

Proceso de Síntesis

Temario

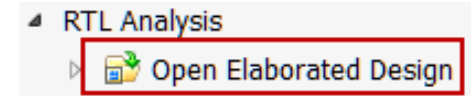
- **Elaboración**
- Síntesis
- Restricciones de tiempo básicas
- Reportes de Síntesis
- Resumen

Elaboración

- **En la Elaboración se optimiza el diseño RTL para una tecnología FPGA específica.**
- **El código fuente RTL puede estar en distintos formatos**
 - Verilog, System Verilog, VHDL, NGC, o archivos de prueba
- **Vistas del código fuente**
 - Vista jerárquica: Muestra los módulos del sistema según su jerarquía
 - Vista de librerías: Muestra el código fuente por categorías

Elaboración y Análisis

- En un sistema basado en RTL, la elaboración es el primer paso
- Presionando en *Open Elaborated Design* dentro de “RTL Analysis”
 - Se compilan los archivos fuente
 - Se carga la netlist RTL para su analisis
- Se puede verificar la estructura, la sintaxis, y las definiciones lógicas
- El analisis y los reportes permiten:
 - Compilar el sistema RTL, validarlo y hacer un chequeo de sintaxis
 - Explorar la Netlist en forma de diagrama esquemático
 - Realizar verificaciones DRC (Design Rule Check)
 - Planificar la ubicacion de los pines de I/O a traves de la lista de puertos
 - Seleccionar un objeto en una vista y verlo en las otras vistas, incluyendo instancias y definiciones lógicas dentro de los archivos fuente (cross probing)



Analisis de un sistema Elaborado

➤ Cuando se abre un sistema elaborado, hay disponibles tres opciones:

– Reporte DRC

- Ejecuta el chequeo de reglas de diseño

– Reporte Noise

- Realiza el analisis de ruido SSO en el sistema
- Simultaneous Switching Output – SSO: es el ruido producido por cambios bruscos de corriente a traves de la inductancia de los circuitos de alimentacion/tierra de la FPGA, lo que produce variaciones transitorias de la tension de alimentaci3n o de la tierra de referencia

– Schematic

- Genera un diagrama esquematico del sistema

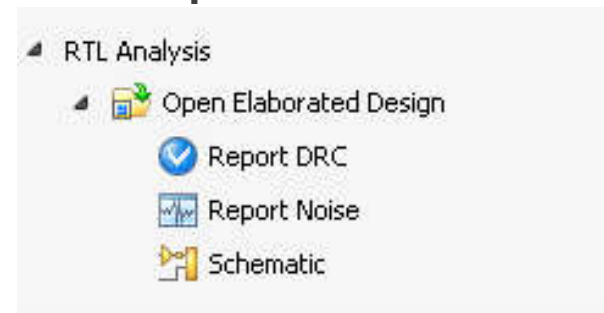
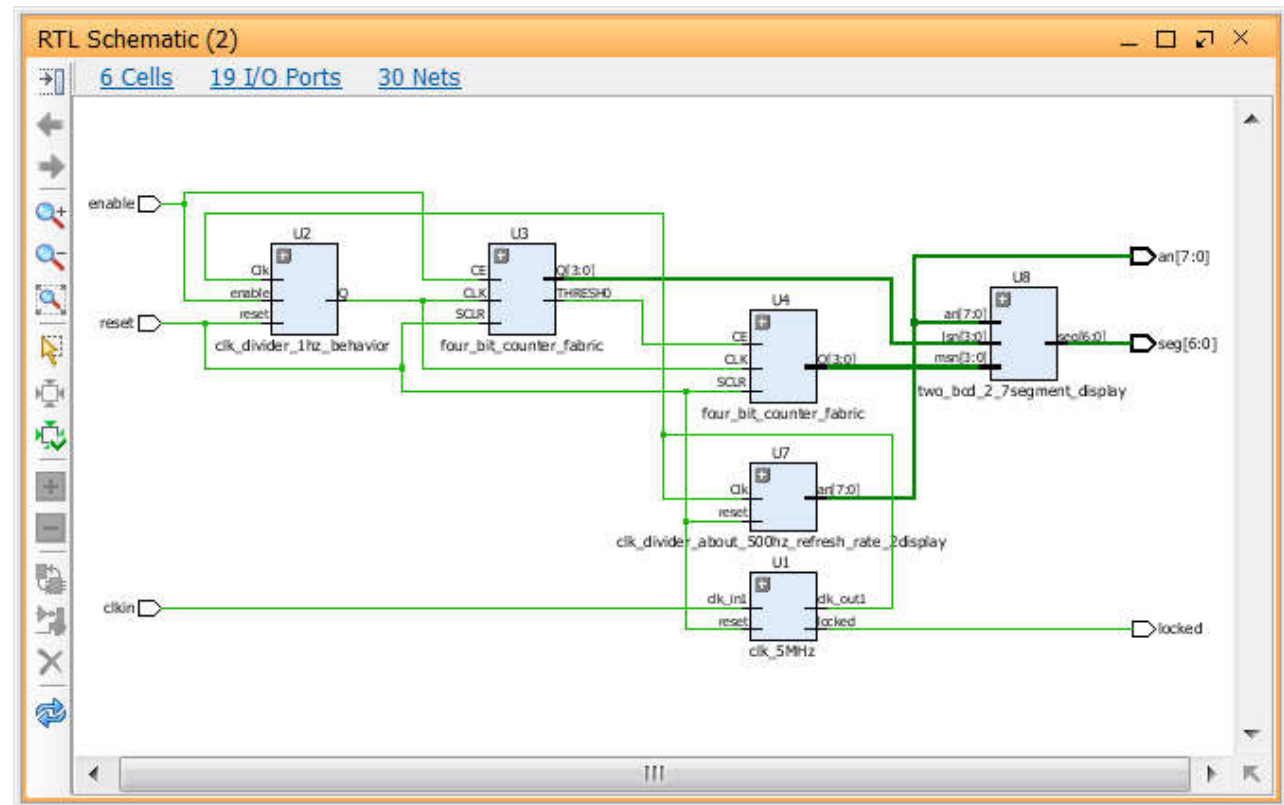


Diagrama esquemático de un sistema Elaborado

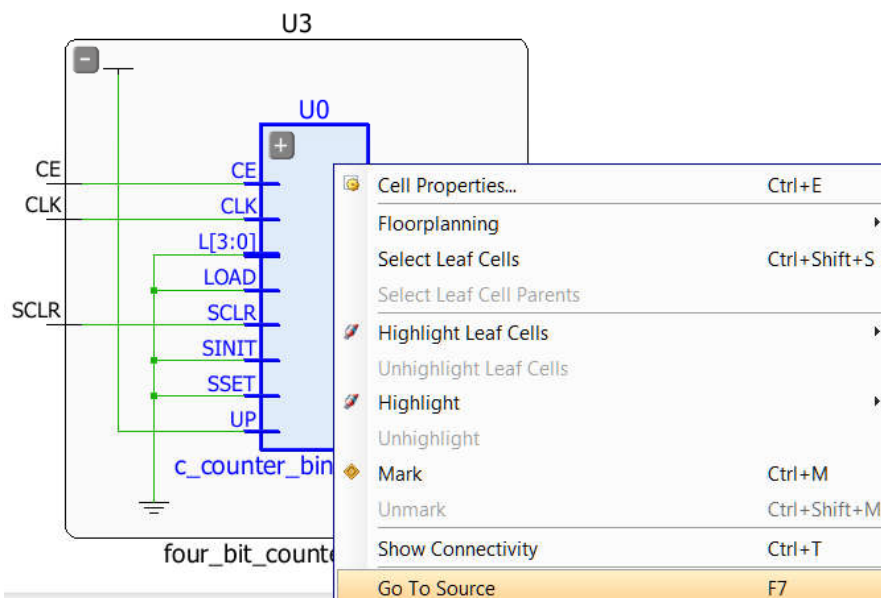
➤ En el diagrama esquemático se ve la jerarquía de bloques y su interconexión

- En esta etapa no se infieren buffers de I/O
- Cada bloque se puede explorar y ver su lógica interna y submodulos



Cross Probing

- Al seleccionar un objeto en el diagrama esquemático, se puede ir al archivo fuente donde este está definido.



```
114 BEGIN
115   U0 : c_counter_binary_v12_0
116   GENERIC MAP (
117     C_IMPLEMENTATION => 0,
118     C_VERBOSITY => 0,
119     C_XDEVICEFAMILY => "artix7",
120     C_WIDTH => 4,
121     C_HAS_CE => 1,
122     C_HAS_SCLR => 1,
123     C_RESTRICT_COUNT => 1,
124     C_COUNT_TO => "1001",
125     C_COUNT_BY => "1",
126     C_COUNT_MODE => 0,
127     C_THRESH0_VALUE => "1001",
128     C_CE_OVERRIDES_SYNC => 0,
129     C_HAS_THRESH0 => 1,
130     C_HAS_LOAD => 0,
131     C_LOAD_LOW => 0,
132     C_LATENCY => 1,
133     C_FB_LATENCY => 0,
134     C_AINIT_VAL => "0",
135     C_SINIT_VAL => "0",
136     C_SCLR_OVERRIDES_SSET => 1,
137     C_HAS_SSET => 0,
138     C_HAS_SINIT => 0
139   )
```

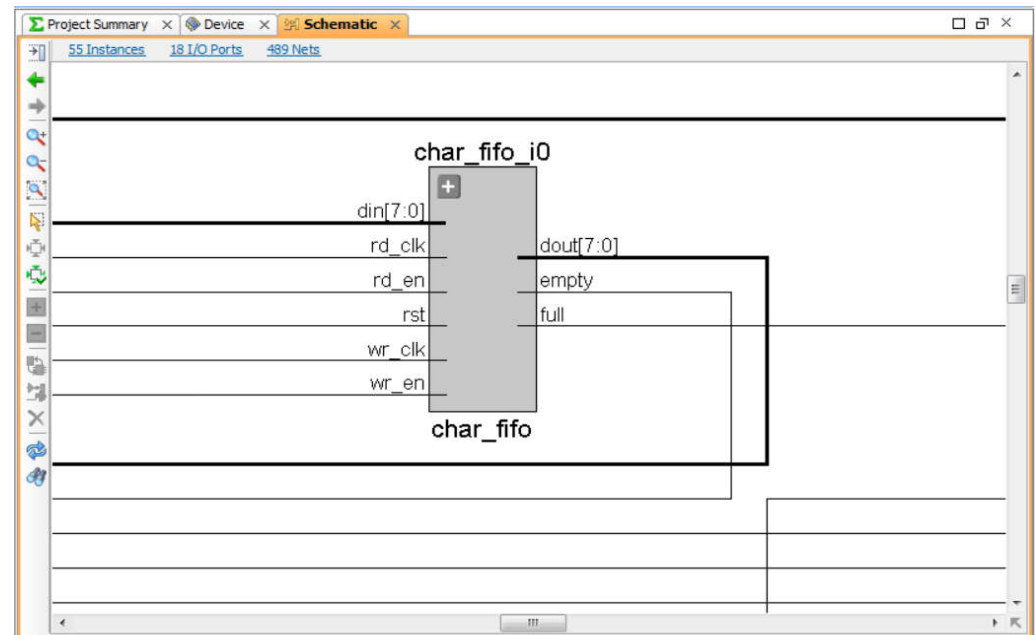
Temario

- Elaboración
- **Síntesis**
- Restricciones de tiempo básicas
- Reportes de Síntesis
- Resumen

Proceso de Síntesis: Optimización de la lógica y mapeo a componentes del dispositivo FPGA

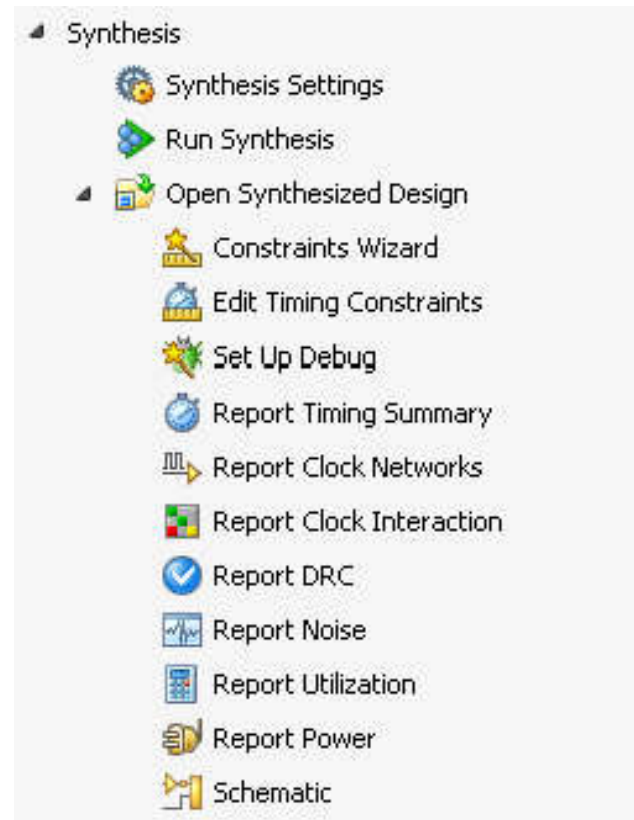
➤ Síntesis de un sistema RTL

- Se optimiza el sistema a nivel compuertas
 - Se mapea la netlist a elementos físicos dentro de la FPGA cuando esto es posible
 - A esto también se lo denomina “technology mapping”
- En la figura se ve una memoria FIFO generica, la cual se mapea a una memoria FIFO física dentro de la FPGA



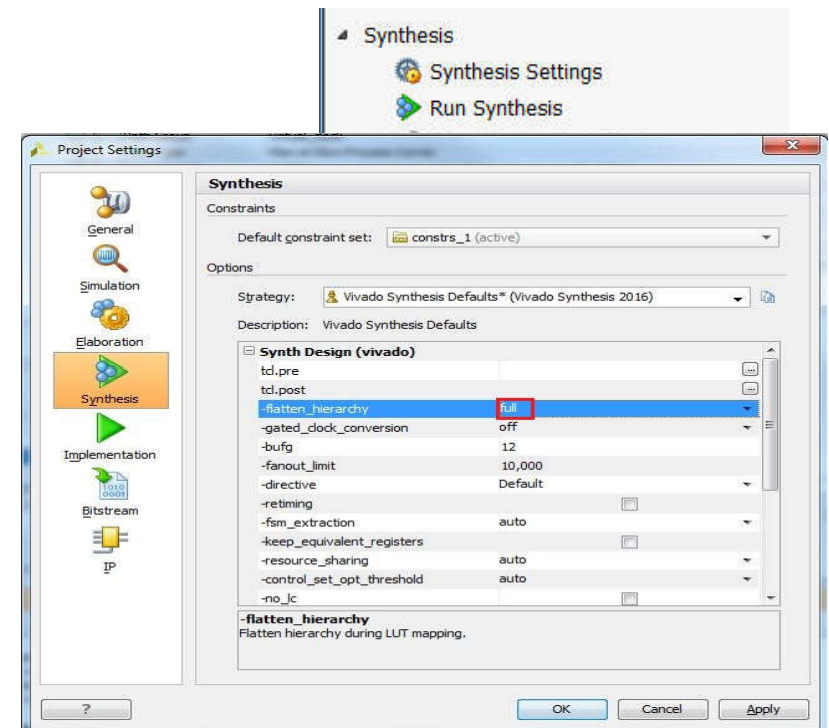
Proceso de Síntesis

- **Se aplica a sistemas RTL (descritos mediante HDL)**
 - Los componentes descritos en EDIF se integran al diseño en una etapa posterior
- **La herramienta de síntesis usa restricciones para controlar las optimizaciones**
 - Es necesario un archivo XDC
- **Restricciones de temporización**
 - El sistema primero debe sintetizarse antes de agregar las restricciones
 - Se puede utilizar un asistente para restricciones simples
- **Luego de sintetizado, se agrega la opción de "Set Up Debug" para agregar características de depuración**



Opciones de Síntesis

- Las opciones de síntesis forman parte de las opciones de proyecto
- Se puede seleccionar el “Default Constraint Set” como el conjunto de restricciones activas para la mayoría de los proyectos. Hay 2 tipos de restricciones
 - **Restricciones físicas (Physical constraints):** definen la ubicación de los pines y la ubicación absoluta o relativa de block RAMs, LUTs, Flip-Flops, etc.
 - **Restricciones de temporización (Timing constraints):** se describen en formato estándar (SDC), y definen los requisitos de temporización del sistema (frecuencia de trabajo, retardos, etc).



Opciones de Síntesis

- Los archivos **tcl.pre** y **tcl.post** se ejecutan antes y despues del proceso de síntesis y sirven para la ejecucion de comandos TCL antes y despues de la síntesis
- **flatten_hierarchy**: Determina como afecta la jerarquia al proceso de síntesis
 - none: no se aplanar el sistema.
 - Se mantiene la jerarquia con la misma estructura que los archivos fuente RTL
 - full: Se aplanar la jerarquia antes del proceso de síntesis a nivel del top level
 - rebuilt: Se aplanar la jerarquia, se realiza la síntesis, y se reconstruye la estructura original RTL
 - Permite una optimizacion por encima de la jerarquia, permitiendo una estructura final similar al codigo fuente RTL
- **gated_clock_conversion**: habilita o no la opcion de generar automaticamente lógica de reloj con habilitaciones

Synth Design (vivado)	
tcl.pre	
tcl.post	
-flatten_hierarchy	rebuilt
-gated_clock_conversion	off
-bufg	12
-fanout_limit	10,000
-directive	Default
-fsm_extraction	auto
-keep_equivalent_registers	<input type="checkbox"/>
-resource_sharing	auto
-control_set_opt_threshold	auto
-no_lc	<input type="checkbox"/>
-shreg_min_size	3
-max_bram	-1
-max_dsp	-1
-cascade_dsp	auto
More Options	

Opciones de Síntesis

- **bufg**: Controla cuantos BUFGs se pueden inferir para el sistema
 - Esta opcion se utiliza cuando otros BUFGs en la netlist no son visibles al proceso de sintesis
- **fsm_extraction**: Codifica las maquinas de estado con una codificacion especifica: one_hot, secuencial, johnson, gray, o auto
- **keep_equivalent_registers**: Evita que los registros con la misma logica de entrada se integren (se utiliza cuando se necesita mayor velocidad en el sistema)
- **resource_sharing**: Configura el uso compartido de operadores aritmeticos entre distintas señales
 - Los valores posibles son auto, on y off

Synth Design (vivado)	
tcl.pre	
tcl.post	
-flatten_hierarchy	rebuilt
-gated_clock_conversion	off
-bufg	12
-fanout_limit	10,000
-directive	Default
-fsm_extraction	auto
-keep_equivalent_registers	<input type="checkbox"/>
-resource_sharing	auto
-control_set_opt_threshold	auto
-no_lc	<input type="checkbox"/>
-shreg_min_size	3
-max_bram	-1
-max_dsp	-1
-cascade_dsp	auto
More Options	

Opciones de Síntesis

- **control_set_opt_threshold**: configura la optimización de la lógica de habilitación de reloj
- **no_lc**: Evita que se combinen LUTs
- **shreg_min_size**: es el mínimo de registros para inferir un SRL
- **max_bram**: Limita la cantidad máxima de BRAM permitida en el sistema
 - El valor -1 establece que se toma tanta BRAM como se pueda, limitada solo por la cantidad disponible en el dispositivo
- **max_dsp**: Limita la cantidad máxima de DSP permitida en el sistema.

Synth Design (vivado)	
tcl.pre	
tcl.post	
-flatten_hierarchy	rebuilt
-gated_clock_conversion	off
-bufg	12
-fanout_limit	10,000
-directive	Default
-fsm_extraction	auto
-keep_equivalent_registers	<input type="checkbox"/>
-resource_sharing	auto
-control_set_opt_threshold	auto
-no_lc	<input type="checkbox"/>
-shreg_min_size	3
-max_bram	-1
-max_dsp	-1
-cascade_dsp	auto
More Options	

Atributos de Síntesis que se pueden incorporar en el código HDL

Attribute	Description
translate_off/_on	Tells the tool to ignore blocks of code
full_case	Tells that all possible case values are specified
parallel_case	Case statement should be built as a parallel structure
keep	Tells tool to keep the signal the attribute is placed on
keep_hierarchy	Used to prevent optimizations along the hierarchy boundaries
buffer_type	Tells tool what buffer type to use on an input
max_fanout	Tells the tool the limits for fanout on registers and signals
ram_style	Tells the tool how to infer memory
rom_style	Tells the tool how to infer ROM memory
use_dsp48	Tells the tool how to deal synthesis arithmetic structures
black_box	Turns a whole level of hierarchy off and enables synthesis to create a black box for that module/entity
gated_clock	Allows the conversion of gated clocks; must be enabled
shreg_extract	Tells the tool on whether to infer structures
iob	Not a synthesis attribute but is passed to the implementation tool indicating if a register should be in IOB

Atributos de Síntesis que se pueden incorporar en el código HDL

Attribute	Description
async_reg	Tells the tool that a register can receive asynchronous data at D input
srl_style	Specifies how SRL is inferred in design
clock_buffer_type	Specifies a buffer other than the (default) BUFG for synthesis
dont_touch	Similar to KEEP attribute, use in place of KEEP. Attribute is forward-annotated to place & route
fsm_encoding	Specifies a specific FSM encoding scheme: one_hot, sequential, johnson, gray, auto (default), none
fsm_safe_state	Place on state machine state registers, used to define a safe state in the machine
IOB	Not a synthesis attribute, used by Vivado implementation. Specifies if a register is packed into IOB
io_buffer_type	Instructs the tool to not automatically infer I/O buffers for a specific top-level port.
MARK_DEBUG	Specifies that a net to be marked for debug.

Ejemplo de uso de atributos en código HDL

- No se pueden agregar atributos de temporización
- Ejemplo del atributo KEEP (mantener una señal aunque pueda ser optimizada)

VHDL

```
signal sig1 : std_logic;  
attribute KEEP : string;  
attribute KEEP of sig1 : signal is "true";  
sig1 <= in1 and in2;  
out1 <= sig1 and in3;
```

Verilog

```
(* KEEP = "true" *) wire sig1;  
assign sig1 = in1 & in2;  
assign out1 = sig1 & in3;
```

// Sin el atributo, la señal sig1 sería optimizada

Atributo Mark_Debug

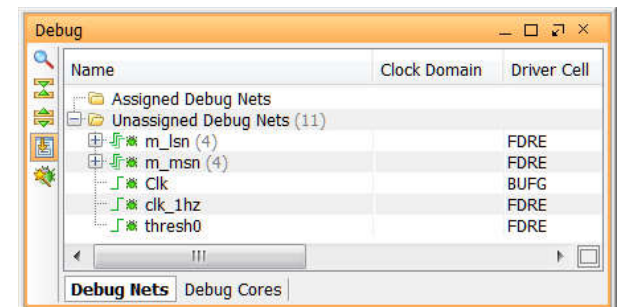
➤ MARK_DEBUG se puede utilizar en el proceso de síntesis

- Marca la señal en el código HDL
- Las redes correspondientes se preservan en la netlist (comportamiento similar a DONT_TOUCH)
- Las señales marcadas aparecen en el analizador lógico

Apply attribute in HDL

```
6 module two_digits_counter_on_2_7segment_display(  
7     input clkin,  
8     input reset,  
9     input enable,  
10    output locked,  
11    output [7:0] an,  
12    output [6:0] seg  
13 );  
14  
15 (* mark_debug = "TRUE" *) wire [3:0] m_lsn, m_msn;  
16 (* mark_debug = "TRUE" *) wire clk_1hz;  
17 (* mark_debug = "TRUE" *) wire Clk, thresh0;  
18
```

After Synthesis



Temario

- Elaboración
- Síntesis
- **Restricciones de tiempo básicas**
- Reportes de Síntesis
- Resumen

Restricciones de tiempo basicas

➤ En un circuito secuencial se pueden aplicar tres restricciones basicas

– Period

- Afecta al camino entre elementos sincronicos que utilizan el reloj de referencia del sistema
 - Los elementos sincronicos pueden ser flip-flops, latches, RAM sincronica, y bloques DSP
- Se usa `create_clock` para crear la restriccion

– Input Delay

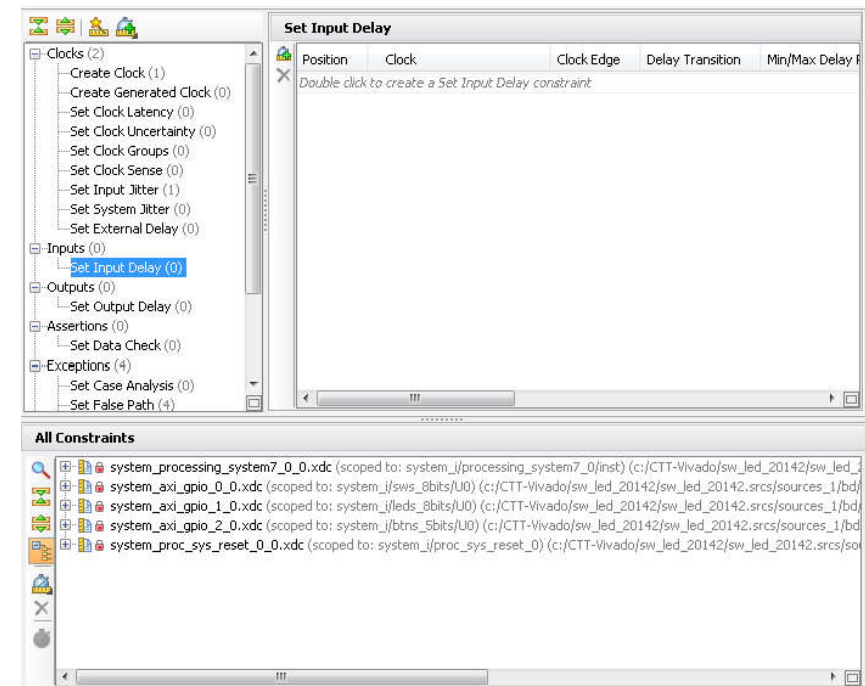
- Afecta al camino entre un pin de entrada y un elemento sincronico
- Se usa `set_input_delay` para crear la restriccion

– Output delay

- Afecta al camino entre un elemento sincronico y un pin de salida
- Se usa `set_output_delay` para crear la restriccion

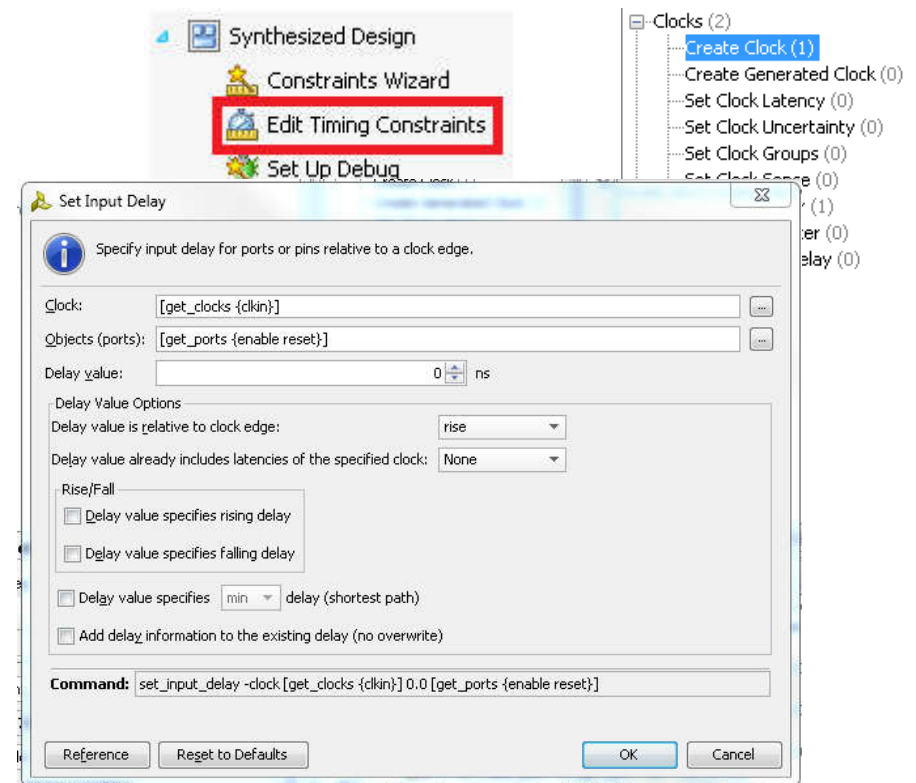
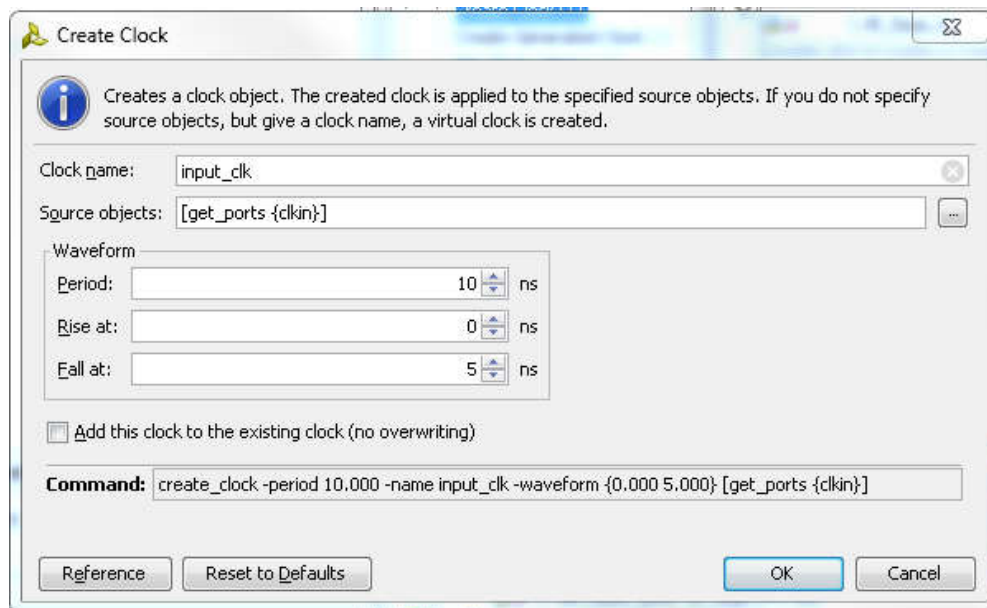
Visor de restricciones (Constraints Viewer)

- **El visor de restricciones permite ver/editar las restricciones de tiempo establecidas en el archivo XDC**
 - Todas las restricciones pueden agregarse a través de la interface grafica (GUI)
 - Presionando el boton derecho sobre la restriccion permite modificarla o borrarla
 - Cada restriccion puede editarse en forma individual en la ventana *All Constraints*
 - Se crea una restriccion seleccionando su tipo en la interface grafica
- **Una vez que se agrego la restriccion, se guarda la misma para que se agregue al archivo XDC**
 - Los cambios se agregan al archivo XDC especifico (en proyectos con mas de una implementacion)



Creación de restricciones básicas de temporización con la interface gráfica

- Ejecutar el proceso de síntesis
- Abrir el sistema sintetizado
- Ejecutar el editor de restricciones



Creación de restricciones básicas de temporización con Tcl

➤ Crear una restriccion Period usando create_clock

- create_clock –period 10 [get_ports clk_pin]
- create_clock –period 10 –waveform {0.000 5.000} –name clk [get_ports clk_pin]

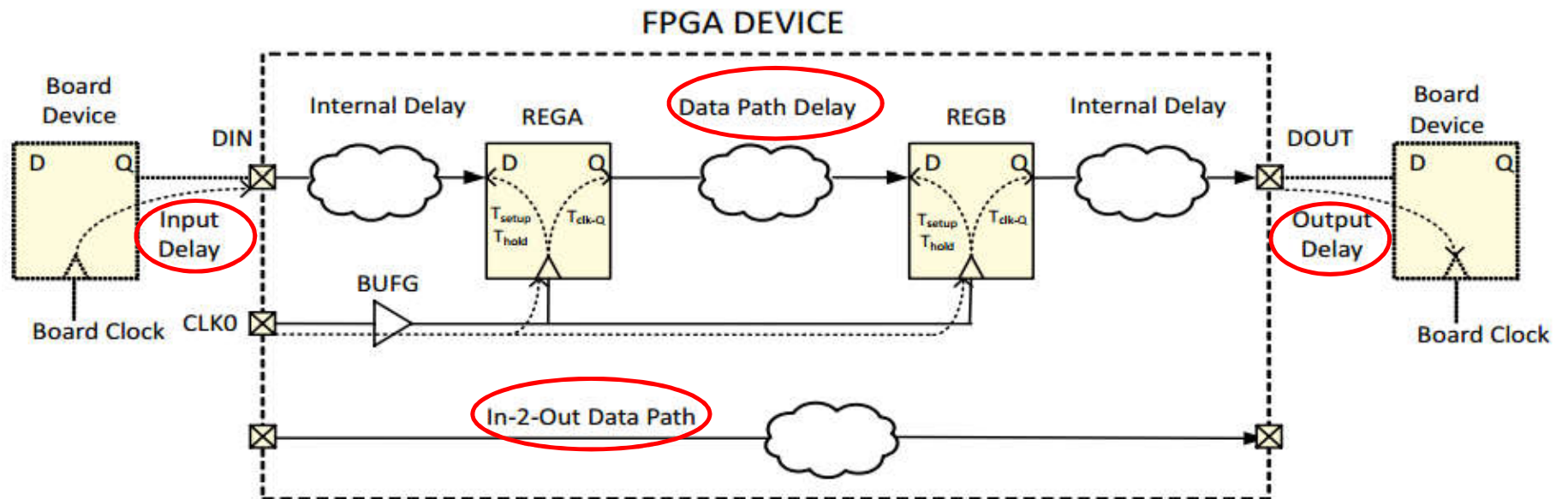
➤ Crear una restriccion de retardo de entrada usando set_input_delay

- set_input_delay –clock clk_pin 2.000 [all_inputs]
- set_input_delay –clock clk_pin 3.000 [get_ports in1]
- set_input_delay –clock clk_pin 2 [get_ports l_msn*]

➤ Crear una restriccion de retardo de salida usando set_output_delay

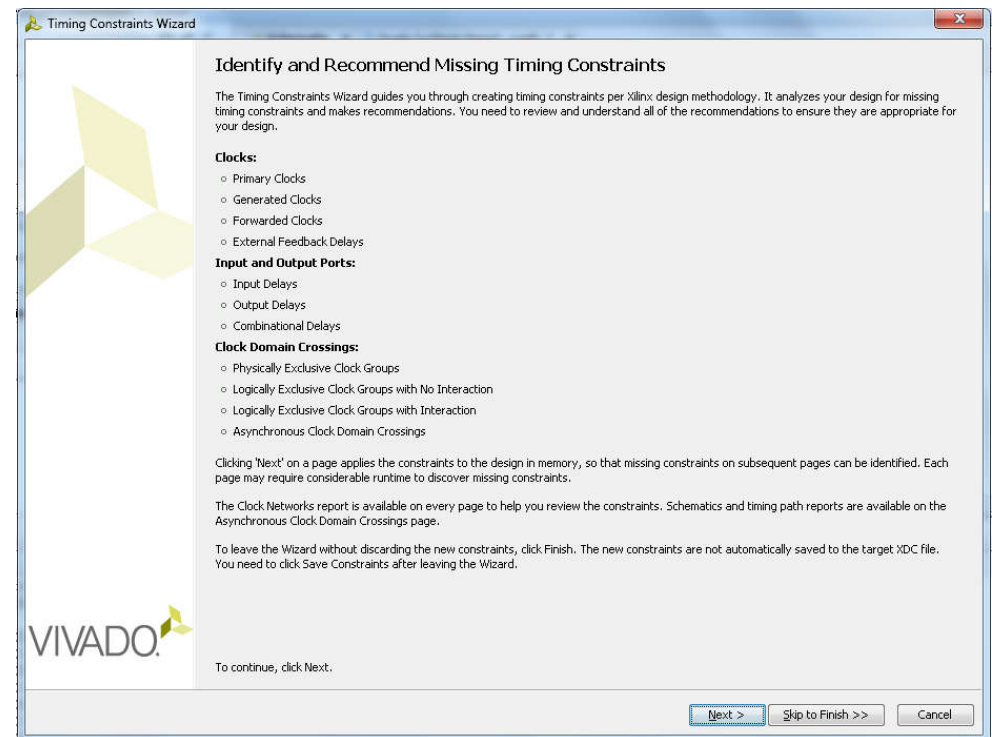
- set_output_delay –clock clk_pin -2 [all_output]
- set_output_delay –clock [get_clocks clk_pin] -2 [get_ports out1]

Ejemplo de retardos



Asistente de restricciones (Constraints Wizard)

- **Permite a las herramientas sugerir restricciones que puedan faltar en el sistema**
 - Recomendado para proyectos que no tienen restricciones iniciales
 - Se pueden ignorar las restricciones recomendadas
 - Es posible desactivar las restricciones sugeridas en cada etapa del asistente
 - Abre el editor de restricciones con las restricciones recomendadas
 - Se pueden modificar o agregar restricciones adicionales al archivo XDC



Temario

- Elaboración
- Síntesis
- Restricciones de tiempo básicas
- **Reportes de Síntesis**
- Resumen

Luego del proceso de síntesis:

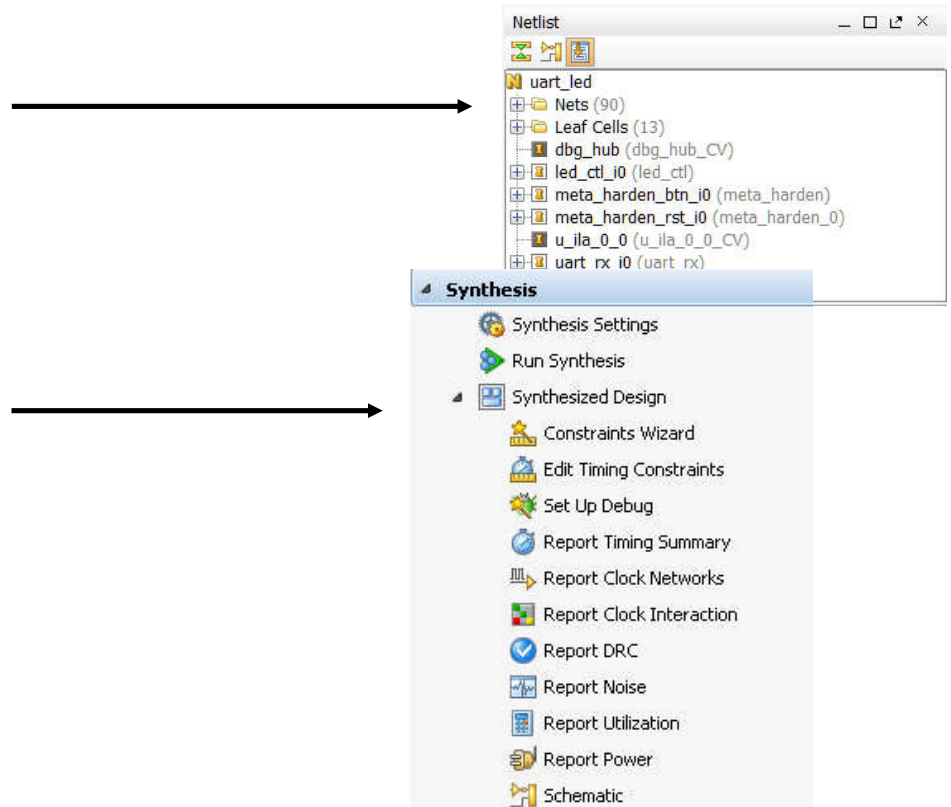
En la vista de sistema sintetizado:

➤ La pestaña Sources no cambia

- La pestaña RTL Netlist (Elaborated Design) cambia a Netlist (Synthesized Design)

➤ El menu Flow Navigator ahora incluye:

- Asistente de restricciones; Editor de restricciones, Configurar depuración, Reporte de temporización, Reporte de redes de reloj, Reporte de interacción de relojes, Reporte DRC, Reporte de ruido, Reporte de Utilización, Reporte de consumo, Diagrama esquemático
- Toda la información de temporización es estimada (hasta que se complete la implementación)



Sistema sintetizado

- **Se accede en el menu Flow Navigator a traves de Open Synthesized Design**
- **Representacion del sistema despues de la sintesis**
 - Netlist con las inteconexiones y jerarquia de BELs (Basic ELeMentS)
 - Instancias de modulos/entitdades
 - BELs:
 - LUTs, flip-flops, carry chain elements, wide MUXes
 - Block RAMs, DSP cells
 - Elemento de la logica de reloj (BUFG, BUFR, MMCM, ...)
 - Elementos de I/O (IBUF, OBUF, I/O flip-flops)
- **Los nombres de los objetos son los mismos que en la netlist del sistema elaborado, cuando esto es posible**

Reportes de síntesis

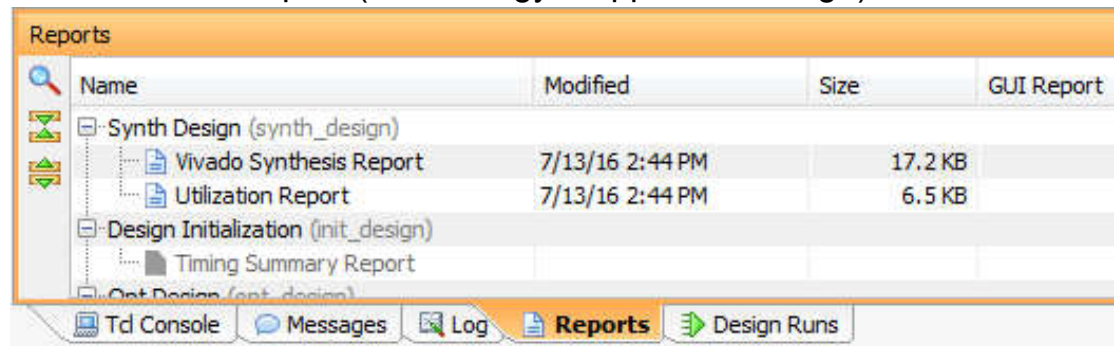
➤ En el menu *Flow Navigator* estan los reportes mas representativos, y en la pestaña *Reports* hay mas reportes

– El reporte *Synthesis Report* muestra:

- Archivos fuente HDL sintetizados, progreso del proceso de síntesis, restricciones de tiempo aplicadas, y primitivas del sistema
- Restricciones de tiempo obtenidas, mapeo a componentes de la FPGA (technology mapping), pines/puertos removidos, uso de bloques (technology-mapped cell usage)

– El reporte *Utilization* muestra

- Uso de bloques (Technology-mapped cell usage) en un formato tabular resumido



Name	Modified	Size	GUI Report
Synth Design (synth_design)			
Vivado Synthesis Report	7/13/16 2:44 PM	17.2 KB	
Utilization Report	7/13/16 2:44 PM	6.5 KB	
Design Initialization (init_design)			
Timing Summary Report			
Opt Design (opt_design)			

Reporte de sintesis

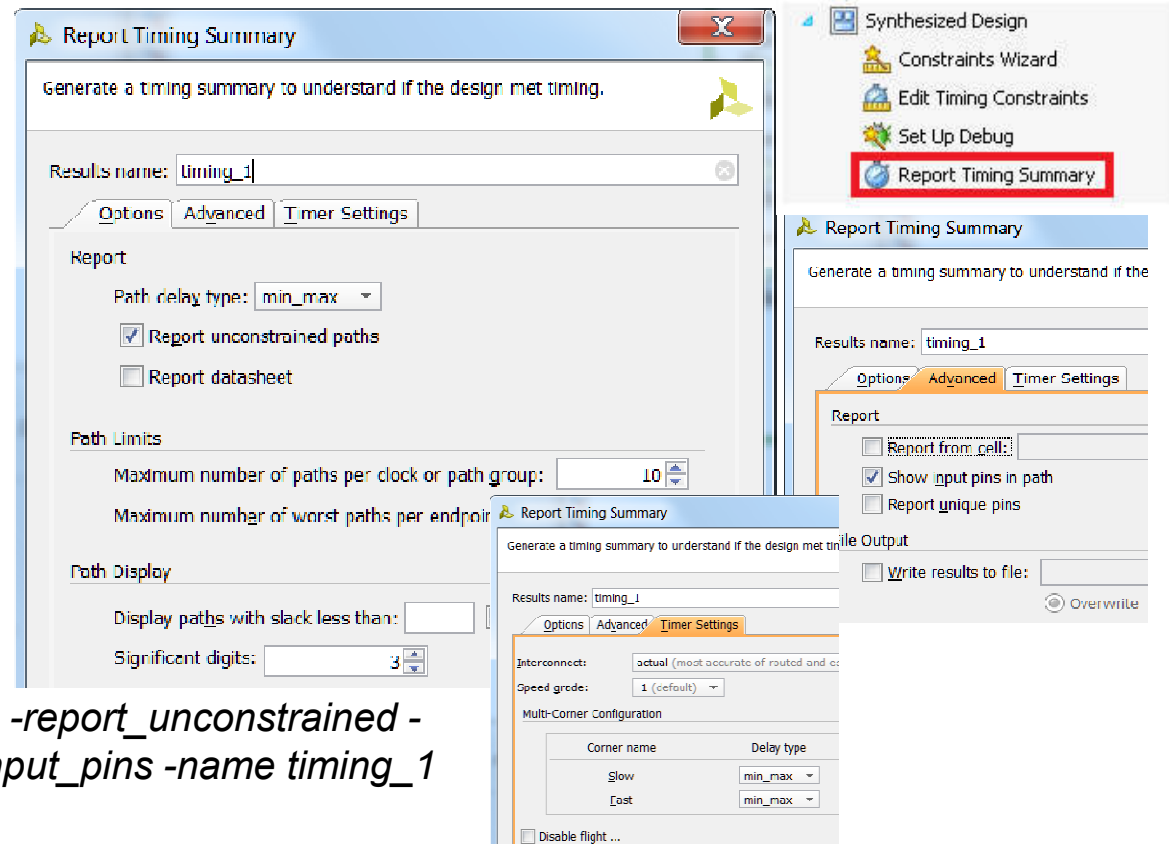
- Muestra el uso de slices, memoria, bloques DSP, bloques de I/O, reloj y otros recursos utilizados por el sistema

```
C:/xup/fpga_flow/2016_2_ZYNQ_labs/zed/lab1/lab1.runs/synth_1/lab1_utilization_synth.rpt
1 Copyright 1986-2016 Xilinx, Inc. All Rights Reserved.
2
3 Tool Version : Vivado v.2016.2 (win64) Build 1577090 Thu Jun  2 16:32:40 MDT 2016
4 Date       : Wed Jul 13 14:44:05 2016
5 Host       : XSJNPURUSH030 running 64-bit Service Pack 1 (build 7601)
6 Command    : report_utilization -file lab1_utilization_synth.rpt -pb lab1_utilization_synth.pb
7 Design     : lab1
8 Device     : 7z020clg484-1
9 Design State : Synthesized
10
11
12 Utilization Design Information
13
14 Table of Contents
15 -----
16 1. Slice Logic
17 1.1 Summary of Registers by Type
18 2. Memory
19 3. DSP
20 4. IO and GT Specific
21 5. Clocking
22 6. Specific Feature
23 7. Primitives
24 8. Black Boxes
25 9. Instantiated Netlists
26
27 1. Slice Logic
28 -----
29
30 +-----+-----+-----+-----+-----+
31 | Site Type | Used | Fixed | Available | Util% |
32 +-----+-----+-----+-----+-----+
33 | Slice LUTs* | 3 | 0 | 53200 | <0.01 |
34 | LUT as Logic | 3 | 0 | 53200 | <0.01 |
35 | LUT as Memory | 0 | 0 | 17400 | 0.00 |
36 | Slice Registers | 0 | 0 | 106400 | 0.00 |
37 | Register as Flip Flop | 0 | 0 | 106400 | 0.00 |
38 | Register as Latch | 0 | 0 | 106400 | 0.00 |
39 | F7 Muxes | 0 | 0 | 26600 | 0.00 |
40 | F8 Muxes | 0 | 0 | 13300 | 0.00 |
41 +-----+-----+-----+-----+-----+
```

Reporte Timing Summary

- **Pestaña Options**
 - Maximo número de caminos
- **Pestaña Advanced**
 - Configuración específica del reporte
- **Pestaña Timer Settings**
 - Estimación de tiempos de interconexión
 - Estimación de retardos de propagación

Comando Tcl: `report_timing_summary`
`report_timing_summary -delay_type max -report_unconstrained -check_timing_verbose -max_paths 10 -input_pins -name timing_1`



Reporte Timing Summary

➤ Design Timing Summary

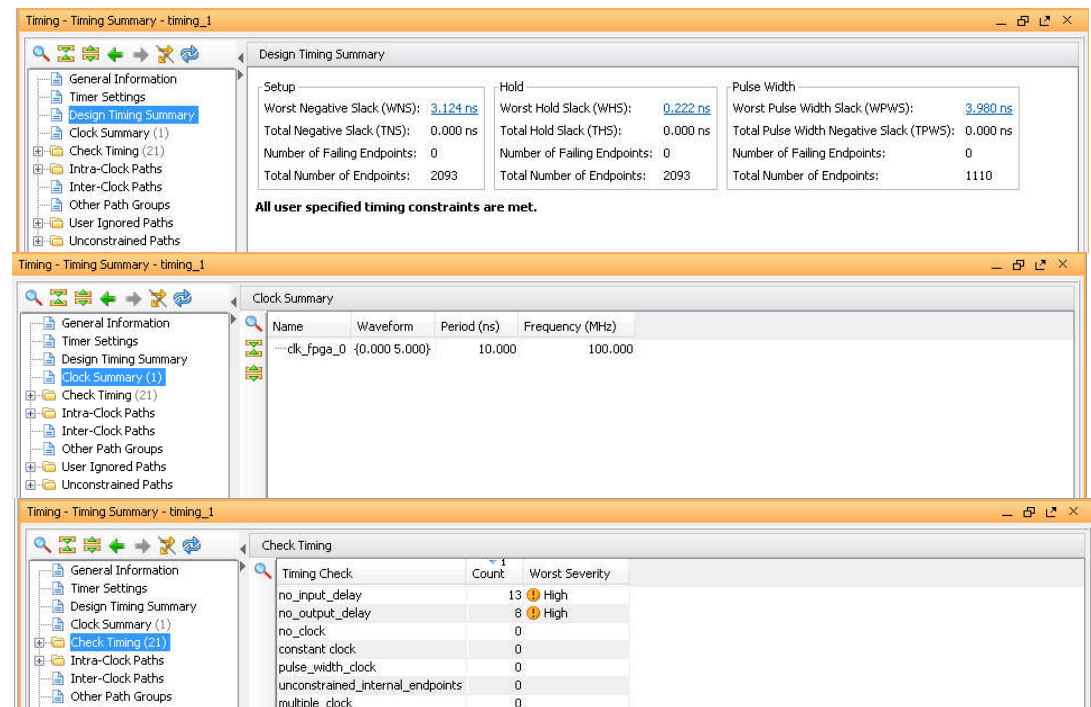
- Worst Negative Slack (WNS), peor retardo; Total Negative Slack (TNS), suma de todos los retardos,

➤ Clock Summary

- Resultados relativos al reloj principal del sistema y sus derivados

➤ Check Timing

- Numero de puntos internos del sistema que no tienen restricciones

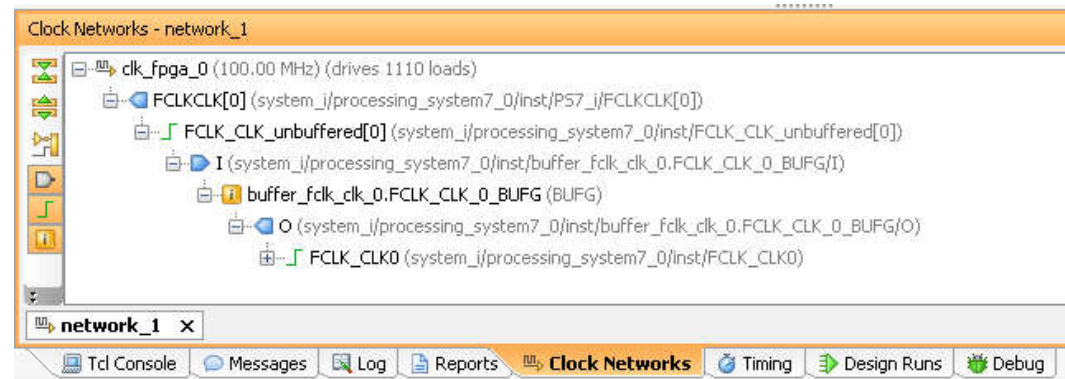
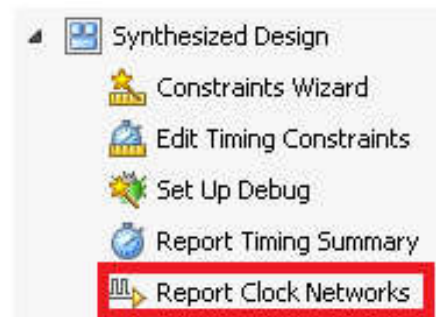


Reporte Clock Networks

➤ Se puede generar mediante comandos Tcl:

- `report_clock_networks`
- Crea una vista de arbol de todas las señales de reloj del sistema, junto con sus definiciones

➤ Tambien se puede crear desde el entorno grafico



Reporte Clock Interaction

➤ Se puede generar mediante comandos Tcl:

```
-report_clock_interaction -  
significant_digits 3 -name  
timing_1
```

➤ Muestra los caminos que cruzan dominios de reloj y registros sin señal de reloj

- Usa una matriz para mostrar la relacion entre dominios de reloj
- Permite ver si la temporizacion es sincronica y si se cumplen las restricciones
- Usa el peor valor de retardo para cada dominio de reloj



Reporte DRC

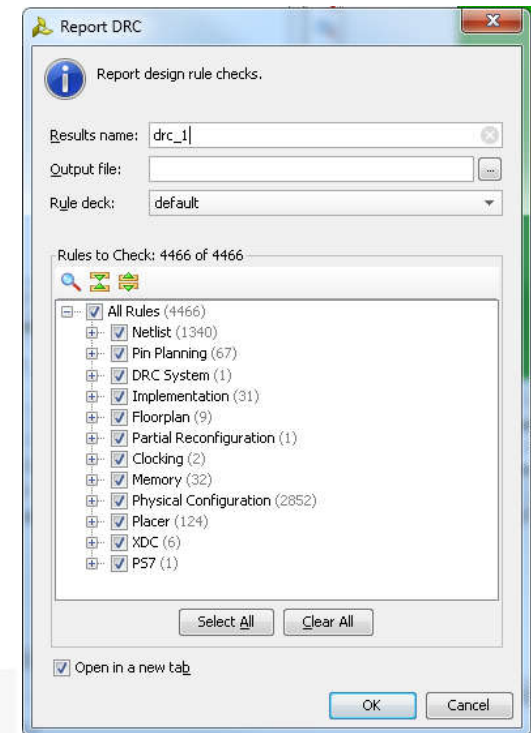
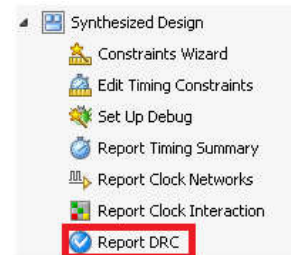
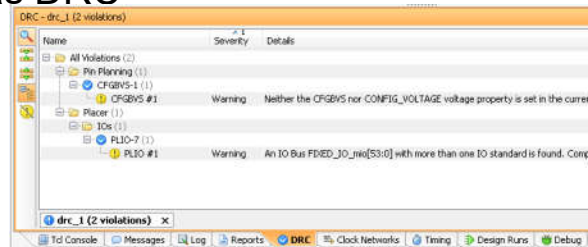
➤ Se puede generar mediante comandos Tcl:

```
-report_drc -name drc_1
```

➤ Realizar chequeos DRCs en las primeras etapas permite realizar correcciones antes de la implementacion

- Los objetos que no cumplen las reglas se pueden ver en sus archivos fuente HDL
- Se puede elegir cuales reglas DRC verificar
 - Este chequeo es mas estricto que el que chequeo DRC de I/O que se realiza durante la planificacion de la ubicacion de los pines de I/O
- Cualquier problema genera una ventana DRC en la parte inferior del entorno grafico
- Se debe realizar el proceso de implementacion para la verificacion final de las reglas DRC

➤ En el IDE:



Reporte Noise

➤ Se puede generar mediante comandos Tcl:

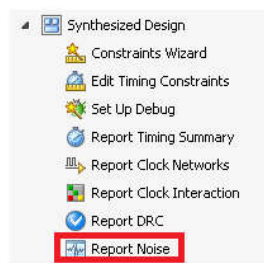
```
-report_ssn -name ssn_1
```

➤ Realiza un analisis SSN en los pines de I/O definidos

– Este reporte analiza el numero de pines, estandard de I/O, y consumo en cada banco de I/O del dispositivo

- Los bancos que exceden el consumo recomendado se pueden ver en la pestaña *Summary*
- El analisis SSN se realiza sobre pines de salida o bidireccionales

➤ En el entorno grafico:



A screenshot of the 'Noise - ssn_1' report window. The window has a left sidebar with 'Summary', 'Messages (1)', 'I/O Bank Details', and 'Links'. The 'I/O Bank Details' tab is selected, showing a table of I/O bank information.

Name	Port	I/O Std	Vcco	Slew	Dr
I/O Bank: 0 (Dedicated) (0)					
I/O Bank: 14 (High Range) (0)					
I/O Bank: 15 (High Range) (0)					
I/O Bank: 16 (High Range) (0)					
I/O Bank: 34 (High Range) (16)		LVC MOS33	3.30	SLOW	
N6	an[0]	LVC MOS33	3.30	SLOW	
M6	an[1]	LVC MOS33	3.30	SLOW	
M3	an[2]	LVC MOS33	3.30	SLOW	

Reporte Power

➤ Se puede generar mediante comandos Tcl:

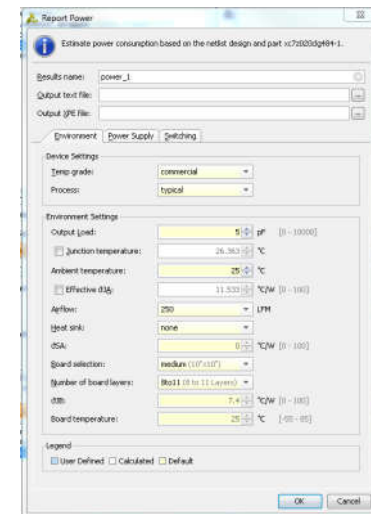
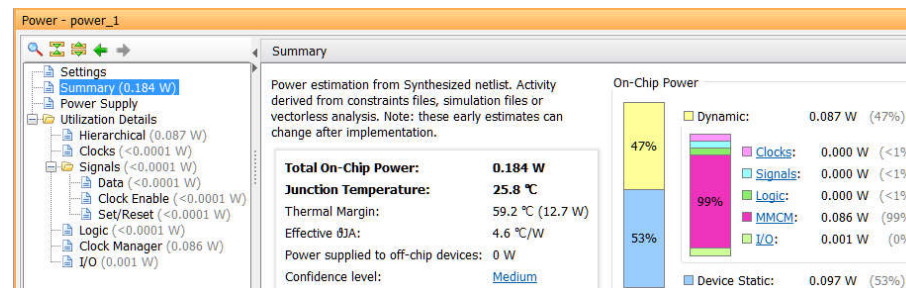
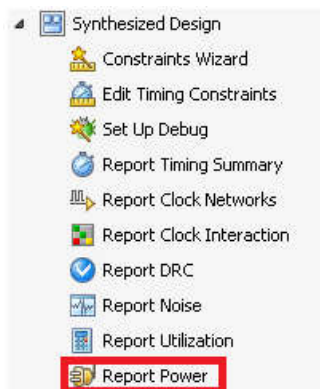
```
-report_power -results {power_1}
```

➤ Estimacion de consumo y generacion de calor

➤ Estimaciones de consumo en cada etapa despues del proceso de sintesis

➤ Realiza analisis variando la actividad de conmutacion de cada pin

➤ Exporta datos a la herramienta de estimacion de consumos (Xilins Power Estimator – XPE)



Detalles del consumo

- Al seleccionar distintos componentes/bloques, se puede tener mayores detalles acerca del consumo

Power - power_1

Utilization Details - Hierarchical

Utilization	Name	Clocks (W)	Signals (W)	Data (W)	Clock Enable (W)	Set/Reset (W)	Logic (W)	Clock Manager (W)
0.087 W (47% of total)	two_digits_counter_on_2_7segment_display							
0.086 W (47% of total)	U1 (clk_5MHz)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.086
<0.0001 W (<1% of total)	Leaf Cells (19)							
<0.0001 W (<1% of total)	U2 (clk_divider_1hz_behavior)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
<0.0001 W (<1% of total)	U7 (clk_divider_about_500hz_refresh_rate_2display)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
<0.0001 W (<1% of total)	U3 (four_bit_counter_fabric_1)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
<0.0001 W (<1% of total)	U4 (four_bit_counter_fabric)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

Power - power_1

Utilization Details - Logic

Utilization	Name	Type	Clock (MHz)	Clock Name	Signal Rate (Mtr/s)	% High
<0.0001 W (<1% of total)	seg_OBUF[3]_inst_i_2 (LUT5)	LUT		N/A Async	2.126	53.032
<0.0001 W (<1% of total)	count_reg[3]_i_2 (CARRY4)	CARRY4		N/A Async	0.558	19.088
<0.0001 W (<1% of total)	seg_OBUF[1]_inst_i_1 (LUT6)	LUT		N/A Async	1.447	61.548
<0.0001 W (<1% of total)	count_reg[2]_i_1 (LUT3)	LUT		N/A Async	0.770	23.701
<0.0001 W (<1% of total)	an_reg[1]_i_1 (LUT3)	LUT		N/A Async	0.739	75.632
<0.0001 W (<1% of total)	seg_OBUF[6]_inst_i_2 (LUT5)	LUT		N/A Async	1.655	52.987
<0.0001 W (<1% of total)	count_reg[13]_i_1 (LUT3)	LUT		N/A Async	0.528	25.913
<0.0001 W (<1% of total)	count_reg[12]_i_1 (LUT3)	LUT		N/A Async	0.520	25.142
<0.0001 W (<1% of total)	count_reg[0] (FDRE)	FF	5.000	U1/inst/clk_out1	4.583	53.594
<0.0001 W (<1% of total)	count_reg[3]_i_1 (LUT3)	LUT		N/A Async	0.497	18.754
<0.0001 W (<1% of total)	count_reg[0]_i_6 (LUT6)	LUT		N/A Async	0.124	2.705

Temario

- Elaboración
- Síntesis
- Restricciones de tiempo básicas
- Reportes de Síntesis
- **Resumen**

Resumen

- Un sistema “Elaborado” permite realizar chequeos DRC, analisis de ruido SSN, y ver las relaciones entre los elementos de los archivos fuente y la jerarquia del sistema
- El proceso de sintesis realizar optimizaciones en la logica y mapeo a elementos especificos de la FPGA
- El codigo fuente puede estar en VHDL, Verilog, y SystemVerilog
- Las restricciones (definidas en un archivo XDC) determinan las optimizaciones aplicadas al sistema
- El reporte *check_timing* es el que permite verificar si se cumplieron las restricciones del sistema
- Hay distintos tipos de reportes que permiten mejorar el desempeño del sistema y evitar los errores mas comunes