



Práctica 3

Redes iterativas y redes en árbol

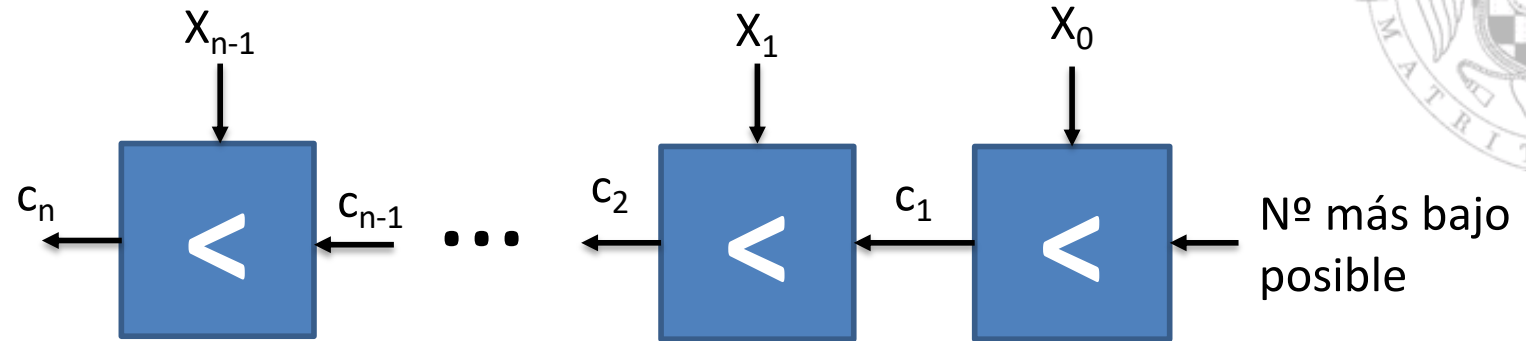


Objetivo

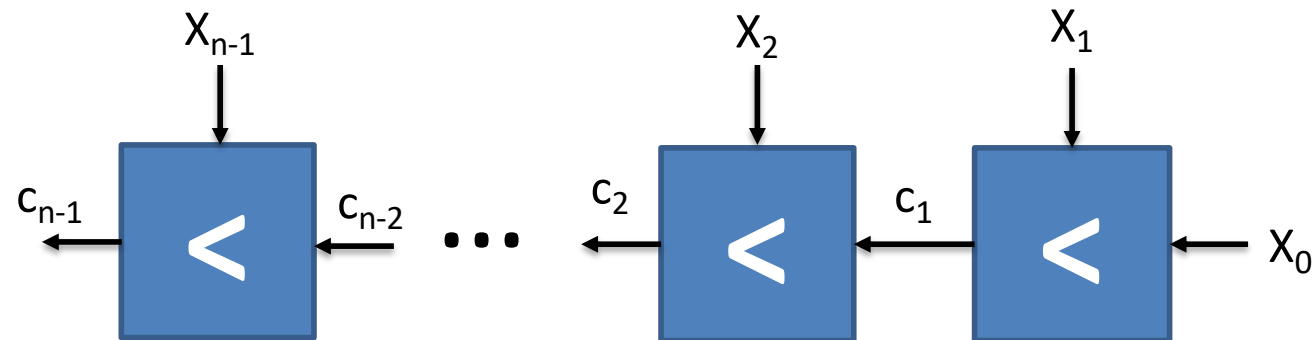
- Implementar una red iterativa y una red en árbol para encontrar el mayor de una serie de números:
 - Ambas redes deben poder generarse para cualquier cantidad (potencia de dos) de números mediante la sentencia 'generate'
- Estudie el informe de síntesis de HDL para encontrar:
 - Los elementos combinacionales instanciados
 - El camino crítico (máximo retardo combinacional)

Red iterativa

Opción 1

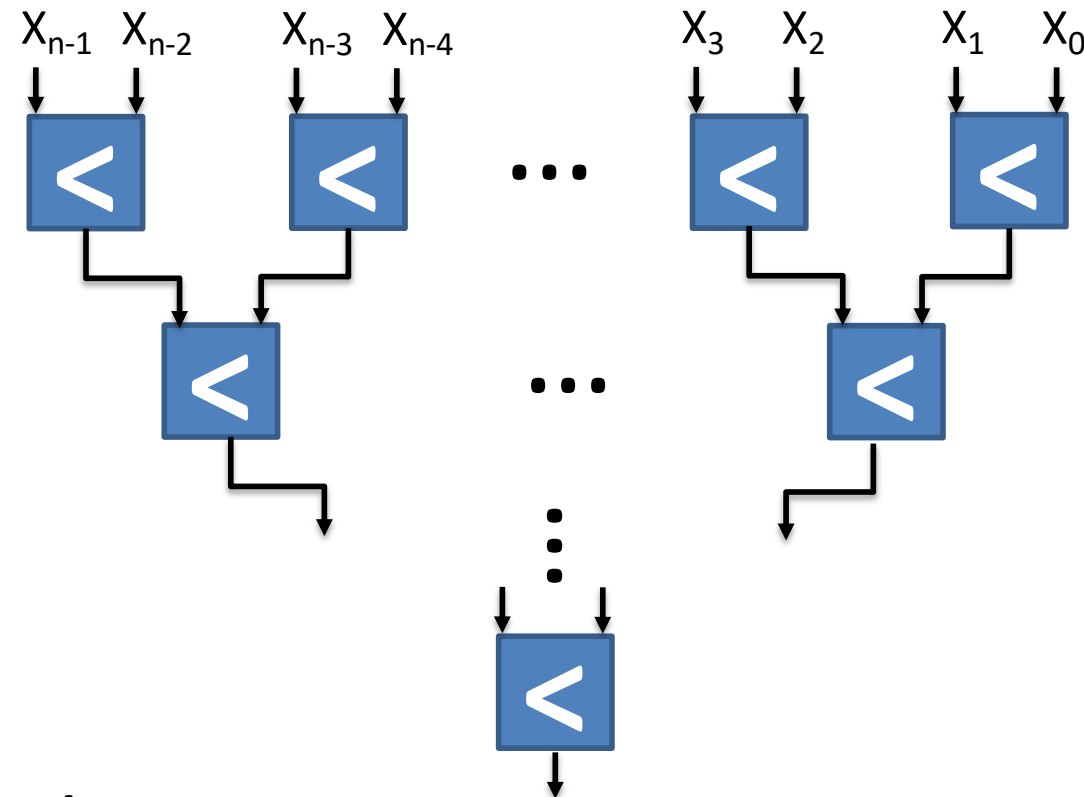


Opción 2
(con una
celda
menos)



- Simulación
 - 16 números representados en C2 de 4 bits
- Placa
 - 4 números representados en C2 de 4 bits

Red en árbol



- Simulación
 - 16 números representados en C2 de 4 bits
- Placa
 - 4 números representados en C2 de 4 bits



Red en árbol

- Generación de una red en árbol mediante 'generate'

```
gen_niveles : for i in 0 to Log2(num_entradas)-1 generate
  gen_compradores : for j in 0 to (num_entradas/2**(i+1))-1 generate
    comparador_i : comparador
      generic map (
        n =>                                -- A completar por el alumno
      )
      port map(
        A =>                                , -- A completar por el alumno
        B =>                                , -- A completar por el alumno
        S =>                                -- A completar por el alumno
      );
    end generate gen_compradores;
  end generate gen_niveles;
```

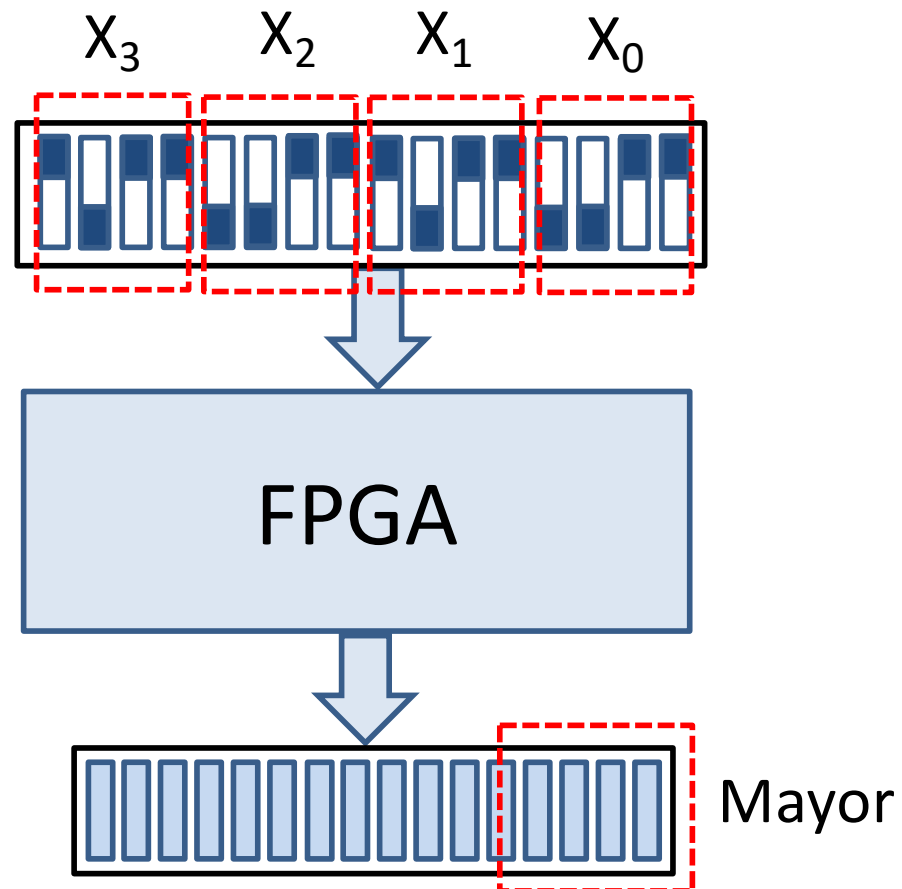


Función log2

```
function Log2( input:integer ) return integer is
    variable temp,log:integer;
begin
    temp:=input;
    log:=0;
    while (temp > 1) loop
        temp:=temp/2;
        log:=log+1;
    end loop;
    return log;
end function log2;
```

- No va a generar nada hardware
- Le pasáis como parámetro el número de entradas (potencia de dos) y os devuelve el log2 que usaréis como una constante (como si estuviera definido en generic)

Implementación





Informe síntesis

- Después de la síntesis, en la pestaña 'Reports' (debajo), se encuentran dos informes de síntesis:

Tcl ConsoleMessagesLogReportsDesign RunsMethodologyPowerTiming

🔍⏏⚖️+−✎▶

Report	Type	Options	Modified	Size
▼ Synthesis				
▼ Synth Design (synth_design)				
📄 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
▼ Implementation				

Este es un informe con los recursos utilizados

Este es un informe con toda la información

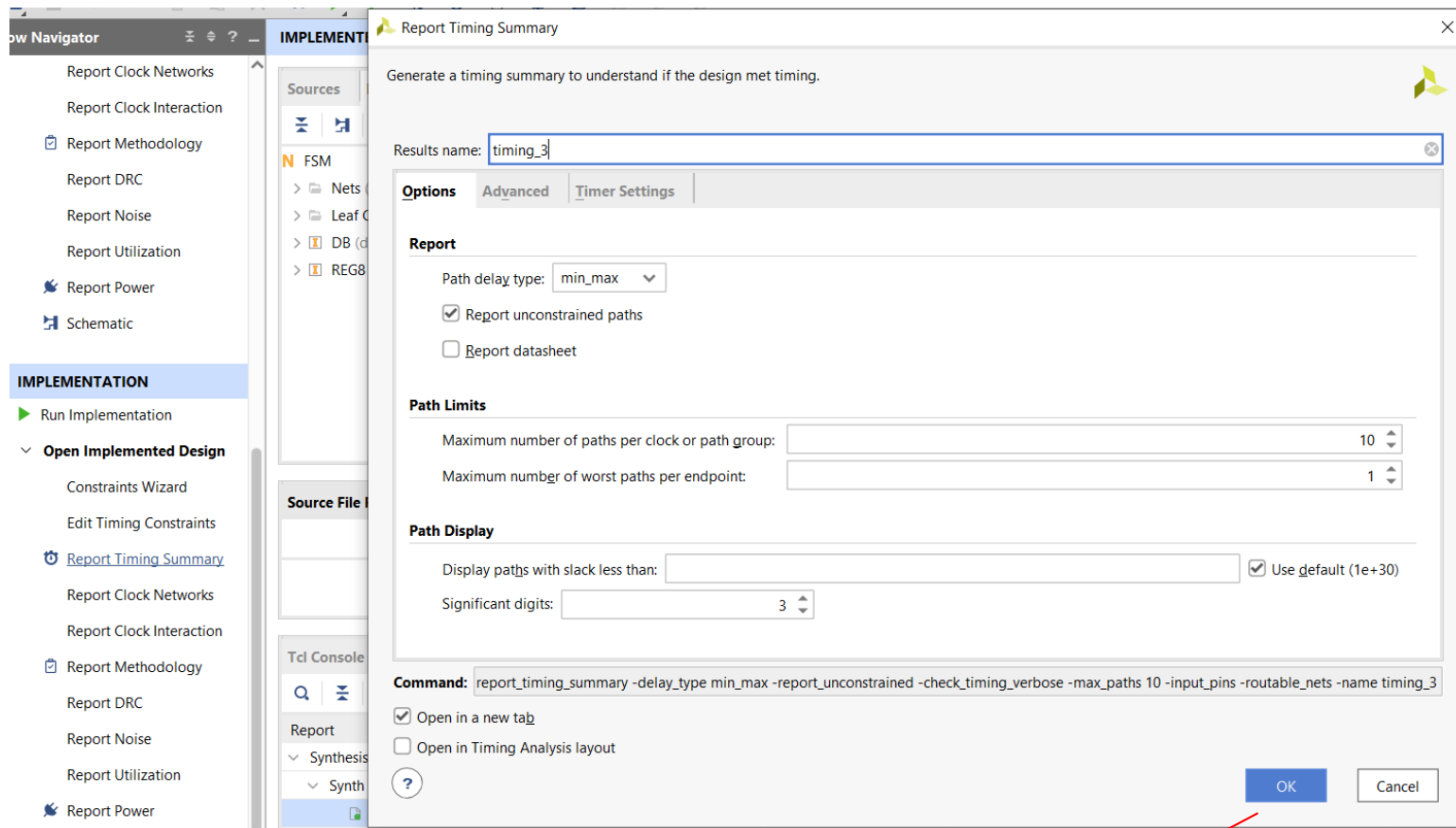
- Por ejemplo (informe de recursos, obtenido de la práctica 2):

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

Informe temporal



■ Debajo de 'Open Implemented Design', tanto en síntesis como en implementación, se puede hacer click en 'Report Timing Summary' (el resumen de tiempos después de la implementación es más preciso)



Aparece esta ventana, al hacer clic en OK, se genera el informe



Informe temporal

- Estos informes son visibles haciendo click en la pestaña 'Timing' (debajo en la ventana principal de VIVADO):

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" → ¿Cómo sabe la herramienta cuál es el período del reloj de entrada que se utiliza en este diseño? Por medio del archivo .xdc:

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

- Esta línea caracteriza la señal clk, que se toma del oscilador que proviene del pin W5 (en la placa Basys-3).
 - Define un período de reloj (10 ns), cuando se convierte en 0 (0 ns) y cuando se convierte en 1 (5 ns)
- No es posible realizar un análisis de tiempo sin esta línea, ni la herramienta puede ubicar y enrutar el hardware de acuerdo con las restricciones temporales.
- Esta línea también es visible al hacer click en "Clock Summary" (una de las secciones del informe - izquierda)

Informe temporal



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

Tcl ConsoleMessagesLogReportsDesign RunsTiming ×

- 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:

Tcl ConsoleMessagesLogReportsDesign RunsTiming x

Unconstrained Paths - NONE - sys_clk_pin - Setup

User Ignored Paths

Unconstrained Paths

NONE to sys_clk_pin

Setup (10)

Hold (10)

sys_clk_pin to NONE

Setup (8)

Hold (8)

Name	Slack ^1	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

Timing Summary - timing_1

Calificación



- Debéis acudir al laboratorio con ambas redes implementadas y simuladas desde casa
- Debéis entregar un documento donde se recoja la información proporcionada por VIVADO sobre los márgenes de setup y de hold (en síntesis e implementación) para ambas redes en la versión de simulación
- Debéis mostrar el funcionamiento de ambas redes y debéis comprender la implementación y la funcionalidad:
 - Si funciona correctamente en la FPGA o en simulación (0.15 pts)
- La práctica 3 presenta una parte avanzada (+0.15 puntos)
- La práctica 3 no se recupera