^ c[29]; newcrc8[19] <=
xgmiirev56[54] ^ xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[49] ^
xgmiirev56[47] ^ xgmiirev56[40] ^ xgmiirev56[38] ^ xgmiirev56[35] ^
xgmiirev56[33] ^ xgmiirev56[32] ^ xgmiirev56[29] ^ xgmiirev56[27] ^
xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[22] ^ xgmiirev56[20] ^
xgmiirev56[16] ^ xgmiirev56[15] ^ xgmiirev56[11] ^ xgmiirev56[8] ^ xgmiirev56[7]
^ xgmiirev56[3] ^ c[0] ^ c[1] ^ c[3] ^ c[5] ^ c[8] ^ c[9] ^ c[11] ^ c[14] ^
c[16] ^ c[23] ^ c[25] ^ c[26] ^ c[27] ^ c[30]; newcrc8[20] <= xgmiirev56[55] ^
xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[48] ^
xgmiirev56[41] ^ xgmiirev56[39] ^ xgmiirev56[36] ^ xgmiirev56[34] ^
xgmiirev56[33] ^ xgmiirev56[30] ^ xgmiirev56[28] ^ xgmiirev56[26] ^
xgmiirev56[25] ^ xgmiirev56[23] ^ xgmiirev56[21] ^ xgmiirev56[17] ^
xgmiirev56[16] ^ xgmiirev56[12] ^ xgmiirev56[9] ^ xgmiirev56[8] ^ xgmiirev56[4]
^ c[1] ^ c[2] ^ c[4] ^ c[6] ^ c[9] ^ c[10] ^ c[12] ^ c[15] ^ c[17] ^ c[24] ^
c[26] ^ c[27] ^ c[28] ^ c[31]; newcrc8[21] <= xgmiirev56[53] ^ xgmiirev56[52] ^
xgmiirev56[51] ^ xgmiirev56[49] ^ xgmiirev56[42] ^ xgmiirev56[40] ^
xgmiirev56[37] ^ xgmiirev56[35] ^ xgmiirev56[34] ^ xgmiirev56[31] ^
xgmiirev56[29] ^ xgmiirev56[27] ^ xgmiirev56[26] ^ xgmiirev56[24] ^
xgmiirev56[22] ^ xgmiirev56[18] ^ xgmiirev56[17] ^ xgmiirev56[13] ^
xgmiirev56[10] ^ xgmiirev56[9] ^ xgmiirev56[5] ^ c[0] ^ c[2] ^ c[3] ^ c[5] ^
c[7] ^ c[10] ^ c[11] ^ c[13] ^ c[16] ^ c[18] ^ c[25] ^ c[27] ^ c[28] ^ c[29];
newcrc8[22] <= xgmiirev56[55] ^ xgmiirev56[52] ^ xgmiirev56[48] ^ xgmiirev56[47]
^ xgmiirev56[45] ^ xgmiirev56[44] ^ xgmiirev56[43] ^ xgmiirev56[41] ^
xgmiirev56[38] ^ xgmiirev56[37] ^ xgmiirev56[36] ^ xgmiirev56[35] ^
xgmiirev56[34] ^ xgmiirev56[31] ^ xgmiirev56[29] ^ xgmiirev56[27] ^
xgmiirev56[26] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[19] ^
xgmiirev56[18] ^ xgmiirev56[16] ^ xgmiirev56[14] ^ xgmiirev56[12] ^
xgmiirev56[11] ^ xgmiirev56[9] ^ xgmiirev56[0] ^ c[0] ^ c[2] ^ c[3] ^ c[5] ^
c[7] ^ c[10] ^ c[11] ^ c[12] ^ c[13] ^ c[14] ^ c[17] ^ c[19] ^ c[20] ^ c[21] ^
c[23] ^ c[24] ^ c[28] ^ c[31]; newcrc8[23] <= xgmiirev56[55] ^ xgmiirev56[54] ^
xgmiirev56[50] ^ xgmiirev56[49] ^ xgmiirev56[47] ^ xgmiirev56[46] ^
xgmiirev56[42] ^ xgmiirev56[39] ^ xgmiirev56[38] ^ xgmiirev56[36] ^
xgmiirev56[35] ^ xgmiirev56[34] ^ xgmiirev56[31] ^ xgmiirev56[29] ^
xgmiirev56[27] ^ xgmiirev56[26] ^ xgmiirev56[20] ^ xgmiirev56[19] ^
xgmiirev56[17] ^ xgmiirev56[16] ^ xgmiirev56[15] ^ xgmiirev56[13] ^
xgmiirev56[9] ^ xgmiirev56[6] ^ xgmiirev56[1] ^ xgmiirev56[0] ^ c[2] ^ c[3] ^
c[5] ^ c[7] ^ c[10] ^ c[11] ^ c[12] ^ c[14] ^ c[15] ^ c[18] ^ c[22] ^ c[23] ^
c[25] ^ c[26] ^ c[30] ^ c[31]; newcrc8[24] <= xgmiirev56[55] ^ xgmiirev56[51] ^
```

```
xgmiirev56[50] ^ xgmiirev56[48] ^ xgmiirev56[47] ^ xgmiirev56[43] ^
xgmiirev56[40] ^ xgmiirev56[39] ^ xgmiirev56[37] ^ xgmiirev56[36] ^
xgmiirev56[35] ^ xgmiirev56[32] ^ xgmiirev56[30] ^ xgmiirev56[28] ^
xgmiirev56[27] ^ xgmiirev56[21] ^ xgmiirev56[20] ^ xgmiirev56[18] ^
xgmiirev56[17] ^ xgmiirev56[16] ^ xgmiirev56[14] ^ xgmiirev56[10] ^
xgmiirev56[7] ^ xgmiirev56[2] ^ xgmiirev56[1] ^ c[3] ^ c[4] ^ c[6] ^ c[8] ^
c[11] ^ c[12] ^ c[13] ^ c[15] ^ c[16] ^ c[19] ^ c[23] ^ c[24] ^ c[26] ^ c[27] ^
c[31]; newcrc8[25] <= xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[49] ^
xgmiirev56[48] ^ xgmiirev56[44] ^ xgmiirev56[41] ^ xgmiirev56[40] ^
xgmiirev56[38] ^ xgmiirev56[37] ^ xgmiirev56[36] ^ xgmiirev56[33] ^
xgmiirev56[31] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[22] ^
xgmiirev56[21] ^ xgmiirev56[19] ^ xgmiirev56[18] ^ xgmiirev56[17] ^
xgmiirev56[15] ^ xgmiirev56[11] ^ xgmiirev56[8] ^ xgmiirev56[3] ^ xgmiirev56[2]
^ c[4] ^ c[5] ^ c[7] ^ c[9] ^ c[12] ^ c[13] ^ c[14] ^ c[16] ^ c[17] ^ c[20] ^
c[24] ^ c[25] ^ c[27] ^ c[28]; newcrc8[26] <= xgmiirev56[55] ^ xgmiirev56[54] ^
xgmiirev56[52] ^ xgmiirev56[49] ^ xgmiirev56[48] ^ xgmiirev56[47] ^
xgmiirev56[44] ^ xgmiirev56[42] ^ xgmiirev56[41] ^ xgmiirev56[39] ^
xgmiirev56[38] ^ xgmiirev56[31] ^ xgmiirev56[28] ^ xgmiirev56[26] ^
xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[22] ^
xgmiirev56[20] ^ xgmiirev56[19] ^ xgmiirev56[18] ^ xgmiirev56[10] ^
xgmiirev56[6] ^ xgmiirev56[4] ^ xgmiirev56[3] ^ xgmiirev56[0] ^ c[0] ^ c[1] ^
c[2] ^ c[4] ^ c[7] ^ c[14] ^ c[15] ^ c[17] ^ c[18] ^ c[20] ^ c[23] ^ c[24] ^
c[25] ^ c[28] ^ c[30] ^ c[31]; newcrc8[27] <= xgmiirev56[55] ^ xgmiirev56[53] ^
xgmiirev56[50] ^ xgmiirev56[49] ^ xgmiirev56[48] ^ xgmiirev56[45] ^
xgmiirev56[43] ^ xgmiirev56[42] ^ xgmiirev56[40] ^ xgmiirev56[39] ^
xgmiirev56[32] ^ xgmiirev56[29] ^ xgmiirev56[27] ^ xgmiirev56[26] ^
xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[21] ^
xgmiirev56[20] ^ xgmiirev56[19] ^ xgmiirev56[11] ^ xgmiirev56[7] ^ xgmiirev56[5]
^ xgmiirev56[4] ^ xgmiirev56[1] ^ c[0] ^ c[1] ^ c[2] ^ c[3] ^ c[5] ^ c[8] ^
c[15] ^ c[16] ^ c[18] ^ c[19] ^ c[21] ^ c[24] ^ c[25] ^ c[26] ^ c[29] ^ c[31];
newcrc8[28] <= xgmiirev56[54] ^ xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[49]
^ xgmiirev56[46] ^ xgmiirev56[44] ^ xgmiirev56[43] ^ xgmiirev56[41] ^
xgmiirev56[40] ^ xgmiirev56[33] ^ xgmiirev56[30] ^ xgmiirev56[28] ^
xgmiirev56[27] ^ xgmiirev56[26] ^ xgmiirev56[25] ^ xgmiirev56[24] ^
xgmiirev56[22] ^ xgmiirev56[21] ^ xgmiirev56[20] ^ xgmiirev56[12] ^
xgmiirev56[8] ^ xgmiirev56[6] ^ xgmiirev56[5] ^ xgmiirev56[2] ^ c[0] ^ c[1] ^
c[2] ^ c[3] ^ c[4] ^ c[6] ^ c[9] ^ c[16] ^ c[17] ^ c[19] ^ c[20] ^ c[22] ^ c[25]
^ c[26] ^ c[27] ^ c[30]; newcrc8[29] <= xgmiirev56[55] ^ xgmiirev56[52] ^
xgmiirev56[51] ^ xgmiirev56[50] ^ xgmiirev56[47] ^ xgmiirev56[45] ^
xgmiirev56[44] ^ xgmiirev56[42] ^ xgmiirev56[41] ^ xgmiirev56[34] ^
xgmiirev56[31] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[27] ^
xgmiirev56[26] ^ xgmiirev56[25] ^ xgmiirev56[23] ^ xgmiirev56[22] ^
xgmiirev56[21] ^ xgmiirev56[13] ^ xgmiirev56[9] ^ xgmiirev56[7] ^ xgmiirev56[6]
^ xgmiirev56[3] ^ c[1] ^ c[2] ^ c[3] ^ c[4] ^ c[5] ^ c[7] ^ c[10] ^ c[17] ^
c[18] ^ c[20] ^ c[21] ^ c[23] ^ c[26] ^ c[27] ^ c[28] ^ c[31]; newcrc8[30] <=
xgmiirev56[53] ^ xgmiirev56[52] ^ xgmiirev56[51] ^ xgmiirev56[48] ^
xgmiirev56[46] ^ xgmiirev56[45] ^ xgmiirev56[43] ^ xgmiirev56[42] ^
xgmiirev56[35] ^ xgmiirev56[32] ^ xgmiirev56[30] ^ xgmiirev56[29] ^
```

```
xgmiirev56[28] ^ xgmiirev56[27] ^ xgmiirev56[26] ^ xgmiirev56[24] ^
xgmiirev56[23] ^ xgmiirev56[22] ^ xgmiirev56[14] ^ xgmiirev56[10] ^
xgmiirev56[8] ^ xgmiirev56[7] ^ xgmiirev56[4] ^ c[0] ^ c[2] ^ c[3] ^ c[4] ^ c[5]
^ c[6] ^ c[8] ^ c[11] ^ c[18] ^ c[19] ^ c[21] ^ c[22] ^ c[24] ^ c[27] ^ c[28] ^
c[29]; newcrc8[31] <= xgmiirev56[54] ^ xgmiirev56[53] ^ xgmiirev56[52] ^
xgmiirev56[49] ^ xgmiirev56[47] ^ xgmiirev56[46] ^ xgmiirev56[44] ^
xgmiirev56[43] ^ xgmiirev56[36] ^ xgmiirev56[33] ^ xgmiirev56[31] ^
xgmiirev56[30] ^ xgmiirev56[29] ^ xgmiirev56[28] ^ xgmiirev56[27] ^
xgmiirev56[25] ^ xgmiirev56[24] ^ xgmiirev56[23] ^ xgmiirev56[15] ^
xgmiirev56[11] ^ xgmiirev56[9] ^ xgmiirev56[8] ^ xgmiirev56[5] ^ c[0] ^ c[1] ^
c[3] ^ c[4] ^ c[5] ^ c[6] ^ c[7] ^ c[9] ^ c[12] ^ c[19] ^ c[20] ^ c[22] ^ c[23]
^ c[25] ^ c[28] ^ c[29] ^ c[30];


checksum_buff<=1;
end
//
else if(eop_flag==0 && eop_flag_buff==0) begin
eop <= 0;
valid <=0;
empty   <=0;
data    <= 63’b0;
            newcrc  <=32’hFFFFFFFF;
newcrc8 <= 32’hFFFFFFFF;
end
if(eop_flag)begin
eop<=1;
eop_flag<=0;
eop_flag_buff<=0;
valid_flag<=0;
valid_flag_buff<=0;
end
///
if(checksum_flag==1)begin
checksum_flag<=0;
checksum_err <= !(newcrc8==32’hC704DD7B);
end
else begin
checksum_err <=0;
end
if(checksum_buff != 0)begin
checksum_flag <= 1;
checksum_buff<=0;
end
end
end
endmodule
```

align.v (adapter from previous project)

```
module xgmii(
  input logic clk,
input logic [71:0] xgmiidata,// input data
input logic reset,
output logic flag,

output logic [71:0] xgmii//output data
);

logic flag2;
logic [35:0] buff;
logic[7:0] xgmiicontrol;
assign xgmiicontrol = {xgmiidata[44], xgmiidata[53],xgmiidata[62], xgmiidata[71],xgmiidata[8],

initial begin
flag<=0;
flag2<=0;
buff<=0;
end
always@(posedge clk) begin
if(reset) begin
flag<=0;
buff<=0;
end
else begin
  if(flag==0 && xgmiicontrol !=8’h1F) begin
 xgmii<=xgmiidata;

  end
  if(xgmiicontrol == 8’h1F)begin
 xgmii[34:27] <= 8’h07;
 xgmii[25:18] <= 8’h07;
 xgmii[16:9]   <=   8’h07;
 xgmii[7:0] <=   8’h07;
 xgmii[70:63] <= 8’h07;
 xgmii[61:54] <=   8’h07;
 xgmii[52:45] <= 8’h07;
 xgmii[43:36] <= 8’h07;
 {xgmii[44], xgmii[53],xgmii[62], xgmii[71],xgmii[8], xgmii[17],
  xgmii[26], xgmii[35]} <=8’hFF;
 buff<=xgmiidata[71:36];
 flag<=1;
 flag2<=1;
  end
  if(flag2)begin
  {xgmii[44], xgmii[53],xgmii[62], xgmii[71],xgmii[8],
   xgmii[17], xgmii[26], xgmii[35]} <=8'h01;
  flag2<=0;
  end
  if(flag==1)begin
 xgmii[35:0]<=buff;
 xgmii[71:36]<=xgmiidata[35:0];
 buff<=xgmiidata[71:36];
  end
  if(flag==1 && xgmiicontrol == 8'hFF)begin
 flag<=0;
  end
end
end
endmodule
```

align.sv

```
module xgmii(
  input logic clk,
input logic [71:0] xgmiidata,// input data
input logic reset,
output logic flag,

output logic [71:0] xgmii//output data
);

logic flag2;
logic [35:0] buff;
logic[7:0] xgmiicontrol;
assign xgmiicontrol = {xgmiidata[44], xgmiidata[53],xgmiidata[62], xgmiidata[71],xgmiidata[8],

initial begin
flag<=0;
flag2<=0;
buff<=0;
end
always@(posedge clk) begin
if(reset) begin
flag<=0;
buff<=0;
end
else begin
  if(flag==0 && xgmiicontrol !=8'h1F) begin
 xgmii<=xgmiidata;

  end
  if(xgmiicontrol == 8'h1F)begin
 xgmii[34:27] <= 8'h07;
 xgmii[25:18] <= 8'h07;
 xgmii[16:9]   <=   8'h07;
 xgmii[7:0] <=   8'h07;
 xgmii[70:63] <= 8'h07;
 xgmii[61:54] <=   8'h07;
 xgmii[52:45] <= 8'h07;
 xgmii[43:36] <= 8'h07;
 {xgmii[44], xgmii[53],xgmii[62], xgmii[71],xgmii[8], xgmii[17], xgmii[26], xgmii[35]} <=8'hFF
 buff<=xgmiidata[71:36];
 flag<=1;
 flag2<=1;
  end
  if(flag2)begin
  {xgmii[44], xgmii[53],xgmii[62], xgmii[71],xgmii[8], xgmii[17], xgmii[26], xgmii[35]} <=8'h0
  flag2<=0;
  end
  if(flag==1)begin
 xgmii[35:0]<=buff;
 xgmii[71:36]<=xgmiidata[35:0];
 buff<=xgmiidata[71:36];
  end
  if(flag==1 && xgmiicontrol == 8'hFF)begin
 flag<=0;
  end
end
end
endmodule
```

fast_mac_tx_loopback.sv

```
module fast_mac_tx_loopback(
    input  logic        clk,
    input  logic [71:0] tx_out,
    output logic [71:0] swap_in);

    always @(posedge clk) begin
        swap_in <= tx_out;
    end
endmodule
```

fast_mac_rx_loopback.sv

```
module fast_mac_rx_loopback(
    input  logic        clk,
    input  logic [63:0] rx_out,
    output logic [63:0] txbuff_in);
    always @(posedge clk) begin
        txbuff_in <= rx_out;
    end
endmodule
```

blazepps.qsys

```
<?xml version="1.0" encoding="UTF-8"?>
<system name="$${FILENAME}">
 <component
   name="$${FILENAME}"
   displayName="$${FILENAME}"
   version="1.0"
   description=""
   tags=""
   categories="System" />
 <parameter name="bonusData"><![CDATA[bonusData
{
   element $${FILENAME}
   {
   }
   element button_pio
   {
      datum _sortIndex
      {
         value = "9";
         type = "int";
      }
   }
   element clk_0
   {
      datum _sortIndex
      {
         value = "6";
         type = "int";
      }
   }
   element eth_10g_mac
   {
      datum _sortIndex
      {
         value = "5";
         type = "int";
      }
   }
   element hps_0.f2h_axi_slave
   {
```

```
   datum baseAddress
   {
      value = "4294967296";
      type = "String";
   }
}
element hps_0
{
   datum _sortIndex
   {
      value = "7";
      type = "int";
   }
}
element tx_oc_fifo.in
{
   datum baseAddress
   {
      value = "32";
      type = "String";
   }
}
element master_0
{
   datum _sortIndex
   {
      value = "8";
      type = "int";
   }
}
element rx_oc_fifo.out
{
   datum baseAddress
   {
      value = "40";
      type = "String";
   }
}
element rx_dc_fifo
{
   datum _sortIndex
   {
      value = "4";
      type = "int";
   }
}
element rx_dc_fifo_loopback
{
   datum _sortIndex
   {
      value = "12";
      type = "int";
   }
}
element rx_oc_fifo
{
   datum _sortIndex
   {
      value = "2";
      type = "int";
   }
}
element button_pio.s1
{
   datum baseAddress
   {
      value = "0";
      type = "String";
   }
}
element tx_dc_fifo
{
   datum _sortIndex
   {
      value = "3";
      type = "int";
   }
}
element tx_dc_fifo_loopback
{
   datum _sortIndex
   {
      value = "11";
      type = "int";
   }
}
element tx_oc_fifo
{
   datum _sortIndex
   {
      value = "1";
      type = "int";
   }
}
element tx_oc_fifo_loopback
{
      datum _sortIndex
      {
         value = "10";
         type = "int";
      }
   }
   element xaui_clk
   {
      datum _sortIndex
      {
         value = "0";
         type = "int";
      }
   }
}
]]></parameter>
 <parameter name="clockCrossingAdapter" value="HANDSHAKE" />
 <parameter name="device" value="5CSXFC6D6F31C8ES" />
 <parameter name="deviceFamily" value="Cyclone V" />
 <parameter name="deviceSpeedGrade" value="8_H6" />
 <parameter name="fabricMode" value="QSYS" />
 <parameter name="generateLegacySim" value="false" />
 <parameter name="generationId" value="0" />
 <parameter name="globalResetBus" value="false" />
 <parameter name="hdlLanguage" value="VERILOG" />
 <parameter name="maxAdditionalLatency" value="1" />
 <parameter name="projectName" value="blazepps.qpf" />
 <parameter name="sopcBorderPoints" value="false" />
 <parameter name="systemHash" value="0" />
 <parameter name="timeStamp" value="0" />
 <parameter name="useTestBenchNamingPattern" value="false" />
 <instanceScript></instanceScript>
 <interface name="clk" internal="clk_0.clk_in" type="clock" dir="end" />
 <interface name="reset" internal="clk_0.clk_in_reset" type="reset" dir="end" />
 <interface name="memory" internal="hps_0.memory" type="conduit" dir="end" />
 <interface name="hps_io" internal="hps_0.hps_io" type="conduit" dir="end" />
 <interface
   name="button_pio_0_external_connection"
   internal="button_pio.external_connection" />
 <interface name="xaui_clk" internal="xaui_clk.clk_in" type="clock" dir="end" />
 <interface
   name="xaui_reset"
   internal="xaui_clk.clk_in_reset"
   type="reset"
   dir="end" />
 <interface name="eth_10g_mac_xgmii_tx" internal="eth_10g_mac.xgmii_tx" />
 <interface name="eth_10g_mac_xgmii_rx" internal="eth_10g_mac.xgmii_rx" />
 <interface name="eth_10g_mac_avalon_st_tx" internal="eth_10g_mac.avalon_st_tx" />
```

```
<interface
  name="eth_10g_mac_avalon_st_rxstatus"
  internal="eth_10g_mac.avalon_st_rxstatus" />
<interface
  name="eth_10g_mac_link_fault_status_xgmii_rx"
  internal="eth_10g_mac.link_fault_status_xgmii_rx" />
<interface
  name="stream_src"
  internal="tx_dc_fifo.out"
  type="avalon_streaming"
  dir="start" />
<interface
  name="stream_sink"
  internal="rx_dc_fifo.in"
  type="avalon_streaming"
  dir="end" />
<module kind="clock_source" version="13.1" enabled="1" name="clk_0">
 <parameter name="clockFrequency" value="50000000" />
 <parameter name="clockFrequencyKnown" value="true" />
 <parameter name="inputClockFrequency" value="0" />
 <parameter name="resetSynchronousEdges" value="NONE" />
</module>
<module kind="altera_hps" version="13.1" enabled="1" name="hps_0">
 <parameter name="MEM_VENDOR" value="Micron" />
 <parameter name="MEM_FORMAT" value="DISCRETE" />
 <parameter name="RDIMM_CONFIG" value="0000000000000000" />
 <parameter name="LRDIMM_EXTENDED_CONFIG">0x000000000000000000</parameter>
 <parameter name="DISCRETE_FLY_BY" value="true" />
 <parameter name="DEVICE_DEPTH" value="1" />
 <parameter name="MEM_MIRROR_ADDRESSING" value="0" />
 <parameter name="MEM_CLK_FREQ_MAX" value="800.0" />
 <parameter name="MEM_ROW_ADDR_WIDTH" value="15" />
 <parameter name="MEM_COL_ADDR_WIDTH" value="10" />
 <parameter name="MEM_DQ_WIDTH" value="32" />
 <parameter name="MEM_DQ_PER_DQS" value="8" />
 <parameter name="MEM_BANKADDR_WIDTH" value="3" />
 <parameter name="MEM_IF_DM_PINS_EN" value="true" />
 <parameter name="MEM_IF_DQSN_EN" value="true" />
 <parameter name="MEM_NUMBER_OF_DIMMS" value="1" />
 <parameter name="MEM_NUMBER_OF_RANKS_PER_DIMM" value="1" />
 <parameter name="MEM_NUMBER_OF_RANKS_PER_DEVICE" value="1" />
 <parameter name="MEM_RANK_MULTIPLICATION_FACTOR" value="1" />
 <parameter name="MEM_CK_WIDTH" value="1" />
 <parameter name="MEM_CS_WIDTH" value="1" />
 <parameter name="MEM_CLK_EN_WIDTH" value="1" />
 <parameter name="ALTMEMPHY_COMPATIBLE_MODE" value="false" />
 <parameter name="NEXTGEN" value="true" />
 <parameter name="MEM_IF_BOARD_BASE_DELAY" value="10" />
```

```
<parameter name="MEM_IF_SIM_VALID_WINDOW" value="0" />
<parameter name="MEM_GUARANTEED_WRITE_INIT" value="false" />
<parameter name="MEM_VERBOSE" value="true" />
<parameter name="PINGPONGPHY_EN" value="false" />
<parameter name="REFRESH_BURST_VALIDATION" value="false" />
<parameter name="MEM_BL" value="OTF" />
<parameter name="MEM_BT" value="Sequential" />
<parameter name="MEM_ASR" value="Manual" />
<parameter name="MEM_SRT" value="Normal" />
<parameter name="MEM_PD" value="DLL off" />
<parameter name="MEM_DRV_STR" value="RZQ/7" />
<parameter name="MEM_DLL_EN" value="true" />
<parameter name="MEM_RTT_NOM" value="RZQ/4" />
<parameter name="MEM_RTT_WR" value="RZQ/4" />
<parameter name="MEM_WTCL" value="8" />
<parameter name="MEM_ATCL" value="Disabled" />
<parameter name="MEM_TCL" value="11" />
<parameter name="MEM_AUTO_LEVELING_MODE" value="true" />
<parameter name="MEM_USER_LEVELING_MODE" value="Leveling" />
<parameter name="MEM_INIT_EN" value="false" />
<parameter name="MEM_INIT_FILE" value="" />
<parameter name="DAT_DATA_WIDTH" value="32" />
<parameter name="TIMING_TIS" value="180" />
<parameter name="TIMING_TIH" value="140" />
<parameter name="TIMING_TDS" value="30" />
<parameter name="TIMING_TDH" value="65" />
<parameter name="TIMING_TDQSQ" value="125" />
<parameter name="TIMING_TQHS" value="300" />
<parameter name="TIMING_TQH" value="0.38" />
<parameter name="TIMING_TDQSCK" value="255" />
<parameter name="TIMING_TDQSCKDS" value="450" />
<parameter name="TIMING_TDQSCKDM" value="900" />
<parameter name="TIMING_TDQSCKDL" value="1200" />
<parameter name="TIMING_TDQSS" value="0.25" />
<parameter name="TIMING_TDQSH" value="0.35" />
<parameter name="TIMING_TQSH" value="0.4" />
<parameter name="TIMING_TDSH" value="0.2" />
<parameter name="TIMING_TDSS" value="0.2" />
<parameter name="MEM_TINIT_US" value="500" />
<parameter name="MEM_TMRD_CK" value="4" />
<parameter name="MEM_TRAS_NS" value="35.0" />
<parameter name="MEM_TRCD_NS" value="13.75" />
<parameter name="MEM_TRP_NS" value="13.75" />
<parameter name="MEM_TREFI_US" value="7.8" />
<parameter name="MEM_TRFC_NS" value="260.0" />
<parameter name="CFG_TCCD_NS" value="2.5" />
<parameter name="MEM_TWR_NS" value="15.0" />
<parameter name="MEM_TWTR" value="4" />
```

```
<parameter name="MEM_TFAW_NS" value="30.0" />
<parameter name="MEM_TRRD_NS" value="7.5" />
<parameter name="MEM_TRTP_NS" value="7.5" />
<parameter name="POWER_OF_TWO_BUS" value="false" />
<parameter name="SOPC_COMPAT_RESET" value="false" />
<parameter name="AVL_MAX_SIZE" value="4" />
<parameter name="BYTE_ENABLE" value="true" />
<parameter name="ENABLE_CTRL_AVALON_INTERFACE" value="true" />
<parameter name="CTL_DEEP_POWERDN_EN" value="false" />
<parameter name="CTL_SELF_REFRESH_EN" value="false" />
<parameter name="AUTO_POWERDN_EN" value="false" />
<parameter name="AUTO_PD_CYCLES" value="0" />
<parameter name="CTL_USR_REFRESH_EN" value="false" />
<parameter name="CTL_AUTOPCH_EN" value="false" />
<parameter name="CTL_ZQCAL_EN" value="false" />
<parameter name="ADDR_ORDER" value="0" />
<parameter name="CTL_LOOK_AHEAD_DEPTH" value="4" />
<parameter name="CONTROLLER_LATENCY" value="5" />
<parameter name="CFG_REORDER_DATA" value="true" />
<parameter name="STARVE_LIMIT" value="10" />
<parameter name="CTL_CSR_ENABLED" value="false" />
<parameter name="CTL_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="CTL_ECC_ENABLED" value="false" />
<parameter name="CTL_HRB_ENABLED" value="false" />
<parameter name="CTL_ECC_AUTO_CORRECTION_ENABLED" value="false" />
<parameter name="MULTICAST_EN" value="false" />
<parameter name="CTL_DYNAMIC_BANK_ALLOCATION" value="false" />
<parameter name="CTL_DYNAMIC_BANK_NUM" value="4" />
<parameter name="DEBUG_MODE" value="false" />
<parameter name="ENABLE_BURST_MERGE" value="false" />
<parameter name="CTL_ENABLE_BURST_INTERRUPT" value="false" />
<parameter name="CTL_ENABLE_BURST_TERMINATE" value="false" />
<parameter name="LOCAL_ID_WIDTH" value="8" />
<parameter name="WRBUFFER_ADDR_WIDTH" value="6" />
<parameter name="MAX_PENDING_WR_CMD" value="16" />
<parameter name="MAX_PENDING_RD_CMD" value="32" />
<parameter name="USE_MM_ADAPTOR" value="true" />
<parameter name="USE_AXI_ADAPTOR" value="false" />
<parameter name="HCX_COMPAT_MODE" value="false" />
<parameter name="CTL_CMD_QUEUE_DEPTH" value="8" />
<parameter name="CTL_CSR_READ_ONLY" value="1" />
<parameter name="CFG_DATA_REORDERING_TYPE" value="INTER_BANK" />
<parameter name="NUM_OF_PORTS" value="1" />
<parameter name="ENABLE_BONDING" value="false" />
<parameter name="ENABLE_USER_ECC" value="false" />
<parameter name="AVL_DATA_WIDTH_PORT" value="32,32,32,32,32,32" />
<parameter name="PRIORITY_PORT" value="1,1,1,1,1,1" />
<parameter name="WEIGHT_PORT" value="0,0,0,0,0,0" />
```

```
<parameter name="CPORT_TYPE_PORT">Bidirectional,Bidirectional,Bidirectional,Bidirectional,
Bidirectional,Bidirectional</parameter>
<parameter name="ENABLE_EMIT_BFM_MASTER" value="false" />
<parameter name="FORCE_SEQUENCER_TCL_DEBUG_MODE" value="false" />
<parameter name="ENABLE_SEQUENCER_MARGINING_ON_BY_DEFAULT" value="false" />
<parameter name="REF_CLK_FREQ" value="25.0" />
<parameter name="REF_CLK_FREQ_PARAM_VALID" value="false" />
<parameter name="REF_CLK_FREQ_MIN_PARAM" value="0.0" />
<parameter name="REF_CLK_FREQ_MAX_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_DR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_DR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_DR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_DR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_MEM_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_MEM_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_MEM_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_ADDR_CMD_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_ADDR_CMD_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_ADDR_CMD_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_HALF_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_HALF_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_HALF_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_NIOS_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_PHASE_PS_PARAM" value="0" />
```

```
<parameter name="PLL_NIOS_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_NIOS_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_NIOS_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_CONFIG_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_CONFIG_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_CONFIG_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_P2C_READ_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_P2C_READ_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_P2C_READ_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_C2P_WRITE_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_C2P_WRITE_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_C2P_WRITE_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_HR_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_HR_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_HR_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_HR_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_HR_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_HR_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_PARAM" value="0.0" />
<parameter name="PLL_AFI_PHY_CLK_FREQ_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_PHASE_PS_SIM_STR_PARAM" value="" />
<parameter name="PLL_AFI_PHY_CLK_MULT_PARAM" value="0" />
<parameter name="PLL_AFI_PHY_CLK_DIV_PARAM" value="0" />
<parameter name="PLL_CLK_PARAM_VALID" value="false" />
<parameter name="ENABLE_EXTRA_REPORTING" value="false" />
<parameter name="NUM_EXTRA_REPORT_PATH" value="10" />
<parameter name="ENABLE_ISS_PROBES" value="false" />
<parameter name="CALIB_REG_WIDTH" value="8" />
<parameter name="USE_SEQUENCER_BFM" value="false" />
<parameter name="DEFAULT_FAST_SIM_MODEL" value="true" />
<parameter name="PLL_SHARING_MODE" value="None" />
<parameter name="NUM_PLL_SHARING_INTERFACES" value="1" />
<parameter name="EXPORT_AFI_HALF_CLK" value="false" />
<parameter name="ABSTRACT_REAL_COMPARE_TEST" value="false" />
<parameter name="INCLUDE_BOARD_DELAY_MODEL" value="false" />
<parameter name="INCLUDE_MULTIRANK_BOARD_DELAY_MODEL" value="false" />
<parameter name="USE_FAKE_PHY" value="false" />
<parameter name="FORCE_MAX_LATENCY_COUNT_WIDTH" value="0" />
```

```
<parameter name="ENABLE_NON_DESTRUCTIVE_CALIB" value="false" />
<parameter name="TRACKING_ERROR_TEST" value="false" />
<parameter name="TRACKING_WATCH_TEST" value="false" />
<parameter name="MARGIN_VARIATION_TEST" value="false" />
<parameter name="EXTRA_SETTINGS" value="" />
<parameter name="MEM_DEVICE" value="MISSING_MODEL" />
<parameter name="FORCE_SYNTHESIS_LANGUAGE" value="" />
<parameter name="FORCED_NUM_WRITE_FR_CYCLE_SHIFTS" value="0" />
<parameter name="SEQUENCER_TYPE" value="NIOS" />
<parameter name="ADVERTIZE_SEQUENCER_SW_BUILD_FILES" value="false" />
<parameter name="FORCED_NON_LDC_ADDR_CMD_MEM_CK_INVERT" value="false" />
<parameter name="PHY_ONLY" value="false" />
<parameter name="SEQ_MODE" value="0" />
<parameter name="ADVANCED_CK_PHASES" value="false" />
<parameter name="COMMAND_PHASE" value="0.0" />
<parameter name="MEM_CK_PHASE" value="0.0" />
<parameter name="P2C_READ_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="C2P_WRITE_CLOCK_ADD_PHASE" value="0.0" />
<parameter name="ACV_PHY_CLK_ADD_FR_PHASE" value="0.0" />
<parameter name="MEM_VOLTAGE" value="1.5V DDR3" />
<parameter name="PLL_LOCATION" value="Top_Bottom" />
<parameter name="SKIP_MEM_INIT" value="true" />
<parameter name="READ_DQ_DQS_CLOCK_SOURCE" value="INVERTED_DQS_BUS" />
<parameter name="DQ_INPUT_REG_USE_CLKN" value="false" />
<parameter name="DQS_DQSN_MODE" value="DIFFERENTIAL" />
<parameter name="AFI_DEBUG_INFO_WIDTH" value="32" />
<parameter name="CALIBRATION_MODE" value="Skip" />
<parameter name="NIOS_ROM_DATA_WIDTH" value="32" />
<parameter name="READ_FIFO_SIZE" value="8" />
<parameter name="PHY_CSR_ENABLED" value="false" />
<parameter name="PHY_CSR_CONNECTION" value="INTERNAL_JTAG" />
<parameter name="USER_DEBUG_LEVEL" value="1" />
<parameter name="TIMING_BOARD_DERATE_METHOD" value="AUTO" />
<parameter name="TIMING_BOARD_CK_CKN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_AC_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_DQS_DQSN_SLEW_RATE" value="2.0" />
<parameter name="TIMING_BOARD_DQ_SLEW_RATE" value="1.0" />
<parameter name="TIMING_BOARD_TIS" value="0.0" />
<parameter name="TIMING_BOARD_TIH" value="0.0" />
<parameter name="TIMING_BOARD_TDS" value="0.0" />
<parameter name="TIMING_BOARD_TDH" value="0.0" />
<parameter name="TIMING_BOARD_ISI_METHOD" value="AUTO" />
<parameter name="TIMING_BOARD_AC_EYE_REDUCTION_SU" value="0.0" />
<parameter name="TIMING_BOARD_AC_EYE_REDUCTION_H" value="0.0" />
<parameter name="TIMING_BOARD_DQ_EYE_REDUCTION" value="0.0" />
<parameter name="TIMING_BOARD_DELTA_DQS_ARRIVAL_TIME" value="0.0" />
<parameter name="TIMING_BOARD_READ_DQ_EYE_REDUCTION" value="0.0" />
<parameter name="TIMING_BOARD_DELTA_READ_DQS_ARRIVAL_TIME" value="0.0" />
```

```
<parameter name="PACKAGE_DESKEW" value="false" />
<parameter name="AC_PACKAGE_DESKEW" value="false" />
<parameter name="TIMING_BOARD_MAX_CK_DELAY" value="0.03" />
<parameter name="TIMING_BOARD_MAX_DQS_DELAY" value="0.02" />
<parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MIN" value="0.09" />
<parameter name="TIMING_BOARD_SKEW_CKDQS_DIMM_MAX" value="0.16" />
<parameter name="TIMING_BOARD_SKEW_BETWEEN_DIMMS" value="0.05" />
<parameter name="TIMING_BOARD_SKEW_WITHIN_DQS" value="0.01" />
<parameter name="TIMING_BOARD_SKEW_BETWEEN_DQS" value="0.08" />
<parameter name="TIMING_BOARD_DQ_TO_DQS_SKEW" value="0.0" />
<parameter name="TIMING_BOARD_AC_SKEW" value="0.03" />
<parameter name="TIMING_BOARD_AC_TO_CK_SKEW" value="0.0" />
<parameter name="RATE" value="Full" />
<parameter name="MEM_CLK_FREQ" value="400.0" />
<parameter name="USE_MEM_CLK_FREQ" value="false" />
<parameter name="FORCE_DQS_TRACKING" value="AUTO" />
<parameter name="FORCE_SHADOW_REGS" value="AUTO" />
<parameter name="MRS_MIRROR_PING_PONG_ATSO" value="false" />
<parameter name="SYS_INFO_DEVICE_FAMILY" value="Cyclone V" />
<parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM_VALID" value="false" />
<parameter name="PARSE_FRIENDLY_DEVICE_FAMILY_PARAM" value="" />
<parameter name="DEVICE_FAMILY_PARAM" value="" />
<parameter name="SPEED_GRADE" value="7" />
<parameter name="IS_ES_DEVICE" value="false" />
<parameter name="DISABLE_CHILD_MESSAGING" value="false" />
<parameter name="HARD_EMIF" value="true" />
<parameter name="HHP_HPS" value="true" />
<parameter name="HHP_HPS_VERIFICATION" value="false" />
<parameter name="HHP_HPS_SIMULATION" value="false" />
<parameter name="HPS_PROTOCOL" value="DDR3" />
<parameter name="CUT_NEW_FAMILY_TIMING" value="true" />
<parameter name="ENABLE_EXPORT_SEQ_DEBUG_BRIDGE" value="false" />
<parameter name="CORE_DEBUG_CONNECTION" value="EXPORT" />
<parameter name="ADD_EXTERNAL_SEQ_DEBUG_NIOS" value="false" />
<parameter name="ED_EXPORT_SEQ_DEBUG" value="false" />
<parameter name="ADD_EFFICIENCY_MONITOR" value="false" />
<parameter name="ENABLE_ABS_RAM_MEM_INIT" value="false" />
<parameter name="ABS_RAM_MEM_INIT_FILENAME" value="meminit" />
<parameter name="DLL_SHARING_MODE" value="None" />
<parameter name="NUM_DLL_SHARING_INTERFACES" value="1" />
<parameter name="OCT_SHARING_MODE" value="None" />
<parameter name="NUM_OCT_SHARING_INTERFACES" value="1" />
<parameter name="MPU_EVENTS_Enable" value="false" />
<parameter name="GP_Enable" value="false" />
<parameter name="DEBUGAPB_Enable" value="false" />
<parameter name="STM_Enable" value="false" />
<parameter name="CTI_Enable" value="false" />
<parameter name="TPIUFPGA_Enable" value="false" />
```

```
<parameter name="BOOTFROMFPGA_Enable" value="false" />
<parameter name="TEST_Enable" value="false" />
<parameter name="HLGPI_Enable" value="false" />
<parameter name="BSEL_EN" value="false" />
<parameter name="BSEL" value="1" />
<parameter name="CSEL_EN" value="false" />
<parameter name="CSEL" value="0" />
<parameter name="F2S_Width" value="2" />
<parameter name="S2F_Width" value="2" />
<parameter name="LWH2F_Enable" value="true" />
<parameter name="F2SDRAM_Type" value="" />
<parameter name="F2SDRAM_Width" value="" />
<parameter name="BONDING_OUT_ENABLED" value="false" />
<parameter name="S2FCLK_COLDRST_Enable" value="false" />
<parameter name="S2FCLK_PENDINGRST_Enable" value="false" />
<parameter name="F2SCLK_DBGRST_Enable" value="false" />
<parameter name="F2SCLK_WARMRST_Enable" value="false" />
<parameter name="F2SCLK_COLDRST_Enable" value="false" />
<parameter name="DMA_Enable">No,No,No,No,No,No,No,No</parameter>
<parameter name="F2SINTERRUPT_Enable" value="true" />
<parameter name="S2FINTERRUPT_CAN_Enable" value="false" />
<parameter name="S2FINTERRUPT_CLOCKPERIPHERAL_Enable" value="false" />
<parameter name="S2FINTERRUPT_CTI_Enable" value="false" />
<parameter name="S2FINTERRUPT_DMA_Enable" value="false" />
<parameter name="S2FINTERRUPT_EMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_FPGAMANAGER_Enable" value="false" />
<parameter name="S2FINTERRUPT_GPIO_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CEMAC_Enable" value="false" />
<parameter name="S2FINTERRUPT_I2CPERIPHERAL_Enable" value="false" />
<parameter name="S2FINTERRUPT_L4TIMER_Enable" value="false" />
<parameter name="S2FINTERRUPT_NAND_Enable" value="false" />
<parameter name="S2FINTERRUPT_OSCTIMER_Enable" value="false" />
<parameter name="S2FINTERRUPT_QSPI_Enable" value="false" />
<parameter name="S2FINTERRUPT_SDMMC_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPIMASTER_Enable" value="false" />
<parameter name="S2FINTERRUPT_SPISLAVE_Enable" value="false" />
<parameter name="S2FINTERRUPT_UART_Enable" value="false" />
<parameter name="S2FINTERRUPT_USB_Enable" value="false" />
<parameter name="S2FINTERRUPT_WATCHDOG_Enable" value="false" />
<parameter name="EMAC0_PinMuxing" value="Unused" />
<parameter name="EMAC0_Mode" value="N/A" />
<parameter name="EMAC1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="EMAC1_Mode" value="RGMII" />
<parameter name="NAND_PinMuxing" value="Unused" />
<parameter name="NAND_Mode" value="N/A" />
<parameter name="QSPI_PinMuxing" value="HPS I/O Set 0" />
<parameter name="QSPI_Mode" value="1 SS" />
<parameter name="SDIO_PinMuxing" value="HPS I/O Set 0" />
```

```
<parameter name="SDIO_Mode" value="4-bit Data" />
<parameter name="USB0_PinMuxing" value="Unused" />
<parameter name="USB0_Mode" value="N/A" />
<parameter name="USB1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="USB1_Mode" value="SDR" />
<parameter name="SPIM0_PinMuxing" value="HPS I/O Set 0" />
<parameter name="SPIM0_Mode" value="Single Slave Select" />
<parameter name="SPIM1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="SPIM1_Mode" value="Single Slave Select" />
<parameter name="SPIS0_PinMuxing" value="Unused" />
<parameter name="SPIS0_Mode" value="N/A" />
<parameter name="SPIS1_PinMuxing" value="Unused" />
<parameter name="SPIS1_Mode" value="N/A" />
<parameter name="UART0_PinMuxing" value="HPS I/O Set 0" />
<parameter name="UART0_Mode" value="No Flow Control" />
<parameter name="UART1_PinMuxing" value="Unused" />
<parameter name="UART1_Mode" value="N/A" />
<parameter name="I2C0_PinMuxing" value="Unused" />
<parameter name="I2C0_Mode" value="N/A" />
<parameter name="I2C1_PinMuxing" value="HPS I/O Set 0" />
<parameter name="I2C1_Mode" value="I2C" />
<parameter name="I2C2_PinMuxing" value="Unused" />
<parameter name="I2C2_Mode" value="N/A" />
<parameter name="I2C3_PinMuxing" value="Unused" />
<parameter name="I2C3_Mode" value="N/A" />
<parameter name="CAN0_PinMuxing" value="Unused" />
<parameter name="CAN0_Mode" value="N/A" />
<parameter name="CAN1_PinMuxing" value="Unused" />
<parameter name="CAN1_Mode" value="N/A" />
<parameter name="TRACE_PinMuxing" value="Unused" />
<parameter name="TRACE_Mode" value="N/A" />
<parameter name="GPIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No</parameter>
<parameter name="LOANIO_Enable">No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No
,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,No,
No,No,No,No,No,No,No,No,No,No,No</parameter>
<parameter name="S2FCLK_USER0CLK_Enable" value="false" />
<parameter name="S2FCLK_USER0CLK_FREQ" value="100" />
<parameter name="S2FCLK_USER1CLK_Enable" value="false" />
<parameter name="S2FCLK_USER1CLK_FREQ" value="100" />
<parameter name="S2FCLK_USER2CLK_Enable" value="false" />
<parameter name="S2FCLK_USER2CLK_FREQ" value="100" />
<parameter name="F2SCLK_PERIPHCLK_Enable" value="false" />
```

```
<parameter name="F2SCLK_PERIPHCLK_FREQ" value="100" />
<parameter name="F2SCLK_SDRAMCLK_Enable" value="false" />
<parameter name="F2SCLK_SDRAMCLK_FREQ" value="100" />
<parameter name="F2H_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="H2F_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="H2F_LW_AXI_CLOCK_FREQ" value="50000000" />
<parameter name="F2H_SDRAM0_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM1_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM2_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM3_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM4_CLOCK_FREQ" value="100" />
<parameter name="F2H_SDRAM5_CLOCK_FREQ" value="100" />
<parameter name="H2F_CTI_CLOCK_FREQ" value="100" />
<parameter name="H2F_TPIU_CLOCK_IN_FREQ" value="100" />
<parameter name="H2F_DEBUG_APB_CLOCK_FREQ" value="100" />
<parameter
   name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC_PTP_REF_CLOCK"
   value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_RX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC0_TX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_MD_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC0_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_RX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_EMAC1_TX_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_MD_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_EMAC1_GTX_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_QSPI_SCLK_OUT" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SDIO_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SDIO_CCLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB0_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_USB1_CLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM0_SCLK_OUT" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_SPIM1_SCLK_OUT" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIS0_SCLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_SPIS1_SCLK_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C0_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C0_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C1_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C1_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C2_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C2_CLK" value="100" />
<parameter name="FPGA_PERIPHERAL_INPUT_CLOCK_FREQ_I2C3_SCL_IN" value="100" />
<parameter name="FPGA_PERIPHERAL_OUTPUT_CLOCK_FREQ_I2C3_CLK" value="100" />
<parameter name="device_name" value="5CSXFC6D6F31C8ES" />
<parameter
   name="quartus_ini_hps_ip_enable_all_peripheral_fpga_interfaces"
   value="false" />
<parameter
```

```
    name="quartus_ini_hps_ip_enable_emac0_peripheral_fpga_interface"
    value="false" />
 <parameter name="quartus_ini_hps_ip_enable_test_interface" value="false" />
 <parameter name="quartus_ini_hps_ip_fast_f2sdram_sim_model" value="false" />
 <parameter name="quartus_ini_hps_ip_suppress_sdram_synth" value="false" />
 <parameter
    name="quartus_ini_hps_ip_enable_low_speed_serial_fpga_interfaces"
    value="false" />
 <parameter name="quartus_ini_hps_ip_enable_bsel_csel" value="false" />
 <parameter name="quartus_ini_hps_ip_f2sdram_bonding_out" value="false" />
</module>
<module
  kind="altera_jtag_avalon_master"
  version="13.1"
  enabled="1"
  name="master_0">
 <parameter name="USE_PLI" value="0" />
 <parameter name="PLI_PORT" value="50000" />
 <parameter name="COMPONENT_CLOCK" value="0" />
 <parameter name="FAST_VER" value="0" />
 <parameter name="FIFO_DEPTHS" value="2" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
</module>
<module kind="altera_avalon_pio" version="13.1" enabled="0" name="button_pio">
 <parameter name="bitClearingEdgeCapReg" value="false" />
 <parameter name="bitModifyingOutReg" value="true" />
 <parameter name="captureEdge" value="false" />
 <parameter name="direction" value="Output" />
 <parameter name="edgeType" value="RISING" />
 <parameter name="generateIRQ" value="false" />
 <parameter name="irqType" value="LEVEL" />
 <parameter name="resetValue" value="0" />
 <parameter name="simDoTestBenchWiring" value="false" />
 <parameter name="simDrivenValue" value="0" />
 <parameter name="width" value="2" />
 <parameter name="clockRate" value="50000000" />
</module>
<module kind="clock_source" version="13.1" enabled="1" name="xaui_clk">
 <parameter name="clockFrequency" value="15625000" />
 <parameter name="clockFrequencyKnown" value="true" />
 <parameter name="inputClockFrequency" value="0" />
 <parameter name="resetSynchronousEdges" value="NONE" />
</module>
<module
  kind="altera_avalon_fifo"
  version="13.1"
  enabled="1"
  name="tx_oc_fifo">
 <parameter name="avalonMMAvalonMMDataWidth" value="32" />
 <parameter name="avalonMMAvalonSTDataWidth" value="32" />
 <parameter name="bitsPerSymbol" value="8" />
 <parameter name="channelWidth" value="0" />
 <parameter name="errorWidth" value="0" />
 <parameter name="fifoDepth" value="512" />
 <parameter name="fifoInputInterfaceOptions" value="AVALONMM_WRITE" />
 <parameter name="fifoOutputInterfaceOptions" value="AVALONST_SOURCE" />
 <parameter name="showHiddenFeatures" value="false" />
 <parameter name="singleClockMode" value="true" />
 <parameter name="singleResetMode" value="true" />
 <parameter name="symbolsPerBeat" value="4" />
 <parameter name="useBackpressure" value="false" />
 <parameter name="useIRQ" value="true" />
 <parameter name="usePacket" value="true" />
 <parameter name="useReadControl" value="false" />
 <parameter name="useRegister" value="false" />
 <parameter name="useWriteControl" value="false" />
 <parameter name="deviceFamilyString" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_fifo"
  version="13.1"
  enabled="1"
  name="rx_oc_fifo">
 <parameter name="avalonMMAvalonMMDataWidth" value="32" />
 <parameter name="avalonMMAvalonSTDataWidth" value="32" />
 <parameter name="bitsPerSymbol" value="8" />
 <parameter name="channelWidth" value="0" />
 <parameter name="errorWidth" value="0" />
 <parameter name="fifoDepth" value="512" />
 <parameter name="fifoInputInterfaceOptions" value="AVALONST_SINK" />
 <parameter name="fifoOutputInterfaceOptions" value="AVALONMM_READ" />
 <parameter name="showHiddenFeatures" value="false" />
 <parameter name="singleClockMode" value="true" />
 <parameter name="singleResetMode" value="false" />
 <parameter name="symbolsPerBeat" value="4" />
 <parameter name="useBackpressure" value="false" />
 <parameter name="useIRQ" value="true" />
 <parameter name="usePacket" value="true" />
 <parameter name="useReadControl" value="false" />
 <parameter name="useRegister" value="false" />
 <parameter name="useWriteControl" value="false" />
 <parameter name="deviceFamilyString" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_dc_fifo"
  version="13.1"
  enabled="1"
  name="tx_dc_fifo">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_IN_FILL_LEVEL" value="0" />
 <parameter name="USE_OUT_FILL_LEVEL" value="0" />
 <parameter name="WR_SYNC_DEPTH" value="2" />
 <parameter name="RD_SYNC_DEPTH" value="2" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_dc_fifo"
  version="13.1"
  enabled="1"
  name="rx_dc_fifo">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_IN_FILL_LEVEL" value="0" />
 <parameter name="USE_OUT_FILL_LEVEL" value="0" />
 <parameter name="WR_SYNC_DEPTH" value="2" />
 <parameter name="RD_SYNC_DEPTH" value="2" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_eth_10g_mac"
  version="13.1"
  enabled="0"
  name="eth_10g_mac">
 <parameter name="ENABLE_TIMESTAMPING" value="0" />
 <parameter name="ENABLE_PTP_1STEP" value="0" />
 <parameter name="TSTAMP_FP_WIDTH" value="4" />
 <parameter name="PREAMBLE_PASSTHROUGH" value="1" />
 <parameter name="ENABLE_PFC" value="0" />
 <parameter name="PFC_PRIORITY_NUM" value="8" />
 <parameter name="DATAPATH_OPTION" value="3" />
 <parameter name="ENABLE_SUPP_ADDR" value="0" />
 <parameter name="INSTANTIATE_TX_CRC" value="0" />
 <parameter name="INSTANTIATE_STATISTICS" value="0" />
 <parameter name="REGISTER_BASED_STATISTICS" value="0" />
 <parameter name="ENABLE_1G10G_MAC" value="0" />
 <parameter name="DEVICE_FAMILY" value="Cyclone V" />
 <parameter name="AUTO_DEVICE" value="5CSXFC6D6F31C8ES" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="0"
  name="tx_oc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="4" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="0"
  name="tx_dc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<module
  kind="altera_avalon_sc_fifo"
  version="13.1"
  enabled="0"
  name="rx_dc_fifo_loopback">
 <parameter name="SYMBOLS_PER_BEAT" value="8" />
 <parameter name="BITS_PER_SYMBOL" value="8" />
 <parameter name="FIFO_DEPTH" value="512" />
 <parameter name="CHANNEL_WIDTH" value="0" />
 <parameter name="ERROR_WIDTH" value="0" />
 <parameter name="USE_PACKETS" value="1" />
 <parameter name="USE_FILL_LEVEL" value="0" />
 <parameter name="EMPTY_LATENCY" value="3" />
 <parameter name="USE_MEMORY_BLOCKS" value="1" />
 <parameter name="USE_STORE_FORWARD" value="0" />
 <parameter name="USE_ALMOST_FULL_IF" value="0" />
 <parameter name="USE_ALMOST_EMPTY_IF" value="0" />
 <parameter name="ENABLE_EXPLICIT_MAXCHANNEL" value="false" />
 <parameter name="EXPLICIT_MAXCHANNEL" value="0" />
 <parameter name="AUTO_DEVICE_FAMILY" value="Cyclone V" />
</module>
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.h2f_axi_clock" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.f2h_axi_clock" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="hps_0.h2f_lw_axi_clock" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="master_0.clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="master_0.clk_reset" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="button_pio.clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="button_pio.reset" />
<connection
  kind="avalon"
  version="13.1"
  start="hps_0.h2f_lw_axi_master"
  end="button_pio.s1">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0000" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="avalon"
  version="13.1"
  start="master_0.master"
  end="button_pio.s1">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0000" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="tx_dc_fifo.out_clk" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="tx_dc_fifo.in_clk_reset" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="tx_dc_fifo.out_clk_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="rx_dc_fifo.in_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="rx_dc_fifo.in_clk_reset" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="rx_dc_fifo.out_clk_reset" />
<connection
  kind="avalon"
  version="13.1"
  start="master_0.master"
  end="rx_oc_fifo.out">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0028" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="avalon"
  version="13.1"
  start="master_0.master"
  end="tx_oc_fifo.in">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0020" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.csr_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.csr_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.tx_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.tx_reset" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="eth_10g_mac.rx_clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="eth_10g_mac.rx_reset" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="tx_oc_fifo.clk_in" />
<connection
  kind="reset"
  version="13.1"
  start="clk_0.clk_reset"
  end="tx_oc_fifo.reset_in" />
<connection
  kind="avalon"
  version="13.1"
  start="hps_0.h2f_lw_axi_master"
  end="tx_oc_fifo.in">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0020" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="rx_dc_fifo.out_clk_reset" />
<connection
  kind="avalon"
  version="13.1"
  start="hps_0.h2f_lw_axi_master"
  end="rx_oc_fifo.out">
 <parameter name="arbitrationPriority" value="1" />
 <parameter name="baseAddress" value="0x0028" />
 <parameter name="defaultConnection" value="false" />
</connection>
<connection kind="clock" version="13.1" start="clk_0.clk" end="rx_oc_fifo.clk_in" />
<connection kind="clock" version="13.1" start="clk_0.clk" end="tx_dc_fifo.in_clk" />
<connection
  kind="clock"
  version="13.1"
  start="clk_0.clk"
  end="rx_dc_fifo.out_clk" />
<connection
  kind="clock"
  version="13.1"
  start="xaui_clk.clk"
  end="tx_dc_fifo_loopback.clk" />
<connection
  kind="reset"
  version="13.1"
  start="xaui_clk.clk_reset"
  end="tx_dc_fifo_loopback.clk_reset" />
 <connection
   kind="reset"
   version="13.1"
   start="clk_0.clk_reset"
   end="rx_oc_fifo.reset_in" />
 <connection
   kind="clock"
   version="13.1"
   start="clk_0.clk"
   end="tx_oc_fifo_loopback.clk" />
 <connection
   kind="reset"
   version="13.1"
   start="clk_0.clk_reset"
   end="tx_oc_fifo_loopback.clk_reset" />
 <connection
   kind="avalon_streaming"
   version="13.1"
   start="tx_oc_fifo.out"
   end="tx_dc_fifo.in" />
 <connection
   kind="avalon_streaming"
   version="13.1"
   start="rx_dc_fifo.out"
   end="rx_oc_fifo.in" />
 <interconnectRequirement for="$system" name="qsys_mm.clockCrossingAdapter" value="HANDSHAKE" /
 <interconnectRequirement for="$system" name="qsys_mm.maxAdditionalLatency" value="1" />
 <interconnectRequirement for="$system" name="qsys_mm.insertDefaultSlave" value="false" />
</system>
```

5.3 Network Driver

Makefile

```
# Compiles blazepps_ethernet driver
#
# Authors
#
#  Christopher Campbell (cc3769@columbia.edu)
# Valeh Valiollahpour Amiri (vv2252@columbia.edu)
#
# Last Modified
#
#  April 23rd, 2015

ifneq (${KERNELRELEASE},)

# KERNELRELEASE defined: we are being compiled as part of the Kernel
obj-m := blazepps_ethernet.o

else

# We are being compiled as a module: use the Kernel build system
PWD := $(shell pwd)
KDIR := /usr/src/linux
DTC := $(KDIR)/scripts/dtc/dtc

default: module tree

module:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} modules

tree: socfpga.dts
$(DTC) -O dtb -o socfpga.dtb socfpga.dts

clean:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} clean
${RM} socfpga.dtb test_client.o test_server.o *~

endif
```

socfpga.dts

```
/*
 *  Copyright (C) 2012 Altera Corporation <www.altera.com>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 * dtc -O dtb -o socfpga.dtb socfpga.dts
 */

/dts-v1/;
/include/ "socfpga.dtsi"

/ {
model = "Altera SOCFPGA Cyclone V";
compatible = "altr,socfpga-cyclone5", "altr,socfpga";

chosen {
bootargs = "console=ttyS0,57600";
};

memory {
name = "memory";
device_type = "memory";
reg = <0x0 0x40000000>; /* 1 GB */
};

aliases {
/* this allow the ethaddr uboot environmnet variable contents
 * to be added to the gmac1 device tree blob.
 */
ethernet0 = &gmac1;
};

soc {
clkmgr@ffd04000 {
clocks {
osc1 {
clock-frequency = <25000000>;
};
};
};
```

```
dcan0: d_can@ffc00000 {
status = "disabled";
};

dcan1: d_can@ffc10000 {
status = "disabled";
};

dwmmc0@ff704000 {
num-slots = <1>;
supports-highspeed;
broken-cd;
altr,dw-mshc-ciu-div = <4>;
altr,dw-mshc-sdr-timing = <0 3>;

slot@0 {
reg = <0>;
bus-width = <4>;
};
};

ethernet@ff700000 {
status = "disabled";
};

ethernet@ff702000 {
phy-mode = "rgmii";
phy-addr = <0xffffffff>; /* probe for phy addr */
};

i2c1: i2c@ffc05000 {
status = "disabled";
};

i2c2: i2c@ffc06000 {
status = "disabled";
};

i2c3: i2c@ffc07000 {
status = "disabled";
};

qspi: spi@ff705000 {
compatible = "cadence,qspi";
#address-cells = <1>;
#size-cells = <0>;
reg = <0xff705000 0x1000>,
<0xffa00000 0x1000>;
```

```
interrupts = <0 151 4>;
master-ref-clk = <400000000>;
ext-decoder = <0>;  /* external decoder */
num-chipselect = <4>;
fifo-depth = <128>;
bus-num = <2>;

flash0: n25q00@0 {
#address-cells = <1>;
#size-cells = <1>;
compatible = "n25q00";
reg = <0>; /* chip select */
spi-max-frequency = <100000000>;
page-size = <256>;
block-size = <16>; /* 2^16, 64KB */
quad = <1>;    /* 1-support quad */
tshsl-ns = <200>;
tsd2d-ns = <255>;
tchsh-ns = <20>;
tslch-ns = <20>;

partition@0 {
/* 8MB for raw data. */
label = "Flash 0 Raw Data";
reg = <0x0 0x800000>;
};
partition@800000 {
/* 8MB for jffs2 data. */
label = "Flash 0 jffs2 Filesystem";
reg = <0x800000 0x800000>;
};
};

};

sysmgr@ffd08000 {
cpu1-start-addr = <0xffd080c4>;
};

timer0@ffc08000 {
clock-frequency = <100000000>;
};

timer1@ffc09000 {
clock-frequency = <100000000>;
};

timer2@ffd00000 {
```

```
clock-frequency = <25000000>;
};

timer3@ffd01000 {
clock-frequency = <25000000>;
};

serial0@ffc02000 {
clock-frequency = <100000000>;
};

serial1@ffc03000 {
clock-frequency = <100000000>;
};

usb0: usb@ffb00000 {
status = "disabled";
};

usb1: usb@ffb40000 {
ulpi-ddr = <0>;
};

i2c0: i2c@ffc04000 {
speed-mode = <0>;
};

leds {
compatible = "gpio-leds";
hps0 {
label = "hps_led0";
gpios = <&gpio1 15 1>;
};

hps1 {
label = "hps_led1";
gpios = <&gpio1 14 1>;
};

hps2 {
label = "hps_led2";
gpios = <&gpio1 13 1>;
};

hps3 {
label = "hps_led3";
gpios = <&gpio1 12 1>;
};
```

```
};

lightweight_bridge: bridge@0xff200000 {
#address-cells = <1>;
#size-cells = <1>;
ranges = < 0x0 0xff200000 0x200000 >;

compatible = "simple-bus";

blazepps_ethernet: blazepps_ethernet@20{
 compatible = "altr,blazepps_ethernet";
 reg = < 0x20 0x8 0x28 0x8>;
 reg-names = "tx_fifo", "rx_fifo";
 local-mac-address = [ 48 40 48 40 48 40 ];
};
};
};
};

&i2c0 {
lcd: lcd@28 {
compatible = "newhaven,nhd-0216k3z-nsw-bbw";
reg = <0x28>;
height = <2>;
width = <16>;
brightness = <8>;
};

eeprom@51 {
compatible = "atmel,24c32";
reg = <0x51>;
pagesize = <32>;
};

rtc@68 {
compatible = "dallas,ds1339";
reg = <0x68>;
};
};
```

blazepps_ethernet.h

```
/* BlazePPS Ethernet Driver
 * Copyright (C) 2015 Valeh Valiollahpour Amiri, Christopher Campbell, Sheng Qian.
 *
 * Contributors:
 * Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 * Christopher Campbell (cc3769@columbia.edu)
 * Sheng Qian (sq2168@columbia.edu)
 *
 * References:
 *   snull.c
 *  altera_tse_main.c
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 * Last Modified: Math 12th, 2015
 */

#ifndef _BLAZEPPS_ETHERNET_H
#define _BLAZEPPS_ETHERNET_H

/* Default packet pool size (can also be defined at module load time) */
#define BLAZE_POOL_SIZE 8

/* Default timeout period (can also be defined at module load time) */
#define BLAZE_TIMEOUT 5

#endif /* _BLAZEPPS_ETHERNET_H_ */
```

blazepps_ethernet.c

```
/* BlazePPS Ethernet Driver
 * Copyright (C) 2015 Valeh Valiollahpour Amiri, Christopher Campbell, Sheng Qian.
 *
 * Contributors:
 *  Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 *  Christopher Campbell (cc3769@columbia.edu)
 *  Sheng Qian (sq2168@columbia.edu)
 *
 * Referenes:
 *  snull.c
 *  altera_tse_main.c
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 * Last Modified: Math 12th, 2015
 */

#include <linux/types.h>
#include <linux/errno.h>
#include <linux/kernel.h>
#include <linux/interrupt.h>
#include <linux/module.h>
#include <linux/moduleparam.h>
#include <linux/netdevice.h>
#include <linux/etherdevice.h>
#include <linux/platform_device.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/io.h>
#include <linux/of.h>
#include <linux/of_net.h>
#include <linux/skbuff.h>
#include <asm/checksum.h>
#include "blazepps_ethernet.h"

#define SOP 0x1
#define EOP 0x2
#define EMPTY 0x12
#define CSR_OFF 0x4
#define BLAZE_RX_INTR 0x0001
#define BLAZE_TX_INTR 0x0002
#define DRIVER_NAME "blazepps_ethernet"

#define TXRX

#ifdef TXRX
#define PTXRX(fmt, args...) printk( KERN_NOTICE fmt, ## args)
#else
#define PTXRX(fmt, args...)
#endif
```

```
#define DEBUG

#ifdef DEBUG
#define PDEBUG(fmt, args...) printk( KERN_DEBUG "blazepps_ethernet: " fmt, ## args)
#else
#define PDEBUG(fmt, args...)
#endif

/* Module Parameters (exposed in sysfs) */
static int pool_size = BLAZE_POOL_SIZE;
module_param(pool_size, int, S_IRUGO);

static int timeout = BLAZE_TIMEOUT;
module_param(timeout, int, S_IRUGO);

/* Hardware Tx FIFO */
struct hw_tx_fifo {
u32 write_interface;
};

/* Hardware Rx FIFO */
struct hw_rx_fifo {
u32 read_interface;
};

/* In-flight packet */
struct blaze_packet {
u8 data[ETH_DATA_LEN];
int datalen;
struct blaze_packet *next;
struct net_device *dev;
};

/* Private data */
struct blaze_priv {
int status;
int tx_packetlen;
int en_rx_interrupt;
struct device *device;
struct net_device *dev;
struct sk_buff *skb;
struct blaze_packet *pkt_pool;
struct blaze_packet *rx_queue;
struct net_device_stats stats;
struct hw_tx_fifo __iomem *tx_dev;
struct hw_rx_fifo __iomem *rx_dev;
spinlock_t lock;
```

```
};

/* Setup packet pool */
void setup_pool(struct net_device *dev)
{
int i;
struct blaze_packet *pkt;
struct blaze_priv *priv = netdev_priv(dev);

priv->pkt_pool = NULL;
for (i = 0; i < pool_size; i++) {
pkt = kmalloc(sizeof(struct blaze_packet), GFP_KERNEL);
if (!pkt) {
PDEBUG("ran out of memory for packet pool\n");
return;
}

pkt->dev = dev;
pkt->next = priv->pkt_pool;
priv->pkt_pool = pkt;
}
}

/* Teardown packet pool */
void teardown_pool(struct net_device *dev)
{
struct blaze_packet *pkt;
struct blaze_priv *priv = netdev_priv(dev);

pkt = priv->pkt_pool;
while (pkt) {
priv->pkt_pool = pkt->next;
kfree(pkt);
pkt = priv->pkt_pool;
}
}

/* Get packet from packet pool */
struct blaze_packet *get_pool_pkt(struct net_device *dev)
{
unsigned long flags;
struct blaze_packet *pkt;
struct blaze_priv *priv = netdev_priv(dev);

spin_lock_irqsave(&priv->lock, flags);
pkt = priv->pkt_pool;
priv->pkt_pool = pkt->next;
if (!priv->pkt_pool) {
```

```
PDEBUG("packet pool empty\n");
netif_stop_queue(dev);
}
spin_unlock_irqrestore(&priv->lock, flags);

return pkt;
}

/* Return packet to packet pool */
void return_pool_pkt(struct blaze_packet *pkt)
{
unsigned long flags;
struct blaze_priv *priv = netdev_priv(pkt->dev);

spin_lock_irqsave(&priv->lock, flags);
pkt->next = priv->pkt_pool;
priv->pkt_pool = pkt;
spin_unlock_irqrestore(&priv->lock, flags);
if (netif_queue_stopped(pkt->dev) && !pkt->next)
netif_wake_queue(pkt->dev);
}

/* Enqueue rxed packet */
void enqueue_rx(struct net_device *dev, struct blaze_packet *pkt)
{
unsigned long flags;
struct blaze_packet *rx_queue_pkt;
struct blaze_priv *priv = netdev_priv(dev);

pkt->next = NULL;
spin_lock_irqsave(&priv->lock, flags);
if (!priv->rx_queue) {
priv->rx_queue = pkt;
} else {
rx_queue_pkt = priv->rx_queue;
while (rx_queue_pkt->next) {
rx_queue_pkt = rx_queue_pkt->next;
}
rx_queue_pkt->next = pkt;
}
spin_unlock_irqrestore(&priv->lock, flags);
}

/* Enable/disable rx interrupts */
static void rx_ints(struct net_device *dev, int en)
{
struct blaze_priv *priv = netdev_priv(dev);
```

```
priv->en_rx_interrupt = en;
}

/* Open device (called by kernel) */
int blaze_open(struct net_device *dev)
{
netif_start_queue(dev);

return 0;
}

/* Stop tx queue (called by kernel) */
int blaze_release(struct net_device *dev)
{
netif_stop_queue(dev);

return 0;
}

/* Receive a packet */
void passup_rx(struct net_device *dev, struct blaze_packet *pkt)
{
struct sk_buff *skb;
struct blaze_priv *priv = netdev_priv(dev);

skb = dev_alloc_skb(pkt->datalen + 2);
if (!skb) {
PDEBUG("low memory, packet dropped\n");
priv->stats.rx_dropped++;
goto out;
}
skb_reserve(skb, 2);
memcpy(skb_put(skb, pkt->datalen), pkt->data, pkt->datalen);

/* Write metadata and pass to upper level */
skb->dev = dev;
skb->protocol = eth_type_trans(skb, dev);
skb->ip_summed = CHECKSUM_UNNECESSARY;
priv->stats.rx_packets++;
priv->stats.rx_bytes += pkt->datalen;
netif_rx(skb);
 out:
return;
}

/* Receive packet from hardware */
void hw_rx(struct net_device *dev)
{
```

```
int i;
        u32 data[256];
        int num_bytes;
struct blaze_packet *pkt = NULL;
        struct blaze_priv *priv = netdev_priv(dev);

        PTXRX("\n");

i = 0;
        num_bytes = 0;
        while (i < 15) {
                num_bytes += 4;
                data[4 * i] = ioread32(&priv->rx_dev->read_interface);
                PTXRX("rx: 0x%08x\n", data[4 * i]);
i++;
        }

/* Create packet */
pkt = get_pool_pkt(dev);
pkt->datalen = num_bytes;
memcpy(pkt->data, data, num_bytes);

/* Enqueue packet */
enqueue_rx(dev, pkt);
}

/* Typical interrupt handler */
static void blaze_interrupt(int irq, void *dev_id, struct pt_regs *regs)
{
int statusword;
struct blaze_priv *priv;
struct blaze_packet *pkt = NULL;
struct net_device *dev = (struct net_device *)dev_id;

if (!dev)
return;

/* Receive packet from hardware and put it in rx queue */
priv = netdev_priv(dev);
statusword = priv->status;
if (statusword & BLAZE_RX_INTR) {
hw_rx(dev);
}

spin_lock(&priv->lock);
priv->status = 0;
/* Get packet from queue and pass it to upper layers */
if (statusword & BLAZE_RX_INTR) {
```

```
pkt = priv->rx_queue;
if (pkt) {
/* Dequeue packet */
priv->rx_queue = pkt->next;

/* Pass to upper layers */
passup_rx(dev, pkt);
}
}
/* Tx successful */
if (statusword & BLAZE_TX_INTR) {
priv->stats.tx_packets++;
priv->stats.tx_bytes += priv->tx_packetlen;
dev_kfree_skb(priv->skb);
}
spin_unlock(&priv->lock);

if (statusword & BLAZE_RX_INTR)
if (pkt)
return_pool_pkt(pkt);
}

/* Transmit a packet through hardware */
static void hw_tx(char *buf, int len, struct net_device *dev)
{
int i;
int tmp_len;
int remainder_bytes;
u32 buf32;
struct blaze_priv *priv;

PTXRX("\n");

if (len < sizeof(struct ethhdr) + sizeof(struct iphdr)) {
PDEBUG("packet too short\n");
return;
}

priv = netdev_priv(dev);
remainder_bytes = len % 4;
if (remainder_bytes)
tmp_len = len;
else
tmp_len = len - 4;
iowrite32(SOP, &priv->tx_dev->write_interface + CSR_OFF);
for (i = 0; i < (tmp_len / 4); i++) {
memcpy(&buf32, &buf[4 * i], sizeof(u32));
iowrite32(buf32, &priv->tx_dev->write_interface);
```

```
PTXRX("tx: 0x%08x\n", buf32);
}
iowrite32(remainder_bytes, &priv->tx_dev->write_interface + CSR_OFF);
iowrite32(EOP, &priv->tx_dev->write_interface + CSR_OFF);
memcpy(&buf32, &buf[4 * i], sizeof(u32));
iowrite32(buf32, &priv->tx_dev->write_interface);
PTXRX("tx: 0x%08x\n", buf32);

/* Simulate rx interrupt */
if (priv->en_rx_interrupt) {
priv->status |= BLAZE_RX_INTR;
blaze_interrupt(0, dev, NULL);
}

/* Simulate tx interrupt */
priv->status |= BLAZE_TX_INTR;
blaze_interrupt(0, dev, NULL);
}

/* Transmit a packet (called by the kernel) */
int blaze_tx(struct sk_buff *skb, struct net_device *dev)
{
int len;
char *data, shortpkt[ETH_ZLEN];
struct blaze_priv *priv = netdev_priv(dev);

data = skb->data;
len = skb->len;
if (len < ETH_ZLEN) {
memset(shortpkt, 0, ETH_ZLEN);
memcpy(shortpkt, skb->data, skb->len);
len = ETH_ZLEN;
data = shortpkt;
}

dev->trans_start = jiffies;
priv->skb = skb;

/* Transmit data through hardware */
hw_tx(data, len, dev);

return 0;
}

/* Handle tx timeout (called by kernel) */
void blaze_tx_timeout(struct net_device *dev)
{
struct blaze_priv *priv = netdev_priv(dev);
```

```
PDEBUG("Transmit timeout at %ld, latency %ld\n", jiffies,
       jiffies - dev->trans_start);

/* Simulate a transmission interrupt so we don’t have to wait */
priv->status = BLAZE_TX_INTR;
blaze_interrupt(0, dev, NULL);
priv->stats.tx_errors++;
netif_wake_queue(dev);
}

/* Return stats (called by kernel) */
struct net_device_stats *blaze_stats(struct net_device *dev)
{
struct blaze_priv *priv = netdev_priv(dev);

return &priv->stats;
}

static const struct net_device_ops blaze_netdev_ops = {
.ndo_open = blaze_open,
.ndo_stop = blaze_release,
.ndo_start_xmit = blaze_tx,
.ndo_tx_timeout = blaze_tx_timeout,
.ndo_set_mac_address = eth_mac_addr,
.ndo_get_stats = blaze_stats,
};

/* Gets device resources */
static int get_resource(struct platform_device *pdev, const char *name,
struct resource **res, void __iomem **ptr)
{
struct resource *region;
struct device *device = &pdev->dev;

*res = platform_get_resource_byname(pdev, IORESOURCE_MEM, name);
if (*res == NULL) {
dev_err(device, "resource %s not defined\n", name);
return -ENODEV;
}

region =
    devm_request_mem_region(device, (*res)->start, resource_size(*res),
    dev_name(device));
if (region == NULL) {
dev_err(device, "unable to request %s\n", name);
return -EBUSY;
}
```

```
*ptr =
    devm_ioremap_nocache(device, region->start, resource_size(region));
if (*ptr == NULL) {
dev_err(device, "ioremap_nocache of %s failed!", name);
return -ENOMEM;
}

return 0;
}

/* Probes for device */
int blazepps_ethernet_probe(struct platform_device *pdev)
{
int result;
int ret = -ENODEV;
const unsigned char *macaddr;
struct net_device *ndev;
struct blaze_priv *priv;
struct resource *tx_fifo, *rx_fifo;

/* Allocate network device */
ndev = alloc_etherdev(sizeof(struct blaze_priv));
if (!ndev) {
PDEBUG("unabled to allocate network device\n");
return ret;
}

/* Set device custom options */
ndev->watchdog_timeo = timeout;
ndev->netdev_ops = &blaze_netdev_ops;
ndev->features |= NETIF_F_HW_CSUM;
priv = netdev_priv(ndev);

/* Initialize network device */
macaddr = of_get_mac_address(pdev->dev.of_node);
if (macaddr)
memcpy(ndev->dev_addr, macaddr, ETH_ALEN);
else
eth_hw_addr_random(ndev);
memset(priv, 0, sizeof(struct blaze_priv));
spin_lock_init(&priv->lock);
rx_ints(ndev, 1);
setup_pool(ndev);
SET_NETDEV_DEV(ndev, &pdev->dev);
priv->device = &pdev->dev;
priv->dev = ndev;
```

```
/* Get device resources */
ret = get_resource(pdev, "tx_fifo", &tx_fifo,
(void __iomem **)&priv->tx_dev);
if (ret) {
PDEBUG("unable to get tx_fifo resource\n");
goto out;
}
ret = get_resource(pdev, "rx_fifo", &rx_fifo,
(void __iomem **)&priv->rx_dev);
if (ret) {
PDEBUG("unable to get rx_fifo resource\n");
goto out;
}

/* Register device */
ret = -ENODEV;
result = register_netdev(ndev);
if (result) {
PDEBUG("unable to register network device\n");
goto out;
} else {
ret = 0;
}

platform_set_drvdata(pdev, ndev);

 out:
if (ret) {
unregister_netdev(ndev);
teardown_pool(ndev);
free_netdev(ndev);
}

return ret;
}

/* Remove device and resources */
static int blazepps_ethernet_remove(struct platform_device *pdev)
{
struct net_device *ndev = platform_get_drvdata(pdev);

platform_set_drvdata(pdev, NULL);
unregister_netdev(ndev);
teardown_pool(ndev);
free_netdev(ndev);

return 0;
}
```

```
#ifdef CONFIG_OF
static const struct of_device_id blazepps_ethernet_of_match[] = {
{.compatible = "altr,blazepps_ethernet"},
{},
};

MODULE_DEVICE_TABLE(of, blazepps_ethernet_of_match);
#endif

static struct platform_driver blazepps_ethernet_driver = {
.probe = blazepps_ethernet_probe,
.remove = blazepps_ethernet_remove,
.suspend = NULL,
.resume = NULL,
.driver = {
   .name = DRIVER_NAME,
   .of_match_table = of_match_ptr(blazepps_ethernet_of_match),},
};

module_platform_driver(blazepps_ethernet_driver);

MODULE_DESCRIPTION("BlazePPS Ethernet Driver");
MODULE_AUTHOR("BlazePPS Corporation");
MODULE_LICENSE("GPL v2");
```

load.sh

```
#!/bin/bash

insmod ./blazepps_ethernet.ko
ifconfig eth1 192.168.2.1
```

unload.sh

```
#!/bin/bash

ifconfig eth1 down
rmmod blazepps_ethernet.ko
```

test.sh

```
!# /bin/bash

ping -I eth1 -c 1 192.168.1.2
```

5.4 MM Driver

Makefile

```
# Compiles txrx_subsys driver and txrx_test test program
#
# Authors
#
#  Christopher Campbell (cc3769@columbia.edu)
# Valeh Valiollahpour Amiri (vv2252@columbia.edu)
#
# Last Modified
#
#  April 11th, 2015

ifneq (${KERNELRELEASE},)

# KERNELRELEASE defined: we are being compiled as part of the Kernel
obj-m := txrx_subsys.o

else

# We are being compiled as a module: use the Kernel build system
PWD := $(shell pwd)
KDIR := /usr/src/linux
ARCH := arm
KDIR_CROSS := /home/cc3769/ES_Project/linux-socfpga
CROSS_COMP := /usr/local/DS-5/bin/arm-linux-gnueabihf-
DTC := $(KDIR)/scripts/dtc/dtc

default: module txrx_test socfpga.dtb

module:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} modules

txrx_test: txrx_test.o
gcc -o txrx_test txrx_test.o

txrx_test.o: txrx_test.c
gcc -c txrx_test.c

socfpga.dtb: socfpga.dts
$(DTC) -O dtb -o socfpga.dtb socfpga.dts

clean:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} clean
${RM} txrx_test socfpga.dtb *~
```

```
endif
```

socfpga.dts

```
/*
 *  Copyright (C) 2012 Altera Corporation <www.altera.com>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 * dtc -O dtb -o socfpga.dtb socfpga.dts
 */

/dts-v1/;
/include/ "socfpga.dtsi"

/ {
model = "Altera SOCFPGA Cyclone V";
compatible = "altr,socfpga-cyclone5", "altr,socfpga";

chosen {
bootargs = "console=ttyS0,57600";
};

memory {
name = "memory";
device_type = "memory";
reg = <0x0 0x40000000>; /* 1 GB */
};

aliases {
/* this allow the ethaddr uboot environmnet variable contents
 * to be added to the gmac1 device tree blob.
 */
ethernet0 = &gmac1;
};

soc {
```

```
clkmgr@ffd04000 {
clocks {
osc1 {
clock-frequency = <25000000>;
};
};
};

dcan0: d_can@ffc00000 {
status = "disabled";
};

dcan1: d_can@ffc10000 {
status = "disabled";
};

dwmmc0@ff704000 {
num-slots = <1>;
supports-highspeed;
broken-cd;
altr,dw-mshc-ciu-div = <4>;
altr,dw-mshc-sdr-timing = <0 3>;

slot@0 {
reg = <0>;
bus-width = <4>;
};
};

ethernet@ff700000 {
status = "disabled";
};

ethernet@ff702000 {
phy-mode = "rgmii";
phy-addr = <0xffffffff>; /* probe for phy addr */
};

i2c1: i2c@ffc05000 {
status = "disabled";
};

i2c2: i2c@ffc06000 {
status = "disabled";
};

i2c3: i2c@ffc07000 {
status = "disabled";
```

```
};

qspi: spi@ff705000 {
compatible = "cadence,qspi";
#address-cells = <1>;
#size-cells = <0>;
reg = <0xff705000 0x1000>,
<0xffa00000 0x1000>;
interrupts = <0 151 4>;
master-ref-clk = <400000000>;
ext-decoder = <0>;  /* external decoder */
num-chipselect = <4>;
fifo-depth = <128>;
bus-num = <2>;

flash0: n25q00@0 {
#address-cells = <1>;
#size-cells = <1>;
compatible = "n25q00";
reg = <0>; /* chip select */
spi-max-frequency = <100000000>;
page-size = <256>;
block-size = <16>; /* 2^16, 64KB */
quad = <1>;    /* 1-support quad */
tshsl-ns = <200>;
tsd2d-ns = <255>;
tchsh-ns = <20>;
tslch-ns = <20>;

partition@0 {
/* 8MB for raw data. */
label = "Flash 0 Raw Data";
reg = <0x0 0x800000>;
};
partition@800000 {
/* 8MB for jffs2 data. */
label = "Flash 0 jffs2 Filesystem";
reg = <0x800000 0x800000>;
};
};

};

sysmgr@ffd08000 {
cpu1-start-addr = <0xffd080c4>;
};

timer0@ffc08000 {
```

```
clock-frequency = <100000000>;
};

timer1@ffc09000 {
clock-frequency = <100000000>;
};

timer2@ffd00000 {
clock-frequency = <25000000>;
};

timer3@ffd01000 {
clock-frequency = <25000000>;
};

serial0@ffc02000 {
clock-frequency = <100000000>;
};

serial1@ffc03000 {
clock-frequency = <100000000>;
};

usb0: usb@ffb00000 {
status = "disabled";
};

usb1: usb@ffb40000 {
ulpi-ddr = <0>;
};

i2c0: i2c@ffc04000 {
speed-mode = <0>;
};

leds {
compatible = "gpio-leds";
hps0 {
label = "hps_led0";
gpios = <&gpio1 15 1>;
};

hps1 {
label = "hps_led1";
gpios = <&gpio1 14 1>;
};

hps2 {
```

```
label = "hps_led2";
gpios = <&gpio1 13 1>;
};

hps3 {
label = "hps_led3";
gpios = <&gpio1 12 1>;
};
};

lightweight_bridge: bridge@0xff200000 {
#address-cells = <1>;
#size-cells = <1>;
ranges = < 0x0 0xff200000 0x200000 >;

compatible = "simple-bus";

txrx_subsys: txrx_subsys@20{
 compatible = "altr,txrx_subsys";
 reg = < 0x20 0x8 0x28 0x8>;
};
};
};
};

&i2c0 {
lcd: lcd@28 {
compatible = "newhaven,nhd-0216k3z-nsw-bbw";
reg = <0x28>;
height = <2>;
width = <16>;
brightness = <8>;
};

eeprom@51 {
compatible = "atmel,24c32";
reg = <0x51>;
pagesize = <32>;
};

rtc@68 {
compatible = "dallas,ds1339";
reg = <0x68>;
};
};
```

txrx_subsys.h

```
#ifndef _TXRX_SUBSYS_H
#define _TXRX_SUBSYS_H

#include <linux/types.h>
#include <linux/ioctl.h>

#define TXRX_MAGIC 'q'

#define TX_DATA _IOW(TXRX_MAGIC,  1, unsigned int *)
#define RX_DATA _IOR(TXRX_MAGIC, 2, unsigned int *)
#define CONTROL _IOW(TXRX_MAGIC, 3, unsigned int *)
#define STATUS  _IOR(TXRX_MAGIC, 4, unsigned int *)
#define PIO_CLR _IOW(TXRX_MAGIC,  5, unsigned int *)
#define PIO_SET _IOW(TXRX_MAGIC,  6, unsigned int *)

#endif
```

txrx_subsys.c

```
/*
 * Device driver for interacting with a packet processors Tx/Rx FIFOs
 *
 * A Platform device implemented using the misc subsystem
 *
 * Authors:
 *
 * Christopher Campbell (cc3769@columbia.edu)
 * Sheng Qian (sq2168@columbia.edu)
 * Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 *
 * Last Modified:
 *
 * April 11th, 2015
 *
 * Build Instructions:
 *
 * "make" to build
 * insmod txrx_subsys.ko
 *
 * Check code style with
 * checkpatch.pl --file --no-tree txrx_fifo.c
 */

#include <linux/module.h>
#include <linux/init.h>
#include <linux/errno.h>
#include <linux/version.h>
#include <linux/kernel.h>
```

```
#include <linux/platform_device.h>
#include <linux/miscdevice.h>
#include <linux/slab.h>
#include <linux/io.h>
#include <linux/of.h>
#include <linux/of_address.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include "txrx_subsys.h"

#define DRIVER_NAME "txrx_subsys"
#define CLR_OFFSET 0x14
#define SET_OFFSET 0x10
#define CSR_OFFSET 0x4

//#define PIO

/*
 * Information about our device
 */
struct txrx_dev {
struct resource res_rx, res_tx, res_pio;
void __iomem *virtbase_rx, *virtbase_tx;
#ifdef PIO
void __iomem *virtbase_pio;
#endif
u32 tx_data;
u32 rx_data;
        u32 csr_data;
#ifdef PIO
u32 pio_clr;
u32 pio_set;
#endif
} dev;

/*
 * Tx
 *
 * Writes data to Tx FIFO
 */
static void tx(u32 tx_data)
{
iowrite32(tx_data, dev.virtbase_tx);
dev.tx_data = tx_data;
}

/*
 * Rx
 *
 * Reads data from Rx FIFO
 */
static void rx(void)
{
dev.rx_data = ioread32(dev.virtbase_rx);
}

/*
 * Control
 *
 * Write to control & status registers
 */
static void control(u32 csr_data)
{
iowrite32(csr_data, dev.virtbase_tx+CSR_OFFSET);
dev.csr_data = csr_data;
}

/*
 * Status
 *
 * Read control & status registers
 */
static void status(void)
{
dev.csr_data = ioread32(dev.virtbase_rx+CSR_OFFSET);
}

#ifdef PIO
/*
 * Pio clear and set
 *
 * Writes clear and set values in the pio register
 * (clr_set==0) => pio clear
 * (clr_set==1) => pio set
 *
 */
static void pio_clr_set(u32 value, u8 clr_set)
{
if (clr_set==0) {
iowrite32(value, dev.virtbase_pio + CLR_OFFSET);
dev.pio_clr = value;
}
if (clr_set==1) {
iowrite32(value, dev.virtbase_pio + SET_OFFSET);
dev.pio_set = value;
}
```

```
}
#endif

/*
 * Handle ioctl() calls from userspace:
 *
 * write data to Tx FIFO - read data from Rx FIFO
 */
static long txrx_ioctl(struct file *f, unsigned int cmd, unsigned long arg)
{
unsigned int txrx;
unsigned int csr;
//unsigned int pio_clr, pio_set;

switch (cmd) {
case (TX_DATA):
if (copy_from_user(&txrx, (unsigned int *) arg, sizeof(int)))
return -EACCES;
tx(txrx);
break;

case (RX_DATA):
rx();
txrx = dev.rx_data;
if (copy_to_user((unsigned int *) arg, &txrx, sizeof(int)))
return -EACCES;
break;

case (CONTROL):
if (copy_from_user(&csr, (unsigned int *) arg, sizeof(int)))
return -EACCES;
control(csr);
break;

case (STATUS):
status();
csr = dev.csr_data;
if (copy_to_user((unsigned int *) arg, &csr, sizeof(int)))
return -EACCES;
break;
#ifdef PIO
case (PIO_CLR):
if (copy_from_user(&pio_clr, (unsigned int *) arg, sizeof(int)))
return -EACCES;
pio_clr_set(pio_clr, 0);
break;

case (PIO_SET):
```

```
if (copy_from_user(&pio_set, (unsigned int *) arg, sizeof(int)))
return -EACCES;
pio_clr_set(pio_set, 1);
break;
#endif

default:
return -EINVAL;
}

return 0;
}

/* The operations our device knows how to do */
static const struct file_operations txrx_subsys_fops = {
.owner = THIS_MODULE,
.unlocked_ioctl = txrx_ioctl,
};

/* Information about our device for the "misc" framework -- like a char dev */
static struct miscdevice txrx_subsys_misc_device = {
.minor = MISC_DYNAMIC_MINOR,
.name = DRIVER_NAME,
.fops = &txrx_subsys_fops,
};

/*
 * Initialization code: get resources (registers)
 */
static int __init txrx_subsys_probe(struct platform_device *pdev)
{
int ret;

/* Register ourselves as a misc device: creates /dev/txrx_subsys */
ret = misc_register(&txrx_subsys_misc_device);

/* Rx registers */
ret = of_address_to_resource(pdev->dev.of_node, 0, &dev.res_rx);
if (ret) {
ret = -ENOENT;
goto out_deregister_rx;
}
if (request_mem_region(dev.res_rx.start, resource_size(&dev.res_rx),
       DRIVER_NAME) == NULL) {
ret = -EBUSY;
goto out_deregister_rx;
}
```

```
dev.virtbase_rx = of_iomap(pdev->dev.of_node, 0);
if (dev.virtbase_rx == NULL) {
ret = -ENOMEM;
goto out_release_rx;
}

/* Tx registers */
ret = of_address_to_resource(pdev->dev.of_node, 1, &dev.res_tx);
if (ret) {
ret = -ENOENT;
goto out_deregister_tx;
}
if (request_mem_region(dev.res_tx.start, resource_size(&dev.res_tx),
       DRIVER_NAME) == NULL) {
ret = -EBUSY;
goto out_deregister_tx;
}

dev.virtbase_tx = of_iomap(pdev->dev.of_node, 1);
if (dev.virtbase_tx == NULL) {
ret = -ENOMEM;
goto out_release_tx;
}

#ifdef PIO
/* Pio registers */
ret = of_address_to_resource(pdev->dev.of_node, 0, &dev.res_pio);
if (ret) {
ret = -ENOENT;
goto out_deregister_pio;
}
if (request_mem_region(dev.res_pio.start,
       resource_size(&dev.res_pio),
       DRIVER_NAME) == NULL) {
ret = -EBUSY;
goto out_deregister_pio;
}
dev.virtbase_pio = of_iomap(pdev->dev.of_node, 0);
if (dev.virtbase_pio == NULL) {
ret = -ENOMEM;
goto out_release_pio;
}
#endif

return 0;

#ifdef PIO
 out_release_pio:
```

```
release_mem_region(dev.res_pio.start, resource_size(&dev.res_pio));
 out_deregister_pio:
#endif

 out_release_tx:
release_mem_region(dev.res_tx.start, resource_size(&dev.res_tx));
 out_deregister_tx:
 out_release_rx:
release_mem_region(dev.res_rx.start, resource_size(&dev.res_rx));
 out_deregister_rx:
misc_deregister(&txrx_subsys_misc_device);
return ret;
}

/* Clean-up code: release resources */
static int txrx_subsys_remove(struct platform_device *pdev)
{
iounmap(dev.virtbase_rx);
release_mem_region(dev.res_rx.start, resource_size(&dev.res_rx));
iounmap(dev.virtbase_tx);
release_mem_region(dev.res_tx.start, resource_size(&dev.res_tx));
#ifdef PIO
iounmap(dev.virtbase_pio);
release_mem_region(dev.res_pio.start,
   resource_size(&dev.res_pio));
#endif
misc_deregister(&txrx_subsys_misc_device);
return 0;
}

/* Which "compatible" string(s) to search for in the Device Tree */
#ifdef CONFIG_OF
static const struct of_device_id txrx_subsys_of_match[] = {
{.compatible = "altr,txrx_subsys"},
{},
};

MODULE_DEVICE_TABLE(of, txrx_subsys_of_match);
#endif

/* Information for registering ourselves as a "platform" driver */
static struct platform_driver txrx_subsys_driver = {
.driver = {
.name = DRIVER_NAME,
.owner = THIS_MODULE,
.of_match_table = of_match_ptr(txrx_subsys_of_match),
},
.remove = __exit_p(txrx_subsys_remove),
};

/* Called when the module is loaded: set things up */
static int __init txrx_subsys_init(void)
{
pr_info(DRIVER_NAME ": init\n");
return platform_driver_probe(&txrx_subsys_driver, txrx_subsys_probe);
}

/* Called when the module is unloaded: release resources */
static void __exit txrx_subsys_exit(void)
{
platform_driver_unregister(&txrx_subsys_driver);
pr_info(DRIVER_NAME ": exit\n");
}

module_init(txrx_subsys_init);
module_exit(txrx_subsys_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Christopher Campbell, Sheng Qian, Valeh Valiollahpour Amiri");
MODULE_DESCRIPTION("TxRx Subsystem");
```

txrx_test.c

```
/*******************************************************************************
 * TxRx Test
 *
 * Description:
 *
 *  This program tests the txrx_fifos driver
 *
 * Authors:
 *
 *  Christopher Campbell (cc3769@columbia.edu)
 *  Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 *
 * Last Modified:
 *
 *  April 4th, 2015
 * ****************************************************************************/

#include <stdio.h>
#include "txrx_subsys.h"
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
```

```
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <stdlib.h>

/* Comment next line if the Enable Packet Data
 * feature is disabled.
 */
#define EN_PACKET_DATA
#define NUM_PKTS_READ 20
//#define PIO

int txrx_subsys_fd;

int main(int argc, const char *argv[])
{
int i;
char txrx_choice;
static const char filename[] = "/dev/txrx_subsys";
unsigned int pio_clr, pio_set;
/* pkt_info contains information related to
 * SOP, EOP, empty, error and channel. It is used
 * when the enable packet data feature is used.
 */
unsigned int pkt_info;
/* Hold data read in receive mode */
unsigned int data_read;

if (argc!= 2) {
printf("usage: %s 0 for tx, 1 for rx\n", argv[0]);
exit(1);
}
else
txrx_choice = atoi(argv[1]);

printf("%d\n", txrx_choice);

if ( (txrx_subsys_fd = open(filename, O_RDWR)) == -1 ) {
fprintf(stderr, "could not open %s\n", filename);
return -1;
}

#ifdef PIO
/* Clear pio */
pio_clr = 2;
if (ioctl(txrx_subsys_fd, PIO_CLR, &pio_clr)) {
perror("ioctl(PIO_CLR) failed");
return;
```

```
}

/* Set pio */
pio_set = 2;
if (ioctl(txrx_subsys_fd, PIO_SET, &pio_set)) {
perror("ioctl(PIO_SET) failed");
return;
}
#endif

/*  TRANSMIT SOME DATA */
if (!txrx_choice) {

#ifndef EN_PACKET_DATA
/* Send: SOP, 21 bytes of data, EOP, valid mask for the last byte, last byte of data */
unsigned int tx_data[] = {0x7a7a7a7a,
  0x01005e00, 0x00010017, 0xf293c2d5, 0x08004500,
  0x0046417c, 0x00000111, 0x032d803b, 0x14c2e000,
  0x00010272, 0x02720032, 0x2fd6534e, 0x51554552,
  0x593a6468, 0x63703437, 0x2e63732e, 0x636f6c75,
  0x6d626961, 0x2e656475, 0x3a725a77, 0x6472533a,
  0x78737672,
  0x7b7b7b7b,  0xffffffff, 0xA916C57B};

for (i=0; i<26; i++) {
if (ioctl(txrx_subsys_fd, TX_DATA, &tx_data[i])) {
perror("ioctl(TX_DATA) failed");
return;
}
}
#else
/* Set SOP bit in the packet_info register */
pkt_info = 0x00000001;
if (ioctl(txrx_subsys_fd, CONTROL, &pkt_info)) {
perror("ioctl(CONTROL) failed");
return;
}
/* Send 21 bytes of data */
unsigned int tx_data[] = {0x01005e00, 0x00010017, 0xf293c2d5, 0x08004500,
  0x0046417c, 0x00000111, 0x032d803b, 0x14c2e000,
  0x00010272, 0x02720032, 0x2fd6534e, 0x51554552,
  0x593a6468, 0x63703437, 0x2e63732e, 0x636f6c75,
  0x6d626961, 0x2e656475, 0x3a725a77, 0x6472533a,
  0x78737672,
  // last byte of data
  0xA916C57B};
for (i=0; i<21; i++) {
if (ioctl(txrx_subsys_fd, TX_DATA, &tx_data[i])) {
```

```
perror("ioctl(TX_DATA) failed");
return;
}
}
/* Set EOP bit and the two empty bits for the last byte in the packet_info register,
 * then send the last byte of data
 */
pkt_info = 0x00000002;
if (ioctl(txrx_subsys_fd, CONTROL, &pkt_info)) {
perror("ioctl(CONTROL) failed");
return;
}
if (ioctl(txrx_subsys_fd, TX_DATA, &tx_data[21])) {
perror("ioctl(TX_DATA) failed");
return;
}
#endif
}

/* RECEIVE SOME DATA */
else {
for (i=0; i<NUM_PKTS_READ; i++) {
if (ioctl(txrx_subsys_fd, RX_DATA, &data_read)) {
perror("ioctl(RX_DATA) failed");
return;
}
printf("data received = %04x\n", data_read);
}
}

return 0;
}
```

5.5 Loopback Driver

Makefile

```
# Compiles blazepps_loopback driver
#
# Authors
#
#  Christopher Campbell (cc3769@columbia.edu)
# Valeh Valiollahpour Amiri (vv2252@columbia.edu)
#
# Last Modified
#
#  April 23rd, 2015

ifneq (${KERNELRELEASE},)

# KERNELRELEASE defined: we are being compiled as part of the Kernel
obj-m := blazepps_loopback.o

else

# We are being compiled as a module: use the Kernel build system
PWD := $(shell pwd)
KDIR := /usr/src/linux
DTC := $(KDIR)/scripts/dtc/dtc

default: module $(DRV)

module:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} modules

# NOTE: Actually, for the network device driver we don't modify the device tree
socfpga.dtb: socfpga.dts
$(DTC) -O dtb -o socfpga.dtb socfpga.dts

clean:
${MAKE} -C ${KDIR} SUBDIRS=${PWD} clean
${RM} socfpga.dtb *˜

endif
```

blazepps_loopback.h

```
/* BlazePPS Software Loopback Driver Header
 *
 * Authors:
```

```
 * Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 * Christopher Campbell (cc3769@columbia.edu)
 *
 * Last Modified:
 * April 23rd, 2015
 */

extern struct net_device *blazedrv;

/* Macro for debug printk's */
#define PDEBUG(fmt, args...) printk( KERN_DEBUG "blaze: " fmt, ## args)

#define NUM_DEVS 2

/* These are the flags in the statusword */
#define BLAZE_RX_INTR 0x0001
#define BLAZE_TX_INTR 0x0002

/* Default use of napi */
#define USE_NAPI 0

/* Default size of the packet pool */
#define POOL_SZ 8

/* Default use of lockup */
#define LOCKUP 0

/* Default timeout period */
#define BLAZE_TIMEOUT 5   /* In jiffies */
```

blazepps_loopback.c

```
/* BlazePPS Software Loopback Driver
 *
 * Authors:
 * Valeh Valiollahpour Amiri (vv2252@columbia.edu)
 * Christopher Campbell (cc3769@columbia.edu)
 *
 * References:
 *  snull.c
 *
 * Last Modified:
 * April 23rd, 2015
 */

/* Network and Host Address Scheme
 *
 * Networks:
```

```
 *     blazenet0    192.168.2.0
 *     blazenet1    192.168.3.0
 *
 * Hosts:
 *     local0       192.168.2.1
 *     remote0      192.168.2.2
 *     local1       192.168.3.2
 *     remote1      192.168.3.1
 *
 * The network addresses have been added to /etc/networks and the host
 * addresses have been added to /etc/hosts so all of these addresses can
 * be referrenced by name
 *
 * Once the driver is running, you should see the bz0 and bz1 interfaces
 * using the command 'ifconfig'.
 *
 * Assigning addresses to the interfaces (bz0 and bz1):
 *     $ ifconfig bz0 local0
 *     $ ifconfig bz1 local1
 *
 * Run load.sh script which loads the module and then runs the two commandes above.
 *
 * Sending packets from bz0 to bz1:
 *     $ ping -c 2 remote0
 *
 * Sending packets from bz1 to bz0:
 *     $ ping -c 2 remote1
 */

/* References
 * Network drivers
 *   http://www.oreilly.com/openbook/linuxdrive3/book/ch17.pdf
 *   http://www.cubrid.org/blog/dev-platform/
understanding-tcp-ip-network-stack/
 * blaze.c
 *   http://www.xml.com/ldd/chapter/book/ch14.html
 *
 * loopback.c
 *   http://lxr.free-electrons.com/source/drivers/net/loopback.c?v=3.8
 *
 * NAPI:
 *   http://en.wikipedia.org/wiki/New_API#Compliant_drivers
 *   http://www.linuxfoundation.org/collaborate/workgroups/networking/napi
 */


#include <linux/module.h>
#include <linux/init.h>
```

```
#include <linux/moduleparam.h>
#include <linux/version.h>

#include <linux/sched.h>
#include <linux/kernel.h>
#include <linux/slab.h>
#include <linux/errno.h>
#include <linux/types.h>
#include <linux/interrupt.h>

#include <linux/in.h>
#include <linux/netdevice.h>
#include <linux/etherdevice.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/skbuff.h>
#include <linux/in6.h>
#include <asm/checksum.h>

#include "blazepps_ethernet.h"

MODULE_DESCRIPTION("BlazePPS Ethernet Driver");
MODULE_AUTHOR("Valeh Valiollahpour Amiri, Christopher Campbell");
MODULE_LICENSE("GPL");

/* static int use_napi = 0;
module_param(use_napi, int, S_IRUGO);

int pool_size = 8;
module_param(pool_size, int, S_IRUGO);

static int lockup = 0;
module_param(lockup, int, S_IRUGO);

static int timeout = BLAZE_TIMEOUT;
module_param(timeout, int, S_IRUGO); */

struct net_device *blaze_devs[NUM_DEVS];

/* In-flight packet */
struct blaze_packet {
struct blaze_packet *next;
struct net_device *dev;
int datalen;
/* ETH_DATA_LEN := Max octets in payload (1500 bytes)*/
u8 data[ETH_DATA_LEN];
};
```

```
/* Private device data */
struct blaze_priv {
struct net_device_stats stats; /* device stats */
int status; /* device status (soft emu) */
struct blaze_packet *ppool; /* packet pool */
struct blaze_packet *rx_queue;  /* incoming packets */
int rx_int_enabled; /* receive enabled */
struct sk_buff *skb; /* socket buff for snt/recv pckt */
int tx_packetlen; /* transmission packet length */
u8 *tx_packetdata; /* transmission packet data */
spinlock_t lock; /* blaze_priv lock */
struct net_device *dev; /* net device */
struct napi_struct napi; /* used for NAPI compliance */
};

static void blaze_tx_timeout(struct net_device *dev);
static void (*blaze_interrupt)(int, void *, struct pt_regs *);


/* Set up device packet pool */
void blaze_setup_pool(struct net_device *dev)
{
int i;
struct blaze_packet *pkt;
struct blaze_priv *priv = netdev_priv(dev);

priv->ppool = NULL;
for (i = 0; i < POOL_SZ; i++) {
pkt = kmalloc (sizeof (struct blaze_packet), GFP_KERNEL);
if (pkt == NULL) {
printk(KERN_NOTICE "Ran out of memory" \
"allocating packet pool\n");
return;
}

pkt->dev = dev;
pkt->next = priv->ppool;
priv->ppool = pkt;
}
}

/* Teardown device pool */
void blaze_teardown_pool(struct net_device *dev)
{
struct blaze_packet *pkt;
struct blaze_priv *priv = netdev_priv(dev);

while ((pkt = priv->ppool)) {
```

```
priv->ppool = pkt->next;
kfree(pkt);
/* FIXME - in-flight packets ? */
}
}

/* Get buffer from the head of the packet pool to use for xmission. */
struct blaze_packet *blaze_get_tx_buffer(struct net_device *dev)
{
unsigned long flags;
struct blaze_packet *pkt;

struct blaze_priv *priv = netdev_priv(dev);

spin_lock_irqsave(&priv->lock, flags);
pkt = priv->ppool;
priv->ppool = pkt->next;
if (priv->ppool == NULL) {
printk (KERN_INFO "Pool empty\n");
netif_stop_queue(dev);
}
spin_unlock_irqrestore(&priv->lock, flags);

return pkt;
}

/* Append a new packet at the head of the packet pool. */
void blaze_release_buffer(struct blaze_packet *pkt)
{
unsigned long flags;
struct blaze_priv *priv = netdev_priv(pkt->dev);

spin_lock_irqsave(&priv->lock, flags);
pkt->next = priv->ppool;
priv->ppool = pkt;
spin_unlock_irqrestore(&priv->lock, flags);
if (netif_queue_stopped(pkt->dev) && pkt->next == NULL)
netif_wake_queue(pkt->dev);
}

/* Enqueue a received packet to the queue of received packets. */
void blaze_enqueue_buf(struct net_device *dev, struct blaze_packet *pkt)
{
unsigned long flags;
struct blaze_priv *priv = netdev_priv(dev);

spin_lock_irqsave(&priv->lock, flags);
pkt->next = priv->rx_queue;  /* FIXME - misorders packets */
```

```
priv->rx_queue = pkt;
spin_unlock_irqrestore(&priv->lock, flags);
}

/* Dequeue a packet from the queue of received packets. */
struct blaze_packet *blaze_dequeue_buf(struct net_device *dev)
{
struct blaze_priv *priv = netdev_priv(dev);
struct blaze_packet *pkt;
unsigned long flags;

spin_lock_irqsave(&priv->lock, flags);
pkt = priv->rx_queue;
if (pkt != NULL)
priv->rx_queue = pkt->next;
spin_unlock_irqrestore(&priv->lock, flags);
return pkt;
}

/* Enable and disable receive interrupts */
static void blaze_rx_ints(struct net_device *dev, int enable)
{
struct blaze_priv *priv = netdev_priv(dev);
priv->rx_int_enabled = enable; //.ndo_change_mtu      = blaze_change_mtu,

}

/* Open device */
int blaze_open(struct net_device *dev)
{
/*
 * Hardware specific code will go here: request_region(),
 * request_irq(), ....  (like fops->open). Assign the hardware
 * address of the board: use "\0BZx", where x is 0 or 1. The
 * first byte is '\0' to avoid being a multicast address (the
 * first byte of multicast addrs is odd).
 */

memcpy(dev->dev_addr, "\0BLZE0", ETH_ALEN);
if (dev == blaze_devs[1])
dev->dev_addr[ETH_ALEN-1]++; /* \0BLZE1 */
netif_start_queue(dev);
return 0;
}

/* Stop transmit queue */
int blaze_release(struct net_device *dev)
{
```

```
/*
 * Hardware specific code will go here:
 * release ports, irq and such -- like fops->close
 */

netif_stop_queue(dev); /* can't transmit anymore */

return 0;
}

/* Configuration via ifconfig */
int blaze_config(struct net_device *dev, struct ifmap *map)
{
if (dev->flags & IFF_UP) /* can't act on a running interface */
return -EBUSY;

/* Don't allow changing the I/O address */
if (map->base_addr != dev->base_addr) {
printk(KERN_WARNING "blaze: Can't change I/O address\n");
return -EOPNOTSUPP;
}

/* Allow changing the IRQ */
if (map->irq != dev->irq) {
dev->irq = map->irq;
         /* request_irq() is delayed to open-time */
}

/* Ignore other fields */
return 0;
}

/* Receive a packet (retrieve, encapsulate and pass over to upper levels) */
void blaze_rx(struct net_device *dev, struct blaze_packet *pkt)
{
struct sk_buff *skb;
struct blaze_priv *priv = netdev_priv(dev);

/*
 * Packet received from transmission medium
         * (build an skb around it, so upper layers can handle it)
 */
skb = dev_alloc_skb(pkt->datalen + 2);
if (!skb) {
if (printk_ratelimit())
printk(KERN_NOTICE "blaze rx: low on mem - " \
"packet dropped\n");
priv->stats.rx_dropped++;
```

```
goto out;
}
skb_reserve(skb, 2); /* align IP on 16B boundary */
memcpy(skb_put(skb, pkt->datalen), pkt->data, pkt->datalen);

/* Write metadata, and then pass to the receive level */
skb->dev = dev;
skb->protocol = eth_type_trans(skb, dev);
skb->ip_summed = CHECKSUM_UNNECESSARY; /* don’t check it */
priv->stats.rx_packets++;
priv->stats.rx_bytes += pkt->datalen;
netif_rx(skb);
out:
return;
}

/* Polling function */
static int blaze_poll(struct napi_struct *napi, int budget)
{
int npackets = 0;
struct sk_buff *skb;
struct blaze_priv *priv = container_of(napi, struct blaze_priv,
napi);
struct net_device *dev = priv->dev;
struct blaze_packet *pkt;

while (npackets < budget && priv->rx_queue) {
pkt = blaze_dequeue_buf(dev);
skb = dev_alloc_skb(pkt->datalen + 2);
if (! skb) {
if (printk_ratelimit())
printk(KERN_NOTICE "blaze: packet " \
"dropped\n");
priv->stats.rx_dropped++;
blaze_release_buffer(pkt);
continue;
}
skb_reserve(skb, 2); /* align IP on 16B boundary */
memcpy(skb_put(skb, pkt->datalen), pkt->data, pkt->datalen);
skb->dev = dev;
skb->protocol = eth_type_trans(skb, dev);
skb->ip_summed = CHECKSUM_UNNECESSARY; /* don’t check it */
netif_receive_skb(skb);

         /* Maintain stats */
npackets++;
priv->stats.rx_packets++;
priv->stats.rx_bytes += pkt->datalen;
```

```
blaze_release_buffer(pkt);
}
/* If we processed all packets, tell the kernel and reenable ints */
if (! priv->rx_queue) {
napi_complete(napi);
blaze_rx_ints(dev, 1);
return 0;
}

/* We couldn't process everything. */
return npackets;
}

/* Typical interrupt handler */
static void blaze_regular_interrupt(int irq, void *dev_id,
    struct pt_regs *regs)
{
int statusword;
struct blaze_priv *priv;
struct blaze_packet *pkt = NULL;

/*
 * As usual, check the "device" pointer to be sure it is
 * really interrupting.
 * Then assign "struct device *dev"
 */
struct net_device *dev = (struct net_device *)dev_id;

/* Paranoid */
if (!dev)
return;

/* Lock the device */
priv = netdev_priv(dev);
spin_lock(&priv->lock);

/* Retrieve statusword: real netdevices use I/O instructions */
statusword = priv->status;
priv->status = 0;
/* rx interrupt */
if (statusword & BLAZE_RX_INTR) {
/* A new packet has been added and enqueued tio the receive queue. Take it and send it to blaze
pkt = priv->rx_queue;
if (pkt) {
priv->rx_queue = pkt->next;
blaze_rx(dev, pkt);
}
}
```

```
/* tx interrupt */
if (statusword & BLAZE_TX_INTR) {
/* a transmission is over: update the tx stats and free the associated skb */
priv->stats.tx_packets++;
priv->stats.tx_bytes += priv->tx_packetlen;
dev_kfree_skb(priv->skb);
}

/* Unlock the device and we are done */
spin_unlock(&priv->lock);
if (pkt) blaze_release_buffer(pkt); /* Do this outside the lock! */
return;
}

/* NAPI Interrupt handler */
static void blaze_napi_interrupt(int irq, void *dev_id,
struct pt_regs *regs)
{
int statusword;
struct blaze_priv *priv;

/* Check the "device" pointer for shared handlers */
struct net_device *dev = (struct net_device *) dev_id;

/*
 * Hardware related code goes here: check with hw if it's
 * really ours
 */

/* Paranoid */
if (!dev)
return;

priv = netdev_priv(dev);
spin_lock(&priv->lock);

/* Retrieve statusword */
/*
 * Hardware related code goes here (real netdevices use
 * I/O instructions)
 */
statusword = priv->status;
priv->status = 0;
if (statusword & BLAZE_RX_INTR) {
blaze_rx_ints(dev, 0);  /* Disable further interrupts */
napi_schedule(&priv->napi);
}
if (statusword & BLAZE_TX_INTR) {
```

```
         /* Transmission is over, free the skb */
priv->stats.tx_packets++;
priv->stats.tx_bytes += priv->tx_packetlen;
dev_kfree_skb(priv->skb);
}

spin_unlock(&priv->lock);
return;
}

/* Transmit a packet (low level interface) */
static void blaze_hw_tx(char *buf, int len, struct net_device *dev)
{
/* This function deals with hw details of transmission. */
struct iphdr *ih;
struct net_device *dest;
struct blaze_priv *priv;
u32 *saddr, *daddr;
struct blaze_packet *tx_buffer;

/* Paranoid! */
if (len < sizeof(struct ethhdr) + sizeof(struct iphdr)) {
printk("blaze: Hmm... packet too short (%i octets)\n",
len);
return;
}

/* Enable this conditional to print out the payload data */
if (0) {
int i;
PDEBUG("len is %i\n" KERN_DEBUG "data:",len);
for (i=0; i<len; i++)
printk(" %02x",buf[i]);
printk("\n");
}

/*
 * Ethhdr is 14 bytes, but the kernel arranges for iphdr
 * to be aligned (i.e., ethhdr is unaligned)
 */
ih = (struct iphdr *)(buf+sizeof(struct ethhdr));
saddr = &ih->saddr;
daddr = &ih->daddr;

/* Why??? */
((u8 *)saddr)[2] ^= 1; /* change the third octet (class C) */
((u8 *)daddr)[2] ^= 1;
```

```
ih->check = 0;         /* and rebuild the checksum (ip needs it) */
ih->check = ip_fast_csum((unsigned char *)ih, ih->ihl);

/* Print out IP addr of source and destination */
if (dev == blaze_devs[0])
PDEBUG("%08x:%05i --> %08x:%05i\n",
ntohl(ih->saddr),ntohs(((struct tcphdr *)(ih+1))->source),
ntohl(ih->daddr),ntohs(((struct tcphdr *)(ih+1))->dest));
else
PDEBUG("%08x:%05i <-- %08x:%05i\n",
ntohl(ih->daddr),ntohs(((struct tcphdr *)(ih+1))->dest),
ntohl(ih->saddr),ntohs(((struct tcphdr *)(ih+1))->source));

/*
 * Ok, now the packet is ready for transmission: first simulate a
 * receive interrupt on the destination device, then a
 * transmission-done interrupt on the transmitting device
 */
dest = blaze_devs[dev == blaze_devs[0] ? 1 : 0];
priv = netdev_priv(dest);
priv = netdev_priv(dest);
tx_buffer = blaze_get_tx_buffer(dev);
tx_buffer->datalen = len;
memcpy(tx_buffer->data, buf, len);
/* Enqueue received packet on the destination's receive queue */
blaze_enqueue_buf(dest, tx_buffer);
if (priv->rx_int_enabled) {
priv->status |= BLAZE_RX_INTR;
blaze_interrupt(0, dest, NULL);
}

priv = netdev_priv(dev);
priv->tx_packetlen = len;
priv->tx_packetdata = buf;
priv->status |= BLAZE_TX_INTR;
/* Drop packet if too many stats.tx_packets */
if (LOCKUP && ((priv->stats.tx_packets + 1) % LOCKUP) == 0) {
/* Simulate a dropped transmit interrupt */
netif_stop_queue(dev);
PDEBUG("Simulate lockup at %ld, txp %ld\n", jiffies,
       (unsigned long) priv->stats.tx_packets);
}
blaze_interrupt(0, dev, NULL);
}

/* Transmit a packet (called by the kernel) */
int blaze_tx(struct sk_buff *skb, struct net_device *dev)
{
```

```
int len;
/* ETH_ZLEN := Min. octets in frame sans FCS (Frame Checking Sequence) (60 bytes) */
char *data, shortpkt[ETH_ZLEN];
struct blaze_priv *priv = netdev_priv(dev);

data = skb->data;
len = skb->len;
if (len < ETH_ZLEN) {
memset(shortpkt, 0, ETH_ZLEN);
memcpy(shortpkt, skb->data, skb->len);
len = ETH_ZLEN;
data = shortpkt;
}

dev->trans_start = jiffies; /* save the timestamp */

/* Remember the skb, so we can free it at interrupt time */
priv->skb = skb;

/* Actual delivery of the data (low level xmit) - device-specific */
blaze_hw_tx(data, len, dev);

return 0; /* our simple device cannot fail */
}

/* Handle a transmit timeout */
void blaze_tx_timeout(struct net_device *dev)
{
struct blaze_priv *priv = netdev_priv(dev);

PDEBUG("Transmit timeout at %ld, latency %ld\n", jiffies,
jiffies - dev->trans_start);

        /* Simulate a transmission interrupt to get things moving */
priv->status = BLAZE_TX_INTR;
blaze_interrupt(0, dev, NULL);
priv->stats.tx_errors++;
netif_wake_queue(dev);
return;
}

/* ioctl commands */
int blaze_ioctl(struct net_device *dev, struct ifreq *rq, int cmd)
{
/* ioctl commands go here */
PDEBUG("ioctl\n");
return 0;
}
```

```
/* Return stats */
struct net_device_stats *blaze_stats(struct net_device *dev)
{
struct blaze_priv *priv = netdev_priv(dev);
return &priv->stats;
}

/*
 * Manually fills up ethernet header because ARP is not used
 * (probably won't be necessary in our actual driver)
 */
int blaze_rebuild_header(struct sk_buff *skb)
{
struct ethhdr *eth = (struct ethhdr *) skb->data;
struct net_device *dev = skb->dev;

memcpy(eth->h_source, dev->dev_addr, dev->addr_len);
memcpy(eth->h_dest, dev->dev_addr, dev->addr_len);
/* ETH_ALEN := Octets in one ethernet addr (6 bytes) */
eth->h_dest[ETH_ALEN-1] ^= 0x01;   /* dest is us xor 1 */
return 0;
}

/* Creates header (probably won't be necessary in our actual driver) */
int blaze_header(struct sk_buff *skb, struct net_device *dev,
unsigned short type, const void *daddr, const void *saddr,unsigned len)
{
struct ethhdr *eth = (struct ethhdr *) skb_push(skb, ETH_HLEN);

eth->h_proto = htons(type);
memcpy(eth->h_source, saddr ? saddr : dev->dev_addr, dev->addr_len);
memcpy(eth->h_dest,   daddr ? daddr : dev->dev_addr, dev->addr_len);
eth->h_dest[ETH_ALEN-1] ^= 0x01;   /* dest is us xor 1 */
return (dev->hard_header_len);
}

/*
 * The "change_mtu" method is usually not needed.
 * If you need it, it must be like this.
 */
int blaze_change_mtu(struct net_device *dev, int new_mtu)
{
unsigned long flags;
struct blaze_priv *priv = netdev_priv(dev);
spinlock_t *lock = &priv->lock;

/* check ranges */
```

```
if ((new_mtu < 68) || (new_mtu > 1500))
return -EINVAL;
/*
 * Do anything you need, and the accept the value
 */
spin_lock_irqsave(lock, flags);
dev->mtu = new_mtu;
spin_unlock_irqrestore(lock, flags);
return 0; /* success */
}

static const struct header_ops blaze_header_ops = {
        .create  = blaze_header,
.rebuild = blaze_rebuild_header
};

static const struct net_device_ops blaze_netdev_ops = {
.ndo_open            = blaze_open,
.ndo_stop            = blaze_release,
.ndo_start_xmit      = blaze_tx,
.ndo_do_ioctl        = blaze_ioctl,
.ndo_set_config      = blaze_config,
.ndo_get_stats       = blaze_stats,
.ndo_change_mtu      = blaze_change_mtu,
.ndo_tx_timeout      = blaze_tx_timeout
};

/* Init/Probe function (invoked by register_netdev()) */
void blaze_init(struct net_device *dev)
{
struct blaze_priv *priv;

/*
 * Hardware specific code will go here: check_region(),
 * probe irq, ... Return -ENODEV if no device is found.
 * No resource should be grabbed (this is done on open())
 */

     /* Setup dev fields */
ether_setup(dev); /* set some of these manually? */
dev->watchdog_timeo = BLAZE_TIMEOUT;
dev->netdev_ops = &blaze_netdev_ops;
dev->header_ops = &blaze_header_ops;
dev->flags           |= IFF_NOARP; /* use ARP? */
dev->features        |= NETIF_F_HW_CSUM;

/* Intialize priv field */
priv = netdev_priv(dev);
```

```
if (USE_NAPI) {
/* weight = 2 (> weight = interface more important) */
netif_napi_add(dev, &priv->napi, blaze_poll, 2);
}
memset(priv, 0, sizeof(struct blaze_priv));

spin_lock_init(&priv->lock);
/* Enable receive interrupts */
blaze_rx_ints(dev, 1);
blaze_setup_pool(dev);
}

/* Cleanup */
void blaze_cleanup(void)
{
int i;

for (i = 0; i < NUM_DEVS; i++) {
if (blaze_devs[i]) {
unregister_netdev(blaze_devs[i]);
blaze_teardown_pool(blaze_devs[i]);
free_netdev(blaze_devs[i]);
}
}

return;
}

/* Initializes device */
int blaze_init_module(void)
{
int result, i, ret = -ENOMEM;

blaze_interrupt = USE_NAPI ? blaze_napi_interrupt : blaze_regular_interrupt;

/* Allocate the devices */
blaze_devs[0] = alloc_netdev(sizeof(struct blaze_priv), "bz%d",
     blaze_init);
blaze_devs[1] = alloc_netdev(sizeof(struct blaze_priv), "bz%d",
     blaze_init);
if (blaze_devs[0] == NULL || blaze_devs[1] == NULL)
goto out;

ret = -ENODEV;
for (i = 0; i < 2;  i++)
if ((result = register_netdev(blaze_devs[i])))
printk("blaze: error %i registering device \"%s\"\n",
       result, blaze_devs[i]->name);
```

```
else
ret = 0;
 out:
if (ret)
blaze_cleanup();
return ret;
}

module_init(blaze_init_module);
module_exit(blaze_cleanup);
```

load.sh

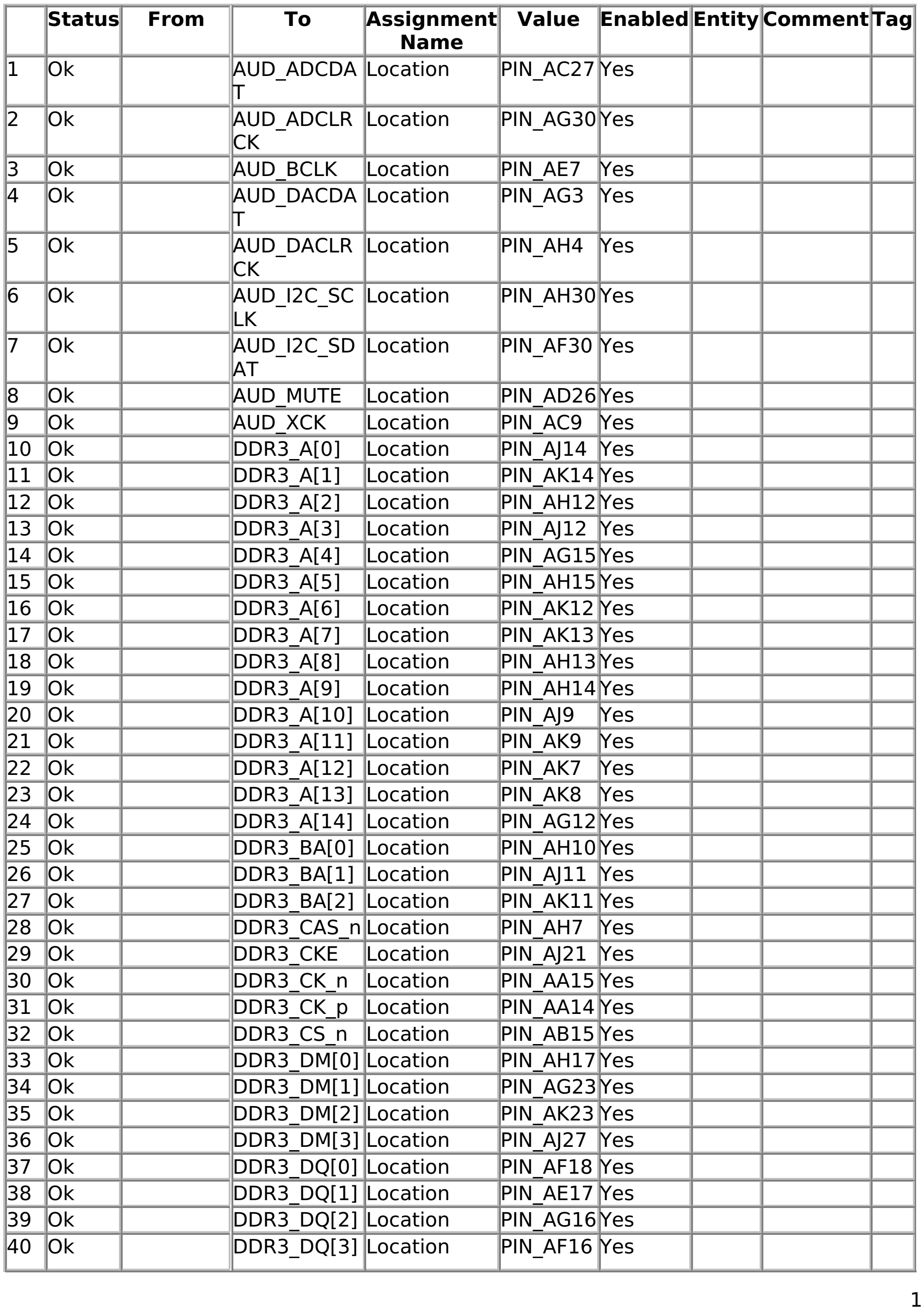
```
#!/bin/bash

insmod ./blazepps_loopback.ko
ifconfig bz0 192.168.2.1
ifconfig bz1 192.168.3.2
```

unload.sh

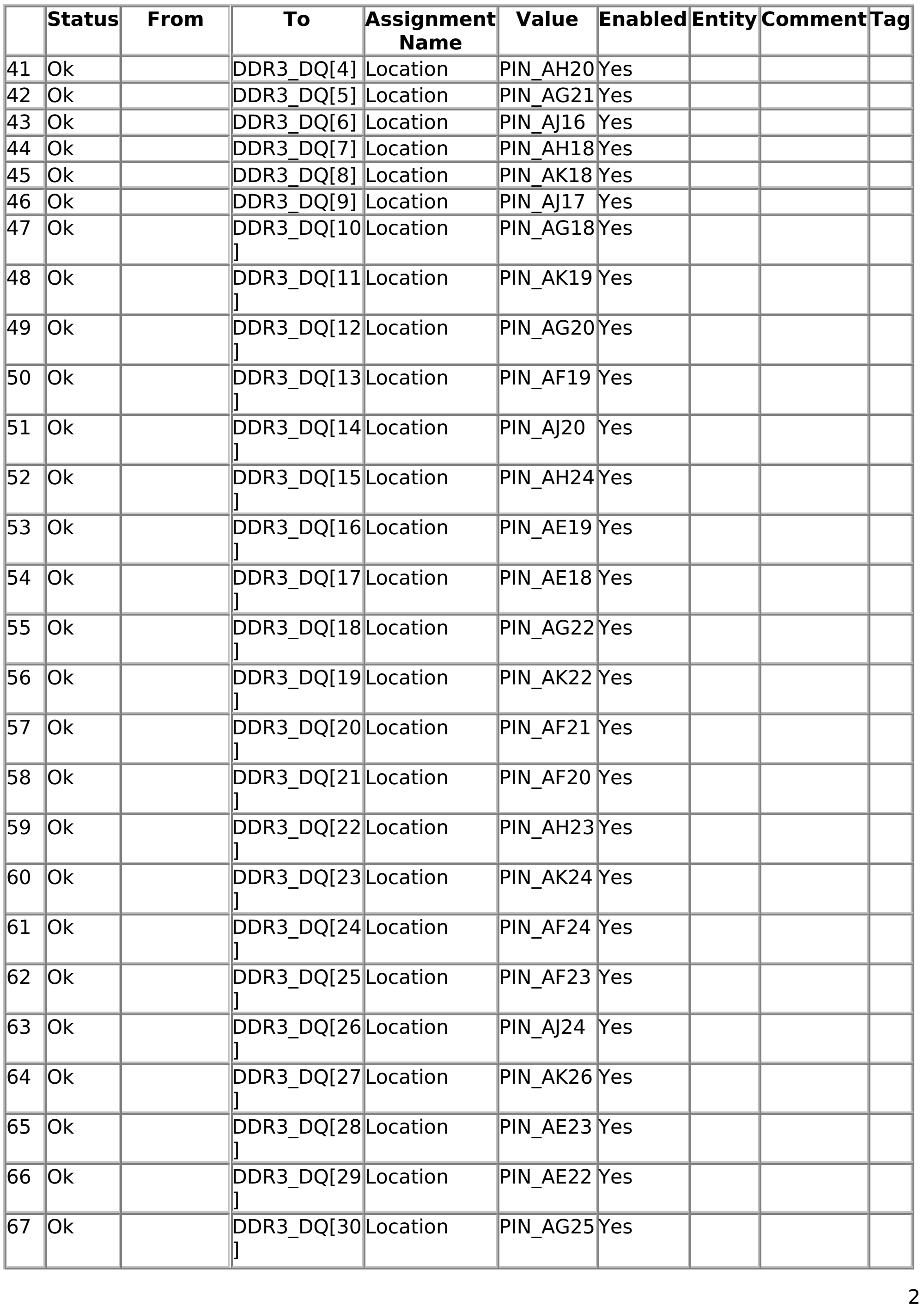
```
#!/bin/bash

ifconfig bz0 down
ifconfig bz1 down
rmmod blazepps_loopback.ko
```

6 Appendix

6.1 A - Dual XAUI Pin Assignments

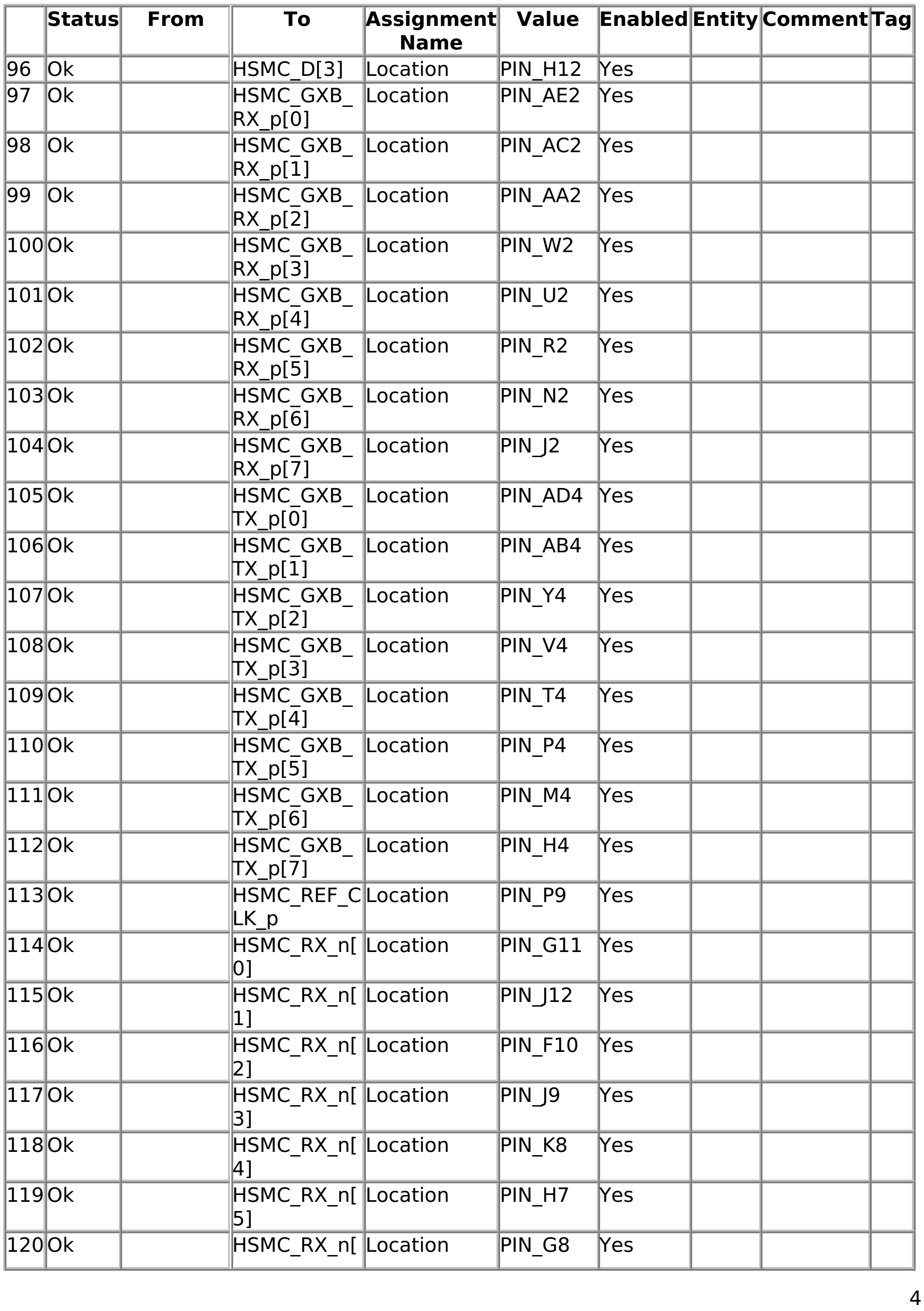


	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	Ok		AUD_ADCDAT	Location	PIN_AC27	Yes			
2	Ok		AUD_ADCLRCK	Location	PIN_AG30	Yes			
3	Ok		AUD_BCLK	Location	PIN_AE7	Yes			
4	Ok		AUD_DACDAT	Location	PIN_AG3	Yes			
5	Ok		AUD_DACLRCK	Location	PIN_AH4	Yes			
6	Ok		AUD_I2C_SCLK	Location	PIN_AH30	Yes			
7	Ok		AUD_I2C_SDAT	Location	PIN_AF30	Yes			
8	Ok		AUD_MUTE	Location	PIN_AD26	Yes			
9	Ok		AUD_XCK	Location	PIN_AC9	Yes			
10	Ok		DDR3_A[0]	Location	PIN_AJ14	Yes			
11	Ok		DDR3_A[1]	Location	PIN_AK14	Yes			
12	Ok		DDR3_A[2]	Location	PIN_AH12	Yes			
13	Ok		DDR3_A[3]	Location	PIN_AJ12	Yes			
14	Ok		DDR3_A[4]	Location	PIN_AG15	Yes			
15	Ok		DDR3_A[5]	Location	PIN_AH15	Yes			
16	Ok		DDR3_A[6]	Location	PIN_AK12	Yes			
17	Ok		DDR3_A[7]	Location	PIN_AK13	Yes			
18	Ok		DDR3_A[8]	Location	PIN_AH13	Yes			
19	Ok		DDR3_A[9]	Location	PIN_AH14	Yes			
20	Ok		DDR3_A[10]	Location	PIN_AJ9	Yes			
21	Ok		DDR3_A[11]	Location	PIN_AK9	Yes			
22	Ok		DDR3_A[12]	Location	PIN_AK7	Yes			
23	Ok		DDR3_A[13]	Location	PIN_AK8	Yes			
24	Ok		DDR3_A[14]	Location	PIN_AG12	Yes			
25	Ok		DDR3_BA[0]	Location	PIN_AH10	Yes			
26	Ok		DDR3_BA[1]	Location	PIN_AJ11	Yes			
27	Ok		DDR3_BA[2]	Location	PIN_AK11	Yes			
28	Ok		DDR3_CAS_n	Location	PIN_AH7	Yes			
29	Ok		DDR3_CKE	Location	PIN_AJ21	Yes			
30	Ok		DDR3_CK_n	Location	PIN_AA15	Yes			
31	Ok		DDR3_CK_p	Location	PIN_AA14	Yes			
32	Ok		DDR3_CS_n	Location	PIN_AB15	Yes			
33	Ok		DDR3_DM[0]	Location	PIN_AH17	Yes			
34	Ok		DDR3_DM[1]	Location	PIN_AG23	Yes			
35	Ok		DDR3_DM[2]	Location	PIN_AK23	Yes			
36	Ok		DDR3_DM[3]	Location	PIN_AJ27	Yes			
37	Ok		DDR3_DQ[0]	Location	PIN_AF18	Yes			
38	Ok		DDR3_DQ[1]	Location	PIN_AE17	Yes			
39	Ok		DDR3_DQ[2]	Location	PIN_AG16	Yes			
40	Ok		DDR3_DQ[3]	Location	PIN_AF16	Yes			



	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
41	Ok		DDR3_DQ[4]	Location	PIN_AH20	Yes			
42	Ok		DDR3_DQ[5]	Location	PIN_AG21	Yes			
43	Ok		DDR3_DQ[6]	Location	PIN_AJ16	Yes			
44	Ok		DDR3_DQ[7]	Location	PIN_AH18	Yes			
45	Ok		DDR3_DQ[8]	Location	PIN_AK18	Yes			
46	Ok		DDR3_DQ[9]	Location	PIN_AJ17	Yes			
47	Ok		DDR3_DQ[10]	Location	PIN_AG18	Yes			
48	Ok		DDR3_DQ[11]	Location	PIN_AK19	Yes			
49	Ok		DDR3_DQ[12]	Location	PIN_AG20	Yes			
50	Ok		DDR3_DQ[13]	Location	PIN_AF19	Yes			
51	Ok		DDR3_DQ[14]	Location	PIN_AJ20	Yes			
52	Ok		DDR3_DQ[15]	Location	PIN_AH24	Yes			
53	Ok		DDR3_DQ[16]	Location	PIN_AE19	Yes			
54	Ok		DDR3_DQ[17]	Location	PIN_AE18	Yes			
55	Ok		DDR3_DQ[18]	Location	PIN_AG22	Yes			
56	Ok		DDR3_DQ[19]	Location	PIN_AK22	Yes			
57	Ok		DDR3_DQ[20]	Location	PIN_AF21	Yes			
58	Ok		DDR3_DQ[21]	Location	PIN_AF20	Yes			
59	Ok		DDR3_DQ[22]	Location	PIN_AH23	Yes			
60	Ok		DDR3_DQ[23]	Location	PIN_AK24	Yes			
61	Ok		DDR3_DQ[24]	Location	PIN_AF24	Yes			
62	Ok		DDR3_DQ[25]	Location	PIN_AF23	Yes			
63	Ok		DDR3_DQ[26]	Location	PIN_AJ24	Yes			
64	Ok		DDR3_DQ[27]	Location	PIN_AK26	Yes			
65	Ok		DDR3_DQ[28]	Location	PIN_AE23	Yes			
66	Ok		DDR3_DQ[29]	Location	PIN_AE22	Yes			
67	Ok		DDR3_DQ[30]	Location	PIN_AG25	Yes			

	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
68	Ok		DDR3_DQ[31]	Location	PIN_AK27	Yes			
69	Ok		DDR3_DQS_n[0]	Location	PIN_W16	Yes			
70	Ok		DDR3_DQS_n[1]	Location	PIN_W17	Yes			
71	Ok		DDR3_DQS_n[2]	Location	PIN_AA18	Yes			
72	Ok		DDR3_DQS_n[3]	Location	PIN_AD19	Yes			
73	Ok		DDR3_DQS_p[0]	Location	PIN_V16	Yes			
74	Ok		DDR3_DQS_p[1]	Location	PIN_V17	Yes			
75	Ok		DDR3_DQS_p[2]	Location	PIN_Y17	Yes			
76	Ok		DDR3_DQS_p[3]	Location	PIN_AC20	Yes			
77	Ok		DDR3_ODT	Location	PIN_AE16	Yes			
78	Ok		DDR3_RAS_n	Location	PIN_AH8	Yes			
79	Ok		DDR3_RESET_n	Location	PIN_AK21	Yes			
80	Ok		DDR3_RZQ	Location	PIN_AG17	Yes			
81	Ok		DDR3_WE_n	Location	PIN_AJ6	Yes			
82	Ok		FAN_CTRL	Location	PIN_AG27	Yes			
83	Ok		HSMC_CLKIN_n[1]	Location	PIN_AB27	Yes			
84	Ok		HSMC_CLKIN_n[2]	Location	PIN_G15	Yes			
85	Ok		HSMC_CLKIN_p[1]	Location	PIN_AA26	Yes			
86	Ok		HSMC_CLKIN_p[2]	Location	PIN_H15	Yes			
87	Ok		HSMC_CLKOUT_n[1]	Location	PIN_E6	Yes			
88	Ok		HSMC_CLKOUT_n[2]	Location	PIN_A10	Yes			
89	Ok		HSMC_CLKOUT_p[1]	Location	PIN_E7	Yes			
90	Ok		HSMC_CLKOUT_p[2]	Location	PIN_A11	Yes			
91	Ok		HSMC_CLK_IN0	Location	PIN_J14	Yes			
92	Ok		HSMC_CLK_OUT0	Location	PIN_AD29	Yes			
93	Ok		HSMC_D[0]	Location	PIN_C10	Yes			
94	Ok		HSMC_D[1]	Location	PIN_H13	Yes			
95	Ok		HSMC_D[2]	Location	PIN_C9	Yes			

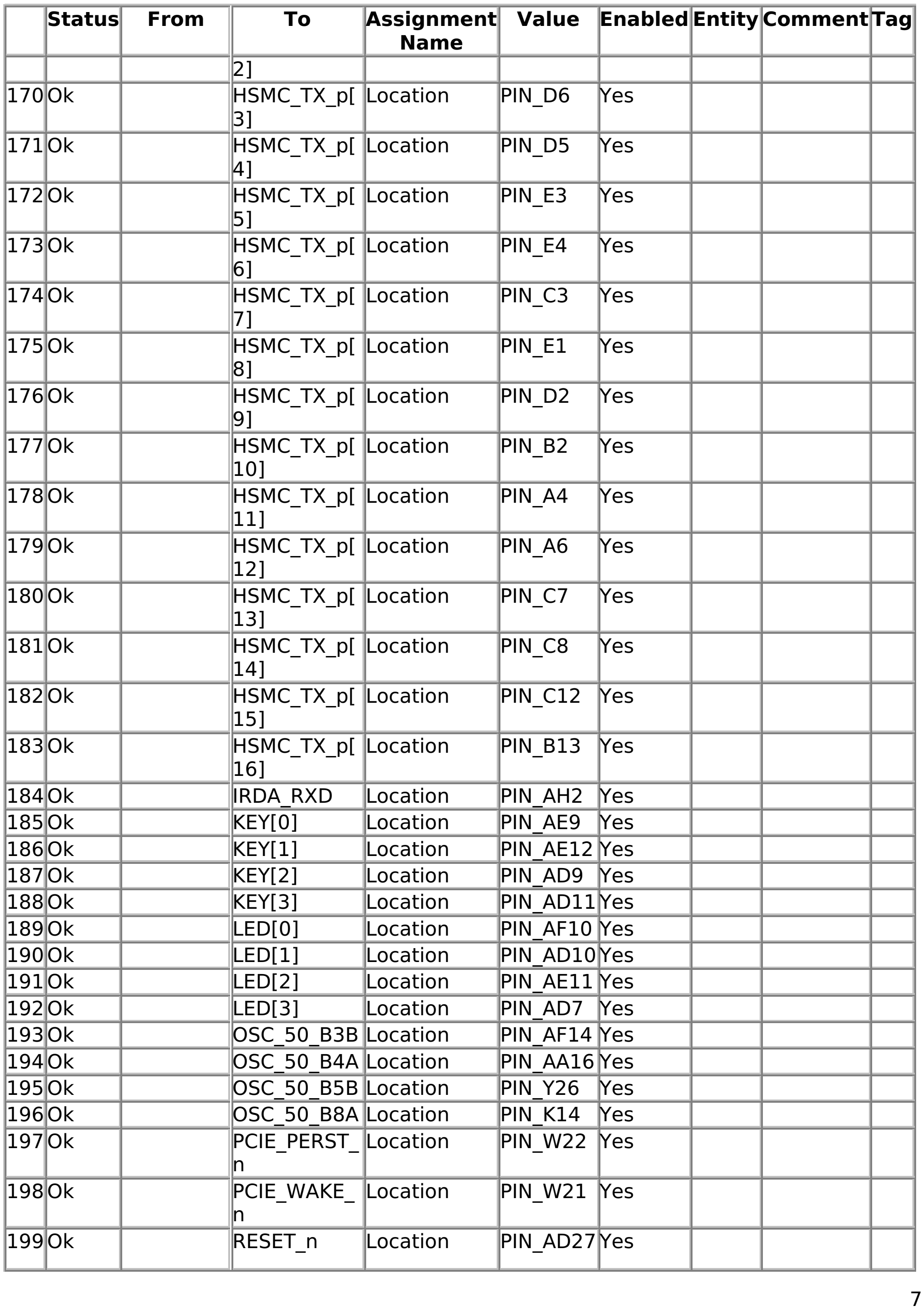


	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
96	Ok		HSMC_D[3]	Location	PIN_H12	Yes			
97	Ok		HSMC_GXB_RX_p[0]	Location	PIN_AE2	Yes			
98	Ok		HSMC_GXB_RX_p[1]	Location	PIN_AC2	Yes			
99	Ok		HSMC_GXB_RX_p[2]	Location	PIN_AA2	Yes			
100	Ok		HSMC_GXB_RX_p[3]	Location	PIN_W2	Yes			
101	Ok		HSMC_GXB_RX_p[4]	Location	PIN_U2	Yes			
102	Ok		HSMC_GXB_RX_p[5]	Location	PIN_R2	Yes			
103	Ok		HSMC_GXB_RX_p[6]	Location	PIN_N2	Yes			
104	Ok		HSMC_GXB_RX_p[7]	Location	PIN_J2	Yes			
105	Ok		HSMC_GXB_TX_p[0]	Location	PIN_AD4	Yes			
106	Ok		HSMC_GXB_TX_p[1]	Location	PIN_AB4	Yes			
107	Ok		HSMC_GXB_TX_p[2]	Location	PIN_Y4	Yes			
108	Ok		HSMC_GXB_TX_p[3]	Location	PIN_V4	Yes			
109	Ok		HSMC_GXB_TX_p[4]	Location	PIN_T4	Yes			
110	Ok		HSMC_GXB_TX_p[5]	Location	PIN_P4	Yes			
111	Ok		HSMC_GXB_TX_p[6]	Location	PIN_M4	Yes			
112	Ok		HSMC_GXB_TX_p[7]	Location	PIN_H4	Yes			
113	Ok		HSMC_REF_CLK_p	Location	PIN_P9	Yes			
114	Ok		HSMC_RX_n[0]	Location	PIN_G11	Yes			
115	Ok		HSMC_RX_n[1]	Location	PIN_J12	Yes			
116	Ok		HSMC_RX_n[2]	Location	PIN_F10	Yes			
117	Ok		HSMC_RX_n[3]	Location	PIN_J9	Yes			
118	Ok		HSMC_RX_n[4]	Location	PIN_K8	Yes			
119	Ok		HSMC_RX_n[5]	Location	PIN_H7	Yes			
120	Ok		HSMC_RX_n[	Location	PIN_G8	Yes			

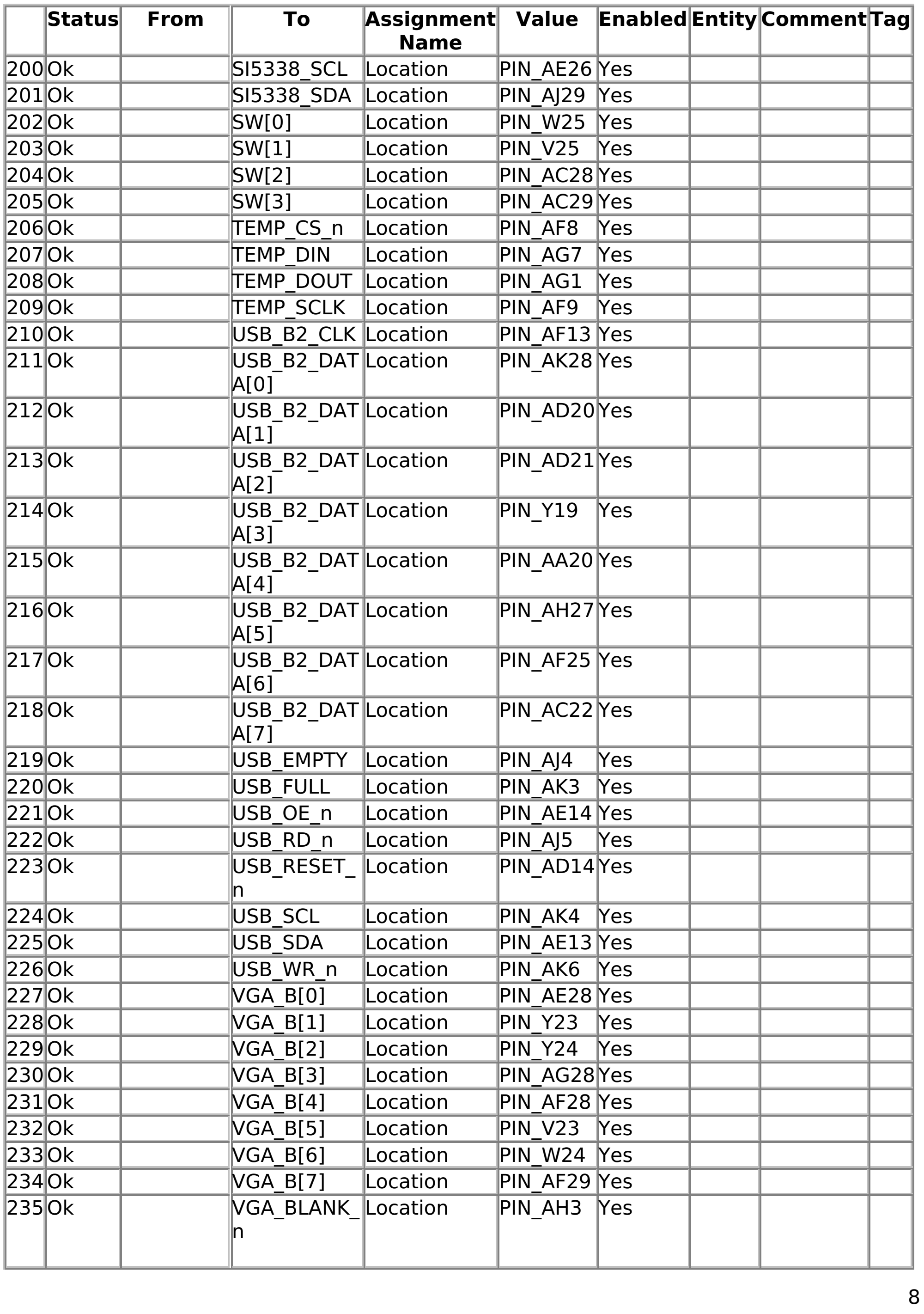
4

	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
			6]						
121	Ok		HSMC_RX_n[7]	Location	PIN_F8	Yes			
122	Ok		HSMC_RX_n[8]	Location	PIN_E11	Yes			
123	Ok		HSMC_RX_n[9]	Location	PIN_B5	Yes			
124	Ok		HSMC_RX_n[10]	Location	PIN_D9	Yes			
125	Ok		HSMC_RX_n[11]	Location	PIN_D12	Yes			
126	Ok		HSMC_RX_n[12]	Location	PIN_D10	Yes			
127	Ok		HSMC_RX_n[13]	Location	PIN_B12	Yes			
128	Ok		HSMC_RX_n[14]	Location	PIN_E13	Yes			
129	Ok		HSMC_RX_n[15]	Location	PIN_G13	Yes			
130	Ok		HSMC_RX_n[16]	Location	PIN_F14	Yes			
131	Ok		HSMC_RX_p[0]	Location	PIN_G12	Yes			
132	Ok		HSMC_RX_p[1]	Location	PIN_K12	Yes			
133	Ok		HSMC_RX_p[2]	Location	PIN_G10	Yes			
134	Ok		HSMC_RX_p[3]	Location	PIN_J10	Yes			
135	Ok		HSMC_RX_p[4]	Location	PIN_K7	Yes			
136	Ok		HSMC_RX_p[5]	Location	PIN_J7	Yes			
137	Ok		HSMC_RX_p[6]	Location	PIN_H8	Yes			
138	Ok		HSMC_RX_p[7]	Location	PIN_F9	Yes			
139	Ok		HSMC_RX_p[8]	Location	PIN_F11	Yes			
140	Ok		HSMC_RX_p[9]	Location	PIN_B6	Yes			
141	Ok		HSMC_RX_p[10]	Location	PIN_E9	Yes			
142	Ok		HSMC_RX_p[11]	Location	PIN_E12	Yes			
143	Ok		HSMC_RX_p[12]	Location	PIN_D11	Yes			
144	Ok		HSMC_RX_p[	Location	PIN_C13	Yes			

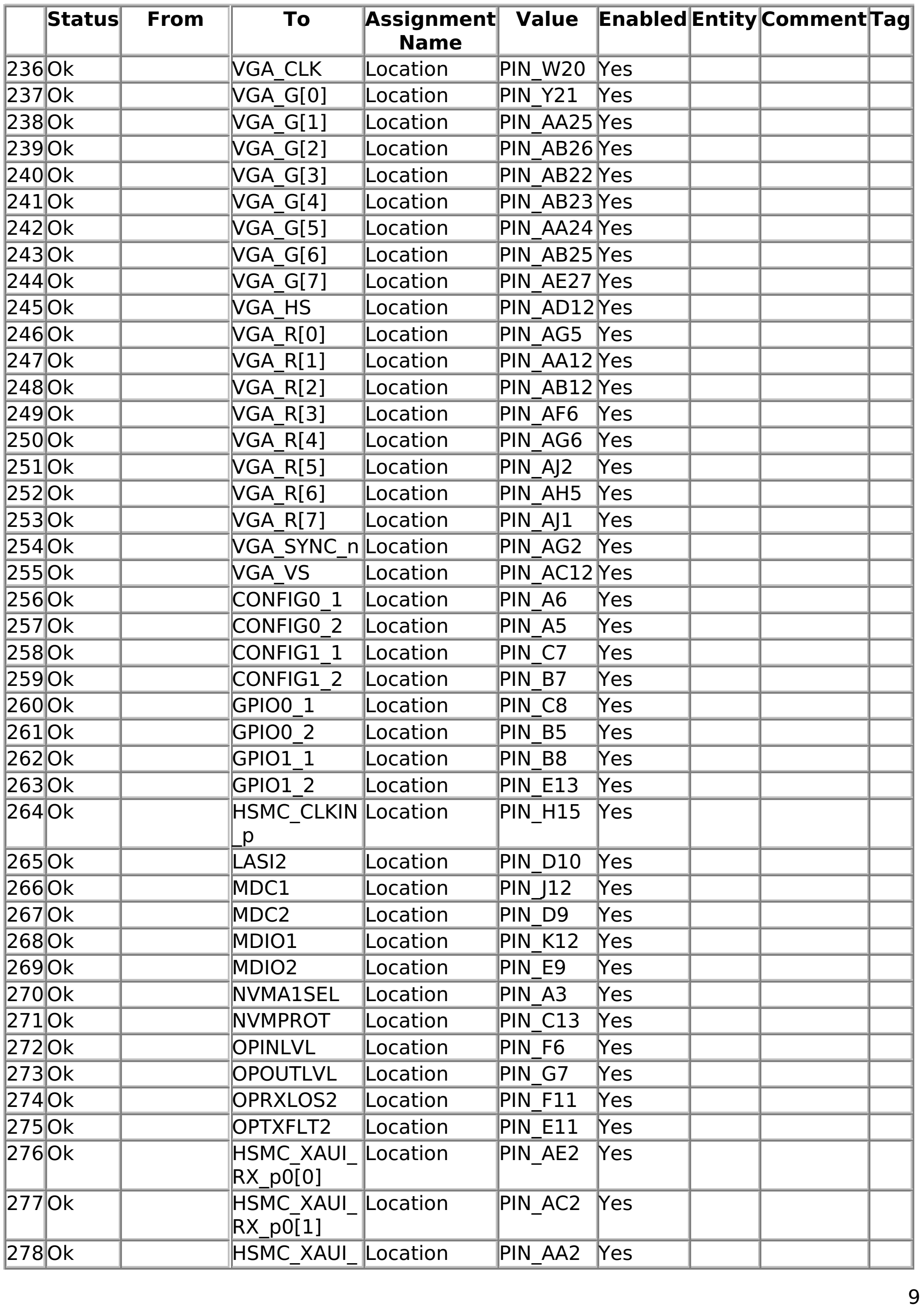
	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
			13]						
145	Ok		HSMC_RX_p[14]	Location	PIN_F13	Yes			
146	Ok		HSMC_RX_p[15]	Location	PIN_H14	Yes			
147	Ok		HSMC_RX_p[16]	Location	PIN_F15	Yes			
148	Ok		HSMC_SCL	Location	PIN_AA28	Yes			
149	Ok		HSMC_SDA	Location	PIN_AE29	Yes			
150	Ok		HSMC_TX_n[0]	Location	PIN_A8	Yes			
151	Ok		HSMC_TX_n[1]	Location	PIN_D7	Yes			
152	Ok		HSMC_TX_n[2]	Location	PIN_F6	Yes			
153	Ok		HSMC_TX_n[3]	Location	PIN_C5	Yes			
154	Ok		HSMC_TX_n[4]	Location	PIN_C4	Yes			
155	Ok		HSMC_TX_n[5]	Location	PIN_E2	Yes			
156	Ok		HSMC_TX_n[6]	Location	PIN_D4	Yes			
157	Ok		HSMC_TX_n[7]	Location	PIN_B3	Yes			
158	Ok		HSMC_TX_n[8]	Location	PIN_D1	Yes			
159	Ok		HSMC_TX_n[9]	Location	PIN_C2	Yes			
160	Ok		HSMC_TX_n[10]	Location	PIN_B1	Yes			
161	Ok		HSMC_TX_n[11]	Location	PIN_A3	Yes			
162	Ok		HSMC_TX_n[12]	Location	PIN_A5	Yes			
163	Ok		HSMC_TX_n[13]	Location	PIN_B7	Yes			
164	Ok		HSMC_TX_n[14]	Location	PIN_B8	Yes			
165	Ok		HSMC_TX_n[15]	Location	PIN_B11	Yes			
166	Ok		HSMC_TX_n[16]	Location	PIN_A13	Yes			
167	Ok		HSMC_TX_p[0]	Location	PIN_A9	Yes			
168	Ok		HSMC_TX_p[1]	Location	PIN_E8	Yes			
169	Ok		HSMC_TX_p[	Location	PIN_G7	Yes			



	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
			2]						
170	Ok		HSMC_TX_p[3]	Location	PIN_D6	Yes			
171	Ok		HSMC_TX_p[4]	Location	PIN_D5	Yes			
172	Ok		HSMC_TX_p[5]	Location	PIN_E3	Yes			
173	Ok		HSMC_TX_p[6]	Location	PIN_E4	Yes			
174	Ok		HSMC_TX_p[7]	Location	PIN_C3	Yes			
175	Ok		HSMC_TX_p[8]	Location	PIN_E1	Yes			
176	Ok		HSMC_TX_p[9]	Location	PIN_D2	Yes			
177	Ok		HSMC_TX_p[10]	Location	PIN_B2	Yes			
178	Ok		HSMC_TX_p[11]	Location	PIN_A4	Yes			
179	Ok		HSMC_TX_p[12]	Location	PIN_A6	Yes			
180	Ok		HSMC_TX_p[13]	Location	PIN_C7	Yes			
181	Ok		HSMC_TX_p[14]	Location	PIN_C8	Yes			
182	Ok		HSMC_TX_p[15]	Location	PIN_C12	Yes			
183	Ok		HSMC_TX_p[16]	Location	PIN_B13	Yes			
184	Ok		IRDA_RXD	Location	PIN_AH2	Yes			
185	Ok		KEY[0]	Location	PIN_AE9	Yes			
186	Ok		KEY[1]	Location	PIN_AE12	Yes			
187	Ok		KEY[2]	Location	PIN_AD9	Yes			
188	Ok		KEY[3]	Location	PIN_AD11	Yes			
189	Ok		LED[0]	Location	PIN_AF10	Yes			
190	Ok		LED[1]	Location	PIN_AD10	Yes			
191	Ok		LED[2]	Location	PIN_AE11	Yes			
192	Ok		LED[3]	Location	PIN_AD7	Yes			
193	Ok		OSC_50_B3B	Location	PIN_AF14	Yes			
194	Ok		OSC_50_B4A	Location	PIN_AA16	Yes			
195	Ok		OSC_50_B5B	Location	PIN_Y26	Yes			
196	Ok		OSC_50_B8A	Location	PIN_K14	Yes			
197	Ok		PCIE_PERST_n	Location	PIN_W22	Yes			
198	Ok		PCIE_WAKE_n	Location	PIN_W21	Yes			
199	Ok		RESET_n	Location	PIN_AD27	Yes			

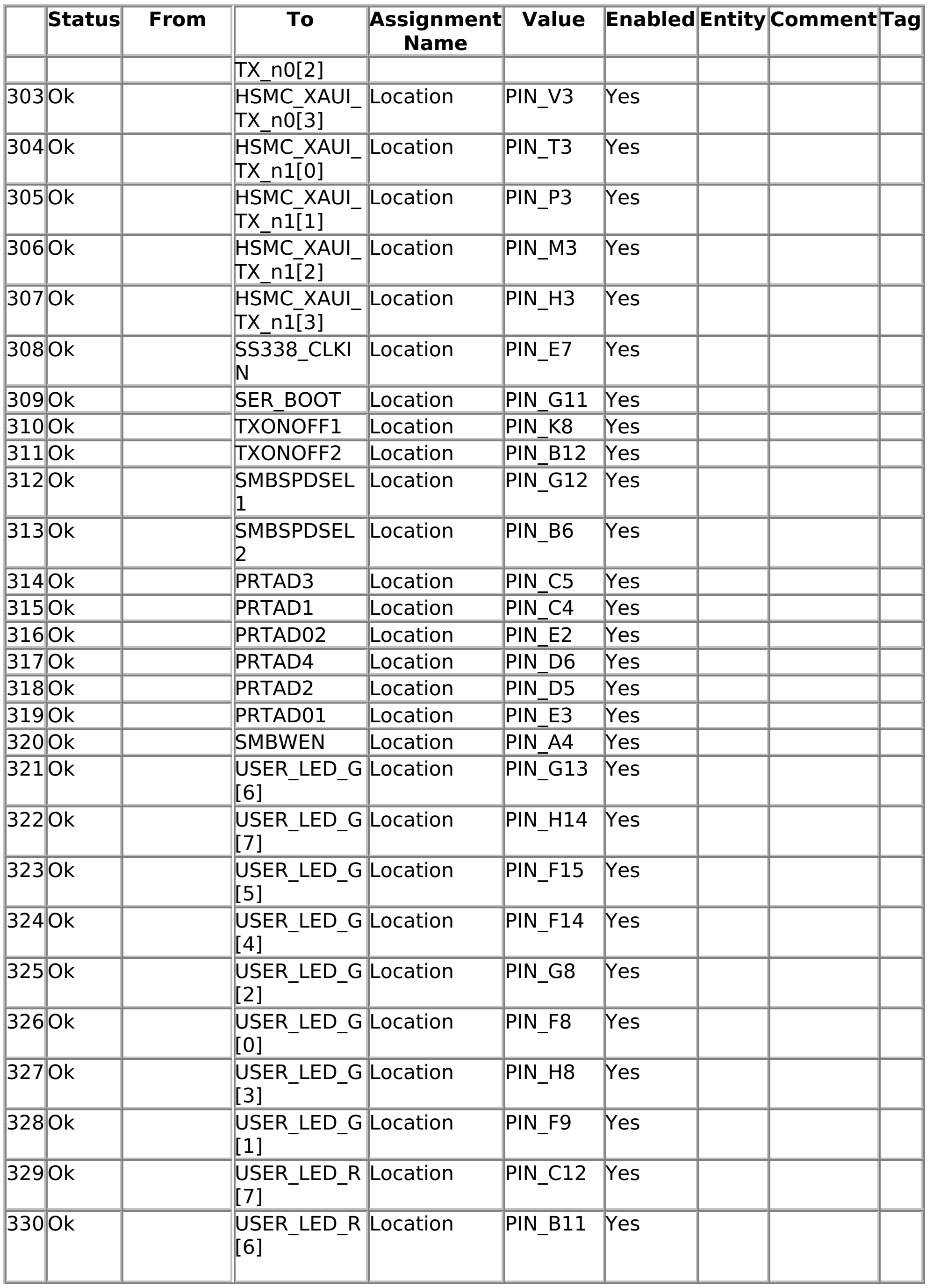


	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
200	Ok		SI5338_SCL	Location	PIN_AE26	Yes			
201	Ok		SI5338_SDA	Location	PIN_AJ29	Yes			
202	Ok		SW[0]	Location	PIN_W25	Yes			
203	Ok		SW[1]	Location	PIN_V25	Yes			
204	Ok		SW[2]	Location	PIN_AC28	Yes			
205	Ok		SW[3]	Location	PIN_AC29	Yes			
206	Ok		TEMP_CS_n	Location	PIN_AF8	Yes			
207	Ok		TEMP_DIN	Location	PIN_AG7	Yes			
208	Ok		TEMP_DOUT	Location	PIN_AG1	Yes			
209	Ok		TEMP_SCLK	Location	PIN_AF9	Yes			
210	Ok		USB_B2_CLK	Location	PIN_AF13	Yes			
211	Ok		USB_B2_DATA[0]	Location	PIN_AK28	Yes			
212	Ok		USB_B2_DATA[1]	Location	PIN_AD20	Yes			
213	Ok		USB_B2_DATA[2]	Location	PIN_AD21	Yes			
214	Ok		USB_B2_DATA[3]	Location	PIN_Y19	Yes			
215	Ok		USB_B2_DATA[4]	Location	PIN_AA20	Yes			
216	Ok		USB_B2_DATA[5]	Location	PIN_AH27	Yes			
217	Ok		USB_B2_DATA[6]	Location	PIN_AF25	Yes			
218	Ok		USB_B2_DATA[7]	Location	PIN_AC22	Yes			
219	Ok		USB_EMPTY	Location	PIN_AJ4	Yes			
220	Ok		USB_FULL	Location	PIN_AK3	Yes			
221	Ok		USB_OE_n	Location	PIN_AE14	Yes			
222	Ok		USB_RD_n	Location	PIN_AJ5	Yes			
223	Ok		USB_RESET_n	Location	PIN_AD14	Yes			
224	Ok		USB_SCL	Location	PIN_AK4	Yes			
225	Ok		USB_SDA	Location	PIN_AE13	Yes			
226	Ok		USB_WR_n	Location	PIN_AK6	Yes			
227	Ok		VGA_B[0]	Location	PIN_AE28	Yes			
228	Ok		VGA_B[1]	Location	PIN_Y23	Yes			
229	Ok		VGA_B[2]	Location	PIN_Y24	Yes			
230	Ok		VGA_B[3]	Location	PIN_AG28	Yes			
231	Ok		VGA_B[4]	Location	PIN_AF28	Yes			
232	Ok		VGA_B[5]	Location	PIN_V23	Yes			
233	Ok		VGA_B[6]	Location	PIN_W24	Yes			
234	Ok		VGA_B[7]	Location	PIN_AF29	Yes			
235	Ok		VGA_BLANK_n	Location	PIN_AH3	Yes			

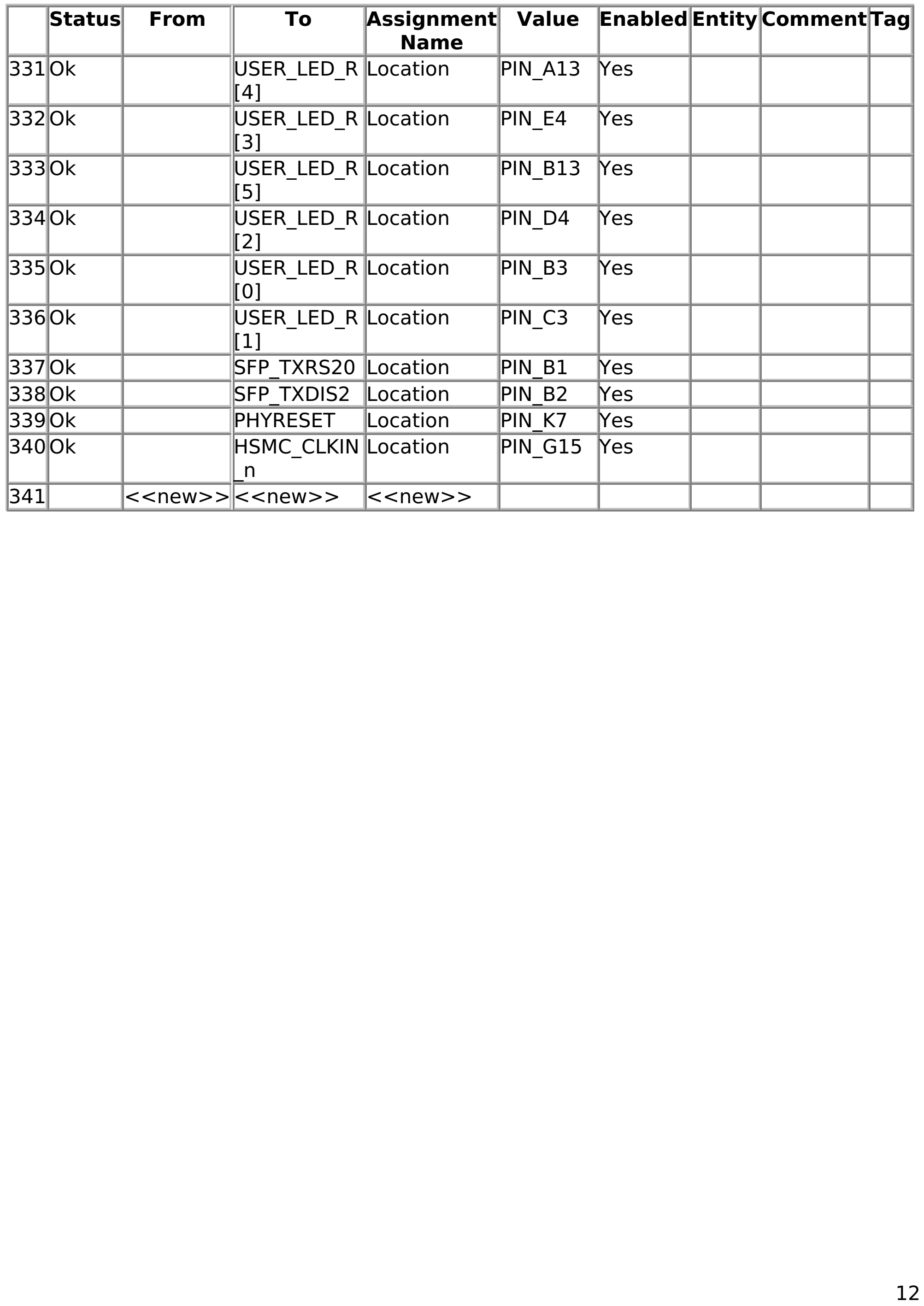


	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
236	Ok		VGA_CLK	Location	PIN_W20	Yes			
237	Ok		VGA_G[0]	Location	PIN_Y21	Yes			
238	Ok		VGA_G[1]	Location	PIN_AA25	Yes			
239	Ok		VGA_G[2]	Location	PIN_AB26	Yes			
240	Ok		VGA_G[3]	Location	PIN_AB22	Yes			
241	Ok		VGA_G[4]	Location	PIN_AB23	Yes			
242	Ok		VGA_G[5]	Location	PIN_AA24	Yes			
243	Ok		VGA_G[6]	Location	PIN_AB25	Yes			
244	Ok		VGA_G[7]	Location	PIN_AE27	Yes			
245	Ok		VGA_HS	Location	PIN_AD12	Yes			
246	Ok		VGA_R[0]	Location	PIN_AG5	Yes			
247	Ok		VGA_R[1]	Location	PIN_AA12	Yes			
248	Ok		VGA_R[2]	Location	PIN_AB12	Yes			
249	Ok		VGA_R[3]	Location	PIN_AF6	Yes			
250	Ok		VGA_R[4]	Location	PIN_AG6	Yes			
251	Ok		VGA_R[5]	Location	PIN_AJ2	Yes			
252	Ok		VGA_R[6]	Location	PIN_AH5	Yes			
253	Ok		VGA_R[7]	Location	PIN_AJ1	Yes			
254	Ok		VGA_SYNC_n	Location	PIN_AG2	Yes			
255	Ok		VGA_VS	Location	PIN_AC12	Yes			
256	Ok		CONFIG0_1	Location	PIN_A6	Yes			
257	Ok		CONFIG0_2	Location	PIN_A5	Yes			
258	Ok		CONFIG1_1	Location	PIN_C7	Yes			
259	Ok		CONFIG1_2	Location	PIN_B7	Yes			
260	Ok		GPIO0_1	Location	PIN_C8	Yes			
261	Ok		GPIO0_2	Location	PIN_B5	Yes			
262	Ok		GPIO1_1	Location	PIN_B8	Yes			
263	Ok		GPIO1_2	Location	PIN_E13	Yes			
264	Ok		HSMC_CLKIN_p	Location	PIN_H15	Yes			
265	Ok		LASI2	Location	PIN_D10	Yes			
266	Ok		MDC1	Location	PIN_J12	Yes			
267	Ok		MDC2	Location	PIN_D9	Yes			
268	Ok		MDIO1	Location	PIN_K12	Yes			
269	Ok		MDIO2	Location	PIN_E9	Yes			
270	Ok		NVMA1SEL	Location	PIN_A3	Yes			
271	Ok		NVMPROT	Location	PIN_C13	Yes			
272	Ok		OPINLVL	Location	PIN_F6	Yes			
273	Ok		OPOUTLVL	Location	PIN_G7	Yes			
274	Ok		OPRXLOS2	Location	PIN_F11	Yes			
275	Ok		OPTXFLT2	Location	PIN_E11	Yes			
276	Ok		HSMC_XAUI_RX_p0[0]	Location	PIN_AE2	Yes			
277	Ok		HSMC_XAUI_RX_p0[1]	Location	PIN_AC2	Yes			
278	Ok		HSMC_XAUI_	Location	PIN_AA2	Yes			

	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
			RX_p0[2]						
279	Ok		HSMC_XAUI_RX_p0[3]	Location	PIN_W2	Yes			
280	Ok		HSMC_XAUI_RX_p1[0]	Location	PIN_U2	Yes			
281	Ok		HSMC_XAUI_RX_p1[1]	Location	PIN_R2	Yes			
282	Ok		HSMC_XAUI_RX_p1[2]	Location	PIN_N2	Yes			
283	Ok		HSMC_XAUI_RX_p1[3]	Location	PIN_J2	Yes			
284	Ok		HSMC_XAUI_TX_p0[0]	Location	PIN_AD4	Yes			
285	Ok		HSMC_XAUI_TX_p0[1]	Location	PIN_AB4	Yes			
286	Ok		HSMC_XAUI_TX_p0[2]	Location	PIN_Y4	Yes			
287	Ok		HSMC_XAUI_TX_p0[3]	Location	PIN_V4	Yes			
288	Ok		HSMC_XAUI_TX_p1[0]	Location	PIN_T4	Yes			
289	Ok		HSMC_XAUI_TX_p1[1]	Location	PIN_P4	Yes			
290	Ok		HSMC_XAUI_TX_p1[2]	Location	PIN_M4	Yes			
291	Ok		HSMC_XAUI_TX_p1[3]	Location	PIN_H4	Yes			
292	Ok		HSMC_XAUI_RX_n0[0]	Location	PIN_AE1	Yes			
293	Ok		HSMC_XAUI_RX_n0[1]	Location	PIN_AC1	Yes			
294	Ok		HSMC_XAUI_RX_n0[2]	Location	PIN_AA1	Yes			
295	Ok		HSMC_XAUI_RX_n0[3]	Location	PIN_W1	Yes			
296	Ok		HSMC_XAUI_RX_n1[0]	Location	PIN_U1	Yes			
297	Ok		HSMC_XAUI_RX_n1[1]	Location	PIN_R1	Yes			
298	Ok		HSMC_XAUI_RX_n1[2]	Location	PIN_N1	Yes			
299	Ok		HSMC_XAUI_RX_n1[3]	Location	PIN_J1	Yes			
300	Ok		HSMC_XAUI_TX_n0[0]	Location	PIN_AD3	Yes			
301	Ok		HSMC_XAUI_TX_n0[1]	Location	PIN_AB3	Yes			
302	Ok		HSMC_XAUI_	Location	PIN_Y3	Yes			



	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
			TX_n0[2]						
303	Ok		HSMC_XAUI_TX_n0[3]	Location	PIN_V3	Yes			
304	Ok		HSMC_XAUI_TX_n1[0]	Location	PIN_T3	Yes			
305	Ok		HSMC_XAUI_TX_n1[1]	Location	PIN_P3	Yes			
306	Ok		HSMC_XAUI_TX_n1[2]	Location	PIN_M3	Yes			
307	Ok		HSMC_XAUI_TX_n1[3]	Location	PIN_H3	Yes			
308	Ok		SS338_CLKIN	Location	PIN_E7	Yes			
309	Ok		SER_BOOT	Location	PIN_G11	Yes			
310	Ok		TXONOFF1	Location	PIN_K8	Yes			
311	Ok		TXONOFF2	Location	PIN_B12	Yes			
312	Ok		SMBSPDSEL1	Location	PIN_G12	Yes			
313	Ok		SMBSPDSEL2	Location	PIN_B6	Yes			
314	Ok		PRTAD3	Location	PIN_C5	Yes			
315	Ok		PRTAD1	Location	PIN_C4	Yes			
316	Ok		PRTAD02	Location	PIN_E2	Yes			
317	Ok		PRTAD4	Location	PIN_D6	Yes			
318	Ok		PRTAD2	Location	PIN_D5	Yes			
319	Ok		PRTAD01	Location	PIN_E3	Yes			
320	Ok		SMBWEN	Location	PIN_A4	Yes			
321	Ok		USER_LED_G[6]	Location	PIN_G13	Yes			
322	Ok		USER_LED_G[7]	Location	PIN_H14	Yes			
323	Ok		USER_LED_G[5]	Location	PIN_F15	Yes			
324	Ok		USER_LED_G[4]	Location	PIN_F14	Yes			
325	Ok		USER_LED_G[2]	Location	PIN_G8	Yes			
326	Ok		USER_LED_G[0]	Location	PIN_F8	Yes			
327	Ok		USER_LED_G[3]	Location	PIN_H8	Yes			
328	Ok		USER_LED_G[1]	Location	PIN_F9	Yes			
329	Ok		USER_LED_R[7]	Location	PIN_C12	Yes			
330	Ok		USER_LED_R[6]	Location	PIN_B11	Yes			



	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
331	Ok		USER_LED_R [4]	Location	PIN_A13	Yes			
332	Ok		USER_LED_R [3]	Location	PIN_E4	Yes			
333	Ok		USER_LED_R [5]	Location	PIN_B13	Yes			
334	Ok		USER_LED_R [2]	Location	PIN_D4	Yes			
335	Ok		USER_LED_R [0]	Location	PIN_B3	Yes			
336	Ok		USER_LED_R [1]	Location	PIN_C3	Yes			
337	Ok		SFP_TXRS20	Location	PIN_B1	Yes			
338	Ok		SFP_TXDIS2	Location	PIN_B2	Yes			
339	Ok		PHYRESET	Location	PIN_K7	Yes			
340	Ok		HSMC_CLKIN _n	Location	PIN_G15	Yes			
341		<<new>>	<<new>>	<<new>>					

7 Bibliography

1. https://www.altera.com/en_US/pdfs/literature/ug/xcvr_user_guide.pdf

2. https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/10gbps_mac.pdf

3. https://www.altera.com/en_US/pdfs/literature/manual/mnl_avalon_spec.pdf

4. http://www.cs.columbia.edu/~sedwards/classes/2014/4840/reports/10Gbs.pdf