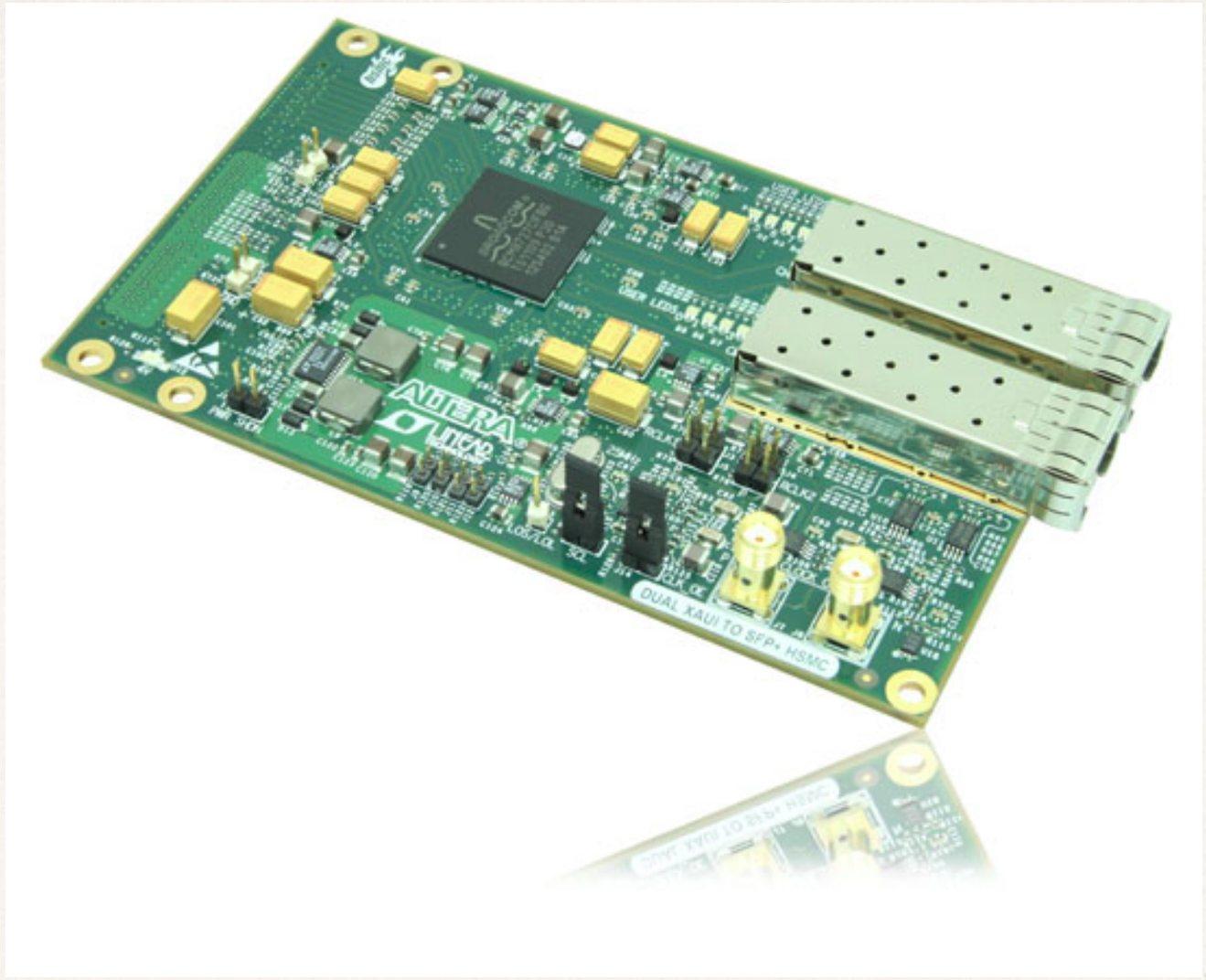


10Gb/s Ethernet Platform Implementation

John Chaiyasarikul, Yumeng Xu, Shuguan Yang, Jian Zhong

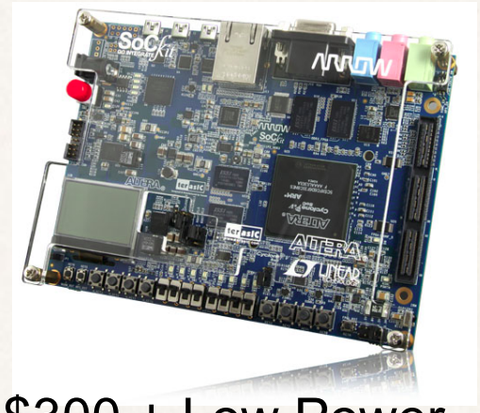
May 31st, 2014

APPLICATIONS

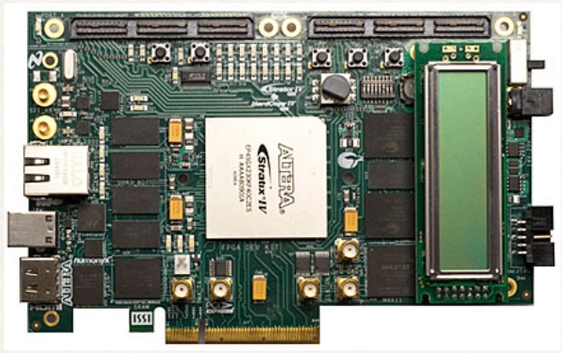
- DATA Centers
- Finance
- Clusters

$1.6k – $3.2k

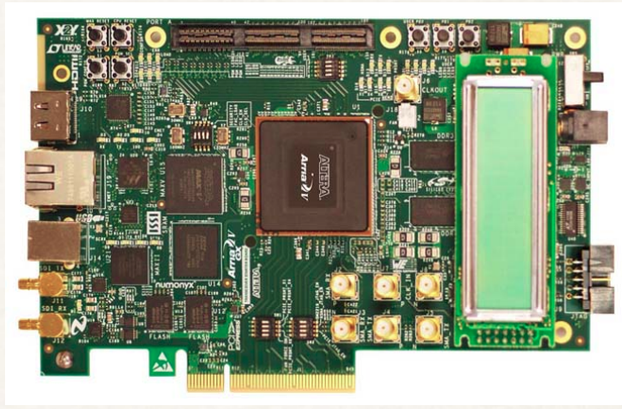
Dev Kit Unavailable



$300 + Low Power



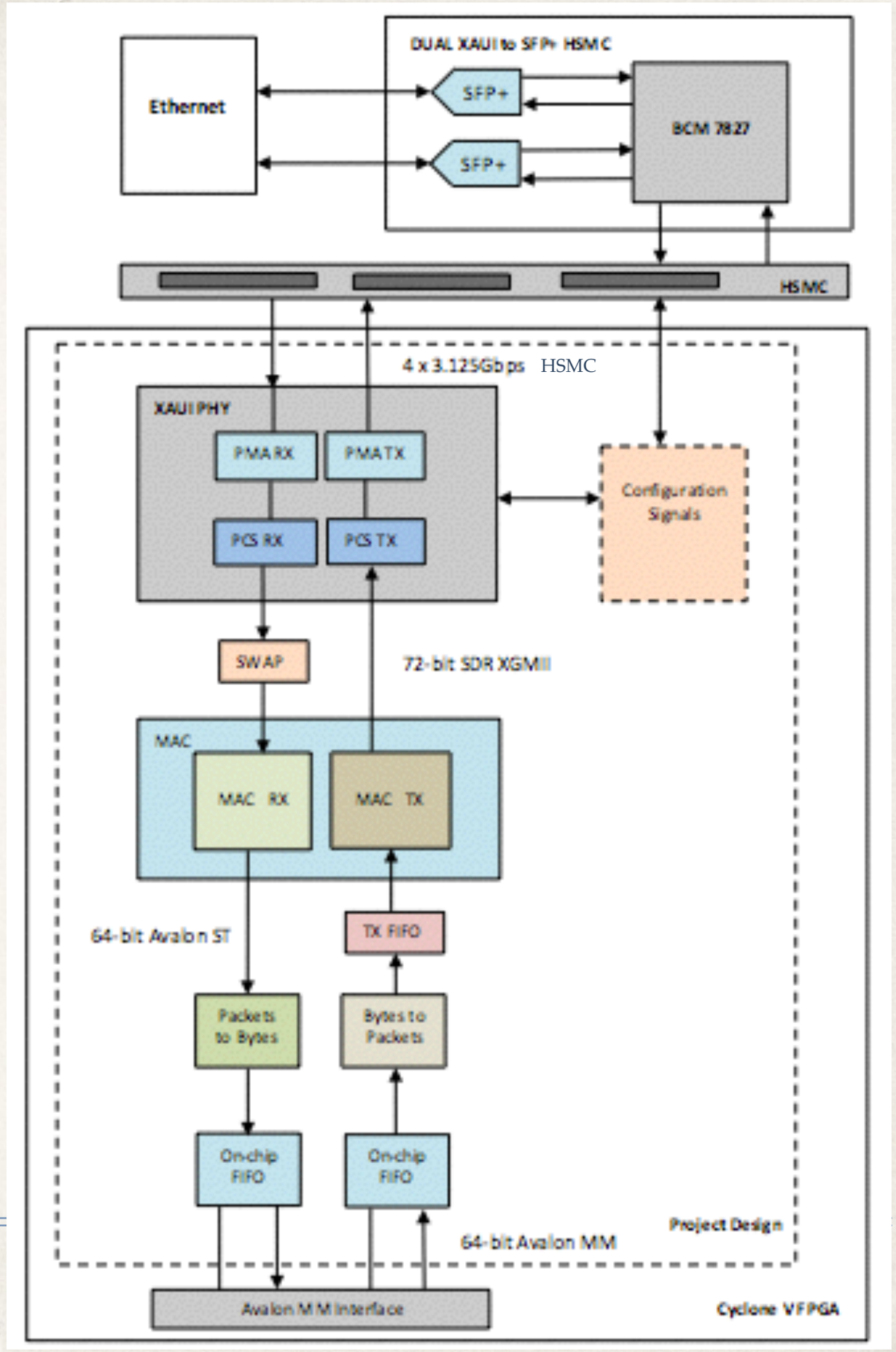
$3.6k



Dev Kit Unavailable

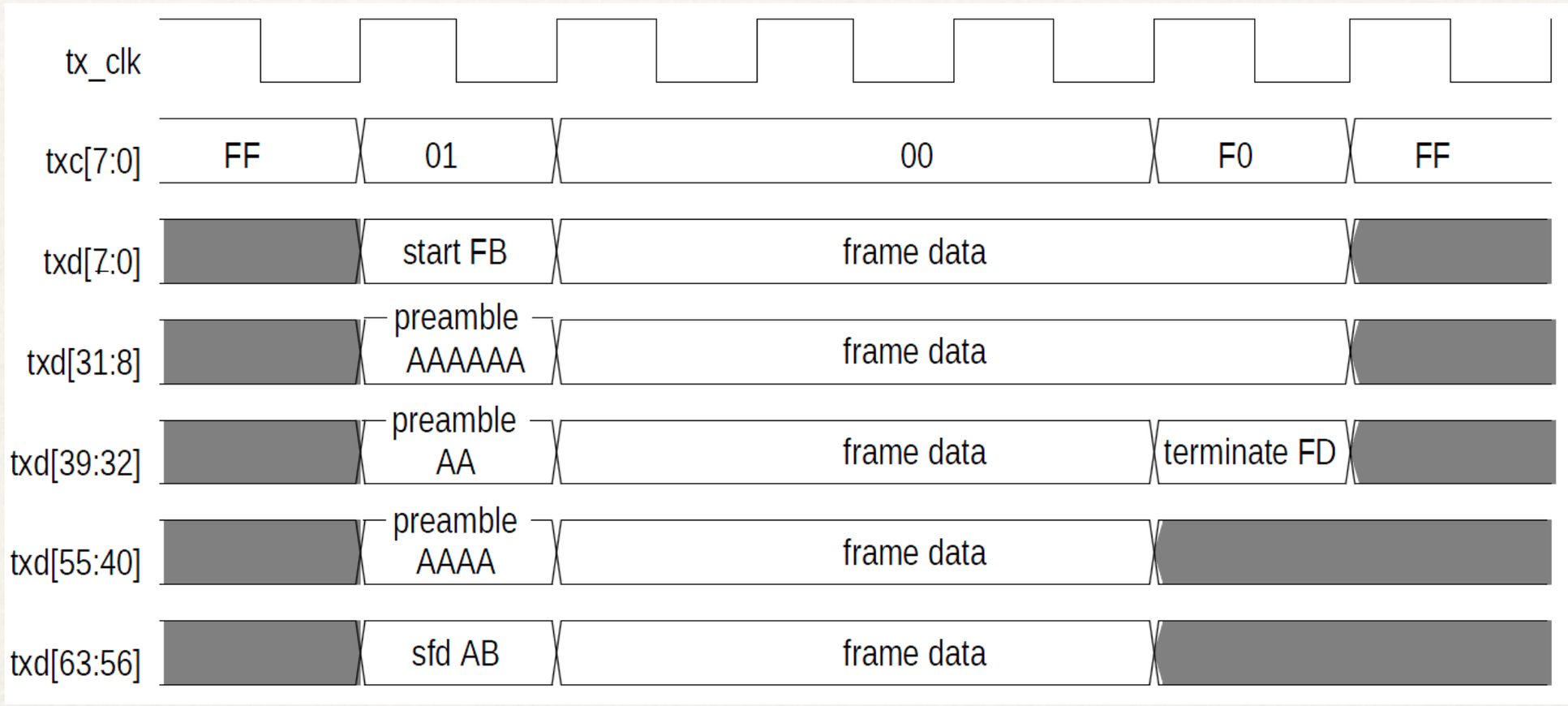
Motivation / Significance

Architecture of the Design



I. HSMC - 4 lanes @ 3.125Gb/s

II. XAUI/XGMII - 8 Bytes with 8bits control signals

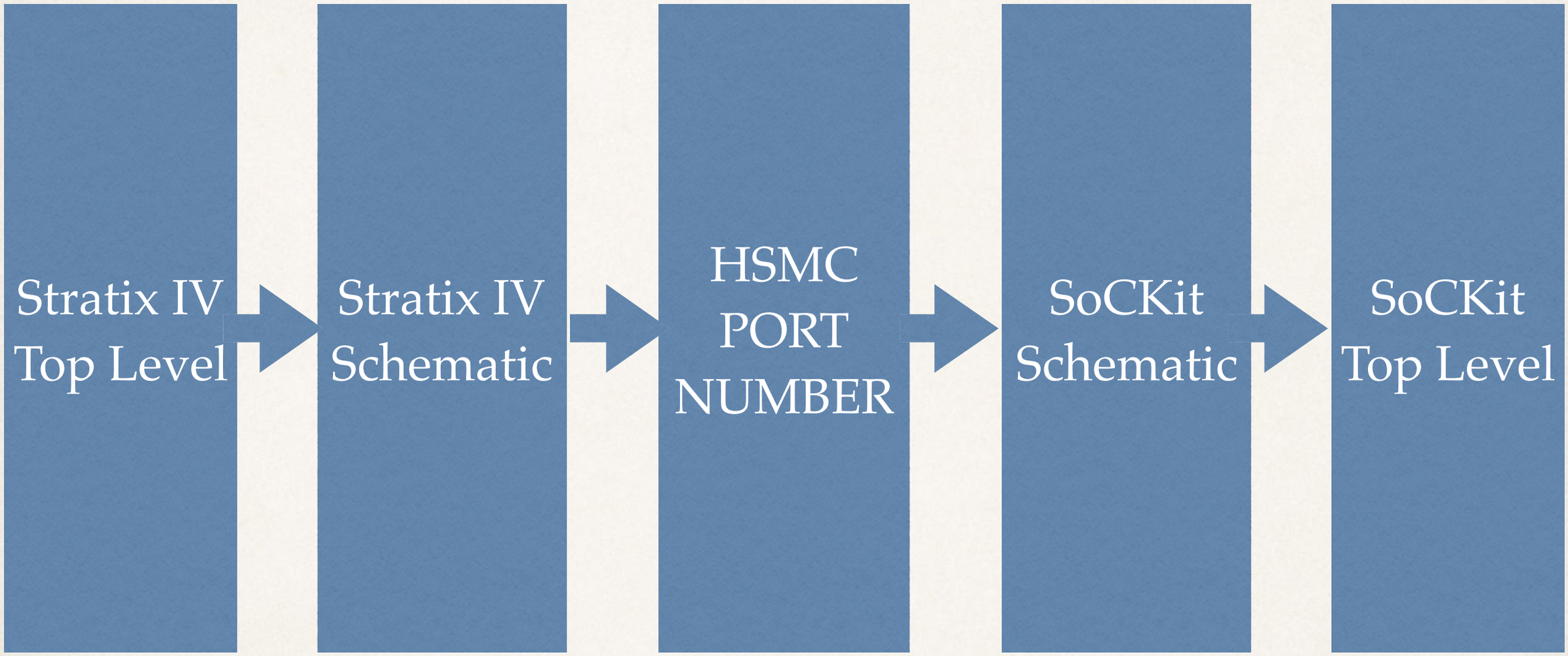


Interfaces

III. Avalon ST - Unidirectional flow of data @ 156.25MHz.

IV. Avalon MM -A standard address-based read/write interface typical of master-slave connections.

Interfaces



Pin Assignment Matching

Configuration

```
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;
```

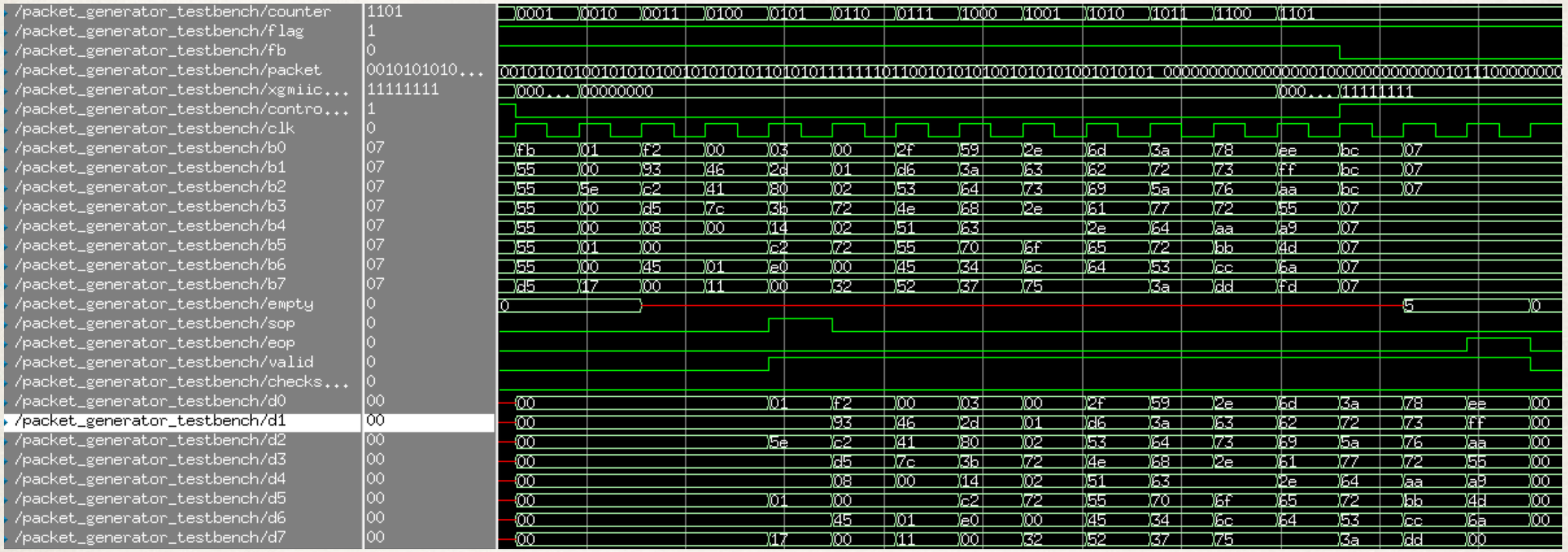
```
assign MDIO1 = !MDOEN1? MDO1 : 1'bz;
assign MDIN1 = MDIO1;
assign MDC2   = 1'bz;
assign MDIO2 = 1'bz;
assign PHYRESET =  KEY[1];
assign STOPMON  = ~KEY[2];

{PRTAD4,PRTAD3,PRTAD2,PRTAD1,PRTAD01} = 5'b00000;
assign  PRTAD02 = 1'b0;
assign TXONOFF1 = 1'b1;
assign TXONOFF2 = 1'b1;
assign OPOUTLVL = 1'b0;
assign OPINLVL  = 1'b1;
assign USER_LED_R = 8'b00001111;
assign USER_LED_G = 8'b11110000;
```

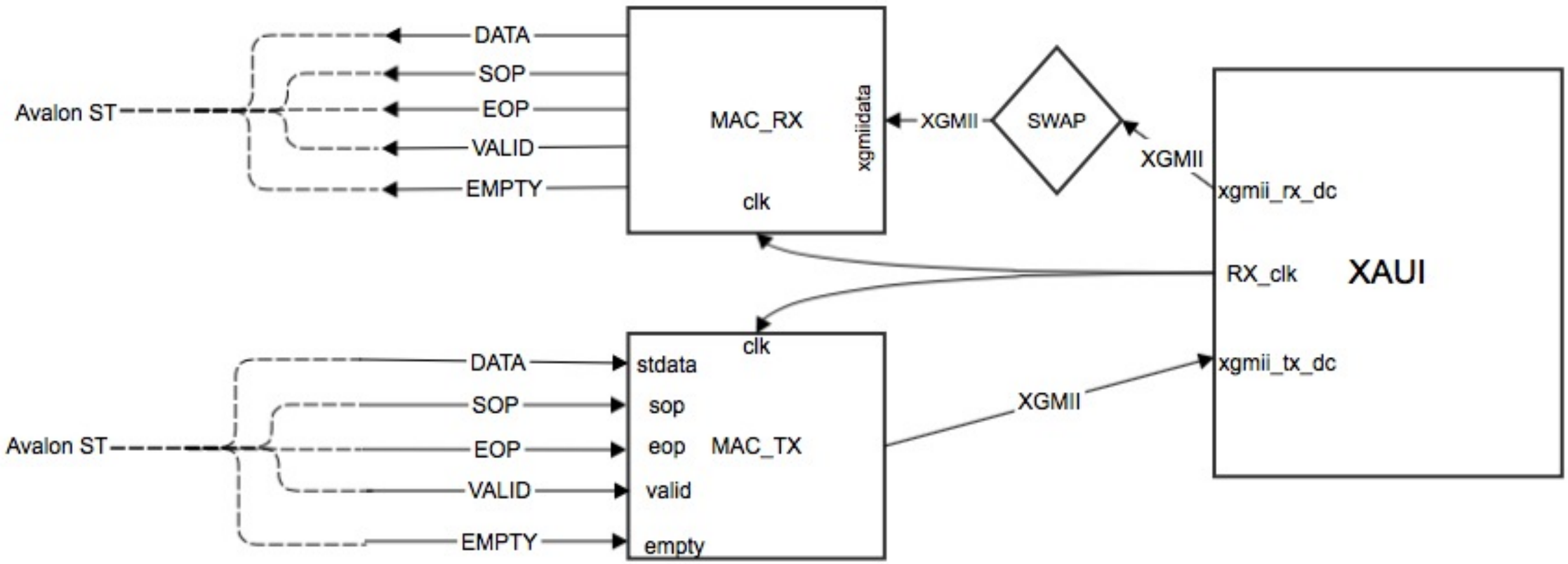
XAUI PHY		HSMC
xaui_rx_seriel_data [3:0]	<<	HSMC_XAUI_RX_p0[3:0]
xaui_tx_seriel_data [3:0]	>>	HSMC_XAUI_TX_p0[3:0]
phy_mgmt_clk_reset	<<	!RESET_N
phy_mgmt_clk	<<	OSC_50_B4A
rx_ready	>>	LED[2]
tx_ready	>>	LED[3]
xgmii_rx_clk	>----------|	
xgmii_tx_clk	<----------|	
phy_mgmt_address [8:0]	<< 3'h000	
phy_mgmt_write	<< 0	
phy_mgmt_writeata [31:0]	<< 8'hFFFFFFFF	
phy_mgmt_read	<< 0	
phy_mgmt_readdata[31:0]	<< 8'h00000000	

- Fast/Transparent MAC
- 3 clock cycle latency rx
- 2 clock cycle latency tx

- Convert between XGMII and Avalon ST Interfaces
- Calculate Checksum
- Check or Append Checksum



MAC


Avalon ST
DATA
SOP
EOP
VALID
EMPTY
MAC_RX
clk
xgmiidata
XGMII
SWAP
XGMII
xgmii_rx_dc
RX_clk
XAUI
clk
xgmii_tx_dc
XGMII
Avalon ST
DATA
SOP
EOP
VALID
EMPTY
stdata
sop
eop
valid
empty
MAC_TX

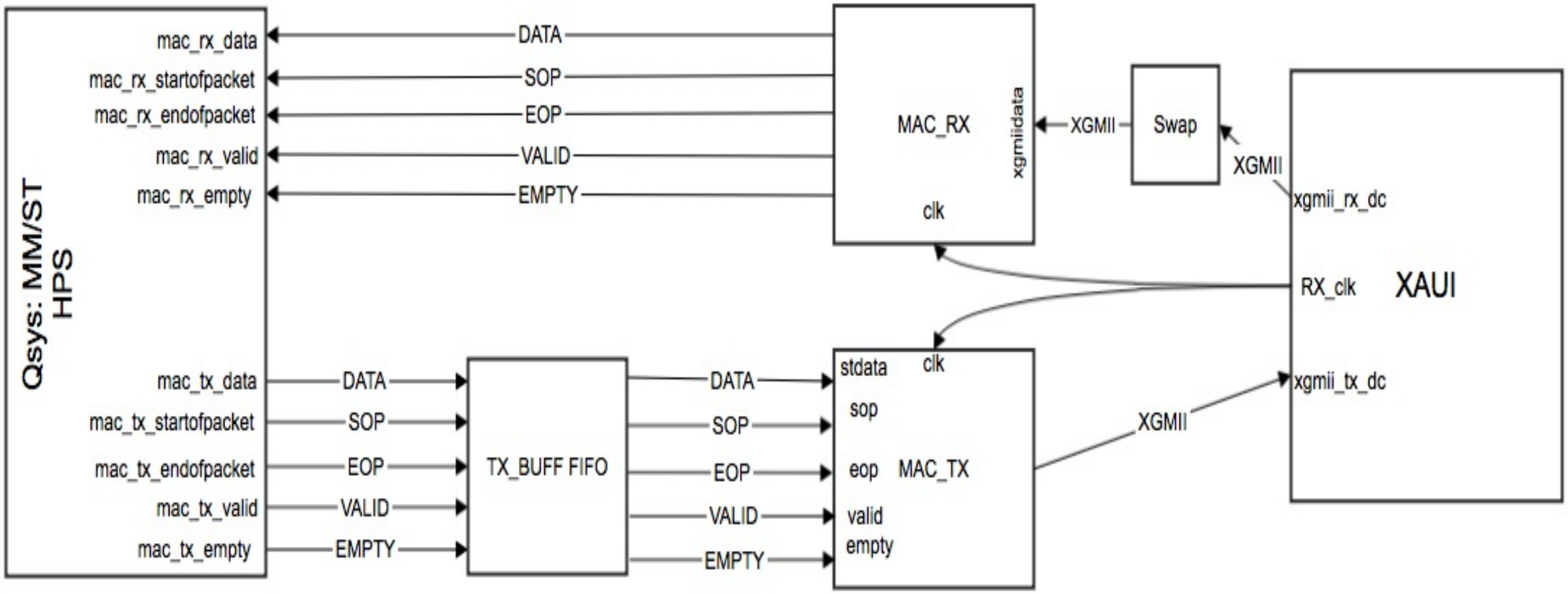
MAC

```
1101 | 01011101 000   ← Append zeros equal to the size of the checksum
       1101
       ------
       1000           ← Perform XOR until more bits are needed
       1101
       ------
       01011          ← Omit leading zeros and bring down next bit
        1101
        ------
        01101
         1101         ←  Repeat steps until run out of bits
         -------------
         0000 0100
              1101
              ------
              1001
              1101
              ------
              01000
               1101
               ------
               0101   ← Remainder is the checksum
```

```
1101 | 01011101 101   ← Append checksum to data
       1101
       ------
       1000           ← Compute as before
       1101
       ------
       01011
        1101
        ------
        01101
         1101
         -------------
         0000 0110
              1101
              ------
              1011
              1101
              ------
              01101
               1101
               ------
               0000   ← Checksum will be zero
```

For parallel computing refer to www.cypress.com/?docID=31573 and easics.com

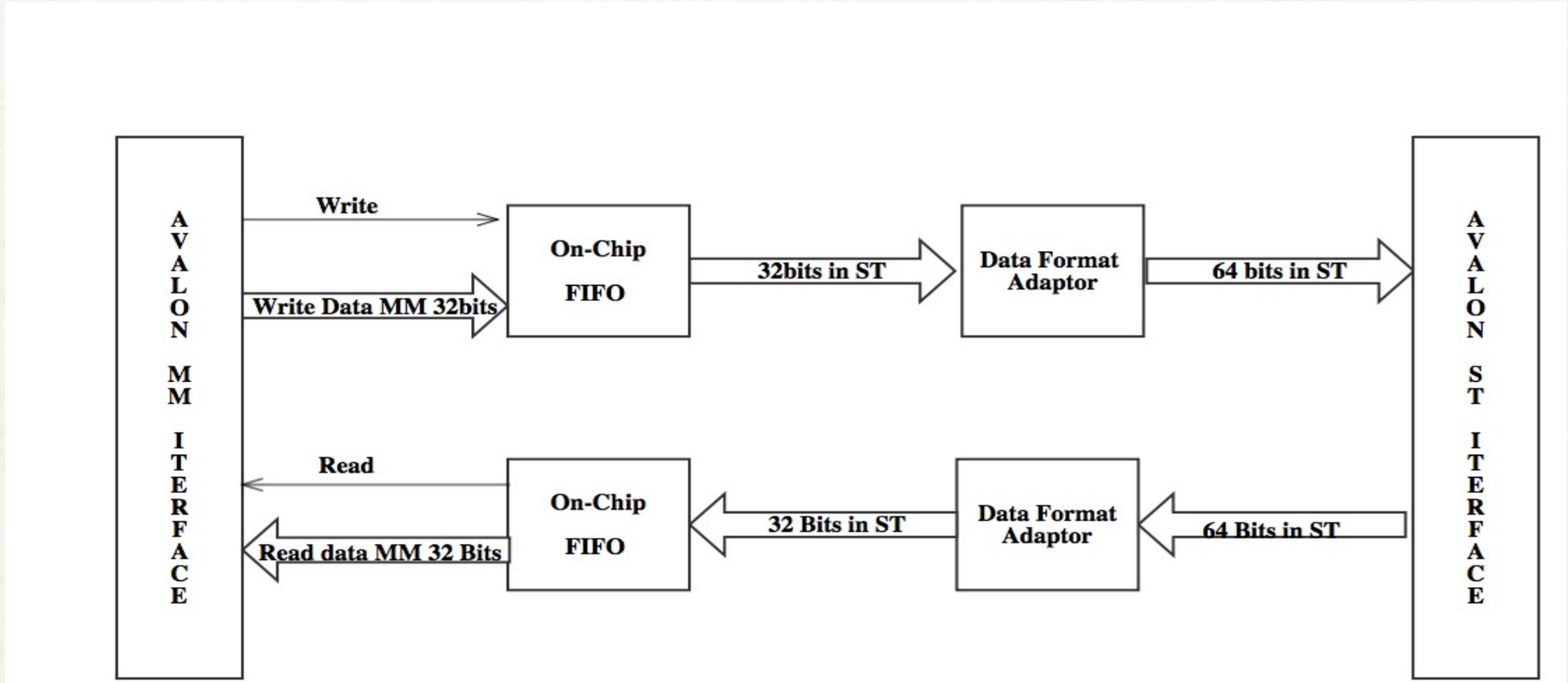
CRC-32



Connections

Problem: Data is received and transmitted with a bus width of 64 bits of data, MM interface supports only 32 bits with no control signals

- Solution for Control
- 7a7a7a7a = SOP
- 7b7b7b7b = EOP
- 7d7d7d7d = escape
- 7e7e7e7e = no data
- Dual Clock FIFO



ST / MM

No/Incorrect documentation
New Board / incompatible IP cores
Burnt Daughter board
Dual Port cant be compiled
CRC computation
Timing constraints
MM and HPS
Unreliable Cables

Challenges / Roadblocks