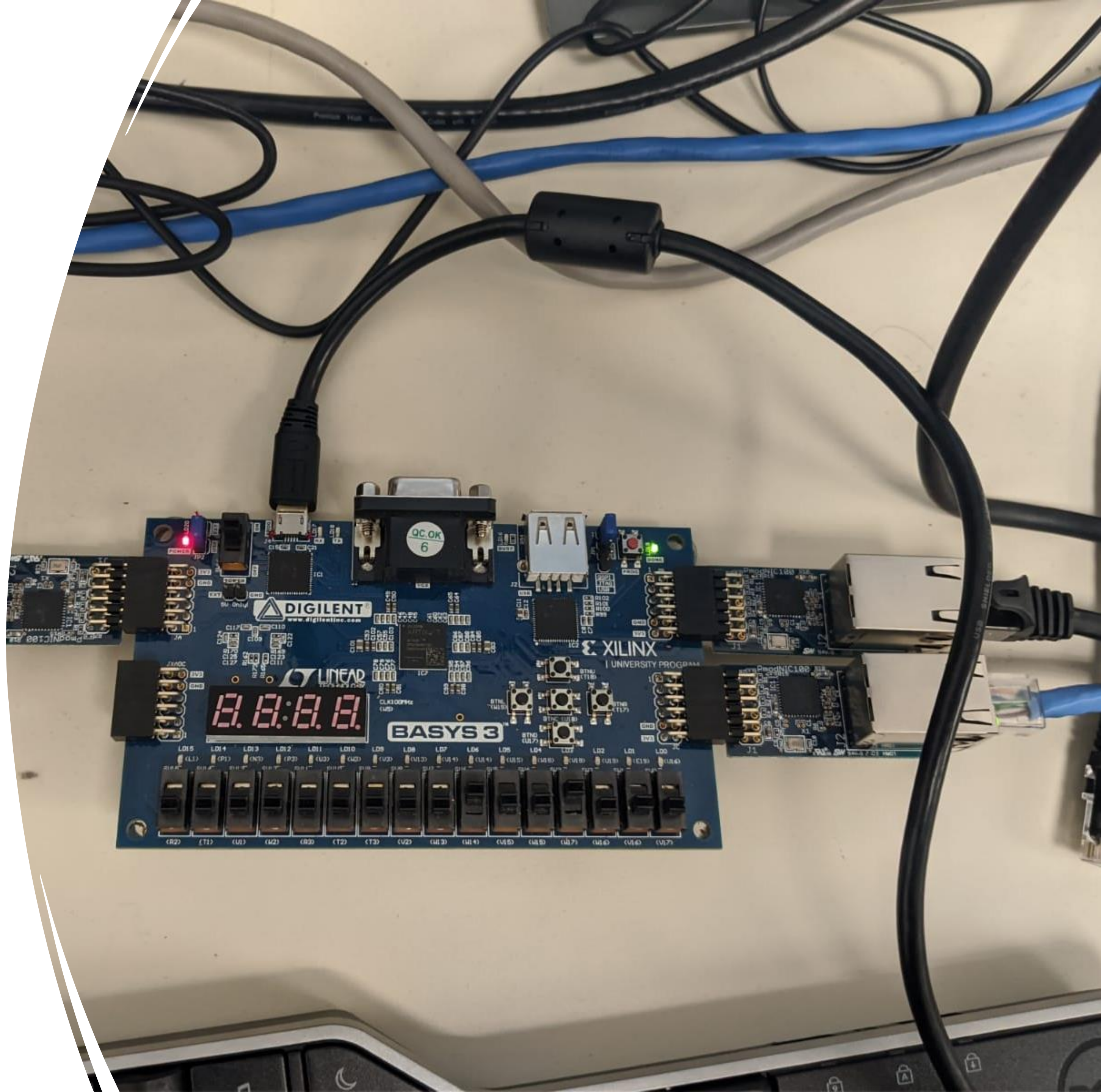
An FPGA-based Network Switch using SPI Protocol

HW&SW co-design class project

Andrea Pinto
12/16/2021

Introduction

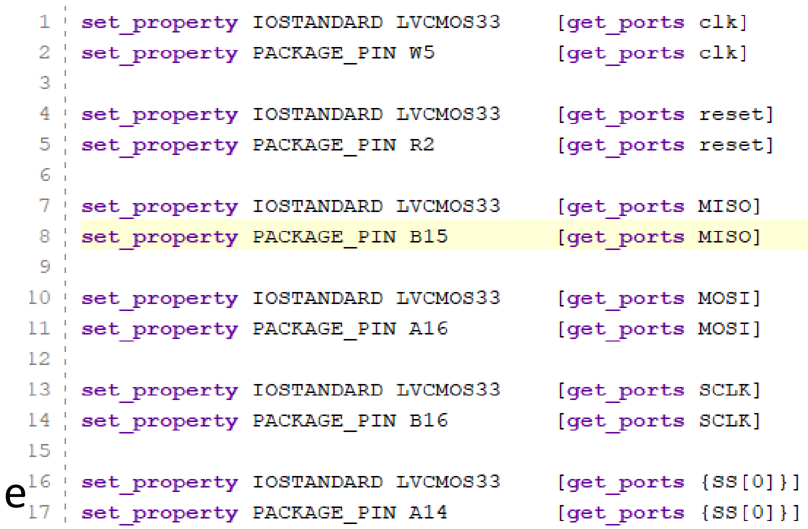
- Basys3
- PMOD Nic 100 Ethernet ports (ENC424J600 processor)
- SPI protocol
- Vivado 2018.3
- Xilinx SDK



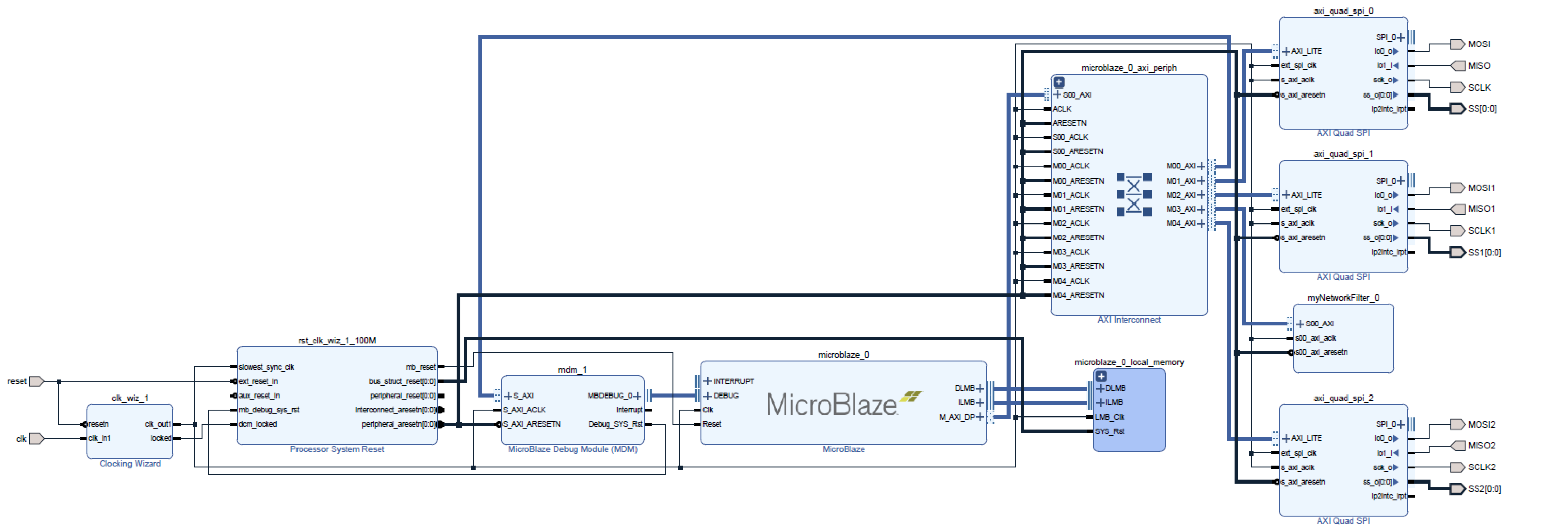
Block Design

- Microblaze
- Clock widzard
- Axi_quad_spi
- Network filter module

Constraints file



```
set_property IOSTANDARD LVCMOS33    [get_ports clk]
set_property PACKAGE_PIN W5         [get_ports clk]

set_property IOSTANDARD LVCMOS33    [get_ports reset]
set_property PACKAGE_PIN R2         [get_ports reset]

set_property IOSTANDARD LVCMOS33    [get_ports MISO]
set_property PACKAGE_PIN B15        [get_ports MISO]

set_property IOSTANDARD LVCMOS33    [get_ports MOSI]
set_property PACKAGE_PIN A16        [get_ports MOSI]

set_property IOSTANDARD LVCMOS33    [get_ports SCLK]
set_property PACKAGE_PIN B16        [get_ports SCLK]

set_property IOSTANDARD LVCMOS33    [get_ports {SS[0]}]
set_property PACKAGE_PIN A14        [get_ports {SS[0]}]
```



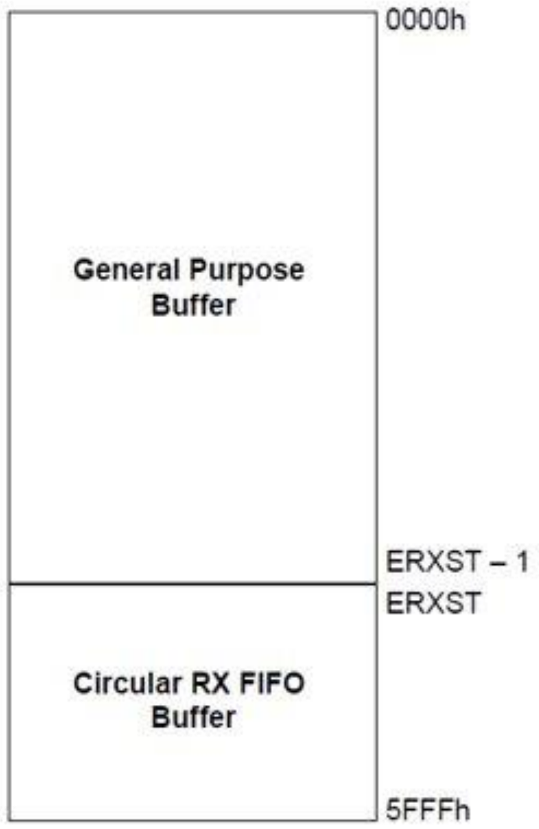
HW SW Functionalities Decomposition

- Software:
  - Setup using Xspi libraries
  - Receive packet
  - Forward packet

- Hardware
  - Filtering behaviour

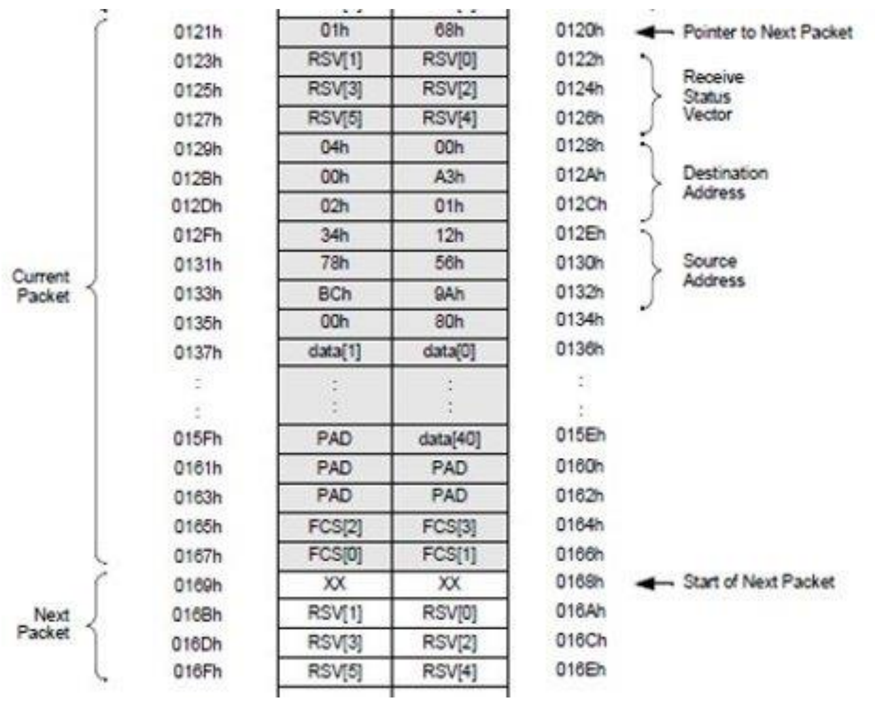
Software Challenges

- Use Xspi library
- Setup phase:
  - Reset
  - Buffer setup
  - Initializing MAC and PHY,
  - Establishing the connection through ETH.
- Implemented methods:
  - Reading and writing from the ENC424J600 memory
  - Receive and forward packet methods
  - Polling ETH interface

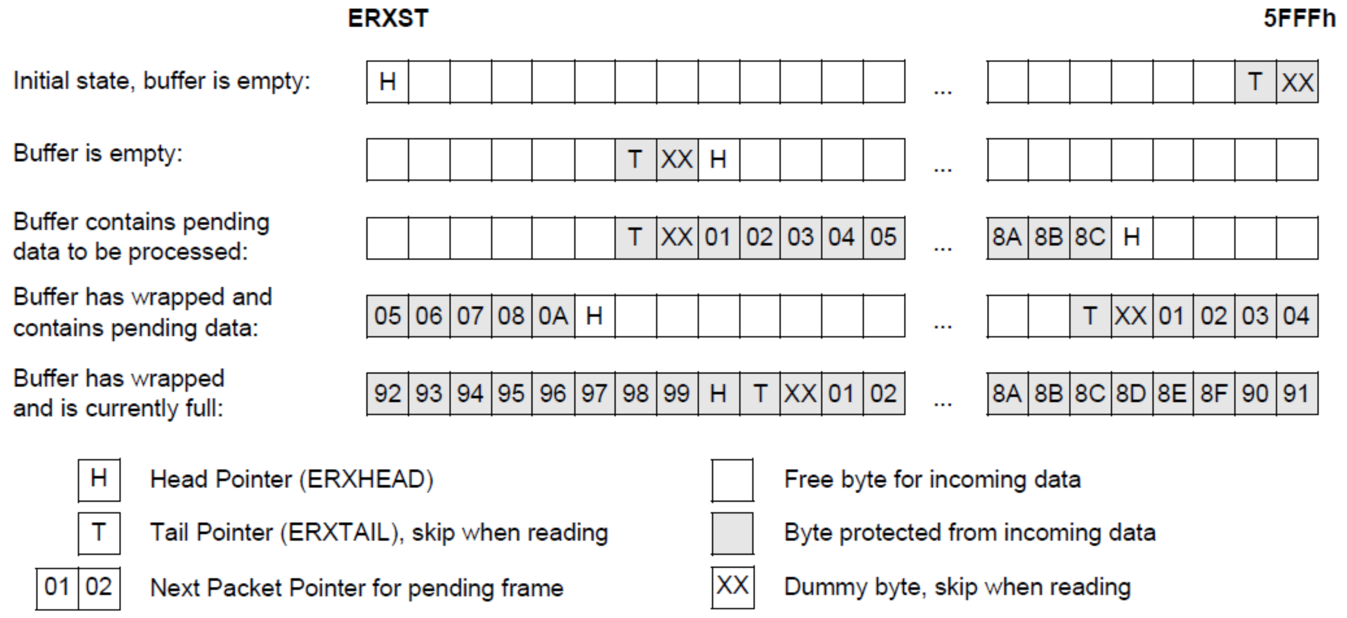


Send and receive buffer

Send packet configuration



0121h	01h	68h	0120h	← Pointer to Next Packet
0123h	RSV[1]	RSV[0]	0122h	Receive Status Vector
0125h	RSV[3]	RSV[2]	0124h	
0127h	RSV[5]	RSV[4]	0126h	
0129h	04h	00h	0128h	Destination Address
012Bh	00h	A3h	012Ah	
012Dh	02h	01h	012Ch	
012Fh	34h	12h	012Eh	Source Address
0131h	78h	56h	0130h	
0133h	BCh	9Ah	0132h	
0135h	00h	80h	0134h	
0137h	data[1]	data[0]	0136h	
:	:	:	:	
015Fh	PAD	data[40]	015Eh	
0161h	PAD	PAD	0160h	
0163h	PAD	PAD	0162h	
0165h	FCS[2]	FCS[3]	0164h	
0167h	FCS[0]	FCS[1]	0166h	
0169h	XX	XX	0168h	← Start of Next Packet
016Bh	RSV[1]	RSV[0]	016Ah	
016Dh	RSV[3]	RSV[2]	016Ch	
016Fh	RSV[5]	RSV[4]	016Eh	



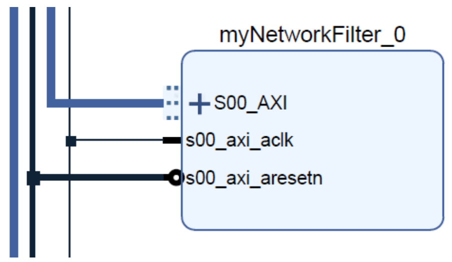
	ERXST																					5FFFh
Initial state, buffer is empty:	H													...							T	XX
Buffer is empty:							T	XX	H					...								
Buffer contains pending data to be processed:							T	XX	01	02	03	04	05	...	8A	8B	8C	H				
Buffer has wrapped and contains pending data:	05	06	07	08	0A	H								...			T	XX	01	02	03	04
Buffer has wrapped and is currently full:	92	93	94	95	96	97	98	99	H	T	XX	01	02	...	8A	8B	8C	8D	8E	8F	90	91

H	Head Pointer (ERXHEAD)	☐	Free byte for incoming data
T	Tail Pointer (ERXTAIL), skip when reading	▨	Byte protected from incoming data
01 02	Next Packet Pointer for pending frame	XX	Dummy byte, skip when reading

head and tail management

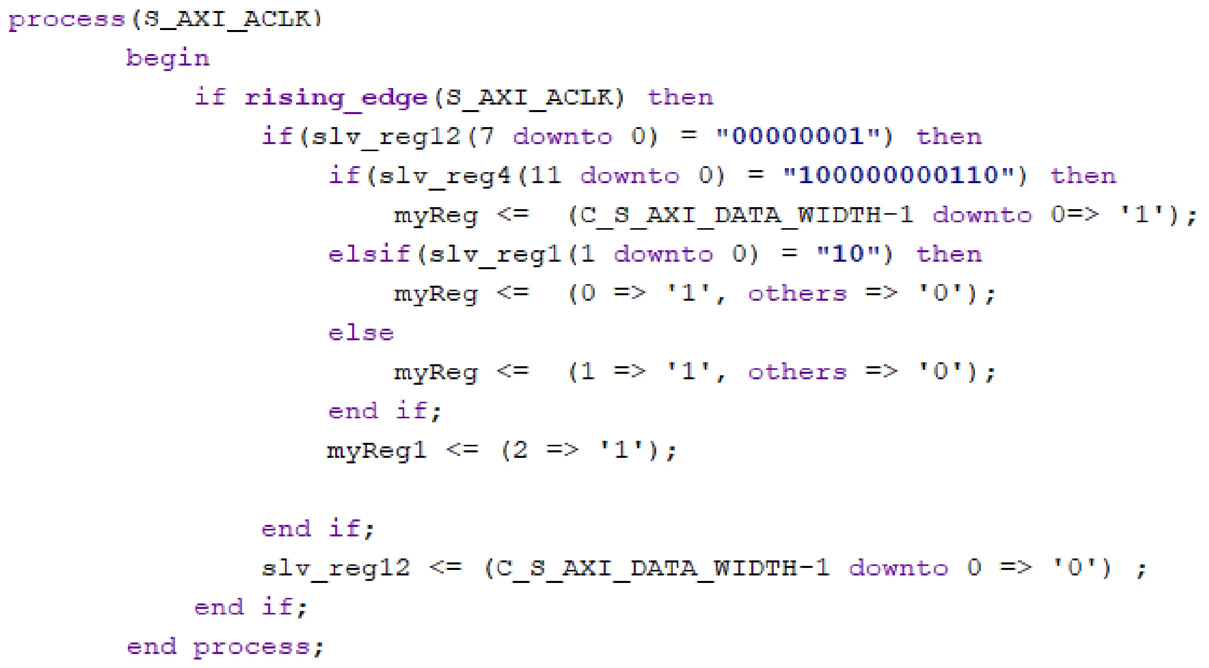
Receive buffer

Hardware Challenges



Hardware Filter diagram block

- Register size 4 bytes
- 16 registers available
- Register12 is a sync register between HW and SW
- MyReg is a temporary register copied inside register15 to indicate wich action the software has to perform



```
process(S_AXI_ACLK)
       begin
           if rising_edge(S_AXI_ACLK) then
               if(slv_reg12(7 downto 0) = "00000001") then
                   if(slv_reg4(11 downto 0) = "100000000110") then
                       myReg <=  (C_S_AXI_DATA_WIDTH-1 downto 0=> '1');
                   elsif(slv_reg1(1 downto 0) = "10") then
                       myReg <=  (0 => '1', others => '0');
                   else
                       myReg <=  (1 => '1', others => '0');
                   end if;
                   myReg1 <= (2 => '1');

               end if;
               slv_reg12 <= (C_S_AXI_DATA_WIDTH-1 downto 0 => '0') ;
           end if;
       end process;
```

VHDL filtering logic

Conclusions

- Pros
  - FPGA advantages : low latency, fast performance, low renewal costs.
  - Easy Sw Implementation using Xspi libraries

- Cons
  - Implementation straight forward in SW
  - though HW SW split.
  - Performances

Thanks !