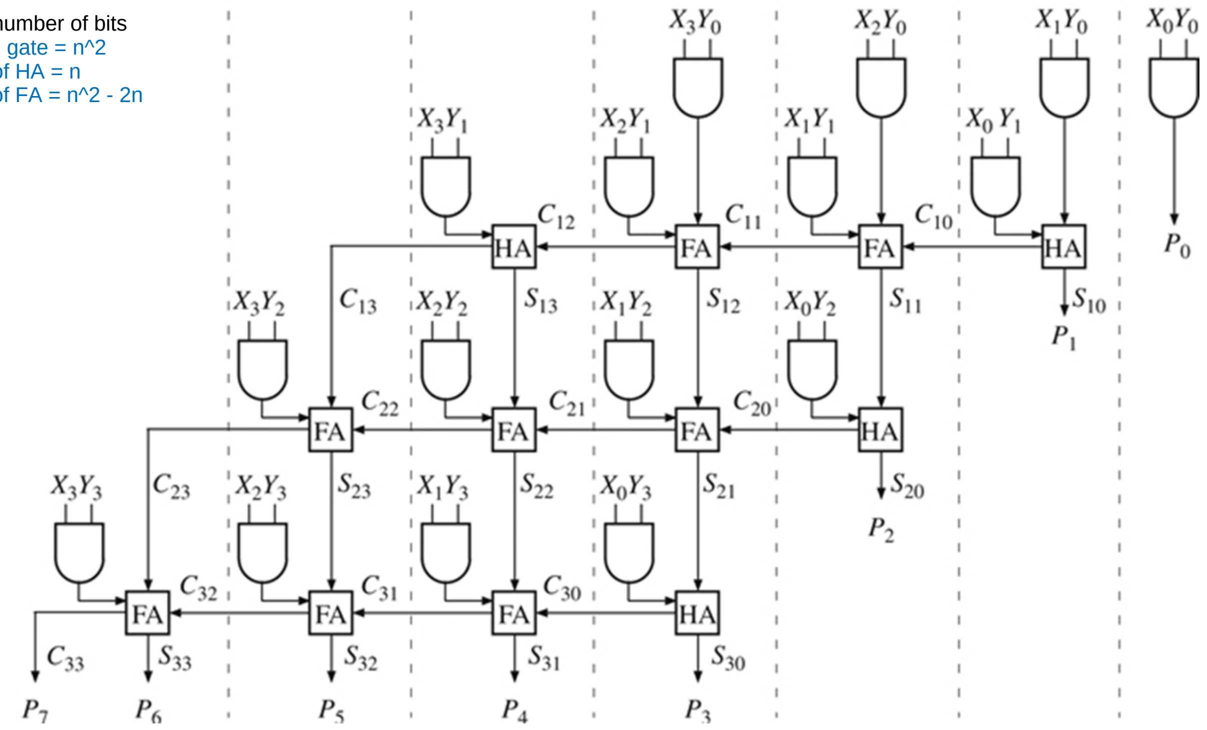
VHDL Exercise 5: Structural VHDL model, Xilinx Vivado design example

1. An array multiplier is a parallel multiplier that generates the partial products in a parallel fashion. The various partial products are added as soon as they are available. Consider the process of multiplication as shown in the figure below. Two 4-bit unsigned numbers are multiplied to generate a final product of 8 bit.

n = number of bits
AND gate = n^2
no. of HA = n
no. of FA = n^2 - 2n



a) After X and Y inputs have been applied, the carry must propagate along each row of cells and the sum must propagate from row to row. If t_{ad} is the worst-case delay through an adder (FA,HA), and t_{and} is the longest AND gate delay, what is the worst-case time to complete multiplication?

b) This 4x4bit array multiplier requires 16 AND gates, 8 full adder (FA) and 4 half-adders (HA). Consider an n-bit by n-bit array multiplier. How many AND gates, FA und HA are required?

c) If you want to get the 4x4bit multiplier implemented like in the figure above, you need to do structural coding. The **structural VHDL model** is shown below. Full adder and half adder are used as components and need to be connected accordingly to the array multiplier topology. Complete the port map statements! Also add the full adder und half adder VHDL models to your ModelSim Project. Compile the project!

```
entity Array_Mult is
  port(X, Y: in bit_vector(3 downto 0);
       P: out bit_vector(7 downto 0));
end Array_Mult;

architecture Behavioral of Array_Mult is
signal C1, C2, C3: bit_vector(3 downto 0);
signal S1, S2, S3: bit_vector(3 downto 0);
signal XY0, XY1, XY2, XY3: bit_vector(3 downto 0);
component FullAdder
```

```
  port(X, Y, Cin: in bit;
       Cout, Sum: out bit);
end component;
component HalfAdder
  port(X, Y: in bit;
       Cout, Sum: out bit);
end component;
begin
  XY0(0) <= X(0) and Y(0);   XY1(0) <= X(0) and Y(1);
  XY0(1) <= X(1) and Y(0);   XY1(1) <= X(1) and Y(1);
  XY0(2) <= X(2) and Y(0);   XY1(2) <= X(2) and Y(1);
  XY0(3) <= X(3) and Y(0);   XY1(3) <= X(3) and Y(1);

  XY2(0) <= X(0) and Y(2);   XY3(0) <= X(0) and Y(3);
  XY2(1) <= X(1) and Y(2);   XY3(1) <= X(1) and Y(3);
  XY2(2) <= X(2) and Y(2);   XY3(2) <= X(2) and Y(3);
  XY2(3) <= X(3) and Y(2);   XY3(3) <= X(3) and Y(3);

  FA1: FullAdder port map (XY0(2), XY1(1), C1(0), C1(1), S1(1));
  FA2: FullAdder port map
  FA3: FullAdder port map
  FA4: FullAdder port map
  FA5: FullAdder port map
  FA6: FullAdder port map
  FA7: FullAdder port map (S2(3), XY3(2), C3(1), C3(2), S3(2));
  FA8: FullAdder port map (C2(3), XY3(3), C3(2), C3(3), S3(3));
  HA1: HalfAdder port map
  HA2: HalfAdder port map
  HA3: HalfAdder port map
  HA4: HalfAdder port map (S2(1), XY3(0), C3(0), S3(0));

  P(0) <= XY0(0);
  P(1) <= S1(0);
  P(2) <= S2(0);
  P(3) <= S3(0);
  P(4) <= S3(1);
  P(5) <= S3(2);
  P(6) <= S3(3);
  P(7) <= C3(3);
end Behavioral;

-- Full Adder and half adder entity and architecture descriptions
-- should be in the project
```

d) Write a self-checking VHDL test bench to verify the functionality on at least four input examples!

2. Install Xilinx Vivado WebPACK 2014.2 with SDK (see Xilinx_Vivado_Installation Guide.pdf).

3. Load project Basic_IPI_Design from file Basic_IPI_Design_Zynq7020_2014p2.zip! Follow instructions 1-16 of Basic_Design_Zedboard_Vivado.pdf. Answer questions Q1 and Q2!

class: Wed,Nov/15