



# TOC – 3ª sesión de Laboratorio

Multiplicador de números de 4x4-bits



# Objetivo

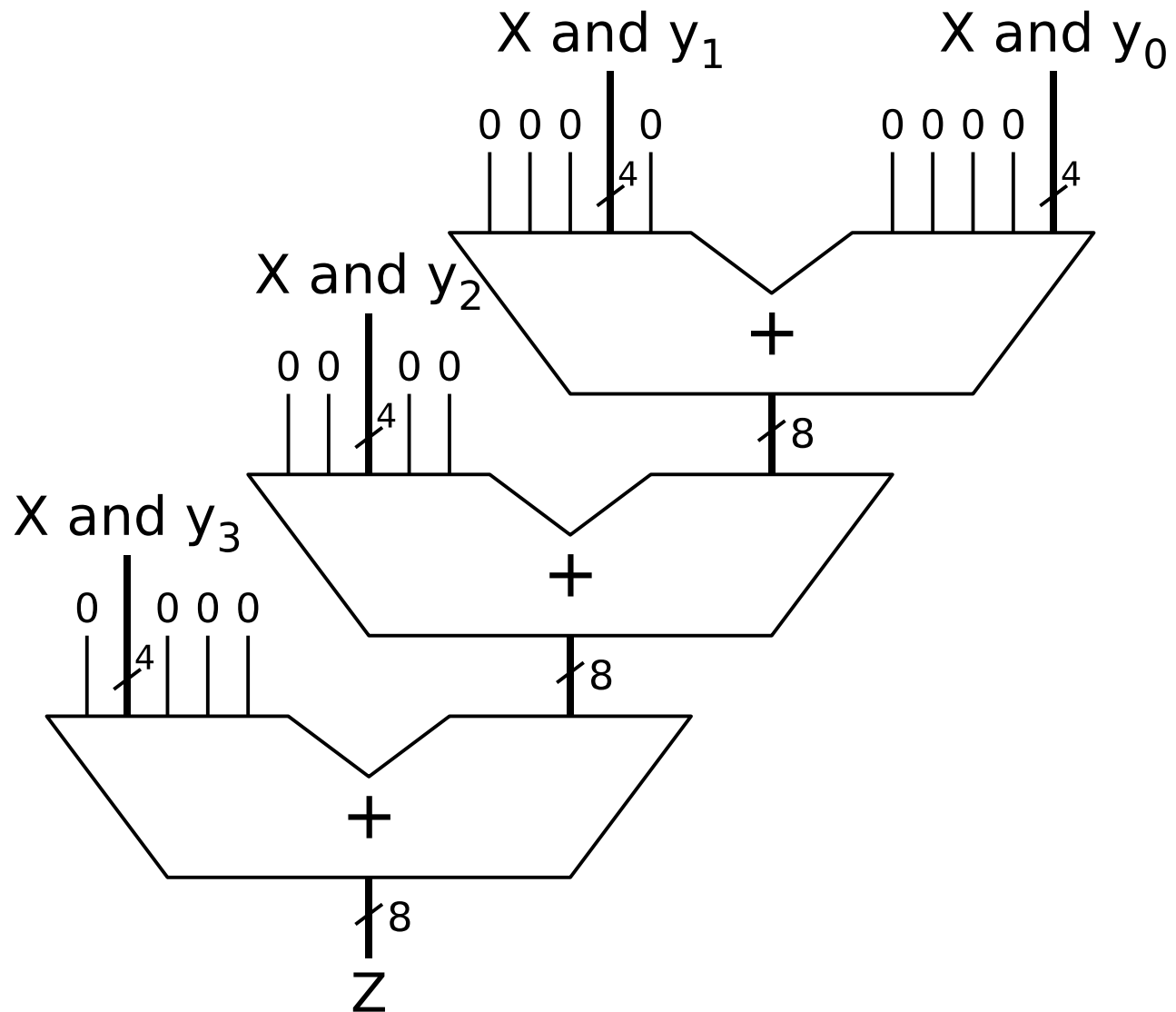
- Implementar de dos formas diferentes un multiplicador de 4x4-bits:
  - 1: Usando la libreria `numeric_std` y el operador `*`
  - 2: Usando los sumadores del laboratorio 1.
- Estudiar los informes de Vivado para encontrar:
  - Los elementos combinacionales usados.
  - El retardo del camino crítico.



# Puertos del Multiplicador

```
entity mult8b is
  port(
    X : in  std_logic_vector(3 downto 0);
    Y : in  std_logic_vector(3 downto 0);
    Z : out std_logic_vector(7 downto 0)
  );
end mult8b_std;
```

# Multiplicador usando sumadores de 8 bits





# Synthesis reports

- Si la síntesis es correcta, en la pestaña “reports” tenemos dos informes:

The screenshot shows the 'Reports' tab in a synthesis tool. It lists two reports under the 'Synth Design (synth\_design)' category:

Report	Type	Options	Modified	Size
synth_2_synth_report_utilization_0	report_utilization		10/2/19, 11:22 AM	7.0 KB
synth_2_synth_synthesis_report_0			10/2/19, 11:22 AM	21.1 KB

Annotations:

- A red box highlights the 'synth\_2\_synth\_report\_utilization\_0' report with the text: "This is a report with the resources usage (“utilization report”)"
- A red box highlights the 'synth\_2\_synth\_synthesis\_report\_0' report with the text: "This is a “raw report” with everything"

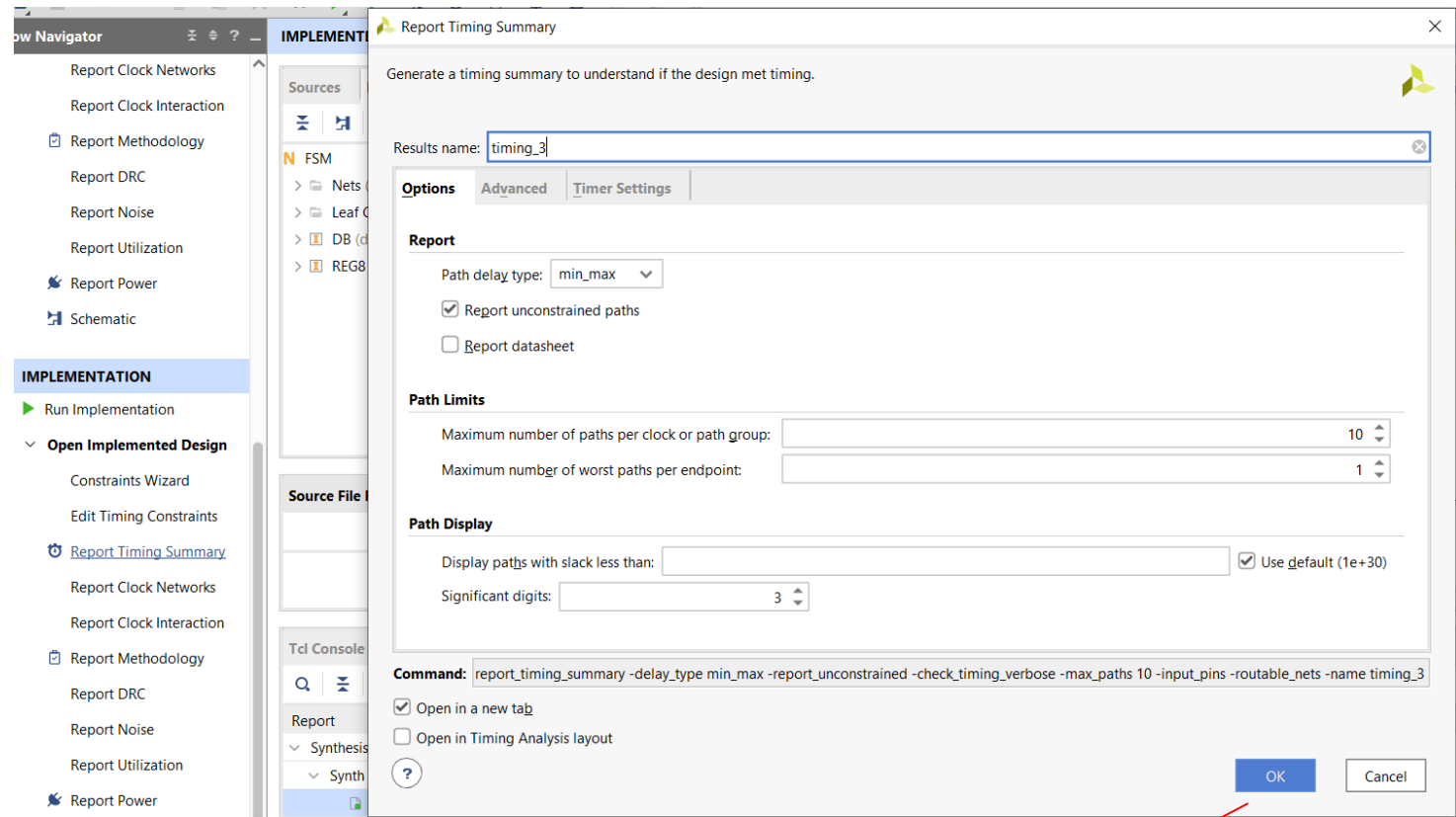
- Ejemplo sacado del laboratorio 2:

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	46	0	20800	0.22
LUT as Logic	46	0	20800	0.22
LUT as Memory	0	0	9600	0.00
Slice Registers	35	0	41600	0.08
Register as Flip Flop	35	0	41600	0.08
Register as Latch	0	0	41600	0.00
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

# Timing reports



- Después de la “IMPLEMENTATION”, podemos ver el “Report Timing Summary”



This window appears, when clicking on  
OK, the report is generated



# Timing reports

- Pestaña de “Timing”:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 6,501 ns	Worst Hold Slack (WHS): 0,217 ns	Worst Pulse Width Slack (WPWS): 4,500 ns
Total Negative Slack (TNS): 0,000 ns	Total Hold Slack (THS): 0,000 ns	Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 54	Total Number of Endpoints: 54	Total Number of Endpoints: 36

**All user specified timing constraints are met.**

- “All user specified timing constraints are met” → La herramienta puede sacar esta conclusión si en el fichero .xdc hemos especificado el periodo de reloj:

```
create_clock -add -name clk -period 10.00 -waveform {0 5} [get_ports clk]
```

- Esta línea define un periodo de reloj de 10 ns. Podemos verlo en la sección del informe llamada: “Clock Summary”



# Informe temporal

- Cuando en el diseño nos encontramos caminos combinacionales entre alguna entrada y alguna salida aparecerá su informe aquí:

Timing Summary - timing\_1

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source
Path 1	∞	5	4	13	Y[1]	Z[6]	8.457	5.944	2.512	∞	input p
Path 2	∞	5	4	13	Y[1]	Z[5]	8.347	5.835	2.512	∞	input p
Path 3	∞	5	4	13	Y[1]	Z[7]	8.259	5.747	2.512	∞	input p
Path 4	∞	5	4	13	Y[1]	Z[4]	8.153	5.641	2.512	∞	input p
Path 5	∞	4	3	13	Y[1]	Z[3]	7.810	5.737	2.073	∞	input p
Path 6	∞	4	3	13	X[1]	Z[2]	7.470	5.870	1.599	∞	input p

- Cuando en el diseño nos encontramos caminos combinacionales entre alguna entrada y un elemento de memoria o entre un elemento de memoria y alguna salida aparecerá su informe aquí:

Timing Summary - timing\_1

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	2	2	5	Y[1]	reg_salida_s...ut_reg[2]/D	2.418	1.618	0.800
Path 2	∞	2	2	10	X[1]	reg_salida_s...ut_reg[4]/D	2.411	1.611	0.800
Path 3	∞	2	2	5	Y[1]	reg_salida_s...ut_reg[1]/D	2.390	1.590	0.800
Path 4	∞	2	2	5	Y[1]	reg_salida_s...ut_reg[3]/D	2.390	1.590	0.800
Path 5	∞	2	2	5	Y[1]	reg_salida_s...ut_reg[4]/D	2.390	1.590	0.800
Path 6	∞	2	2	5	Y[1]	reg_salida_s...ut_reg[5]/D	2.390	1.590	0.800





# Día de la práctica

- Al inicio de la session, el alumno debe presentar al profesor de laboratorio:
  - el multiplicador implementado con sumadores –
    - Simulado con el test\_bench que está en el campus virtual junto con los informes de Vivado con el número de elementos combinatoriales y los retardos (0.05 puntos).
    - Funcionando en la FPGA (0.05 puntos).
- Durante la session los alumnos tienen que realizar una parte extra (0.2 puntos) que se explicará al iniciar el laboratorio.



# Testbench

```
-- Stimulus process
p_stim : process
    variable v_i : natural := 0;
    variable v_j : natural := 0;
begin
    i_loop : for v_i in 0 to 15 loop
        j_loop : for v_j in 0 to 15 loop
            X      <= std_logic_vector(to_unsigned(v_i, 4));
            Y      <= std_logic_vector(to_unsigned(v_j, 4));
            Z_xpct <= std_logic_vector(to_unsigned(v_i * v_j, 8));
            wait for 5 ns;
            assert Z = Z_xpct
                report "Error multiplying, "&integer'image(v_i) & " * " &
                    &integer'image(v_j) & " = "&integer'image(v_i*v_j) &
                    " not "&integer'image(to_integer(unsigned(Z)))
                severity error;
            wait for 5 ns;
        end loop j_loop;
    end loop i_loop;
    wait;
end process p_stim;
```