Tema 3:

El entorno de diseño Xilinx ISE Design Suite

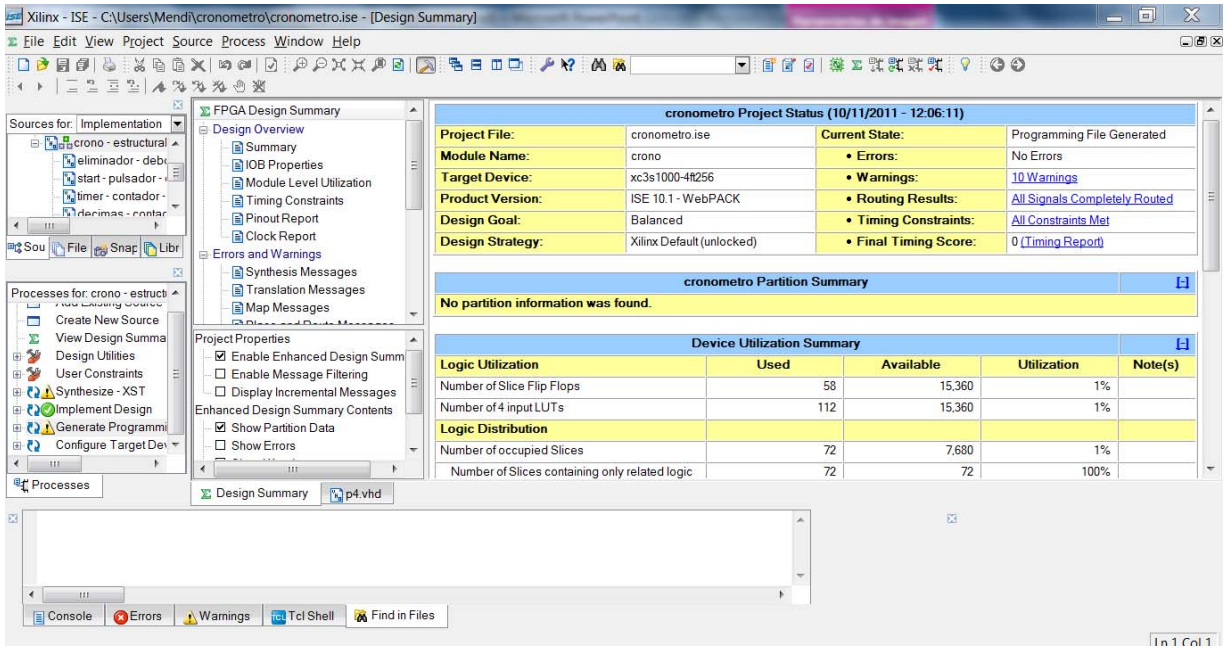
Diseño automático de sistemas

José Manuel Mendías Cuadros
Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Contenidos

- ✓ Xilinx ISE Design Suite
- ✓ Flujo de diseño y validación.
- ✓ Volcado de bitstreams.
- ✓ Síntesis y prototipado.
- ✓ Validación por simulación.
- ✓ Organización de proyectos.
- ✓ Controlando XIlinx ISE.
- ✓ Ligaduras de diseño.
- ✓ Ligaduras de implementación.

Xilinx ISE Design Suite



- Conjunto de herramientas para la síntesis lógica y física de sistemas digitales sobre FPGAS/CPLDs integradas en un interfaz gráfico común:
  - Gestión integral del proyecto.
  - Edición del código fuente.
  - Síntesis.
  - Validación por simulación.
  - Llamada a otras herramientas auxiliares

Xilinx ISE Design Suite

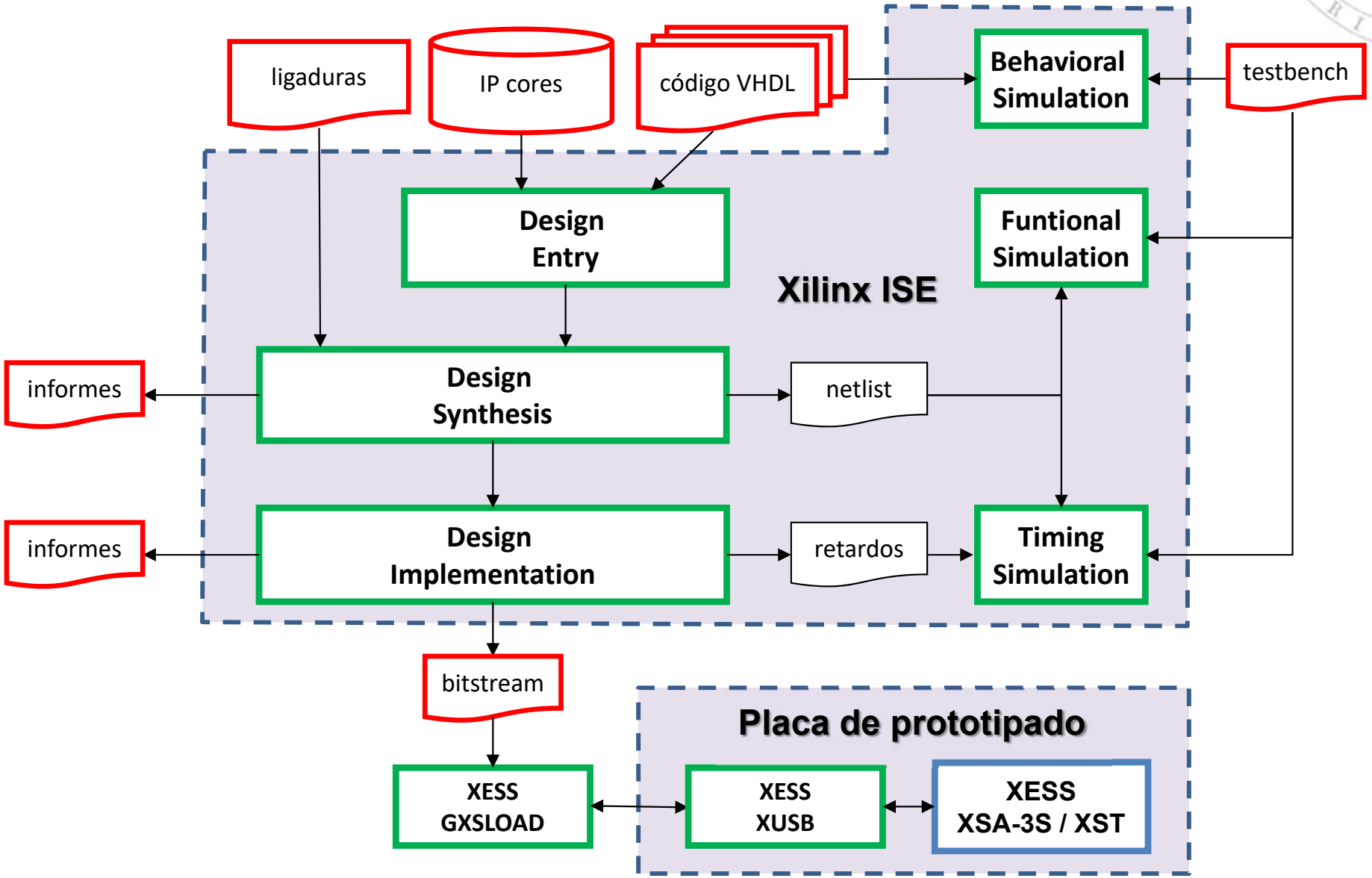
herramientas EDA

- Esta formado por las herramientas:
  - Project Navigator: Interfaz gráfico común de las herramientas EDA de Xilinx.
    - Incluye la gestión de proyectos y los algoritmos de síntesis e implementación.
  - Plan Ahead: Interfaz gráfico alternativo al Project Navigator.
  - iSIM Simulator: Simulador HDL para la simulación funcional y temporal de diseños.
  - Core Generator: Generador de componentes virtuales (Intellectual Property cores).
  - Schematic Viewer: Visualizador del esquemático de un diseño.
  - Timing Analizer: Analizador estático de tiempos.
  - FPGA Editor: Editor físico para el emplazamiento y rutado manual de un diseño.
  - XPower Analyzer: Analizador estático de consumo.
  - iMPACT: Permite el volcado y readback de bitstreams.
  - ChipScope: Permite el análisis lógico y en tiempo real de un diseño.
  - Constraints Editor: Interfaz gráfico para la edición de ligaduras de diseño e implementación.

- Existe una versión gratuita descargable
  - Xilinx ISE Webpack: http://www.xilinx.com/support/download

Xilinx ISE Design Suite

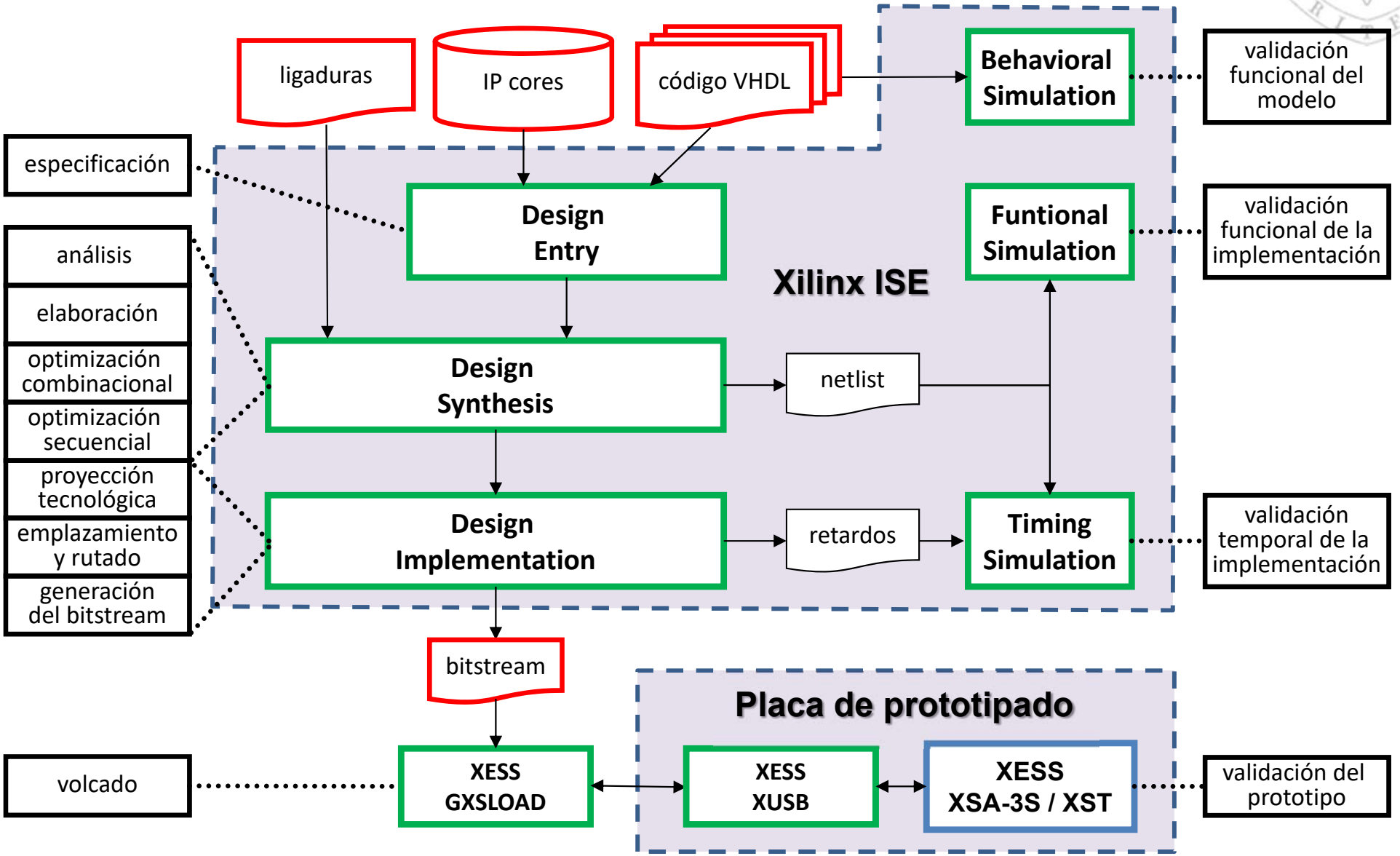
flujo de diseño y validación (i)



J.M. Mendías 2015

Xilinx ISE Design Suite

flujo de diseño y validación (ii)



Xilinx ISE Design Suite

flujo de diseño y validación (iii)

- Las ligaduras (constraints) permiten:
  - Controlar el ciclo de diseño e implementación.
  - Fijar los niveles mínimos de calidad aceptables de una implementación.

- Las ligaduras pueden clasificarse según definan:
  - Técnicas de diseño a aplicar (locales a módulo):
    - Diseño jerárquico, codificación de FSM, estilo de RAM (block o distribuida) ...
  - Estrategias de optimización (aplicables a todo el diseño):
    - Objetivo de la optimización (área o velocidad), esfuerzo (normal o alto) ...
  - Ligaduras físicas:
    - Tiempo de ciclo, retardos, localización (celdas y pines) ...

- Las ligaduras pueden establecerse a través de:
  - GUI: método simple.
  - Atributos de HDL: método asociable a HDL.
  - Fichero de ligaduras: método más completo.

Xilinx ISE Design Suite

flujo de diseño y validación (iv)

- Los informes (reports) son ficheros generados por las herramientas que contienen información sobre los procesos efectuados:
  - Avisos y errores encontrados durante el proceso de diseño e implementación.
  - Herramientas usadas y opciones utilizadas.
  - Decisiones de diseño tomadas (inferencias).
  - Estadísticas de caracterización de la implementación.

- La visualización de informes puede ser:
  - Directa (fichero texto).
  - A través de GUI.

Volcado de bitstreams

volátil sobre FPGA

- Los bitstreams (.BIT) contienen los valores a almacenar en las memorias de configuración de la FPGA.
  - De manera que la FPGA tenga el comportamiento especificado.
- Se vuelcan en la placa XSA-3S a través de un puerto USB.
  - Usando el adaptador a puerto paralelo XESS XSUSB y la herramienta XESS GXSLOAD.
  - Esta configuración puede ir directamente a la FPGA o almacenarse en la Flash ROM.

- Diseños volcados directamente sobre FPGA: es el método habitual usado para el desarrollo y depuración de un diseño.
  - La configuración de la FPGA reside el disco del ordenador:
    - Tras cada inicialización (pulsación de PROG) o encendido de la FPGA, la configuración debe volcarse de nuevo.
    - El CPLD toma la configuración recibida por el cable USB y la envía a la FPGA.
  - Los bitstream deben tener el formato .BIT generado por Xilinx ISE.

Volcado de bitstreams

permanente sobre Flash ROM

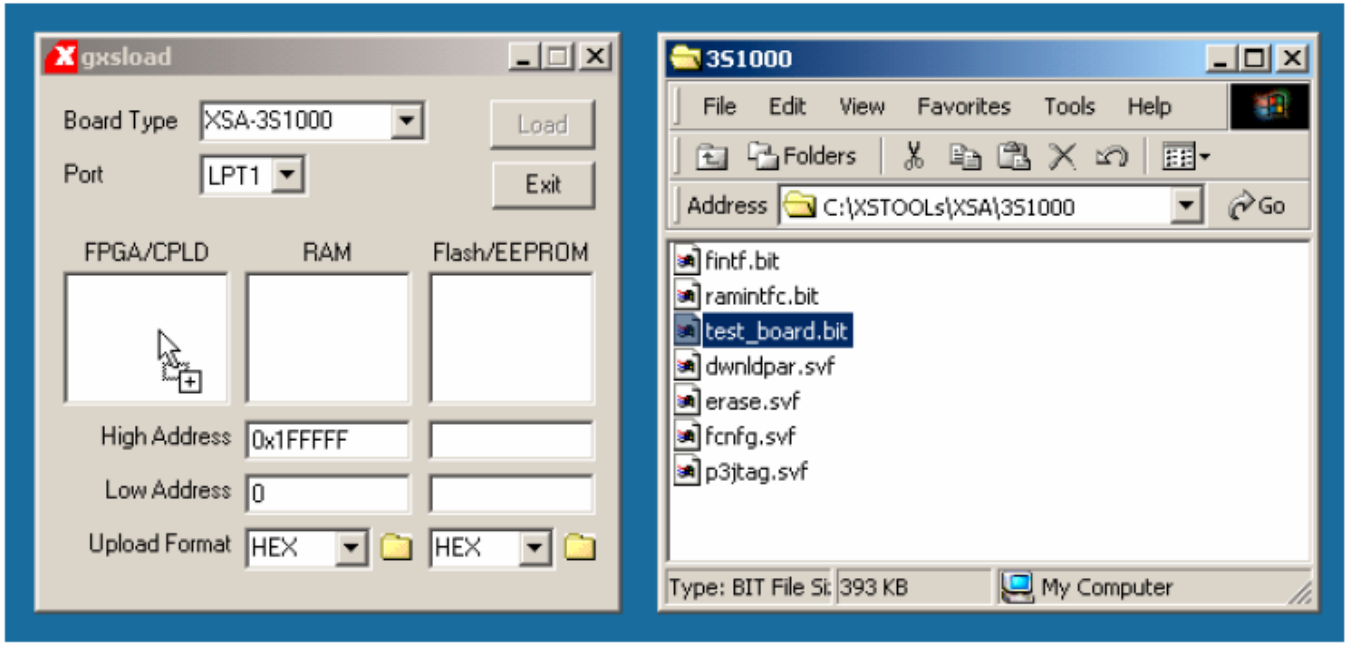
- Diseños volcados sobre Flash ROM: es el método usado para la puesta en producción de un diseño.
  - La configuración de la FPGA reside en la Flash ROM.
    - Tras cada inicialización (pulsación de PROG) o encendido de la placa, no es necesario volcar de nuevo la configuración.
    - El CPLD lee la configuración de la Flash ROM y la envía a la FPGA.
  - En la Flash ROM dispone de 4 cuadrantes en cada uno de los cuales cabe una configuración (diseño) de la FPGA distinta
    - Cada cuadrante es seleccionable con los interruptores SW1 y SW2 de la placa XSA-3S.
  - Antes de que un bitstream pueda cargarse en la Flash ROM el fichero en formato BIT debe convertirse en el formato .EXO o .MSG usando los comandos:

Quadrant	Address Range	Conversion Command	DIP Switch Setting	
			SW1-1	SW1-2
0	0x000000 – 0x07FFFF	promgen –u 0 file.bit –p exo -w promgen –u 0 file.bit –p mcs –w	ON	ON
1	0x080000 – 0x0FFFFF	promgen –u 80000 file.bit –p exo -w promgen –u 80000 file.bit –p mcs –w	ON	OFF
2	0x100000 – 0x17FFFF	promgen –u 100000 file.bit –p exo -w promgen –u 100000 file.bit –p mcs –w	OFF	ON
3	0x180000 – 0x1FFFFF	promgen –u 180000 file.bit –p exo -w promgen –u 180000 file.bit –p mcs -w	OFF	OFF

Volcado de bitstreams

XESS XStools

- GXSLOAD: utilidad para el volcado de bitstreams y datos específica para la placa de prototipado XESS XSA-3S.
  - Pueden volcarse configuraciones en la FPGA (.BIT) y en el CPLD (.SDF).
  - Pueden almacenarse configuraciones de la FPGA (.EXO o .MCS) en la Flash.
  - Pueden bajarse o subirse datos de la SDRAM (.EXO, .MCS, .HEX o .XES).
  - Los bitstreams / datos se arrastran y sueltan sobre la ventana correspondiente.



J.M. Mendías
2015

Síntesis y prototipado

flujo de tareas

1. Crear proyecto HDL: **`File > New Project`**
   - Family: **Spartan3**; Device: **XC3S1000**; Package: **FT256**; Speed: **4**; Language: **VHDL**
2. Añadir ficheros VHDL al proyecto: **`Project > Add Source`**
   - El fichero se añade tanto a la vista **`Implementation`** como a la **`Simulation`**
   - Alternativamente pueden crearse usando un Wizard: **`Project > New Source`**
   - Usar plantillas cuando haya dudas: **`Edit – Language Templates`**
   - Chequear sintáxis y salvar: **`Processes – Synthesize - Check Syntax`**
3. Añadir un fichero UCF al proyecto: **`Project > Add Source`**
   - El fichero se añade a la vista **`Implementation`**
4. Implementar: **`Processes – Generate Programming File`**
5. Volcar fichero BIT (fichero de bitstream) resultante usando *GXSLOAD*
   - Seleccionar placa y arrastrar fichero sobre la ventana FPGA/CPLD

J.M. Mendías 2015

Validación por simulación

flujo de tareas

- Para simulación de la especificación (no requiere implementar):

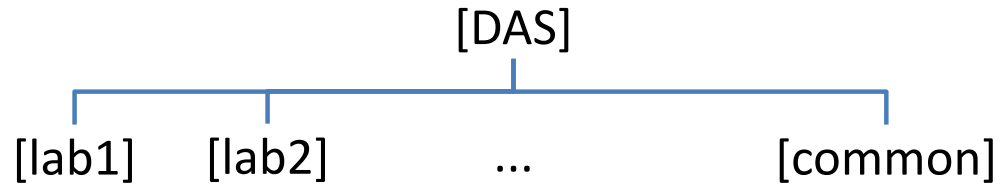
4. Añadir testbench VHDL al proyecto: **`Project > Add Source`**
   - El fichero se añade a la vista **`Implementation`**
5. Simular: **`Processes – Simulate Behavioral Model`**
   - Seleccionando previamente del desplegable **`Behavioral`**

- Para simulación funcional o de tiempos (tras diseñar o implementar):

5. Añadir testbench VHDL al proyecto: **`Project > Add Source`**
   - El fichero se añade a la vista **`Implementation`**
6. Generar modelo VHDL deseado: **`Processes – Implement Design – XXX – Generate XXX Simulation Model`**
   - Pueden ser tras traslación, proyección o emplazamiento
7. Simular: **`Processes – Simulate XXX Simulation Model`**
   - Seleccionando previamente del desplegable correspondiente

Organización de proyectos

- La estructura de directorios de nuestro entorno de trabajo será:



- Para cada laboratorio crearemos un proyecto:
  - Directorio raíz: contiene los ficheros fuente exclusivos de este diseño, el fichero de ligaduras y el bitstream generado.
  - Xilinx ISE creará otros ficheros y directorios auxiliares desde dicho directorio.

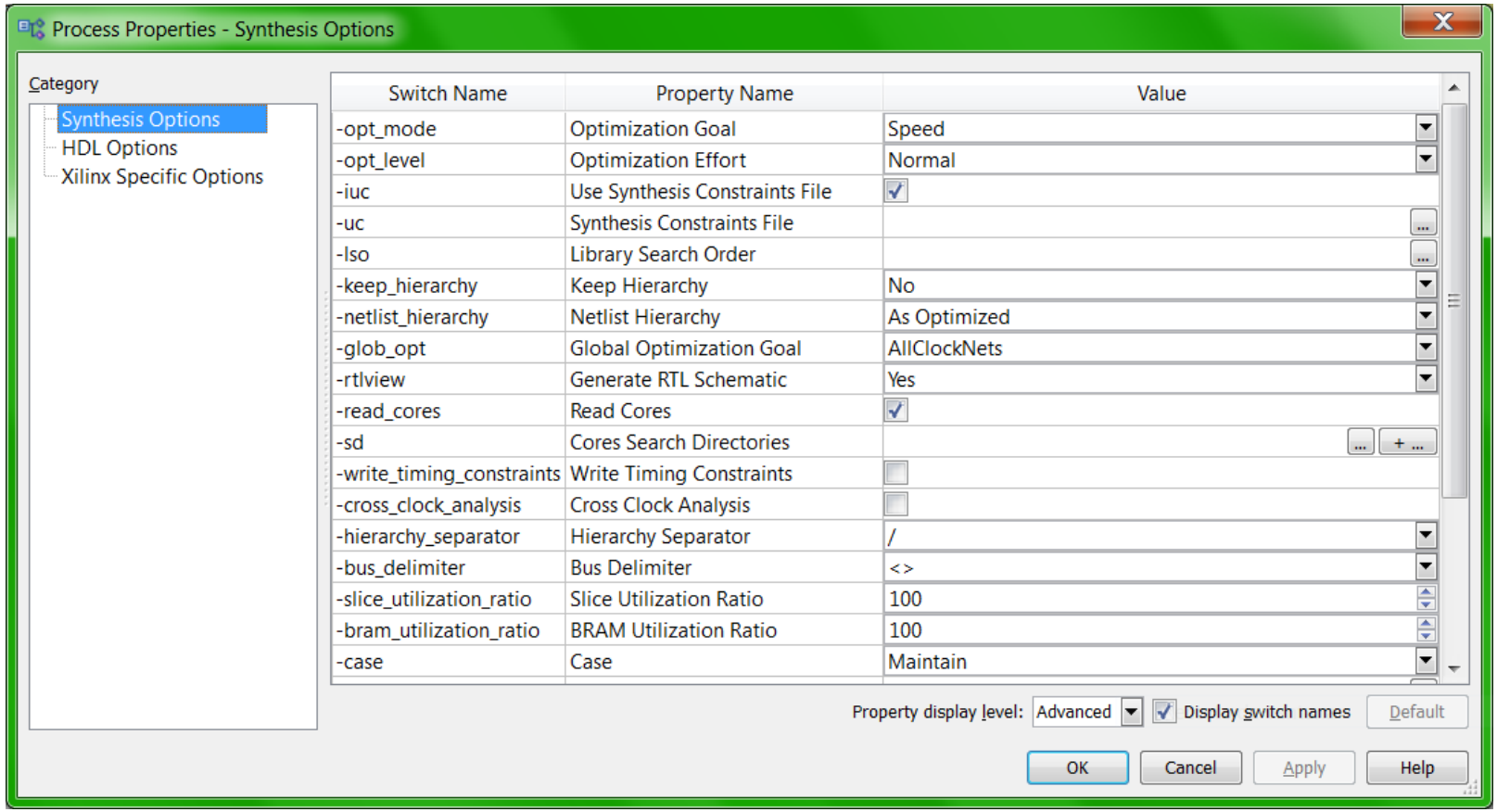
- Progresivamente iremos desarrollando una biblioteca de utilidades y componentes reutilizables denominada common:
  - Directorio raíz: contiene los ficheros fuente correspondientes.

Controlando Xilinx ISE

- Xilinx ISE es una herramienta automática de especificación y síntesis:
  - El diseñador especifica mediante VHDL el comportamiento funcional deseado.
  - La implementación resultante la decide un algoritmo cuyo funcionamiento puede ser condicionado mediante ligaduras
- Atributos: configuran la instancia de un cierto componente primitivo
  - Mediante la construcción "generic map" de VHDL
- Ligaduras de síntesis: controlan el proceso VHDL→ netlist
  - Mediante la construcción "attribute" de VHDL
  - A través del GUI de propiedades de síntesis (afectan al diseño completo)
  - Mediante fichero ".XCF" (debe ser incluido mediante el GUI en el proyecto)
- Ligaduras de implementación: controlan el proceso netlist → FPGA
  - A través del Constraint Editor
  - A través del GUI de propiedades de implementación (afectan al diseño completo)
  - Mediante fichero ".UCF" (debe ser incluido en el proyecto)
  - Algunas pueden aparecer en código VHDL o en el fichero ".XCF"
  - No todas las ligaduras aplican a todas las familias de FPGAs

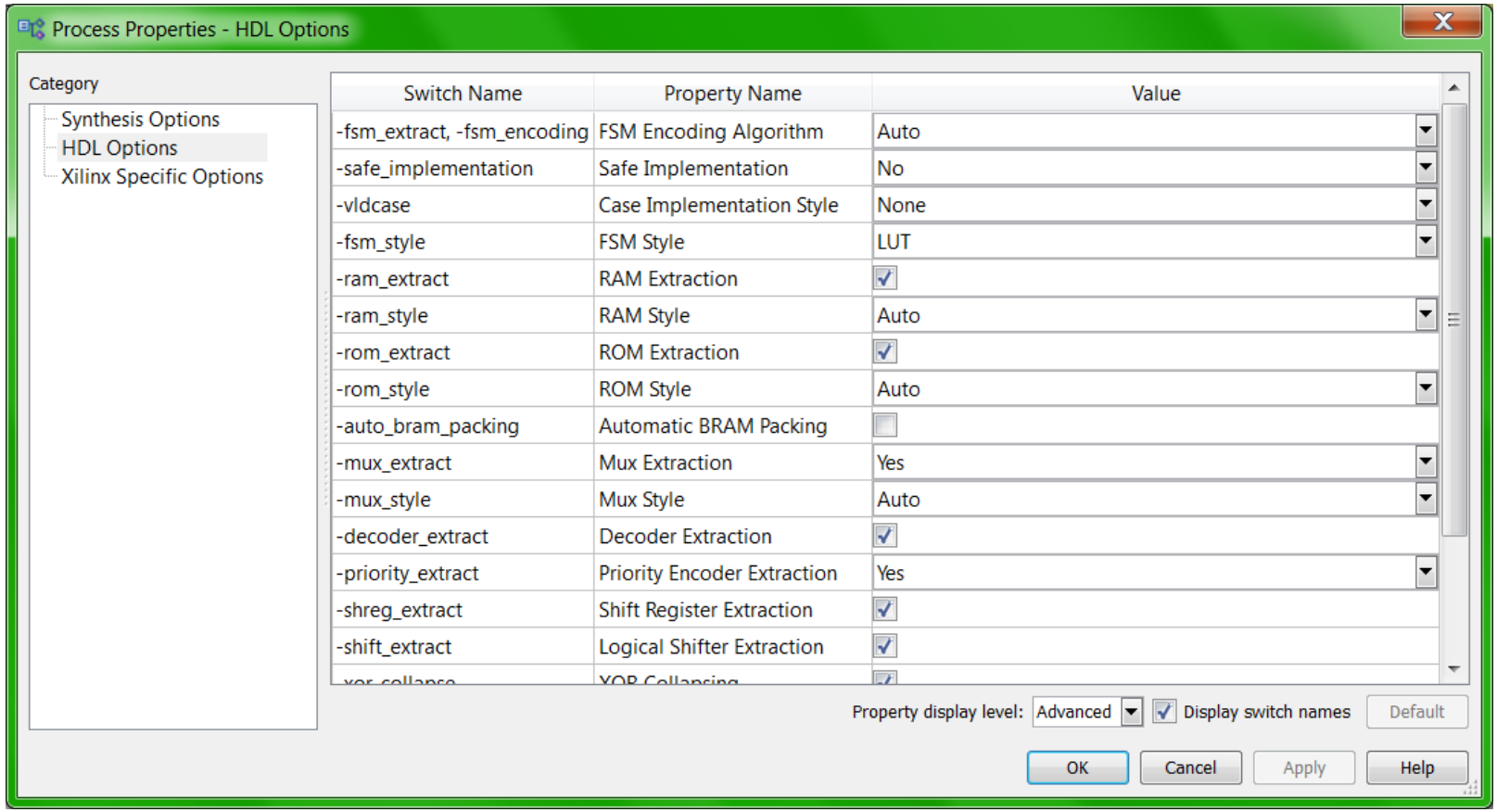
Ligaduras de síntesis

- Relativas al proceso de síntesis – Synthesis options
  - Objetivo de optimización (área o velocidad), esfuerzo en optimización...



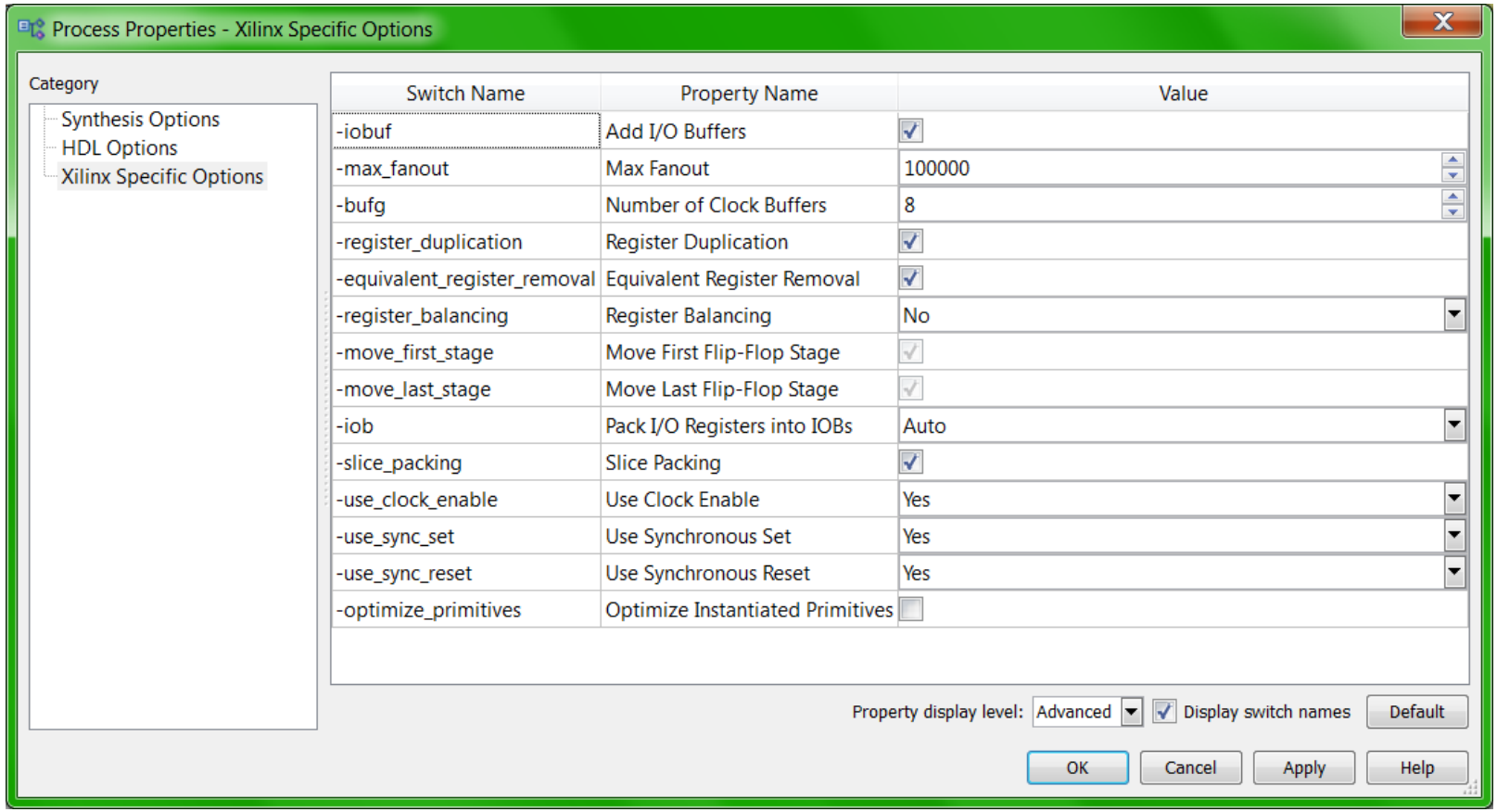
Ligaduras de síntesis

- Relativas a la interpretación del código – HDL options
  - Algoritmo de codificación de estados, inferencia de módulos...



Ligaduras de síntesis

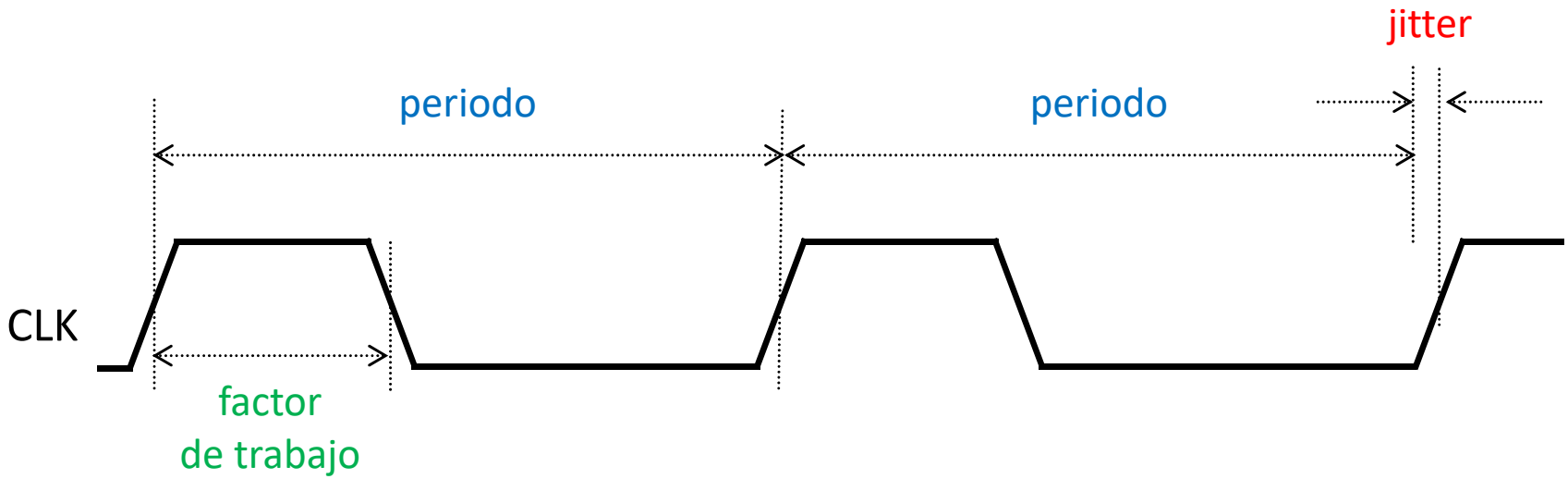
- Específicas de la herramienta – Xilinx Specific Options
  - Balanceo de registros, adición automática de buffers de E/S...



Ligaduras de implementación

especificación (i)

- Especificación de las características de la señal de reloj:
  - **NET** *clk* **TMN_NET =** *grupoClk*
  - **TIMESPEC TS***clk* = **PERIOD** *grupoClk periodo unidad* **HIGH** *factor_de_trabajo*% **INPUT_JITTER** *jitter unidad*;

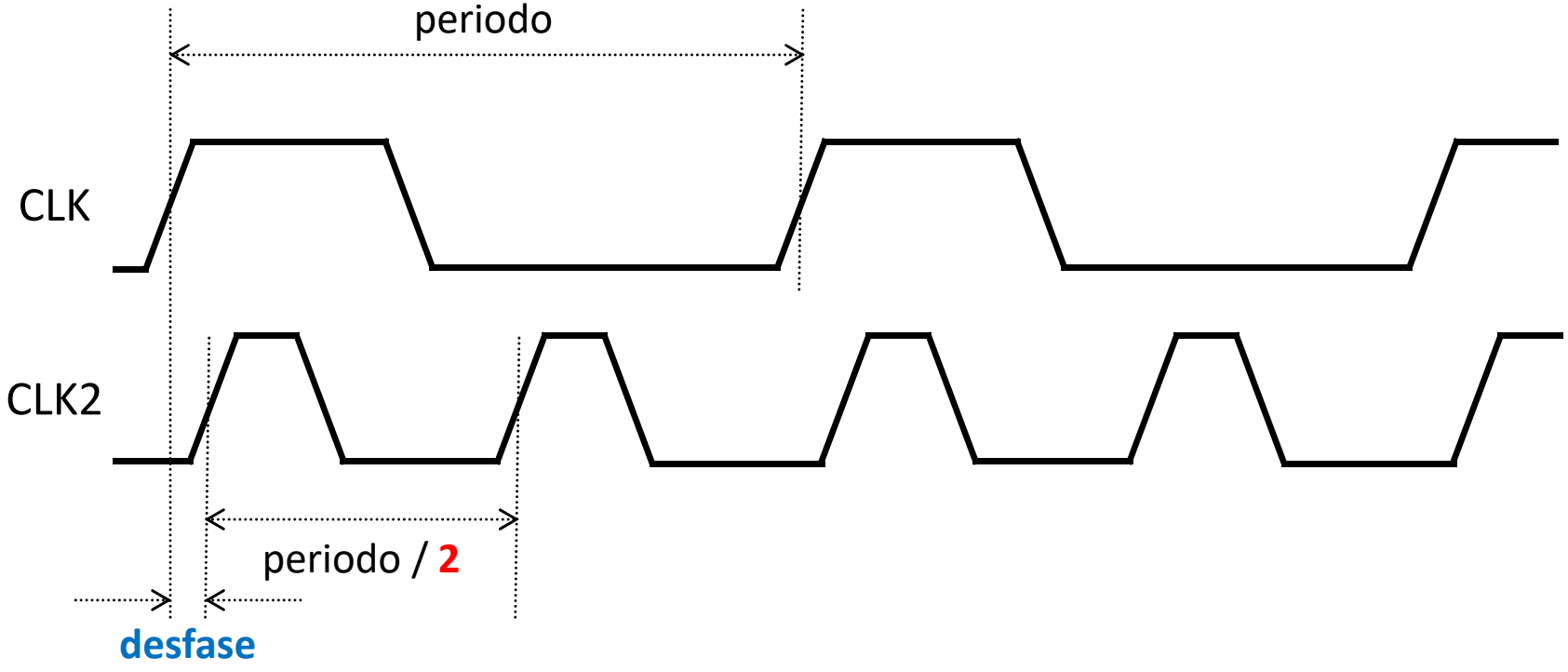


- Jitter: Desviación aleatoria máxima entre la duración de un ciclo de reloj cualquiera y la duración media del ciclo de reloj.
- Factor de trabajo: Fracción del tiempo de ciclo en el que la señal de reloj está en alta.

Ligaduras de implementación

especificación (ii)

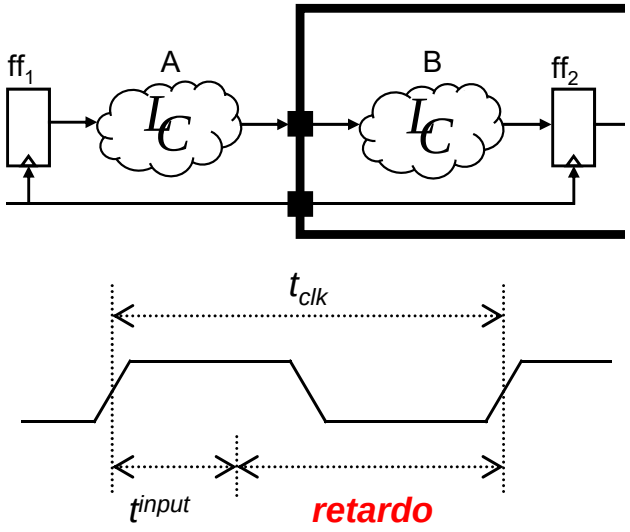
- Se denominan relojes derivados los que se obtienen dividiendo, multiplicando o desfasando un reloj maestro.
- Especificación de reloj derivado:
  - **NET** *clk2* **TMN_NET =** *grupoClk2*
  - **TIMESPEC TS***id* = **PERIOD** *grupoClk2* **TS***clk* [*|/] *factor* **PHASE** [+|-] *desfase unidad*;

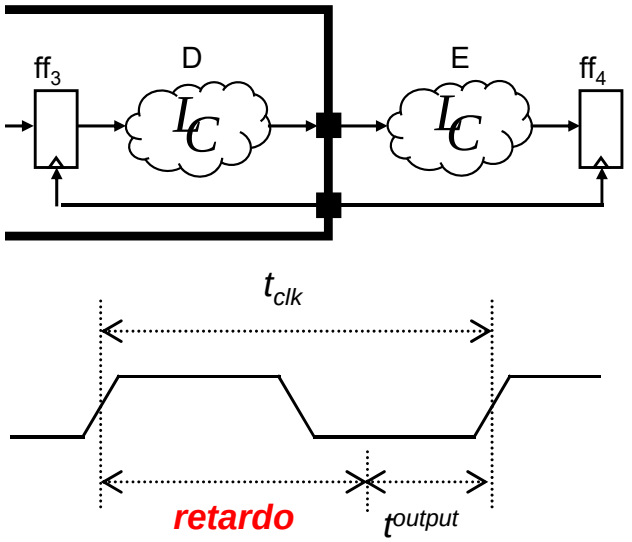


Ligaduras de implementación

especificación (iii)

- Especificación de retardo de llegada máximo a entradas:
  - `[NET puerto] OFFSET = IN retardo unidad BEFORE clk RISING;`
- Especificación de retardo de llegada máximo tolerable a salidas:
  - `[NET puerto] OFFSET = OUT retardo unidad AFTER clk RISING;`
- Especificación de retardo máximo absoluto tolerable:
  - `TIMESPEC TSid = FROM grupo [THRU puntos] TO grupo retardo unidad;`

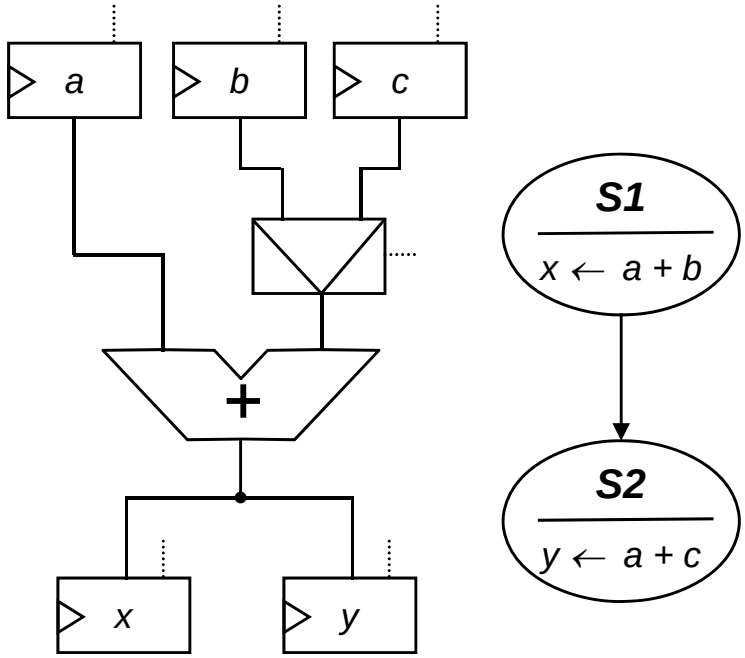




Ligaduras de implementación

especificación (iv)

- Se denomina camino falso a todo camino de un circuito que sea utilizado para propagar señales inútiles.
  - No tiene que ser optimizado bajo los criterios impuestos al resto del diseño.
  - Cualquier diseño con porciones de lógica compartida puede tener caminos falsos.
  - También son caminos falsos los lazos existentes en puertos bidireccionales, los caminos que atraviesan lógica de reset, etc.



- Desde el punto de vista lógico en esta ruta de datos existen 6 caminos registro a registro:
  - a → x, a → y, b → x, b → y, c → x, c → y
- Sin embargo, si sobre ella sólo se realizan las transferencias entre registros indicadas por la máquina de estado, sólo algunos caminos son utilizados para realizar transferencias útiles:
  - a → x, a → y, b → x, c → y
- Luego los restantes caminos, son caminos falsos que no requieren ser optimizados:
  - b → y, c → x

Ligaduras de implementación

especificación (v)

- Se denomina camino multiciclo a todo camino combinacional que tarda en propagar valores más de un ciclo
  - Es una técnica usada para solventar el problema de los grupos de lógica de retardo elevado sin reducir la frecuencia de reloj.
  - Requiere que se diseñe para que el registro ubicado al final del camino no cargue nuevos valores en todos los ciclos.

- Identificación de caminos falsos:
  - `TIMESPEC TS`*`id`*`=FROM` *`grupo`* `[THRU` *`puntos`*`] TO grupoDest TIG;`

- Identificación de caminos multiciclo:
  - `TIMESPEC TS`*`id`*`=FROM` *`grupo`* `TO` *`grupo`* `TS`*`clk * factor;`*

Ligaduras de implementación

especificación (vi)

- Las variaciones en la temperatura ambiente o voltaje de alimentación pueden alterar el retardo de un circuito.
  - Las herramientas usan en los cálculos de cualquier retardo factores de escala dependientes de las condiciones de funcionamiento previstas.
- Especificación de condiciones de funcionamiento:
  - **VOLTAGE** = *valor* **V**;
  - **TEMPERATURE** = *valor* **C**;
- Especificación de localización física de una instancia (en especial pin)
  - **NET** *instancia* **LOC=***localizacion*;
- Especificación de interfaz eléctrico de un pin
  - **NET** *instancia* **IOSTANDARD=***interfaz*;
  - Donde el interfaz depende del dispositivo: **LVTTL**, **LVCMOS33**, **LVCMOS25**...

J.M. Mendías 2015

Ligaduras de implementación

nombrado de elementos

- Muchas ligaduras requieren hacer referencia a grupos de lógica, existen distintas maneras de nombrarlos.
  - Los nombres de las instancias se heredan de los identificadores usados en VHDL
  - Para referirse a elementos concretos pueden usarse caracteres comodín (*, ?)
  - Los niveles de jerarquía se separan con /

- Agrupaciones predefinidas:
  - **`FFS, LATCHES, RAMS, PADS, BRAMS_PORTA, BRAMS_PORTB...`**
- Agrupación de elementos por nombre de interconexión:
  - **`NET`** *`interconexión`* **`TNM_NET =`** *`grupo`*`;`
- Agrupación de elementos por nombre de instancia:
  - **`INST`** *`instancia`* **`TNM =`** *`grupo`*`;`

Ligaduras de implementación

ejemplos (i)

- Indicar que la señal clk es un reloj a 50 MHz con factor de trabajo del 50% (y que esta ligadura afecta a todos los elementos conectados a dicha señal).

```
NET clk TNM_NET = clk;
TIMESPEC TSclk = PERIOD clk 20 ns HIGH 50%;
```

- Indicar que la señal x tiene un retraso de 5 ns respecto al flanco de subida del reloj de 50 MHz (20 ns de periodo):

```
NET x OFFSET = IN 15 ns BEFORE clk RISING;
```

- Indicar que la señal z requiere un margen de 5 ns respecto al flanco de subida del reloj de 50 MHz (20 ns de periodo):

```
NET z OFFSET = OUT 15 ns AFTER clk RISING;
```

J.M. Mendías 2015

Ligaduras de implementación

ejemplos (ii)

- Indicar que la señal clk1 (12.5 MHz) se obtiene por división de la frecuencia de clk0 (50 MHz):

```
NET clk0 TNM_NET = clk0
NET clk1 TNM_NET = clk1
TIMESPEC TSclk0 = PERIOD clk0 20 ns HIGH 50%;
TIMESPEC TSclk1 = PERIOD clk1 TSclk0 * 4;
```

- Indicar que la señal clk1 tiene un desfase de 180º respecto a la señal clk0 (50 MHz):

```
NET clk0 TNM_NET = clk0
NET clk1 TNM_NET = clk1
TIMESPEC TSclk0 = PERIOD clk0 20 ns HIGH 50%;
TIMESPEC TSclk1 = PERIOD clk1 TSclk0 PHASE + 10 ns;
```

J.M. Mendías
2015

Ligaduras de implementación

ejemplos (iii)

- Indicar que los caminos que unen el regA y regB son falsos

```
INST regA* TNM = regA;
INST regB* TNM = regB;
TIMESPEC TSfalso = FROM regA TO regB TIG;
```

- Indicar que regB solo carga valores de regA en 1 de cada 2 ciclos, es decir, que el camino que une regA y regB es multiciclo (de 2 ciclos)

```
NET clk TNM_NET = clk;
TIMESPEC TSclk = PERIOD clk 20 ns HIGH 50%;
INST regA* TNM = regA;
INST regB* TNM = regB;
TIMESPEC TSmulticiclo = FROM regA TO regB TSclk * 2;
```

J.M. Mendías
2015

Ligaduras de implementación

ejemplos (iv)

- Indicar condiciones ambientales.

```
VOLTAGE = 3.3 V;
TEMPERATURE = 20 C;
```

- Indicar el interfaz eléctrico que se desea para el puerto x.

```
NET x IOSTANDARD = LVCMOS25;
```

- Indicar que el puerto clk se ubique en el pin P8 de la FPGA.

```
NET clk LOC = P8;
```

Acerca de *Creative Commons*

- Licencia CC (Creative Commons)
  - Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:

Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.

Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: https://creativecommons.org/licenses/by-nc-sa/4.0/